

USER GROUP NEWS

OCT 7 1983



FORWARD

When the first issues of "USER GROUP NEWS" were compiled there was a concern that there would not be enough new material for the following issues in the span of three months. The major problem, as it turns out, was which of many articles would be printed. Still one problem remains! Almost all articles focus on the 8560. That is not really surprising because of the versatility of the TNIX tool set; but we don't want to ignore the owners of 8002 and 8550 systems. If you are an 8002 or 8550 user please take special note of the following:

CALL FOR ARTICLES

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

ABOUT THE "USER GROUP LIBRARY SECTION"

We have collected a number of application programs since the last issue and we will make the programs reported in the User Group Library available through your Tektronix Applications Engineer. Separate volumes will be compiled for the 8002A, 8550, and TNIX. Each issue will report updates to the library and a separate total listing will be produced annually.

John Owens
Editor

CONTENTS

PRODUCT INFORMATION SECTION	1
TEKTRONIX and DIGITAL MICROPROCESSOR SUPPORT FOR VAX	1
MDP USER GROUP LIBRARY	2
NEW VERSIONS OF CLASS C PRODUCTS	2
10 MHZ FOR THE 68000 EMULATOR IS HERE	3
68000 PASCAL FLOATING POINT LIBRARIES	4
8560 GPIB CONTROL CAPABILITY	4
SPEED OF 8560 DISK BACKUP USING GPIB OPTION	7
SUPPORTING 8001 AND 8002A CUSTOMERS	7
NEW MANUALS ORDERABLE	8
APPLICATIONS SECTION	9
8086/88 ASSEMBLER ASSUME DIRECTIVE	9
IMMEDIATE DATA IN 8086 ASSEMBLER	10
SEGMENTED ADDRESSES AND THE 8560 LINKER	10
A VAX/8560/8540 Configuration	10
SOME TOOLS FOR THE 8560	11
TNIX CONFIGURABILITY - WHEN AND WHERE	16
INCLUDING MATH LIBRARIES IN C PROGRAMS	17
EXTENDING THE LINKER SYMBOL CAPABILITY	17
USE OF PRINTER CAPABILITIES WITH NROFF	19
PDB FOR PAYMENT LEARNING GUIDE DEMONSTRATION RUN	22
PASCAL DEBUG/TERMINAL CONNECTION	33
ACE V.3 & ACECONFIG	34
Z8000 MEMORY ALTERATION PROBLEMS	34
CONSTANTS LOCATION WITH 8086 PASCAL - CORRECTION	34
MDP USER GROUP SOFTWARE LIBRARY/ARTICLE SUBMITTAL FORM	35
THIRD PARTY SOFTWARE	37
MDP SOFTWARE REFERRAL SERVICE	37
MDP SOFTWARE REFERRAL CATALOG UPDATE	39
CHANGES TO MDP SW REFERRAL CATALOG	39
NEW PRODUCTS FOR MDP SW REFERRAL CATALOG	40
REAL TIME OPERATING SYSTEMS	42
APPLICATION SOFTWARE	42
SOFTWARE PRODUCT REFERENCES	43
RUNNING UNIX SOFTWARE ON THE 8560	44
PRODUCT PERFORMANCE SECTION	45
SOFTWARE SUBSCRIPTION SERVICE	45
SLOW 8560 OR EXCESS GETTY TIMES	45
Z8000 ASSEMBLER VERSION 2.0	46
8086 PASCAL/PDB MOD	46
4105 POWER UP PROBLEMS	46
MDP BUG BASE	46
8540 ROMPATCH FAILURE	46
NATIVE PROGRAMMING TOOLS INSTALLATION	47
DIRECTORY PERMISSIONS OF /USR/INCLUDE/SYS	47
GLOBAL ERROR MESSAGES AND NO GLOBALS	47
PASCAL DEBUG ERROR #79	48
16BIT PASCALS SECTION ASSIGNMENT ERROR	48
6801 EMULATOR MODE SELECT ERROR	48
PASCAL AND REDEFINITION OF IDENTIFIERS	49
PASCAL OPTIMIZER ERROR	49
8086 PASCAL TRUNCATION OF IDENTIFIERS TO 16 CHAR.	50
USER GROUP LIBRARY ABSTRACTS	53

ACCOUNT - LOGIN/LOGOUT HISTORY	53
ADMIN - DISPLAY LOGIN STATS	53
ASCII - CONVERTER	54
ASCITABLE - PRINTS ASCII CHART	54
ASMIT - AUTO FILENAME EXTENSION	54
ATS - AT STATUS	54
ATSTATS - AT STATISTICS	54
DECIMAL - CONVERTER	55
DONUM - DO COMMAND X TIMES	55
EXTX - EXTRACT TEXT	55
FDMP - FILE DUMP UTILITY	55
FDUMP - FILEDUMP	55
FMAN - FAST MANUAL PAGE COMMAND	56
FMT - LINK LIST FORMATTER	56
HELP - PERSONAL COMMANDS HELP	56
HEX - CONVERTER	56
IFIX - INTEL FIX	57
IOCFORM - 'FILL OUT' FORM	57
LOAD - OBJECT/SYMBOL LOADER	57
LOG - TERMINAL SESSION LOGGER	57
LOG1 - TERMINAL SESSION LOGGER	57
LOGGER - PIIONE CALL LOGGER	58
MAILALL - MAIL TO ALL USERS	58
MAILTO - MAIL TO USERS ON REMOTE SYSTEMS	58
MDSHEX - INTEL SYMBOL LISTER	59
MLABEL - MAIL LABEL PRINTER	59
MONTH - CALENDAR PRINTER	59
OCTAL - CONVERTER	59
PATCH - FILE PATCH ROUTINE	60
PROLOG - M900 PROM PROGRAMMER COMMUNICATIONS PROGRAM	60
PRQ - PRINTER QUEUE STATUS	60
RESTORE - RESTORE EMU STATUS	60
SAVE - SAVE DEBUG SESSION	61
TELE - TELEPHONE NUMBER	61
TOOLS - PERSONAL COMMANDS LIST	61
TR68000 - 68K TRACE FILTER	61
TRZ80 - Z80 TRACE FILTER	62
TRZ8002 - Z8K TRACE FILTER	62
USPLIT - TEKHEX FILE SPLIT	62
VMORE - 'VANILLA MORE'	62

PRODUCT INFORMATION SECTION

TEKTRONIX and DIGITAL MICROPROCESSOR SUPPORT FOR VAX

Tektronix and Digital Equipment Corporation have just concluded an agreement to jointly market Tektronix Microprocessor support tools on the Digital Equipment VAX series of computers. Through this agreement, Tektronix and Digital will be able to offer designers the most comprehensive set of hardware and software tools available in the industry.

Tektronix currently offers support for the 68000/08/10, 8086/88/186, Z80, NSC800, 8080, 8085, and 6809 microprocessors on the VAX under either VMS or UNIX 4.1bsd operating systems. Included with this support software is a unique interface package (ICOM40) which allows users to access the Tektronix Emulation unit from any terminal on the VAX; thus giving VAX users the same integrated functionality as Tektronix provides with their 8560 systems. High level languages, Pascal and C, are available for the 68000/08 microprocessors, including Tektronix' unique LANDS (Language Development System) package. Tektronix provides Language Directed Editing, Integration Control System, High Level Debug, and Chip Specific Compilers.

For more information contact your local Tektronix representative.

Doug Johnson Software Product Marketing Manager

MDP USER GROUP LIBRARY
CALL FOR PROGRAMS FOR ALL TEK MDP PRODUCTS - NEW AND OLD

The MDP User Group Library is here! The library is provided as a service to MDP users for collecting and distributing user-contributed software for all Tektronix Microprocessor Development Products. Library software will not be supported by Tektronix. However, some application engineers and customers have created some useful programs which may benefit others. The program works like this:

- All users are encouraged to submit their creations to Library Manager, MDP Marketing, P.O. Box 4600, 92-635, Beaverton, OR 97075. All submissions will be considered and are made with the understanding that the software may be placed in the public domain. For your convenience, a software submission form is included in this issue and on the first distribution disk volume. We must have the author's name to consider a submission, but we will withhold it if you prefer not to be contacted by anyone. Please make your entries on disk media.
- We will generally check out the programs, but no guarantees of any kind will be made. We prefer to have the object, source, documentation, and manual page (as applicable) submitted on a floppy disk, but we'll take whatever you have. If the program warrants it, we can add the documentation.
- All accepted programs will be archived in MDP Marketing and as soon as we have enough to reasonably fill a disk, a new volume will be released. Copies of these disks will be available from your Tektronix Application Engineer. Separate distributions will be made for the 8002A, 8550, and TNIX.
- Quarterly we will provide a listing and index of all new entries. Annually we will produce a complete catalog which will be distributed with "USER GROUP NEWS." Each volume distributed will also include a catalog listing and summary of all software included on that disk.

The first TNIX volume is almost ready for release and includes over thirty programs, ranging from useful shell scripts, to complex C programs which allow binary patching of TNIX files. Other useful programs are out there, so send 'em in!

Greg Saville MDP Technical Support Manager

NEW VERSIONS OF CLASS C PRODUCTS

New versions of the Text Processing, Native Programming, and Auxiliary Utilities Packages are available. These versions are modified for compatibility with TNIX v.2 and correct several Software Problem Reports. There are change pages to the user manuals which reflect changes in installation and file content. Order information is available from your Tektronix Sales Engineer.

Product Name	Product #	Upgrade Part #
Text Processing	8560U01	020-1086-00
Native Programming	8560U02	020-1088-00
Auxiliary Utilities	8560U03	020-1087-00

TNIX V.2 COMPATIBILITY

For compatibility with TNIX v.2 there are changes in on-line manual pages and installation procedures. The on-line manual pages are distributed with version 2 of these products instead of with TNIX as they were for version 1. The installation procedures are modified to accommodate installation on previous versions of TNIX.

TEXT PROCESSING

"Nroff" now works with index macros (nroff -T). "Spell" now works for the options: American, British, S (hashed stop), and local.

NATIVE PROGRAMMING

Programs generated by "lex" can now link in functions from /lib/libl.a (ie, "all print", "sprint", "printable", "yyreject", "yyracc", "yyles", and "yywrap").

Updated include files for C programs are provided and are the same as supplied with TNIX v.2. These are the additions and deletions to /usr/include:

Added	Deleted
crtdev.h	SCCS/s.stdio.h
mtab.h	ar.h
sqqpio.h	ident.h
sys/bootblk.h	lp.h
" /cmd_def.h	olddump.h
" /disc.h	sys/tiop.h
" /error.h	
" /iopcmd.h	
" /lp.h	
" /timeblk.h	
" /trm_def.h	

AUXILIARY UTILITIES

"Learn" now recognizes as correct answers for TNIX that it previously recognized as incorrect (the "tail" command was modified).

CUSTOMER UPGRADES

To obtain the updates, order by the part number shown in the table above.

Rodney Bell MDP Product Marketing Manager

10 MHZ FOR THE 68000 EMULATOR IS HERE

An upgrade is available to increase the maximum clock frequency of the 68000 Emulator from 8 MHZ to 10 MHZ. The upgrade is available for ALL 68000 Emulators.

For existing emulators, users can either install an upgrade kit, or have the emulator upgraded via a board exchange through Service Support.

Bill Bevan MDP Product Marketing Manager

68000 PASCAL FLOATING POINT LIBRARIES

A speeded-up version of the floating point libraries for the 68000 PASCAL is available now. If an application depends upon "fast" floating point, Tektronix provides it. Users who have the Software Subscription Service will receive this update automatically.

Marilyn Hanson MDP Product Marketing Manager

8560 GPIB CONTROL CAPABILITY

When the 8560 or 8561 contains the GPIB Interface and the Category C Native Programming Package, it can control GPIB-compatible instruments. Specifically, it can perform the following functions:

- enable instruments to talk
- enable instruments to listen
- perform serial polls of all instruments or selected instruments
- perform parallel polls
- anything else except pass control to another instrument.

The C programming language allows total access to all device functions. With 3 additional commands, control capability comparable to a 4050 Series Graphics Computer is possible using only the TNIX shell.

TNIX/GPIB ARCHITECTURE

All GPIB devices are TNIX special files, and provide the same level of interface as other special files. (For example, the echo command works the same on a GPIB instrument as it does on a terminal.) When you install a GPIB instrument on the system, you can select one of 3 driver routines that tailor the special file to a specific application:

- The "magtape" driver provides an interface to Dylon magnetic tape systems.
- The "instr" driver provides a general interface that accommodates just about any instrument.
- The "direct" driver provides total low-level control of the GPIB. It may be used if the "instr" driver is unable to handle a particular instrument, or it can be used to provide direct access to the bus.

NOTE: The software documentation and the manuals refer to these drivers as "GD"s.

All three drivers can automatically generate device addresses and perform serial polling tasks.

The GPIB Interface itself is an intelligent board, containing an 8088 processor and RAM-based software (only the bootstrap code and diagnostics are in PROM). The combination of this intelligent board, and 11/23 main CPU, hard disk, and the TNIX operating system make this a powerful GPIB control solution.

COMMANDS

These commands come with standard TNIX 1.4 (or above):

- ldgplib - initializes the GPIB Interface
- mkgplib - install/remove/modify GPIB instruments on the system
- statgplib - provides status information about GPIB instruments
- miscellaneous magtape commands
- a detailed description of GPIB special files (type "man 4 gpib")

These commands are insufficient to control instruments from the shell. To do this, you need the following commands:

- `gread` - reads device-dependent data from a specified device
- `srq` - waits for a service request on the specified device, polls the device, writes the status byte in a temporary file, and signals the parent process
- `setgpib` - sets and reports the termination mode and exclusive/inclusive use of a device.

The following is a copy of the man page for "exgpib". This is a command that was created while designing the GPIB Interface software/firmware. `exgpib` allows the user to do just about anything that can be done on the GPIB. The user interface provides full functionality.

Name

`exgpib` - exercise GPIB interface

Syntax

`exgpib [device-name]`

Description

`Exgpib` exercises a direct-type GPIB device based on a command string received from standard input. If an argument is given it is used as the device to be exercised. If no device is specified the device `/dev/ctr` is used. The source and destination of device dependent data is initially set to standard input and output but may be changed with exerciser commands.

Many commands are simply GPIB mnemonics for the command to be sent. The command list follows:

<code>=STRING</code>	send STRING as device data
<code>:HH</code>	send HH (hex) as a character
<code>eoI D</code>	send char D (decimal) with EOI
<code>@STRING</code>	send STRING as device data and accept a one record response.
<code>? D</code>	accept D characters (default 1)
<code>a D</code>	accept D records (default 1)
<code>s D</code>	source D records (default 1)
<code><FILE</code>	set data input file to FILE
<code>>FILE</code>	set data output file to FILE
<code>x</code>	external data xfer
<code>xfer</code>	
<code>xfer D</code>	external data xfer [time]

Status and data transfer control:

<code>stat</code>	print device status
<code>bus</code>	print (GPIB) bus status
<code>htime D</code>	set data xfer time
<code>ptime D</code>	set poll xfer time
<code>term lf</code>	set termination mode
<code>term crl</code>	
<code>term eoi</code>	
<code>term none</code>	
<code>adr</code>	print GPIB address
<code>adr PRI.SEC</code>	set GPIB address (addresses in decimal)

Interface commands:

atnt	set ATN
atnf	
rent	set REN
renf	
ifc	send IFC pulse
ifcf	set IFC false
llo	send universal command
dcl	
ppu	
gtl	send addressed command
get	
sdc	
c D	send byte D (decimal) as a command
ta D	send talk address D
la D	talk listen address D
sa D	send secondary address D
unt	send untalk
unl	send unlisten

Poll commands:

poll	turn on auto poll
poll on	turn on auto poll
poll off	turn off auto poll
spoll	serial poll device
ppoll	parallel poll device
sps	print saved s-poll status
spstat	print saved s-poll status
srq	wait for SRQ
mysrq	wait for device SRQ

Misc commands

v	Verbose mode on
voff	Verbose mode off
ass	assign GPIB
deass	deassign GPIB
excl	set excl open
excl off	set no excl open
quit	exit the exerciser
q	

Files:

/dev/ctr default direct gpib device.

Diagnostics:

A diagnostic "can't open device-name" is printed if a device or source file is requested that cannot be opened.

Commands which are not understood are printed on standard error surrounded with question marks (?? foo ??).

If the execution of a GPIB command fails, a message describing the failure and the returned error code is printed to standard error.

Greg Gadeholt MDP Product Engineer

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 takes about 1 hour, 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.

Chuck Smith MDP Product Marketing Manager

SUPPORTING 8001 AND 8002A CUSTOMERS

The matrix below shows which items to order for support on an 8001 or an 8002A Microprocessor Lab. The part numbers listed under the columns represent manuals, software or proms for the appropriate mainframe.

Microprocessor	ORDER THESE Emulator/Probe	PLUS THESE ITEMS	OR THESE ITEMS
		8002A	8001
Z-80	8300E04 8300P04	062-3511-05 070-2705-01 070-2710-01	160-0163-01 160-0164-01
8085	8300E06 8300P06	062-3750-05 070-2702-01 070-2707-01	160-0165-01 160-0166-01 070-2871-01
6800	8300E02 8300P02 8300P03	062-3735-05 070-2703-02 070-2708-02	160-0004-02 160-0251-00 070-2713-00 070-2951-00
8048	8300E10 8300P10 8300P11 8300P12	062-4401-04 070-2611-00 070-2612-00	160-0763-00 160-0764-00 160-0765-00 070-2855-00 070-2473-00
1802	8300E09 8300P09	062-4169-03 070-2627-00 070-2628-00 070-2628-00	160-0462-00 160-0463-00 070-2854-00 070-2882-01

MEMORY BOARDS - If a user wishes to add memory to either an 8001 or an 8002A, order the 16K memory board from Customer Service under part number 670-5298-02. Users can also order 32K Memory Boards nomenclated as 8550F02. When the user installs an 8550F02, the emulator operation may change. A brief summary of the changes is presented below.

EMULATOR	OPERATING CONDITION IN 8001 or 8002A
Z80	Must be strapped for wait states
8085	Must be strapped for wait states
6800	Must be strapped for wait states
8048	No strapping necessary
1802	No strapping necessary

Roger Hokanson MDP Product Marketing Manager

NEW MANUALS ORDERABLE

8560 Series MUSDU System Users	070-4730-00
8560 Series MUSDU System Managers and Instruction Guide	070-0211-00
8500 MDL Series 68000 Emulator Processor Instruction Sheet	070-4798-00
8561 MUSDU 4-User Upgrade Instruction Sheet	070-1623-00
8561 MUSDU 8-User Upgrade Instruction Sheet	070-1438-00
8561 MUSDU 4-User Upgrade Option User Information Instruction Sheet	070-4764-00
8561 MUSDU 8-User Upgrade Option User Information Instruction Sheet	070-4770-00

Charlene Eason MDP Customer Support

APPLICATIONS SECTION

8086/88 ASSEMBLER ASSUME DIRECTIVE

In the 8086, each memory reference is calculated by adding the contents of a segment register, which contains the starting address of a segment and an offset from the start of a segment. With the Tek Assembler, all offset calculations on relocatables are done at link time. When the linker resolves the address of a relocatable symbol, it determines its 20-bit address. The ASSUME directive is used to tell the assembler what values will be in the segment registers at runtime. The assembler instructs the linker to calculate the offset by subtracting the ASSUMEd value for the start of the segment (also a 20-bit address) from this resolved address. If the memory reference is in the segment, this subtraction yields a 16-bit offset. However, if the memory reference is outside the segment, the linker will issue a truncation error. For more information, see the 8086/88 Assembler Specifics manual, page 9A-41.

Marilyn Hanson MDP Product Marketing Manager

IMMEDIATE DATA IN 8086 ASSEMBLER

Immediate data is handled in the following way by the 8086 assembler.

If a scalar is specified as immediate data, the assembler checks that it is in the correct range (8 or 16 bits, signed or unsigned).

If the immediate is relocatable, a 20-bit value is calculated by the linker. It is the *user's* responsibility to get this into a form appropriate for the instruction. The main reason for doing this is that the 20-bit value contains both the segment and offset parts of an address. By allowing (making) the user to handle immediate data, either part may be extracted and manipulated. You can use the BITS function to access either part.

Marilyn Hanson MDP Product Marketing Manager

SEGMENTED ADDRESSES AND THE 8560 LINKER

If an address is defined in one module, and the user wants to use the upper 16 bits of a 20 bit address to set a segment register in another module, use the bits function. "LO", "HI", and "BITS" operations are resolved by the linker when the definition of the address is contained in one module and is referenced in another module.

CAUTION: When the linker is required to resolve the assembler function, subsequent assembler operations or inclusion in expressions is not allowed; i.e., the result of the linker resolved function must be used directly as in the following example:

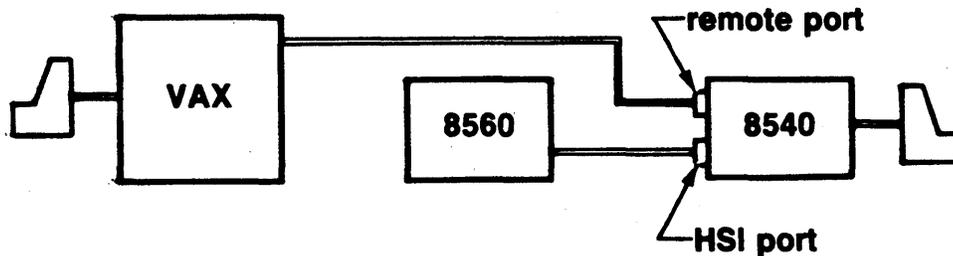
```
global    fred
movw     ss,#bits(fred,4,16)
```

If the resulting 16 bit value is needed in subsequent instructions, it will need to be entered as above again. It would be incorrect in the referencing module to assign the result of the bits expression to a label for later use.

John Owens MDP Technical Support Manager

A VAX/8560/8540 Configuration

The following is a synergistic VAX/8560/8540 configuration:



With ICOM40 on the VAX, from a terminal connected to the 8540, the user can switch between using the 8560/8540 configuration and the VAX/8540 configuration.

```
config term          ; connects the user (8540) to the 8560
{ 8560/8540
```

```

        session }

config local
config term i=r m=4 t=10 ; connects the user (8540) to the VAX, using
                           the remote port on the 8540.

{ VAX/8540
  session }

```

Therefore, both configurations are available from one terminal!

***** NOTE *****

This has been tested for VAX UNIX only.

Diane Wortsman MDP Product Marketing Manager

- - - - -

SOME TOOLS FOR THE 8560

Jean Paul Bonsang of Tek France has provided the several very useful programs that follow.

- 1 FDUMP provides a file dump (or directory) in both hex and ascii format. The format of the output is exactly like the 8550 fdump command. The source for the fdump command follows:

```

/* program to dump a file */
/* fdump 16-mars-83 jpb/france */

#include <stdio.h>

char   line[80];
char   table[] = "0123456789abcdef";
int    nl, c, i, x;
long   n;

FILE   *infp;

main   (argc, argv)
int    argc;
char   *argv[];

{
    if (argc > 2) {
        printf("Usage : fdump [file]\n");
        exit(1);
    }
    if (argc == 2) {
        if ((infp = fopen(argv[1], "r")) == NULL) {
            printf("Inexisting file \n");
            exit(1);
        }
    }
    if (argc == 1) {
        infp=stdin;
    }
    n = 0;
    nl=24;
    raz();
    while ((c = getc(infp)) != EOF) {
        if (nl == 24) {
            printf(" \n");
            printf("      0 1 2 3 4 5 6 7 8 9 a b c d e f \n");
            printf(" \n");
            nl = 0;
        }
        binhex(c);
        if ( c < 32 || c > 128) {

```

```

        line[x++] = ' ';
    }
    else {
        line[x++] = c;
    }
    i++;
    if (i > 57) {
        printf("%s\n", line);
        n = n + 16;
        nl = nl + 1;
        raz();
    }
}
if (i != 11) {
    printf("%s\n", line);
}
}
/* procedure binhex */

int    binhex(tmp)
{
int    h, l;

    h = tmp/16;
    l = tmp%16;
    line[i++] = table[h];
    line[i++] = table[l];
}
/* procedure raz */

int    raz()

{
int    xl, xml, xmh, xh;
long   nn, nnn;
    for (i = 0; i < 78; i++) {
        line[i] = ' ';
    }
    line[78] = '\0';
    i = 0;
    nn = n/256;
    nnn = nn/256;
    xl = n%256;
    xml = nn%256;
    xmh = nnn%256;
    xh = nnn/256;
    binhex(xh);
    binhex(xmh);
    binhex(xml);
    binhex(xl);
    x = 61;
    i = 11;
}

```

2 KEY programs CT8500 keys directly or via a file. The source follows:

```

#include <stdio.h>

#define LF 10
#define CR 13
#define ESC 27
#define US 31

main (argc, argv)
int   argc;
char  *argv[];
{
    int    i, c;
    char   keyc, string[100];
    keyc = 'G';
    if (argc != 2) {
        fprintf(stderr, "Usage: key {nbre} <cr>\n");
        exit(1);
    }
}

```

```

    }
    sscanf(argv[1], "%d", &i);
    if (i < 1 || i > 8) {
        fprintf(stderr, "Key between (1---8)\n");
        exit(1);
    }
    keyc=keyc+i;
    fprintf(stderr, "? ");
    string[0]=ESC;
    string[1]='0';
    string[2]=ESC;
    string[3]=keyc;
    i=4;
    c=1;
    while((c=getc(stdin)) != '\n') {
        if (c == LF) {
            c=CR;
        }
        string[i++] = c;
    }
    string[i++] = US;
    string[i++] = '\0';
    printf("%s", string);
}

```

- 3 ACCOUNT provides a history of a users login and logout activity. The source follows:

```

/* programm account jpb france 17/2/83*/
#include <stdio.h>

char    console[10], oconsole[10], name[10], oname[10], login[30], logout[30];
int     ident, test, flag, ptr, nl;
long    dh, dmn, cumul, delta, otime, ttime;

FILE    *infp;
main (argc, argv)
int     argc;
char    *argv[];
{
    if (argc != 2) {
        printf("Usage : account user <cr>\n");
        exit(1);
    }
    if ((infp = fopen ("/usr/adm/wtmp", "r")) == NULL){
        printf("File : /usr/adm/wtmp not found\n");
        exit(1);
    }
    printf("\n\n");
    printf("                USER NAME : %s\n\n", argv[1]);
    flag=10;
    cumul=0;
    nl=0;
    test=10;
    delta=0;
    printf("-----\n");
    printf("    LOGIN    |    LOGOUT");
    printf("    | TIME | TOTAL |\n");
    printf("-----\n");
LOOP1: while(flag != NULL) {
        fscanf(infp, "%8s%8s", console, name);
        flag=fread(&ttime, 4, 1, infp);
        if ((ident=strcmp(name, argv[1])) == 0) {
            ptr=ctime(&ttime);
            strcpy(login, ptr);
            login[24]='\0';
            strcpy(oconsole, console);
            otime=ttime;
            while(test != 0) {
                fscanf(infp, "%8s%8s", console, name);
                flag=fread(&ttime, 4, 1, infp);
                test=strncmp(console, oconsole, 4);
                if (test == 0) {

```

```

ptr=ctime(&ttime);
strcpy(logout, ptr);
logout[24]='\0';
test=10;
delta=ttime-otime;
delta=delta/60;
cumul=cumul+delta;
dh = cumul/60;
dmm = cumul%60;
printf(" | %s | %s |%4ld mn |%3ld h%3ld mn | \n",login,logout,delta,dh,dmm);
printf("-----\n");
nl=nl+1;
goto LOOP1;
}
if (flag == NULL) {
    exit(1);
}
}
}
}
if(nl == 0) {
    printf("\n");
    printf("          No information for %s \n",argv[1]);
}
}
}

```

- 4 FMAN is identical to the TNIX man command in function, but the manual content is displayed much faster. The command normally displays a page at a time and inquires for next page displays. The -n option suppresses the prompting for each page. The source follows:

```

/* programme fman jpb 10 / 5 /83 */

#include <stdio.h>
#define TRUE 1
#define FALSE 0
#define LF 10
#define CR 13
#define MAX 23 /* max length of a page*/

char table[] = "0123456789";
char post[] = ".1";
char dir[] = "/usr/man/cat1/";
int q_flag; /* question_flag */
int s_flag; /* short_flag */
int d_flag; /* debug_flag */
char line[100];

FILE *infp;

main (argc, argv)
int argc;
char *argv[];
{
    char file[100];
    int test, x, flag;
    char *pname;
    test = 0;
    x = 1;
    s_flag = FALSE;
    q_flag = TRUE;
    d_flag = FALSE;
    /*
     * examine the switches
     */
    pname = *argv++;
    while (*argv[0] == '.') {
        while ( *(++*argv) ) {
            switch (**argv) {
                case 'd':
                    d_flag = TRUE;
                    break;
                case 's':

```

```

        s_flag = TRUE;
        break;
    case 'n':
        q_flag = FALSE;
        break;
    default:
        printf("Usage : fman [-nsd] [chapter] {command}\n");
        exit(1);
    }
}
argv++;
argc--;
}

if (argc < 2 || argc > 3) {
    printf("Usage : fman [-nsd] [chapter] {command}\n");
    exit(1);
}

if (argc == 3) {
    sscanf(argv[0], "%d", &x);
    dir[12] = table[x];
    post[1] = table[x];
    strcpy(file, dir);
    strcat(file, argv[1]);
    strcat(file, post);
    if (d_flag == TRUE) {
        printf("Seeked file : %s\n", file);
    }
    if ((infp = fopen(file, "r")) != NULL) {
        flag = display();
        exit(1);
    } else {
        printf("\n$Command not found. \n");
        exit(1);
    }
}

for (x = 1 ; x < 10 ; x++) {
    dir[12] = table[x];
    post[1] = table[x];
    strcpy(file, dir);
    strcat(file, argv[0]);
    strcat(file, post);
    if (d_flag == TRUE) {
        printf("Seeked file : %s\n", file);
    }
    if ((infp = fopen(file, "r")) != NULL) {
        test = 1;
        if (s_flag == TRUE) {
            fgets(line, 80, infp);
            fgets(line, 80, infp);
            fgets(line, 80, infp);
        }
        flag = display();
        flag = fclose(infp);
    }
}

if (test == 0) {
    printf("\n$Command not known. \n");
}

}

/*
 * sous programme display
 */
int display()
{
    char *ptr;
    int c, nl;
    nl = 0;
    printf("\n");
    while ((ptr = fgets(line, 80, infp)) != NULL) {
        if (line[0] == 'D' && s_flag == TRUE) {
            exit(1);
        }
    }
}

```

```

    }
    printf("%s", line);
    nl++;
    if (nl == MAX && q_flag == TRUE) {
        printf("\n$[Enter CR to continue] ");
        c = getchar();
        printf("R          \r");
        nl = 0;
    }
}
printf("\n");
return(1);
}

```

Jean Paul Bonsang Tek France

TNIX CONFIGURABILITY - WHEN AND WHERE

Since the advent of TNIX 1.4 and 1 mbyte memory for the 856X, TNIX configurability has not been clearly understood by users. The TNIX manual "8560 Series System Manager's operational and installation guide" (Part No. 070-0211-00) for TNIX 2.0 provides an excellent description of TNIX configurability.

Earlier versions of TNIX did not provide TNIX buffer configurability. With the introduction of larger memory, GPIB Tape Controller, Expansion disks, and the 8561, a default configuration could not meet all needs.

The "cvt" command can be invoked by the system superuser (root) or with standalone utilities to provide the configurability needed. The following is a brief summary of when to use cvt, and how to use it.

The default TNIX configuration, as supplied, is satisfactory for most needs. The only need to change TNIX via cvt would occur when:

1. an 8560(61) is restored with a set of dump disks from a 8561(60) or
2. a 1/2 mbyte or 1 mbyte memory is installed and a large number of processes will be created or
3. TNIX file or inode errors occur or
4. several disk intensive jobs running or
5. a number of disk intensive jobs run much slower than expected or
6. no apparent gain in system performance with increased memory.

If a dump series is used from an 8560(61) to restore an 8561(60), the standalone version of cvt will be required to set swplo to 65000(8560) or 24360(8561) and nswap to 4500(8560) or 2250(8561). TNIX will now be able to run. Using a 8560 dump series to restore an 8561, even when smaller than 13 disks, may not work at all. The TNIX restore disks provided with the 8561 is a 8560 dump disk set; thus restoring an 8561 with the supplied disks will require the above step before the system will boot up.

If any of the other conditions are met, then the following is recommended as a starting point.

mount	1	This must be done first to make room for the following.
buffer	30	
inode	80	
file	96	
text	35	
swapmap	90	
coremap	120	

The following relationships must be observed:

- File must be greater than inode and
- Swapmap must be greater than or equal to proc and
- Proc must be greater than or equal to text and
- mount must be greater than or equal to 1.

To make the above changes, use the cvt option "-w". Note: cvt does not prompt the user. The current settings can be viewed by entering an "*". A control D will terminate the cvt command. The system will have to be restarted to use the new settings.

John Owens MDP Technical Support Manager

INCLUDING MATH LIBRARIES IN C PROGRAMS

Including math libraries in C programs is not automatic with the normal cc invocation. To gain access to the math functions, use the -lm option at the end of the command line. The following example shows how to compile and run a short example using a square root function.

C program 'sqrt.c'

```
#include <math.h>
main() {
    float i=2.0;
    printf(" sqrt(2.0) = %f0,sqrt(i));
}
```

Compilation:

```
cc sqrt.c -lm
```

Execution:

```
$ sqrt
sqrt(2.0) = 1.414214 $
```

By the way, the -lm option is not documented under "cc" because it is a loader flag. Use "man ld" or see "ld" in section 6 of the TNIX System Reference Manual.

Greg Saville MDP Technical Support Manager

EXTENDING THE LINKER SYMBOL CAPABILITY

Several methods of effectively extending the global symbol table capability of the linker are presented here. "Module Grouping" and "Data and Declarations Grouping" are the simplest methods to implement. The "Section Names" method will only provide a limited improvement, but is easy to implement. "Separation of Load Files" is the most effective method provided, but requires significant effort to implement.

Module Grouping

Typically a large number of modules are generated to describe an executable program. If the modules which primarily reference each other are grouped together, the number of globals can be significantly reduced without having to combine the modules in a common file or sacrifice code modularity. That is, the symbols which are not referenced outside the collected group of modules should not be global. The modules can remain separate and are referenced for assembly in a new (global) module which is simply a series of "include" directives.

Such a file might look something like:

```
...  
include      IO1.src  
include      IO2.src  
include      IOdata.src  
...
```

This is functionally equivalent to concatenating all the grouped files into a common file or entering an assembler command which lists all the group source files on the command line. Such as:

```
asm object list IO1.src IO2.src IOdata.src ...
```

The module grouping method will reduce the number of globals and filenames which the linker must store, provide better documentation, and minimize the file space required.

Data and Declarations Grouping

Often when organizing a program into manageable units (modules), it is logical to place data and location declarations in a common module(s). This is the case especially when the assignments relate to the configuration of the hardware and memory organization which are usually the most subject to change. Data structures which are referenced from many locations in the program are also combined into a module(s) for accessibility and maintainability. Generally these declarations are referenced from many locations in the code, thus they are declared global.

The assembler allows the declaration of a "common" section. If the above types of declarations were placed in a "common" type section and referenced by each module that needed access to the values via the "include" directive, the labels would not be global. When using include to reference a file more than once, the file cannot contain global directives. The linker assumes that all "common" sections of the same name are, in reality, multiple descriptions of one logical section as in the common directive in Fortran.

The use of this assembler feature will greatly reduce the number of globals required.

Section Names

Often the separation of code between modules is made on the basis of the length of the code rather than the function provided by the code. If this is the case, the modules which contain a single logical function or collections of functions can use the same section name in multiple modules and at link time the sections will be concatenated in the order that the linker command file specifies the file names. **Caution:** The value associated with the section name will be the address of the first of the concatenated sections. Avoid using the section name as a label in the modules which contain portions of the section's descriptions (code).

Section names are also global variables; thus, multiple sections with the same name will conserve global symbol space.

Separation of Load Files

If the modules generated can be divided into satellite groups of modules that are collectively standalone, and if the code contained in each group is primarily called or referenced by a core group, each satellite group and the core group may be linked independently. The resulting load files are independently loaded, thus the programmer must locate the groups carefully in order that gaps and overlaps in code are not generated.

Three link passes are required to build the final load files. The first pass of each group is used to establish the size and location of its sections. The second pass using the -r option is used to locate the sections. The third pass is used to link each of the groups using define directives to resolve the globals that remain undefined.

The resulting load files can be loaded in any sequence.

John Owens MDP Technical Support Manager

USE OF PRINTER CAPABILITIES WITH NROFF

The files contained in /usr/lib/term contain specific printer implementations that can be used with nroff. All too often, those provided are not suited to the printer that you may have available. The following source with comments will provide the foundation for developing a configuration for other printers.

The following is an expanded source file and procedure to install a "taba" file for a Florida Data printer 120/130. The "Native Programming Tools" software must be available to compile the source provided.

Two files are required:

- 1 A file (tabFD.c) which contains the structure definition of the modes of operation, and the character set followed by the code that will cause the printer to assume the selected modes, and an include reference to the second file.
- 2 A file (code.FD) that contains the code required to cause the printer to print the nroff supported character set.

```
#define INCH 240
/*
FLORIDA DATA 120/130
nroff driving tables
width and code tables
*/

struct {
    int termtyp; /*TEK*/
    int bset;
    int breset;
    int Hor;
    int Vert;
    int Newline;
    int Char;
    int Em;
    int Halfline;
    int Adj;
    char *twinit;
    char *twrest;
    char *twnl;
    char *h1r;
    char *h1f;
    char *f1r;
    char *bdon;
    char *bdoff;
    char *ulon; /*TEK*/
    char *uloff; /*TEK*/
    char *plon;
    char *plotoff;
    char *up;
    char *down;
    char *right;
    char *left;
    char *codetab[256-32];
    int sss;
} t = {
/*termtyp*/ 100, /*TEK - 100 means "not specially supported"*/
/*bset*/ 0, /* binary value used to set(mask) the port mode */
/*breset*/ 0, /* user to reset the port mode at end of printing */
/*Hor*/ INCH/60,
/*Vert*/ INCH/48,
/*Newline*/ INCH/6,
/*Char*/ INCH/12,
/*Em*/ INCH/12,
/*Halfline*/ INCH/12,
/*Adj*/ INCH/12,
/*twinit*/ "\0331 \0331 \0331 \0331", /* Initialize the printer for text writing/
/*twrest*/ "\014\033\015P", /* reset the printer after completion of t#
```

```

/*twn!*/      "\015\n", /* text new line */
/*h1r*/      "\033D", /* .move 1/2 line up toward top of page */
/*h1f*/      "\033U", /* move 1/2 line down on page */
/*flr*/      "\033\n", /* move one line toward the top of page */
/*bdon*/     "\033O", /* start bold mode printing */
/*bdoff*/    "\033&", /* terminate bold mode printing */
/*ulon*/     "\033E", /* start underline mode printing */
/*uloff*/    "\033R", /* terminate underline mode printing */
/*ploton*/   "\0333", /* start printer plot(graphics) mode */
/*plotoff*/  "\0334", /* terminate graphics mode */
/*up*/       "\033\n", /* move up one line */
/*down*/     "\n", /* move down a line */
/*right*/    "\b", /* move one position right */
/*left*/     "\b", /* move one position back */
/*codetab*/
#include "code.FD" /* Char. set */
}
    
```

The second file referenced above as "code.FD" follows. Note that the printer is capable of more than is described here. However, the files as shown (in two column format) are sufficient to meet our needs.

```

"\001 ", /*space*/
"\001!", /*!*/
"\001\"", /*"*/
"\001#", /*#*/
"\001$", /*$*/
"\001%", /*%*/
"\001&", /*&*/
"\001'", /*' close*/
"\001(", /*(*/
"\001)", /*)*/
"\001+", /*+*/
"\001,", /*,*/
"\001-", /*- hyphen*/
"\001.", /*.*/
"\001/", /*/*/
"\2010", /*0*/
"\2011", /*1*/
"\2012", /*2*/
"\2013", /*3*/
"\2014", /*4*/
"\2015", /*5*/
"\2016", /*6*/
"\2017", /*7*/
"\2018", /*8*/
"\2019", /*9*/
"\001.", /*.*/
"\001<", /*<*/
"\001=", /*=*/
"\001>", /*>*/
"\001?", /*?*/
"\001@", /*@*/
"\201A", /*A*/
"\201B", /*B*/
"\201C", /*C*/
"\201D", /*D*/
"\201E", /*E*/
"\201F", /*F*/
"\201G", /*G*/
"\201H", /*H*/
"\201I", /*I*/
"\201J", /*J*/
"\201K", /*K*/
"\201L", /*L*/
"\201M", /*M*/
"\201N", /*N*/
"\201O", /*O*/
"\201P", /*P*/
"\201Q", /*Q*/
"\201R", /*R*/
"\201S", /*S*/
"\201T", /*T*/
"\201U", /*U*/
"\201V", /*V*/
"\201W", /*W*/
"\201X", /*X*/
"\201Y", /*Y*/
"\201Z", /*Z*/
"\001|", /*|*/
"\001||", /*||*/
"\001]", /*]*/
"\001^", /*^*/
"\001_", /*_ dash*/
"\001`", /*` open*/
"\201a", /*a*/
"\201b", /*b*/
"\201c", /*c*/
"\201d", /*d*/
"\201e", /*e*/
"\201f", /*f*/
"\201g", /*g*/
"\201h", /*h*/
"\201i", /*i*/
"\201j", /*j*/
"\201k", /*k*/
"\201l", /*l*/
"\201m", /*m*/
"\201n", /*n*/
"\201o", /*o*/
"\201p", /*p*/
"\201q", /*q*/
"\201r", /*r*/
"\201s", /*s*/
"\201t", /*t*/
"\201u", /*u*/
"\201v", /*v*/
"\201w", /*w*/
"\201x", /*x*/
"\201y", /*y*/
"\201z", /*z*/
"\001{", /*{*/
"\001}", /*}*/
"\001~", /*~*/
"\000\0", /*narrow sp*/
"\001.", /*hyphen*/
"\001\033W\033&", /*bullet*/
    
```

```

"002\|",          /*square*/
"002--",         /*3/4 em*/
"001_",         /*rule*/
"0031/4",       /*1/4*/
"0031/2",       /*1/2*/
"0033/4",       /*3/4*/
"001-",         /*minus*/
"202f",         /*fi*/
"202ff",        /*ffi*/
"202fff",       /*fffi*/
"203fff",       /*ffffi*/
"001\033Do\033U", /*degree*/
"001\|b-",      /*dagger*/
"001\|bo",      /*section*/
"001'",         /*foot mark*/
"001'",         /*acute accent*/
"001'",         /*grave accent*/
"001_",         /*underrule*/
"001'",         /*slash (longer)*/
"000\0",        /*half narrow space*/
"001 ",         /*unpaddable space*/
"000\0",        /*alpha*/
"000\0",        /*beta*/
"000\0",        /*gamma*/
"000\0",        /*delta*/
"001<\|b-",     /*epsilon*/
"000\0",        /*zeta*/
"000\0",        /*eta*/
"001O\|b-",    /*theta*/
"001i",         /*iota*/
"001k",         /*kappa*/
"000\0",        /*lambda*/
"000\0",        /*mu*/
"000\0",        /*nu*/
"000\0",        /*xi*/
"001o",         /*omicron*/
"000\0",        /*pi*/
"000\0",        /*rho*/
"000\0",        /*sigma*/
"000\0",        /*tau*/
"001v",         /*upsilon*/
"001o\|b",     /*phi*/
"001x",         /*chi*/
"e000/0",      /*psi*/
"000\0",        /*omega*/
"000\0",        /*Gamma*/
"000\0",        /*Delta*/
"001O\|b=",    /*Theta*/
"000\0",        /*Lambda*/
"001\|b",      /*Xi*/
"000\0",        /*Pi*/
"000\0",        /*Sigma*/
"000\0",        /****/
"001Y",         /*Upsilon*/
"001o\|b\|b", /*Phi*/
"000\0",        /*Psi*/
"000\0",        /*Omega*/
"000\0",        /*square root*/
"000\0",        /*terminal sigma*/
"000\0",        /*root en*/
"001>\|b-",    /*>=*/
"001<\|b-",    /*<=*/
"001=\|b-",    /*identically equal*/
"001-",         /*equation minus*/
"001=\|b-",    /*approx =*/
"000\0",        /*approximates*/
"001=\|b",     /*not equal*/
"002->",       /*right arrow*/
"002<-",       /*left arrow*/
"001\|b^",     /*up arrow*/
"000\0",        /*down arrow*/
"001=",         /*equation equal*/
"001x",         /*multiply*/
"001/",         /*divide*/
"001+\|b-",    /*plus-minus*/
"001U",         /*cup (union)*/
"000\0",        /*cap (intersection)*/
"000\0",        /*subset of*/
"000\0",        /*superset of*/
"000\0",        /*improper subset*/
"000\0",        /*improper superset*/
"002oo",        /*infinity*/
"000\0",        /*partial derivatives*/
"000\0",        /*gradient*/
"000\0",        /*not*/
"000\0",        /*integral sign*/
"000\0",        /*proportional to*/
"000\0",        /*empty set*/
"000\0",        /*angstrom*/ /*TEK - new*/
"001+",         /*equation plus*/
"001r\|bo",    /*registered*/
"001c\|bo",    /*copyright*/
"001|",         /*box rule*/
"001c\|b",     /*cent sign*/
"001\|b=",     /*dbl dagger*/
"002=>",       /*right hand*/
"002=<=",      /*left hand*/
"001*",         /*math */
"000\0",        /*bell system sign*/
"001*",         /*or (was star)*/
"001O",         /*circle*/
"001",          /*left top (of big curly)*/
"001",          /*left bottom*/
"001",          /*right top*/
"001",          /*right bot*/
"001",          /*left center of big curly bracket*/
"001",          /*right center of big curly bracket*/
"001",          /*bold vertical*/
"001",          /*left floor (left bot of big sq bract)*/
"001",          /*right floor (rb of )*/
"001",          /*left ceiling (lt of )*/
"001",          /*right ceiling (rt of )*/
"001",          /*TEK - paragraph sign*/
"001",          /*TEK - trademark sign*/
"001",          /*TEK - double underline*/
"001^",         /*TEK - logical and*/
"001v",         /*TEK - logical or*/
"000"};        /*TEK - tall line character*/
    
```

After modifying the files to reflect the capabilities of your printer, compile the first file.

```
cc -s -n -o tabFD.c -O tabaFD
```

The file should then be located in /usr/lib/term. Note, if you use the lpXr command to print the file the printer daemon will remove most control characters. Thus, until a new daemon (lpd) is available, the file must be sent to the device.

```
cat out.file >/dev/lpX
```

The command to create the output file would be:

```
nroff -ms -TFD <ms.file >out.file
```

John Owens MDP Technical Support Manager

PDB FOR PAYMENT LEARNING GUIDE DEMONSTRATION RUN

The following listings have been prepared which contain working files for Z8000, 68000, and 8086 payment.ps under PDB. There are some confusing and erroneous things in the manual, but if a little time is spent working through the appropriate example with the assistance of these notes, you should be able run them successfully.

Z8001

These listings contain a working set of files to run the Z8001 payment.ps pdb learning guide demo. Note that a new .is and .ic file are used to enable pdb to run correctly. To run the demo, invoke the file setup by: . setup <cr> to initialize. Next run the command file payment.cmd to compile, ics, link, and pdb the program. The file payment.log shows the working output of the pdb execution.

Notes:

The recommended configuration is terminal-8540-8560. If using the 8540 remotely, a terminal is still required on the 8540 to run this program.

The z8K library requires entering a decimal point for real number input, so you must enter 500.0 15.0 12 instead of 500 15 12 as the manual shows.

The modified /lib/z8001/sdefault.is which must be used is called payment.is for ics and payment.ic for the linkage. See the payment.cmd file for proper invocation of the compiler, ics, link, and pdb.

Z8K pdb does not currently work on an 8550 in term mode with the 8560.

setup - initialization command file, modify as required

```
: z8001 Pascal setup
: execute this by: '. setup <cr>'
uP=z8001
TERM=ct8500
LDECONFIG=/usr/lib/lde/lde.$TERM.cfg
export uP TERM LDECONFIG
set $uP
al 0 0fff
reset
```

payment.cmd - compile, ics, link, and pdb command file

```
set -v
: z8001 payment.ps demo run
: assume setup run already
pas -dvslo payment.po payment.ps >payment.pl
ics -olv payment.is
```

```
link -d -o payment.lo -c payment.ic
pdb -v payment.lo
```

payment.is - ics source file

```
PASCAL_CONFIGURATION      payment pdb test
HARDWARE_CONFIGURATION   Z8001, SEGMENTED
EXECUTION_MODE           SYSTEM, SYMBOLIC_DEBUG
EMULATOR_ADDRESS_SPACES SEPARATE
INSTRUCTIONS_ROM         [00012H, 0003FH], [00060H, 07FFFH]
CONSTANTS_ROM            [08000H, 08FFFH]
GLOBAL_VAR_RAM           [09000H, 0D7FFH]
HEAP_STACK_RAM           [0D800H, 0FFFFH]
RESET_MEMORY             YES
DYNAMIC_RAM              NONE
SERVICE_CALLS           40H, 0FFF0H
SOFTWARE_CONFIGURATION   /usr/gregs/Pascal/Z8001/Payment/payment.po
MODULE                   NONE
LIBRARY                  NONE
FLOATING_POINT_SUPPORT   YES
FILE_SUPPORT             DEFAULT
ERROR_HANDLING           DEFAULT
INTERRUPT_CONFIGURATION UNIMPL_INSTR_TRAP, SYSTEM_CALL_TRAP
FAULT_NOTIFICATION       PASCAL_RTS
RESTART_LABEL            PASCAL_BEGIN
END
```

payment.ic - linker command file

```
-O payment.io
-m INSTRQQ.ROM=04000012H-0400003FH 04000060H-04007FFFH
-m CONSTQQ.ROM=01008000H-01008FFFH
-m SRBVQQ.RAM=01000040H-0100005FH
-m DATAQQ.RAM=01009000H-0100D7FFH
-m ZZRESETVQQ.ROM=04000002H-04000007H
-L class=INSTRQQ range INSTRQQ.ROM
-L class=CONSTQQ range CONSTQQ.ROM
-L class=SRBVQQ range SRBVQQ.RAM
-L class=DATAQQ range DATAQQ.RAM
-L class=ZZRESETVQQ range ZZRESETVQQ.ROM
-D HEAPBASEQQ=0100D800H
-D STKBASEQQ=0100FFFFH
-D SVCLOCZZ=0FFF0H
-O /usr/gregs/Pascal/Z8001/Payment/payment.po
-O /lib/z8000/pas.conv.segm
-O /lib/z8000/pas.hiio.segm
-O /lib/z8000/pas.fpsimr4.segm
-O /lib/z8000/pas.rts.segm
-O /lib/z8000/pas.err.segm
-O /lib/z8000/pas.rts.segm
-O /lib/z8000/pas.err.segm
-O /lib/z8000/pas.posi.segm
-O /lib/z8000/pas.conv.segm
-x PASCAL_BEGIN
```

payment.log - program execution log

start log: Sat Jul 9 13:43:33 PDT 1983

load "payment.lo"

go

Enter loan_amount interest_rate and number_of_payments

500.0 15.0 12

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	0.56	41.55	45.13
13	0.00	0.00	0.00	41.55	0.00

Program terminated

reset

break payment#25 after 12

break

```

Name Usage Kind Address
SW1: Used X PAYMENT#25 SC:001660 after 12
HW1: Free R/W
HW2: Free R/W
TTA1: Free R/W
TTA2: Free R/W
TTA3: Free R/W
TTA4: Free R/W

```

go

Enter loan_amount interest_rate and number_of_payments

500.0 15.0 12

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	0.56	41.55	45.13

SW1: Used X PAYMENT#25 SC:001660

principal

PRINCIPAL = 1.991271E-03

log

stop log: Sat Jul 9 13:50:25 PDT 1983

Z8002

These listings contain a working set of files to run the Z8002 payment.ps pdb learning guide demo. Note that a new -.is and -.ic file are used to enable pdb to run correctly. To run the demo, invoke the file setup by: . setup <cr> to initialize. Next run the command file payment.cmd to compile, ics, link, and pdb the program. The file payment.log shows the working output of the pdb execution.

Notes:

The recommended configuration is terminal-8540-8560. If using the 8540 remotely, a terminal is still required on the 8540 to run this program.

The z8K library requires entering a decimal point for real number input, so 500.0 15.0 12 must be entered instead of 500 15 12 as the manual shows.

The modified /lib/z8002/sdefault.is which must be used is called payment.is for ics and payment.ic for the linkage. See the payment.cmd file for proper invocation of the compiler, ics, link, and pdb.

Z8K pdb does not currently work on an 8550 in term mode with the 8560.

setup - initialization command file, modify as required

```
: z8002 Pascal setup
: execute this by: ' . setup <cr> '
uP=z8002
TERM=ct8500
LDECONFIG=/usr/lib/lde/lde.$TERM.cfg
export uP TERM LDECONFIG
sel $uP
al 0 0fff
reset
```

payment.cmd - compile, ics, link, and pdb run command file

```
set -v
: z8002 payment.ps demo run
: assume setup run already
pas -dvslo payment.po payment.ps >payment.pl
ics -olv payment.is
link -d -o payment.lo -c payment.ic
pdb -v payment.lo
```

payment.is - ics source file

```
PASCAL_CONFIGURATION      payment pdb test
HARDWARE_CONFIGURATION    Z8002
EXECUTION_MODE            SYSTEM,SYMBOLIC_DEBUG
EMULATOR_ADDRESS_SPACES  SEPARATE
INSTRUCTIONS_ROM          [00012H,0003FH] , [00060H,07FFFH]
CONSTANTS_ROM             [08000H,08FFFH]
GLOBAL_VAR_RAM            [09000H,0D7FFFH]
HEAP_STACK_RAM           [0D800H,0FFFFH]
RESET_MEMORY              YES
DYNAMIC_RAM               NONE
SERVICE_CALLS            40H,0FFF0H
SOFTWARE_CONFIGURATION    /usr/gregs/Pascal/Z8002/Payment/payment.po
MODULE                    NONE
LIBRARY                   NONE
FLOATING_POINT_SUPPORT    YES
FILE_SUPPORT              DEFAULT
ERROR_HANDLING            DEFAULT
INTERRUPT_CONFIGURATION   UNIMPL_INSTR_TRAP,SYSTEM_CALL_TRAP
```

```

FAULT_NOTIFICATION      PASCAL_RTS
RESTART_LABEL           PASCAL_BEGIN
END

```

payment.ic - linker command file

```

-O payment.io
-m INSTRQQ.ROM=04000012H-0400003FH 04000060H-04007FFFH
-m CONSTQQ.ROM=01008000H-01008FFFH
-m SRBVQQ.RAM=01000040H-0100005FH
-m DATAQQ.RAM=01009000H-0100D7FFFH
-m ZZRESETVQQ.ROM=04000002H-04000005H
-L class=INSTRQQ range INSTRQQ.ROM
-L class=CONSTQQ range CONSTQQ.ROM
-L class=SRBVQQ range SRBVQQ.RAM
-L class=DATAQQ range DATAQQ.RAM
-L class=ZZRESETVQQ range ZZRESETVQQ.ROM
-D HEAPBASEQQ=0100D800H
-D STKBASEQQ=0100FFFFH
-D SVCLOCZZ=0FFF0H
-O /usr/gregs/Pascal/Z8002/Payment/payment.po
-O /lib/z8000/pas.conv.nseg
-O /lib/z8000/pas.hio.nseg
-O /lib/z8000/pas.fpsimr4.nseg
-O /lib/z8000/pas.rts.nseg
-O /lib/z8000/pas.err.nseg
-O /lib/z8000/pas.rts.nseg
-O /lib/z8000/pas.err.nseg
-O /lib/z8000/pas.posi.nseg
-O /lib/z8000/pas.conv.nseg
-x PASCAL_BEGIN

```

payment.log - program execution log

```

start log: Sat Jul 9 16:58:57 PDT 1983
# load "payment.lo"
# go
Enter loan_amount interest_rate and number_of_payments
500.0 15.0 12

```

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	0.56	41.55	45.13
13	0.00	0.00	0.00	41.55	0.00

```

Program terminated
# break payment#25 after 12
# break
Name Usage Kind Address
SW1: Used X PAYMENT#25 SC:0013BA after 12

```

```

HW1: Free R/W
HW2: Free R/W
TTA1: Free R/W
TTA2: Free R/W
TTA3: Free R/W
TTA4: Free R/W

```

```
# reset
```

```
# go
```

```
Enter loan_amount interest_rate and number_of_payments
```

```
500.0 15.0 12
```

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	Breakpoint Encountered:		

```
SW1: Used X PAYMENT#25 SC:0013BA
```

```
# principal
```

```
PRINCIPAL = 1.991271E-03
```

```
# exit
```

```
MDP Pascal Z8000 debug exiting
```

68000

These listings contain a working set of files to run the 68000 payment.ps pdb learning guide demo. Note that a new `-.is` and `-.ic` file are used to enable pdb to run correctly. To run the demo, invoke the file setup by: `. setup <cr>` to initialize. Next run the command file `payment.cmd` to compile, ics, link, and pdb the program. The file `payment.log` shows the working output of the pdb execution.

Notes:

The recommended configuration is terminal-8540-8560. If using the 8540 remotely, a terminal is still required on the 8540 to run this program.

The 68000 library requires entering a decimal point for real number input, so `500.0 15.0 12` must be entered instead of `500 15 12` as the manual shows.

The modified `/lib/68000/default.is` which must be used is called `payment.is` for ics and `payment.ic` for the linkage. See the `payment.cmd` file for proper invocation of the compiler, ics, link, and pdb.

68K pdb can run with an 8550 or 8540 in term mode to the 8560.

setup - initialization command file, modify as required

```
: 68000 pascal setup command file
```

```
: execute this by: '. setup <cr>'
```

```
LDECONFIG=/usr/lib/lde/lde.ct8500.cfg
```

```
TERM=ct8500
```

```
uP=68000
```

```
export uP TERM LDECONFIG
```

```
sel $uP
```

payment.cmd - compile, ics, link, and run command file

```

set -v
: payment.cmd - 68000 learning guide demo run
: assume setup run already
: run compilation..
pas -dvslo payment.po payment.ps >payment.pl
: run ics..
ics -lv payment.is
: perform link...
link -d -o payment.lo -c payment.ic
: run pdb via payment.ie
payment.ie

```

payment.is - ics source file

```

PASCAL_CONFIGURATION      payment - pdb learning guide demo
HARDWARE_CONFIGURATION    68000
PRIVILEGE_STATE           SUPERVISOR,PASCAL_DEBUG
EMULATOR_ADDRESS_SPACES  CONTIGUOUS
INSTRUCTIONS_ROM          00400H-07FFFH
CONSTANTS_ROM             08000H-08FFFH
GLOBAL_VAR_RAM            09000H-0D7FFH
HEAP_STACK_RAM           0D800H-0FFCFH
RESET_MEMORY              YES
SERVICE_CALLS            0FFD0H,0FFF0H
SOFTWARE_CONFIGURATION    /usr/gregs/Pascal/68000/Payment/payment.po
MODULE                    NONE
LIBRARY                   NONE
FLOATING_POINT_SUPPORT    PASCAL_RTS
FILE_SUPPORT              PASCAL_RTS
ERROR_HANDLING            PASCAL_RTS
INTERRUPT_CONFIGURATION   ROM,HIGHEST_VECTOR=47
INTERRUPT_PROCEDURE       NONE
FAULT_NOTIFICATION        PASCAL_RTS, STOP
INITIALIZATION_CODE       PASCAL_BEGIN
END

```

payment.ic - linker command file

```

-d
-O payment.io
-o payment.lo
-O /usr/gregs/Pascal/68000/Payment/payment.po
-O /lib/68000/pas.conv
-O /lib/68000/pas.hio
-O /lib/68000/pas.fpsimr4
-O /lib/68000/pas.rts
-O /lib/68000/pas.err
-O /lib/68000/pas.rts
-O /lib/68000/pas.err
-O /lib/68000/pas.posi
-m RESETVQQ.ROM=08000000H-08000007H
-m VECTRQQ.ROM=04000008H-040000BFH
-m INSTRQQ.ROM=08000400H-08007FFFH
-m CONSTQQ.ROM=04008000H-04008FFFH
-m DATAQQ.RAM=04009000H-0400D7FFH
-m SRBVQQ.RAM=0400FFD0H-0400FFEFH
-L class=RESETVQQ range RESETVQQ.ROM
-L class=VECTRQQ range VECTRQQ.ROM

```

```
-L class=INSTRQQ range INSTRQQ.ROM
-L class=CONSTQQ range CONSTQQ.ROM
-L class=DATAQQ range DATAQQ.RAM
-L class=SRBVQQ range SRBVQQ.RAM
-D HEAPBASEQQ=0400D800H
-D STKBASEQQ=0400FFCCH
-D SVCLOCZZ=0400FFF0H
-x PASCAL_BEGIN
```

payment.ie - emulator setup and download command file

```
cat << eof
Preparing emulator and downloading...
eof
set -x
sel 68000
  if test $? != 0; then exit 1; fi
s sr=2700
  if test $? != 0; then exit 1; fi
a -a
  if test $? != 0; then exit 1; fi
svc on, OFFD0H, OFFF0H
  if test $? != 0; then exit 1; fi
if al > $$; then
  al 00H OFFFFH
  if test $? != 0; then exit 1; fi
fi
rm -f $$
rm -f /tmp/8540$IU
s ssp=OFFCCH
  if test $? != 0; then exit 1; fi
pdb payment.lo
```

payment.log - program execution log

```
start log: Sun Jul 10 12:45:34 PDT 1983
# load "payment.lo"
# go
Enter loan_amount interest_rate and number_of_payments
500.0 15.0 12
```

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	0.56	41.55	45.13
13	0.00	0.00	0.00	41.55	0.00

```
Program terminated
# reset
# break payment#25 after 12
# break
```

```
Name Usage Kind Address
SW1: Used X PAYMENT#25 SP:001D86 after 12
HW1: Free R/W
HW2: Free R/W
HW3: Free R/W
TTA1: Free R/W
TTA2: Free R/W
TTA3: Free R/W
TTA4: Free R/W
```

```
# go
```

```
Enter loan_amount interest_rate and number_of_payments
```

```
500.0 15.0 12
```

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	Breakpoint Encountered:		

```
SW1: Used X PAYMENT#25 SP:001D86
```

```
# principal
```

```
PRINCIPAL = 1.991271E-03
```

```
# tb
```

```
$0: PAYMENT SP:001D86 at statement number 25
```

```
# type payment
```

```
program (statements 2 .. 25)
```

```
var
```

```
INPUT: TEXT
```

```
OUTPUT: TEXT
```

```
PRINCIPAL: REAL
```

```
INTRATE: REAL
```

```
PAYONPRIN: REAL
```

```
PAYMENT: REAL
```

```
INTEREST: REAL
```

```
ACCUMINT: REAL
```

```
NPAY: INTEGER
```

```
PAYNUM: INTEGER
```

```
procedures
```

```
POWER: function returning REAL (statement 1)
```

```
# exit
```

8086

These listings contain a working set of files to run the 8086 payment.ps pdb learning guide demo. Note that a new -.is and -.ic file are used to enable pdb to run correctly. To run the demo, invoke the file setup by: . setup <cr> to initialize. Next run the command file payment.cmd to compile, ics, link, and pdb the program. The file payment.log shows the working output of the pdb execution.

Notes:

The recommended configuration is terminal-8540-8560. If using the 8540 remotely, a terminal is still required on the 8540 to run this program.

8086 pdb will also work with an 8550 in term mode to the 8560 when the next version is released.

The modified /lib/8086/default.is which must be used is called payment.is for ics and payment.ic for the linkage. See the payment.cmd file for proper invocation of the compiler, ics, link, and pdb.

setup - initialization command file, modify as required

```
: execute this by: '. setup <cr> '
LDECONFIG=/usr/lib/lde/lde.ct8500.cfg
TERM=ct8500
uP=8086
export uP TERM LDECONFIG
set $uP
al 0 0fff
```

payment.cmd - compile, ics, link, and pdb command file

```
: assume IU,uP setup
set -v
pas -dvslo payment.po payment.ps >payment.pl
ics -olv payment.is
link -d -o payment.lo -c payment.ic
pdb payment.lo
```

payment.is - ics source file

```
PASCAL_CONFIGURATION      Default Configuration
HARDWARE_CONFIGURATION    8086
INSTRUCTIONS_ROM          [00060H,07FFFH]
CONSTANTS_ROM             [08000H,08FFFH]
GLOBAL_VAR_RAM            [09000H,0D7FFFH]
HEAP_STACK_RAM            [0D800H,0FFFFH]
RESET_MEMORY              NO
SERVICE_CALLS            DEFAULT
SOFTWARE_CONFIGURATION    /usr/gregs/Pascal/8086/Payment/payment.po
MODULE                    NONE
LIBRARY                   NONE
FILE_SUPPORT              DEFAULT
INTERRUPT_CONFIGURATION  NONE
RESTART_LABEL             PASCAL_BEGIN
END
```

payment.ic - linker command file

```
-O payment.io
-D STKBASQQ=0FFFFH
-D HEAPBASQQ=0D800H
-D SVCLOCZZ=0FFF0H
-m INSTRQQ.ROM=060H-07FFFH
-m CONSTQQ.ROM=08000H-08FFFH
-m DATAQQ.RAM=09000H-0D7FFFH
-m SRBVQQ.RAM=040H-05FH
-L class=INSTRQQ range INSTRQQ.ROM
-L class=CONSTQQ range CONSTQQ.ROM
-L class=DATAQQ range DATAQQ.RAM
-L class=SRBVQQ range SRBVQQ.RAM
```

```
-O /usr/gregs/Pascal/8086/Payment/payment.po
-D CODEBASEQQ=060H
-D DATABASEQQ=08000H
-O /lib/8086/pas.hio.scsd
-O /lib/8086/pas.fp86.scsd
-O /lib/8086/pas.rts.scsd
-O /lib/8086/pas.err.scsd
-O /lib/8086/pas.posi.scsd
-O /lib/8086/pas.conv.scsd
-x PASCAL_BEGIN
```

payment.log - program execution log

start log: Wed Jul 20 16:44:55 PDT 1983

load "payment.lo"

go

Enter loan_amount interest_rate and number_of_payments

500 15 12

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	0.56	41.55	45.13
13	0.00	0.00	0.00	41.55	0.00

Program terminated

reset

break payment#25 after 12

break

Name Usage Kind Address

SW1: Used X PAYMENT#25 521 after 12

HW1: Free R/W

HW2: Free R/W

HW3: Free R/W

go

Enter loan_amount interest_rate and number_of_payments

500 15 12

Payment #	Principal	Payment on Principal	Interest	Accumulated Interest	Payment
1	461.12	38.88	6.25	6.25	45.13
2	421.76	39.36	5.76	12.01	45.13
3	381.90	39.86	5.27	17.29	45.13
4	341.54	40.36	4.77	22.06	45.13
5	300.68	40.86	4.27	26.33	45.13
6	259.31	41.37	3.76	30.09	45.13
7	217.43	41.89	3.24	33.33	45.13
8	175.01	42.41	2.72	36.05	45.13
9	132.07	42.94	2.19	38.23	45.13
10	88.60	43.48	1.65	39.89	45.13
11	44.57	44.02	1.11	40.99	45.13
12	0.00	44.57	0.56	41.55	45.13

Breakpoint Encountered:

```
SW1: Used X      PAYMENT#25  521
# principal
PRINCIPAL = 1.991271E-03
# type payment
program (statements 2 .. 25)
var
  INPUT: TEXT
  OUTPUT: TEXT
  PRINCIPAL: REAL
  INTRATE: REAL
  PAYONPRIN: REAL
  PAYMENT: REAL
  INTEREST: REAL
  ACCUMINT: REAL
  NPAY: INTEGER
  PAYNUM: INTEGER
procedures
  POWER: function returning REAL (statement 1)
# tb var
$0: PAYMENT      521 at statement number 25
  Local variable(s):
    INPUT = {file value not displayed}
    OUTPUT = {file value not displayed}
    PRINCIPAL = 1.991271E-03
    INTRATE = 1.500000E+01
    PAYONPRIN = 4.457182E+01
    PAYMENT = 4.512900E+01
    INTEREST = 5.571727E-01
    ACCUMINT = 4.155000E+01
    NPAY = 12
    PAYNUM = 12
# exit
```

Greg Saville MDP Technical Support Manager

PASCAL DEBUG/TERMINAL CONNECTION

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 the terminal is to be connected to the 8560, another terminal (null connector) must be connected to the console port on the 8540.

Marilyn Hanson MDP Product Marketing Manager

ACE V.3 & ACECONFIG

ACE Version 3 is supplied with new configuration files for the CT8500 and 4105 terminals. These new `-.cfg` files **MUST** be used with the new version of ACE. This also means that any user created `-.cfg` files must be rebuilt with the new `aceconfig` to be compatible. Suggestion: Before updating to version 3, print a copy of your existing config file (`aceconfig -v /usr/lib/ace/<terminal>.cfg | lp1r`) to use as an input list when creating the new configuration file.

Greg Saville MDP Technical Support Manager

Z8000 MEMORY ALTERATION PROBLEMS

The Z8000 emulator exhibits some unexpected symptoms under the following conditions:

1. EM 1 with memory mapped to user, or EM 2 .
2. do a download to user (loading a program for example) or a "fill" of user memory or any other memory alteration.
3. press any key on the keyboard at the same time (generating an interrupt to the 2650).

The "unexpected symptom" could be either a memory error message or possibly a complete system lockup which requires re-booting.

Here is a solution. Lift pin #14 on U 4150 on the emulator and connect a wire from it to ground. This mod is unofficial at this point and all the usual cautions apply.

Wolfgang Takatsch MDP Technical Support

CONSTANTS LOCATION WITH 8086 PASCAL - CORRECTION

The program example which appeared in User Group News, Issue 1, Volume 2, page 41, was in error. The corrected listing is as follows:

```
SECTION      CNST_COPY_SEC,CLASS=INSTRQQ
GLOBAL      CNST_COPY,PASCAL_BEGIN,CODEBASEQQ

CNST_COPY XORW     SI,SI          ; CLEAR THE SI REGISTER
MOVW AX,#0A00H     ; LOCATION OF NEW ROM / 16
MOVW DS,AX
MOVW DI,SI        ; CLEAR THE DI REGISTER
MOVW AX,#0F00H    ; DESTINATION OF CONSTANTS_ROM / 16
MOVW ES,AX
MOVW CX,#01000H   ; SIZE OF CONSTANTS_ROM (0FFFFH-0)
CLD
REPZ MOVB         ; BLOCK TRANSFER
JMPS PASCAL_BEGIN,CODEBASEQQ ; TRANSFER TO ICS INITIALIZATION
```

MDP USER GROUP SOFTWARE LIBRARY/ARTICLE SUBMITTAL FORM

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

Entries should be forwarded to:

Library Manager
Tektronix, Inc.
Del. Sta. 92-635
P. O. Box 4600
Beaverton, Oregon 97075

MDP USER'S GROUP SOFTWARE LIBRARY/ARTICLE SUBMITTAL FORM

1. ABSTRACT. _____

2. Execution CPU _____ Primary Language _____
Hardware configuration required _____

Software configuration required (include source if non-Tek) _____

3. Do you want the following to appear in U.G.N.
Author's name _____ 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. I am submitting the program/article described above for possible placement in the MDP User's Group Library. I understand there is no further compensation due to me. This program/article is of my own design, the data contained in this submittal is not copyrighted and does not break any obligation to another person or organization relating to proprietary or confidential information. Tektronix, Inc. is authorized to distribute or publish copies of this program.

Signature _____ Date _____

THIRD PARTY SOFTWARE

**PLM COMPILERS - 100% INTEL-COMPATIBLE - ON THE 8560.
SEE LANGUAGE 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 it's accuracy. Tektronix suggests it's 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 languages
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 computers 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 for which MDP customers may want to use their system.
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, executability, 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.

* This issue of User Group News contains only changes to the tables; for the complete listing see User Group News Issue 1 - Volume 2.

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 it's 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

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

MDP SOFTWARE REFERRAL CATALOG UPDATE

This issue of User Group News contains changes and additions to the Catalog published in User Group News Number 2 (June 83).

CHANGES TO MDP SW REFERRAL CATALOG

Systems & Software, Inc. Moved

SSI's new location (effective 22 Aug 83): 3303 Harbor Blvd., Costa Mesa CA 92626, Ph: 714-241-8650.

Telecon Moved

Telecon is now located at 1155 Meridian, Suite 218, San Jose CA 95125.

IPI Distributors Changed

Industrial Programming Inc. offers Multi-Tasking Operating Systems (see Catalog Additions below). There are several changes in their international distributors; all of them are listed here:

Country	Distributor
France	Celdis SA 53 Rue Charles Frerot 94250 Gentilly ph: (1) 546.13.13
	Metrologie La Tour d'Asnieres 4 av. Laurent Cely 92606 Asnieres Cedex ph: 791.44.44
Germany	Scientific Control Systems GmbH Postfach 62 04 80 Oehleckerring 40 2000 Hamburg 62 Ph: 040 531030 Tx: 841-2174113
	Celdis-Enatechnik-System Vertriebsgesellschaft GmbH Schillerstrasse 14 2085 Quickborn (Hamburg) Ph: 041 06/612-240 Tx: 841-213590

Country	Distributor
India	Saras Electronics 21 Perambur Barracks Road Madras 600012 Ph: 32497
Israel	R. N. Electronics Agencies Ltd. 15 Kineret Street Bney-Brak P. O. Box 10205 Tel-Aviv 61 101 Ph: (03) 79 69 27 Tx: 922-3472107
Japan	Tokyo Electron Limited (MTOS-86, 80, 80MP) Shinjuku Nomura Bldg 1-26-2 Nishi-Shinjuku Shinjuku-Ku Tokyo 160 Ph: (03) 344-5893 C. Itoh Data Systems Limited (MTOS-68K, 68KF, 68, 69) C. Itoh Bldg. 2-5-1 Kita-Aoyama Minato-Ku Tokyo 107 Ph: (03) 497-8096
Scandanavia	Frontec Microdatorcentrum AB Box 204 Malmvagen 28 Sollentuna, Sweden Ph: 08-359360 Tx: 854-15130
Switzerland	Xmit AG Bellikonerstrasse 218 CH-8967 Widen Ph: 057-54656 Tx: 845-59955
U.K.	Itech Bessemer Road Stevenage Herts SG1 2DX Ph: (04380) 66561 Tx: 851-825824

NEW PRODUCTS FOR MDP SW REFERRAL CATALOG

There is information about languages, operating systems, and applications. Contact the vendor for product specs, prices, and references. Advise the vendor that you're interested in product compatibility with Tektronix' Micro-computer Development Products.

PLM on 8560

Caine, Farber, and Gordon (213-449-3070 California) is field testing on the 8560 PLM compilers for the 8080 and 8086. The compilers are 100% Intel compatible and produce Tektronix Linker object module format. The 8080 PLM is also available with Intel-compatible assembler/linker. The compilers will be available as products in the near future. Contact the vendor if you are interested in references or being a test site.

8086 Pascal, FORTRAN, & Assembly

First Systems Corp. (213-546-5581 California) offers Pascal and FORTRAN for the 8086 on VAX and IBM hosts. They offer for an Intel-compatible assembler/linker/locator for the 8086 on VAX. They also offer a sophisticated symbolic debugger that can be used with both their and Intel's products.

C 6809 on 8560

Introl Corp (414-276-2937 Wisconsin) offers a C 6809 cross-compiler on PDP-11 UNIX and 8560 systems. It sells for \$1500; specify 8560 media when ordering. The compiler is said to accept Kernighan & Ritchie C (the Bell Labs standard) and generate very high quality code. The product comes with a 6809 assembler. See SOFTWARE PRODUCT REFERENCES.

JMI Software (215-657-5660 Pennsylvania) distributes in the US the C 6809 compiler from Real Time Systems for \$1550. It runs on DEC minis and the 8560. The 8560 version generates assembly compatible with Tek's 6809 assembler. Real Time Systems (phone: 0624-833403, telex:628356 TOMASS; England) offers this compiler in Europe. See SOFTWARE PRODUCT REFERENCES.

C 16032

JMI Software Consultants (215-657-5660) offers a C 16032 compiler on VAX (VMS and UNIX) for \$1500. The 16032 assembler/linker package is available separately for \$500.

BASTOC

JMI Software Consultants (215-657-5660) offers a Basic to C Translator for any system that supports C. It supports ANSI 78 BASIC with most Microsoft and some CBASIC extensions. BASTOC can be tailored to BASIC dialects, possibly including Tek's 8550 MDL/u.

Jovial

Proprietary Software Systems (429 Santa Monica Blvd., Suite 430, Santa Monica CA 90401, Ph: 213-394-5233) offers Jovial compilers for 1750A, 8086, and z8000 on VAX and IBM systems. Assembler, linker, and simulator are also available with these products. PSS also offers a complete line of full-function assemblers for most micros and running on many hosts.

Ada

Telesoft (619-457-2700 California) offers Telesoft-Ada compilers and other Ada related products. A cross-compiler for the 68000 is available on VAX (VMS and UNIX 4.1bsd) and IBM 370.

Irvine Computer Sciences Corp (714-754-6684 California) offers an Ada to C translator and is working on machine code generators. Their products are available only from distributors today. Contact the vendor for a list of distributors.

8-bit C on 8560

Holos Corp (404-373-0110, Georgia) offers C compilers for 8080, 8085, z80, and 6809. Support for others is available upon request. The compilers run on the 8560 and generate Tek assembly code. Price is \$1200; specify 8560 media when ordering.

8-bit C and Pascal on 8560

Real Time Systems (phone: 0632-733131, telex: 53429 PACE G; England) offers both C and Pascal compilers for 8080, z80, 6502, 8086, 68000, and 6809. The compilers run on DEC minicomputers and the 8560. 8560 versions for 8080, z80, and 6809 produce Tek Assembly language. RTS is seeking 8560 test sites for the 8080 and z80 versions. RTS also offers cross-assemblers for most micros on DEC minis and UNIX systems.

8-bit Pascal on VAX and PDP-11

Boston Systems Office (617-894-7800) now offers Pascal Z80 and 6809 on VAX/VMS and PDP-11/RSX. BSO already offers Pascal support for the 68000, 8086/186. For compatibility with Tektronix' 8540 Integration System, BSO offers download and Extended TEKHEX support.

68000 FORTRAN, C, & PL/M

Green Hills Software (213-796-6543 California) offers FORTRAN, C and PL/M for the 68000 on a VAX/UNIX. They cost \$6000; compilers for VMS are available by special request. Output is either UNIX or Motorola format.

C 8086 + Debug

C Systems (714-637-5362 California) offers a C compiler for the 8086 with a sophisticated symbolic debug capability. The product is available now on IBM PC.

TMS 320 Assembler

PH Associates (703-281-5762, Virginia) offers an assembler for the TMS 320. It runs on the 8560, PDP-11, and CP/M systems. It is very fast and supports TEKHEX (not relocatable). The price is \$1000. TMS 320 from Texas Instruments is high speed single-chip microcomputer for signal processing applications.

REAL TIME OPERATING SYSTEMS**MTOS 68000**

Industrial Programming Institute (516-936-6600 New York) now offers their MTOS 68000 in Tek Assembly source for full compatibility with Tektronix' development tools for the 68000. MTOS is a multi-tasking real-time operating system with support for memory management and multi-processors. There are also options for a file system and floppy disk driver.

VRTX z80

Hunter & Ready (415-326-2950 California) now offer VRTX for the z80. VRTX is already available for 68000, 8086, and z8000. VRTX is provided in a loadable form that is used with Tektronix development tools with minor adaptation: program Tek assembly routines to match the parameter calling conventions of 8560 Pascal with each VRTX function (about 15). Hunter & Ready also now offers TRACER, a debugging tool for VTRX-based systems, for the 68000 (8086 to follow).

REX Tools

Systems & Software (714-241-8650 California) offers REX Tools for the 8086/88. REX Tools include a Pascal compiler, Intel-compatible assembler/linker/locator, operating system kernel, and an interactive symbolic software emulator called Softprobe.

C Executive

JMI Software (215-657-5600) now offers their C Executive for the NS16032. C Executive is also available for LSI-11, MC68000, 8080/85, 8086/88, and Z80.

pSOS 68000

Software Components Group (408-923-2741 California) offers a real-time multiprogramming, multitasking, operating system kernel for the 68000. pSOS is provided in loadable, object form that can be used with 8500 products. There is a companion file system FILES-68K.

RTOS-80

Microsystems, Inc. (213-577-1471, California) now offers a real-time operating system for 8080/8085 and z80. RTOS-80 is compatible with the vendors proFORTH system which is available on the 8550 and 8002A.

APPLICATION SOFTWARE**Comm 60**

Holos Corp. (403 West Ponce de Leon Ave., Decatur GA 30030, 404-373-0110) offers a communication package for the 8560. Comm 60 is a \$1000 general purpose, configurable, communications package connects the 8560 user to a remote host such as VAX/VMS or IBM. It offers functions similar to Tektronix' 8540 COM option and 8560 UNICOM option including downloading object and source files. It is configurable to a variety of hosts via a series of prompts for parameters and control characters. A demo version of the product will be available.

Nex

Holos Corp. (404-373-0110) also offers a screen editor for the 8560. Nex is a \$800 full-featured screen editor supporting user-definable commands (KEYMAP) and over 180 terminals.

Viewcomp

This spreadsheet is available for the 8560 from UNICORP SW (212-307-6800 New York). Different terminals are handled in a way compatible with termcap. The cost is \$500; specify 8560 media when ordering.

Vc

Software Innovations (603-883-9300 New Hampshire) offers a "visual calculator" for UNIX systems. No 8560 installations are known yet.

SCCS

Source Code Control System is available for \$500 from Santa Cruz Operation (408-425-7222 California). Specify 8560 media when ordering from SCO.

UNIX Applications Software

Andyne Computing (613-548-4355 Ontario) offers UNIX applications like: Menu Presentation Software for developing interactive menu systems, RTUTIL to access RT-11 files on UNIX with wildcard notation, RTUNIX to profile time-critical parts of (resident) C programs, eh? to manage on-line documentation, and net for hierarchical data management.

UniPress Software, Inc. (1164 Raritan Ave., Highland Park NJ 08904, 201-985-8000) offers a varied product line for UNIX systems. Their offering includes MIMIX (CP/M Emulator), LEX (word processor), C 8086 compiler, a UNIX menu system, UNICALC spreadsheet, and others.

SOFTWARE PRODUCT REFERENCES

MDP provides to its sales organization references for third party products. Contact your local Tektronix salesperson for these references. References for these products have been published in MDP's sales newsletter:

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
C	6809	8560	Introl Corp.	July 83
C	68000	8560	Alycon	July 83
C	68000	8560	Whitesmiths Ltd.	July 83
FORTRAN	8086	8560	Virtual Systems	July 83
Assembly	6805	8560	Virtual Systems	July 83

Tektronix would like to know your appraisal of third party products used with our Microcomputer Development Products. This information helps us determine which products deserve a stronger commitment from Tektronix. If you want to offer an appraisal, please contact

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

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.
 - Formatting 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 Manager

PRODUCT PERFORMANCE SECTION

SOFTWARE SUBSCRIPTION SERVICE

Customers who have purchased software subscription service for a product(s) will receive the updated versions as performance problems are resolved.

John Owens MDP Technical Support Manager

SLOW 8560 OR EXCESS GETTY TIMES

On a few occasions, poor system performance was observed when little was going on to justify the reduced response. When the process status was displayed, a getty process with a large CPU time was discovered. When the process was killed (kill -9 PID) system performance returned to normal.

One known cause is a terminal (not logged in) which produces a glitch on the DTR line on power up.

If poor system performance is encountered as described above, simply kill the errant getty process. Note: cron, atrun, and swapper will normally accumulate substantial CPU times, do not remove these processes.

John Owens MDP Technical Support Manager

Z8000 ASSEMBLER VERSION 2.0

Version 2.0 of the Z8000 Assembler for the 8560 (ASMZ8K opt 1A) is now available. Version 2.0 includes a virtual symbol table for user-defined symbols. To assemble code with large number of symbols, add a -b to the invocation line.

Marilyn Hanson MDP Product Marketing Manager

8086 PASCAL/PDB MOD

The Pascal Compiler (PAS8086 opt 1A) and PDB (PDB8086 opt 1A) for 8086/8088 have been modded to correct bugs reported on these products. These mods do not reflect any enhancements to support the full addressing of the processor. Those enhancements will be included in the Version 2 release of the product scheduled for 4th quarter of FY400. Customers currently subscribed to the Software Subscription Service will receive the mods automatically. The current version numbers are:

PAS8086	Version 1.10-05
PDB8086	Version 1.11-00

Marilyn Hanson MDP Product Marketing Manager

4105 POWER UP PROBLEMS

Some 4105's will not power up when the LED in the "CAPS LOCK" position flashes continuously.

If this happens, turn off the terminal, remove the ROM access cover on the back of the terminal, and be sure that the ROM's are properly seated.

Gordon Glathar MDP Technical Support

MDP BUG BASE

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

John Owens MDP Technical Support Manager

8540 ROMPATCH FAILURE

PRODUCT

8540 executing rompatch -l

CONFIGURATION

8540 OS-40 version 1 with patch #36 and roms for TTA, COMM (V.-01), prom programmer, 8086/8088 and 68000 (without diags)

PROBLEM

With the comm rom (version -01) and all the rom sockets filled, then executing the "rompatch -l" command will cause the 8540 to halt.

NATIVE PROGRAMMING TOOLS INSTALLATION**PRODUCT**

8560 and installing Class C software

CONFIGURATION

8560 TNIX version 1.4 and Class C software, Native Programming Package version 1.0

PROBLEM

If the Native Programming Package is told "no" to installation of "/usr/include" and "/bin/sed" using the "install" command, all commands or files which use "/usr/include" as a dependent file will not be installed.

DIRECTORY PERMISSIONS OF /USR/INCLUDE/SYS**PRODUCT**

8560 and accessing /usr/include/sys

CONFIGURATION

8560 TNIX version 1.4

PROBLEM

Only root can execute the directory or contents of /usr/include/sys.

COMMENTS

Change attributes of /usr/include/sys from 644 (drw-r--r--) to 755 (drwxr-xr-x).

GLOBAL ERROR MESSAGES AND NO GLOBALS**PRODUCT**

8086 assembler without globals but with global error messages

CONFIGURATION

8550 DOS-50 version 2.1, 8086 assembler version 1.01-04 and linker version 2.05

PROBLEM

At times, a large 8086 assembler program without declared globals will assemble without errors, but during linking may generate an error message "undefined global."

COMMENT

The error is caused by overlapping memory areas defined at link invocation. The user should check section size versus location definitions.

PASCAL DEBUG ERROR #79**PRODUCT**

8086 Pascal Debug with error #79 and internal error

CONFIGURATION

8560 TNIX version 1.3, 8086 Pascal compiler version 1.02-04 and 8086 Pascal Debug version 1.07-00A

PROBLEM

At times, the 8086 Pascal Debug will produce error messages (#79 or internal error) without an obvious reason.

COMMENTS

This problem is corrected in version PDB8086 V1.11-00. This version was released Aug. 25.

16BIT PASCALS SECTION ASSIGNMENT ERROR**PRODUCT**

16bit Pascal & wrong lib sec. name & initialized word variable

CONFIGURATION

8560 TNIX V1.3 and 16bit pascal compiler

PROBLEM

Two sections in the PASCAL library pertaining to floating point computations, have been assigned to the class name "DATAQQ". "DATAQQ" contains RAM variable sections. The two sections should have been assigned the the class name "CONSTQQ" which contains ROM constants. The incorrectly assigned sections are "FPP01QQ" and "FPP02QQ."

COMMENTS The above sections can be explicitly assigned to class name "CONSTQQ" at link invocation.

6801 EMULATOR MODE SELECT ERROR**PRODUCT**

6801 emulator and incorrectly reported 6801 mode

CONFIGURATION

8550 or 8540 with 6801 emulator and probe

PROBLEM

Upon reset, status lines P20, P21 and P22 are sampled by the 6801 to select the mode of operation. Motorola recommends 10K pullup resistors on those lines (see the Motorola Data Manual series B, 1981 page 4-95 for details). However, the 6801 emulator requires a smaller value pullup resistor in order to insure reliable operation.

COMMENTS

This problem manifests itself in that after user hardware reset the emulator may not report the correct 6801 mode.

PASCAL AND REDEFINITION OF IDENTIFIERS
PRODUCT

8086, Z8000, 68000 Pascal compilers and identifier

CONFIGURATION

8550 DOS-50 version 1.4 or 8560 and TNIX version 1.4 pascal compiler (68000 version 1.X, Z8000 version 1.X 8086 V1.01-07

PROBLEM

Redefinition of identifier is not caught.

PASCAL OPTIMIZER ERROR
PRODUCT

Z8000, 68000 and 8086 Pascal compilers and optimization

CONFIGURATION

8560 TNIX version 1.4 and pascal compilers (Z8000 version 1.09-08, 68000 version 1.01-14, 8086 version 1.04)

PROBLEM

Optimization may cause bad code to be generated in some cases. This is only a problem when common sub-expressions are of different ordinal type.

COMMENTS

The following code shows the sequence of code required to demonstrate the problem.

```

type
  int16 = -32767..32767;
var
  i: int16;
  j: int16;
  temp: integer;
begin
  ...
  temp := i; { temp is type integer and i is being      }
  i := j;   { upward coerced from type int16 to integer }
  j := temp; { the problem will occur here unless there }
  ...      { is some intervening jumps or breaks      }
          { or the types are changed to the same type }
end.
```

8086 Pascal is fixed in version 1.1; 68000 Pascal version 1.1 is expected soon.

8086 PASCAL TRUNCATION OF IDENTIFIERS TO 16 CHAR.**PRODUCT**

8086 Pascal compiler and a maximum of 16 character identifiers

CONFIGURATION

8560 TNIX V1.4 or 8550 DOS-50 V2.1 with 8086 Pascal compiler (all versions)

PROBLEM

The debug information which the compiler produces for pdb, allows a maximum of 16 characters in an identifier. To a user using pdb, any identifiers that are more than 16 characters will not be known by pdb except the 16 character truncation.

COMMENTS

Avoid the problem by limiting identifiers to a maximum of 16 characters.

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

USER GROUP LIBRARY ABSTRACTS

ACCOUNT - LOGIN/LOGOUT HISTORY

utility
8560/61

MUGL TNIX Vol I
C

Abstract

This utility provides a history of a user's login and logout activity. C source included.

Author: Jean Paul Bonsang, Tek France

ADMIN - DISPLAY LOGIN STATS

utility
8560/61

MUGL TNIX Vol I
shell script

Abstract

Displays the last 25 logins and some other system stats by examining the /usr/adm/wtmp file. Displays a legend describing what each entry means.

ASCII - CONVERTERutility
8560/61MUGL TNIX Vol I
C**Abstract**

This utility simply converts an ascii input string to its hex, octal, and decimal equivalents. C source included.

ASCIITABLE - PRINTS ASCII CHARTutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

This utility shell script prints a nice ascii table showing the full ascii character set along with each character's hex and decimal equivalent, both with and without the 8th parity bit set.

Author: Greg Saville, MDP Marketing

ASMIT - AUTO FILENAME EXTENSIONutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

This shell script simplifies assembler invocation by providing automatic file extensions of .obj and .lst on your object and listing files. You need specify only the core name of your .asm source file, and you may run multiple assemblies with a single command line invocation.

ATS - AT STATUSutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

A shell script to display any jobs scheduled with the at command. You are given the options of examining any jobs in the list, and/or deleting if desired.

ATSTATS - AT STATISTICSutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

This shell script provides a convenient means to examine the status of any jobs scheduled by the at command by displaying the associated directories involved with at.

DECIMAL - CONVERTERutility
8560/61MUGL TNIX Vol I
C**Abstract**

This C program converts decimal input values to their hex, octal, and character equivalents. C source included.

DONUM - DO COMMAND X TIMESshell utility
856XTNIX Vol I
shell script**Abstract**

This shell script executes any command a given number of times. Useful example for use of eval in shell scripts to implement variable loop counters. See donum shell script for documentation on use.

EXTX - EXTRACT TEXTutility
8560/61MUGL TNIX Vol I
C**Abstract**

C program that extracts and prints ascii character strings from any file. Useful for snooping around in object files looking for character strings for undocumented error messages, instructions, etc. C source included.

FDMP - FILE DUMP UTILITYutility
8560/61MUGL TNIX Vol I
C**Abstract**

Another file dump utility that lets you examine any type file and displays the hex and ascii equivalent for each byte. C source included.

Author: Jean Paul Bonsang, Tek France

FDUMP - FILEDUMPutility
8560/61MUGL TNIX Vol I
C**Abstract**

C program that gives an 8550 fdump style filedump of any given file. Displays hex and ascii values. C source included.

FMAN - FAST MANUAL PAGE COMMAND

Utility
8560/61

MUGL TNIX Vol I
C

Abstract

This program is identical to the standard TNIX man command in function, but the manual content is displayed much faster. The command normally displays a page at a time and inquires for next page displays. The -n option suppresses the prompting for each page. C source included.

Author: Jean Paul Bonsang, Tek France

FMT - LINK LIST FORMATTER

utility
856X

TNIX Vol I
C

Abstract

This utility provides a linker listing similar to the one that was available with the Version V1.XX-XX linker. Good documentation and well commented C source code included.

Author: Matthias Dolder, Tek Switzerland

HELP - PERSONAL COMMANDS HELP

utility
8560/61

MUGL TNIX Vol I
shell script

Abstract

A shell script to put in your .bin directory. When invoked with no command line arguments, it lists your personal .bin directory. When invoked with a command line argument, and if that file name exists in your .bin directory, then the file is displayed for you.

HEX - CONVERTER

utility
8560/61

MUGL TNIX Vol I
C

Abstract

Utility to convert hex input values to their hex, octal, and character equivalents. C source included.

IFIX - INTEL FIXutility
856XTNIX Vol I
shell script**Abstract**

This awk shell script removes the extra line feeds from Intel system files downloaded to the 8560. The extra linefeeds result from Intel's use of a cr-lf sequence, while TNIX/UNIX uses a single new-line character.

IOC.FORM - 'FILL OUT' FORMnroff utility
8560/61MUGL TNIX Vol I
fillout form**Abstract**

This file provides a convenient method of generating Inter Office Communications (IOC or memo) input to nroff. Simply edit the ioc.form file with a screen editor such as ace or lde and 'fill in the blanks' to create an nroff ready input file.

Author: Byron Lunz, MDP Marketing

LOAD - OBJECT/SYMBOL LOADERdebug utility
8560/61MUGL TNIX Vol I
shell script**Abstract**

This shell script does an object file load, symbol table load, and turns on symbolic debug for you all in one command.

LOG - TERMINAL SESSION LOGGERutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

This shell script simulates the 8550 log command which allows you to create a file of an entire terminal session. Useful for documentation purposes or recording of a debug session.

LOG1 - TERMINAL SESSION LOGGERutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

Another version of a shell script which simulates the 8550 log command. This allows you to create a file of an entire terminal session. Useful for documentation purposes or recording of a debug

session.

Author: John Owens, MDP Marketing

LOGGER - PHONE CALL LOGGER

utility
8560/61

MUGL TNIX Vol I
C

Abstract

Nice program for maintaining a log of telephone calls, including who called, date/time, subject, and resolution. When invoked with a name, searches through a database for that person and displays their full name, title, and location. If not in the database, takes your input directly for the record header. Next, you enter input mode where you can explain details of the call, suggestions, solutions, etc. Provides several convenience shorthand entry features which may be used if desired. You may also request help during the session and get a summary of all the options and commands available in logger. When exiting, the information is appended to the file 'logfile', located in your home directory, as well as two other places for administrative purposes. This program is used in MDP marketing for maintaining a log of all customer and sales calls which is circulated through the department to share the information. The program as is, is set up for use on our machines in marketing, and makes specific reference to particular database and destination directories/files. However, since the commented C source is supplied, it is an easy matter to alter a few definitions and recompile to suit your installation. A manual page is also included.

Author: Jeff Francis, Phoenix FO

MAILALL - MAIL TO ALL USERS

utility
8560/61

MUGL TNIX Vol I
shell script

Abstract

This shell script simplifies sending the same mail to all users on the system. It examines /etc/passwd and uses sed to extract a user list and then sends the message to each user. Does take into account the fact that you don't normally want to send mail to root, bin, etc.

Author: Byron Lunz, MDP Marketing

MAILTO - MAIL TO USERS ON REMOTE SYSTEMS

shell utility
856X

MUGL TNIX Vol I
shell script

Abstract

This shell script makes it easy to send mail to users on remote systems without having to remember the exact system path for each user. You will need to alter the case statements in the script to reflect the paths of the users you frequently send mail to.

Author: Byron Lunz, MDP Marketing

MDSHEX - INTEL SYMBOL LISTER

utility
856X/Intel MDS/ICE51

TNIX Vol I
shell script, C

Abstract

This utility reads an Intel hex file from an MDS with ICE51 and prints a symbol listing and Intel-hex file. Requires the Tektronix Intel Vendor Interface Package (VIP) for downloading to 8560. Two versions supplied, one a shell script, and the other a well commented C program.

Author: Matthias Dolder, TEK Switzerland

MLABEL - MAIL LABEL PRINTER

utility
8560/61

MUGL TNIX Vol I
shell script grep/awk

Abstract

A shell script using grep and awk which selectively extracts, formats, and prints mailing labels from a master database file. The database file consists of records of 8 fields separated by commas containing the information. The format is documented in a sample database file which is included along with the shell source and a .doc file. The script may easily be modified to provide different formats for other purposes.

Author: Greg Saville, MDP Marketing

MONTH - CALENDAR PRINTER

utility
8560/61

MUGL TNIX Vol I
shell script

Abstract

This utility prints a nice calendar of the current month with the current day blinking in inverse video. Handy to put in your .profile to display the date when you login each day. Currently designed to work with the Tektronix CT-8500 or 4105 terminals, however the shell script is easily modified to work with any other terminal which has blink and/or inverse video attributes. Requires the optional TNIX software package, Auxiliary Utilities.

Author: John Owens, MDP Marketing

OCTAL - CONVERTER

utility
8560/61

MUGL TNIX Vol I
shell script

Abstract

This C program takes octal numbers as input and displays their hex, decimal, and character equivalents. C source included.

PATCH - FILE PATCH ROUTINEutility
8560/61MUGL TNIX Vol I
C**Abstract**

This C program allows you to snoop around in any type file, and if invoked in write mode, and if you are superuser, and if you think you know what you're doing, allows you to patch the file. This can be a very useful program is used with care on a scratch copy of the file you're working with. It allows you change anything, even directory entries, so be careful. The program is written in C using a menu approach which makes it even easier for the unwary to botch things up. Be carefull! C source included.

PROLOG - M900 PROM PROGRAMMER COMMUNICATIONS PROGRAMcomm utility
8560/61MUGL TNIX Vol I
C**Abstract**

This program manages communications between a Prolog M900 prom programmer and an 8560 port. The C source is included along with a manual page describing its use.

Author: Jim Willey, Santa Clara FO

PRQ - PRINTER QUEUE STATUSutility
8560/61MUGL TNIX Vol I
shell script**Abstract**

Handy shell script which prints the queue for the two spooler directories. Useful for determining your position in the queue for your job.

RESTORE - RESTORE EMU STATUSdebug utility
8560/61MUGL TNIX Vol I
shell script**Abstract**

This shell script provides a convenient method of restoring your emulator setup from a previous debug session saved with the 'save' utility. Your memory image is reloaded, registers restored to previous values, and the old program counter value is displayed so you may continue right where you left off. This version is designed for the Z80 emulator, but may be modified to work with others. Also see the 'save' utility for more information.

Author: Greg Saville, MDP Marketing

SAVE - SAVE DEBUG SESSIONdebug utility
8560/61MUGL TNIX Vol I
Shell script**Abstract**

This shell script saves your current 8540 debug status, including a complete memory image and current register status. It is useful when you are interrupted or must shutdown, but wish to later continue from the same point. You may also wish to create a known starting point from which to start several debug runs. This version is designed for the Z80 emulator, but may easily be adapted to other emulators. Could easily be expanded to also save your emulation mode, memory map, svc status, etc. for later restoration by a similarly expanded 'restore' command. See the 'restore' utility for additional information.

Author: Greg Saville, MDP Marketing

TELE - TELEPHONE NUMBERutility
8560/61MUGL TNIX Vol I
Shell Script**Abstract**

Shell script command that uses grep and awk to search through a database of names, addresses, and phone numbers and prints the desired person's phone number. Uses the same database format as in the mailing label printer, 'mlable'. A sample database file is included.

Author: Greg Saville, MDP Marketing

TOOLS - PERSONAL COMMANDS LISTUtility
8560/61MUGL TNIX Vol I
Shell Script**Abstract**

Simple shell script that conveniently prints a directory of any personal commands you may have placed in your .bin directory.

TR88000 - 68K TRACE FILTERdebug utility
8560/61MUGL TNIX Vol I
C**Abstract**

This program was created by John Simmons, SE Newport News. They do selective filtering of "tra all" type output from an emulation session. All read stdin and write to stdout. C source included.

Author: John Simmons, Newport News FO

TRZ80 - Z80 TRACE FILTERdebug utility
8560/81MUGL TNIX Vol I
C**Abstract**

This program was created by John Simmons, SE Newport News. They do selective filtering of "tra all" type output from an emulation session. All read stdin and write to stdout. C source included.

Author: John Simmons, Newport News FO

TRZ8002 - Z8K TRACE FILTERdebug utility
8560/81MUGL TNIX Vol I
C**Abstract**

This program was created by John Simmons, SE Newport News. They do selective filtering of "tra all" type output from an emulation session. All read stdin and write to stdout. C source included.

Author: John Simmons, Newport News FO

USPLIT - TEKHEX FILE SPLITutility
856XTNIX Vol I
C**Abstract**

This program splits a texhex file into two parts, useful for programming eproms where one prom has the hi byte, while the other contains the lo byte. See the tpsplit.doc and C source files for documentation on use.

VMORE - 'VANILLA MORE'utility
8560/81MUGL TNIX Vol I
C**Abstract**

A simple version of the standard TNIX 'more' command which pages output to the terminal a screenful at a time. Eliminates the problem of trying to ctrl-s/ctrl-q the screen without the text flying by. This version, written in C, is designed to work with any 'vanilla' type terminal and makes no use of any terminal dependent cursor controls, etc. As such, there are no prompts; a screenful is displayed and it waits for you to type a space for the next screen, a carriage return for the next line or a q to quit. See the C source and doc files for more information. Designed for 25 line terminals, but may easily be modified and recompiled for any screen length. Also useful for viewing nroff generated output on a screen, since the standard more trashes the display when overprinting and backspacing is used in ioc headers, etc.

Authors: Jeff Francis, Phoenix FO & Greg Saville, MDP Marketing

INDEX

16 bit Pascals	48	PDB	22, 46
4105 Power up problem	46	PDB Step	33
68000 Emulator at 10 MHz	3	PDB Trace	33
68000 FORTRAN, C, & PL/M - 3rd Party SW	42	PLM on 8560 - 3rd Party SW	40
68000 Pascal	4, 22, 49	PSOS 68000 - 3rd Party SW	42
68000 Pascal Debug	33	Pascal	46
6801 Emulator	48	Pascal Debug	50
8-bit C and Pascal on 8560 - 3rd Party SW	41	Pascal Floating Point	4
8-bit C on 8560 - 3rd Party SW	41	Printer	19
8-bit Pascal on VAX and PDP-11 - 3rd Party SW	41	Printer Configuration	19
8001	7	REX Tools - 3rd Party SW	42
8002A	7	ROMPATCH Failure	46
8086 Assembler	9-10, 47	RTOS-80 - 3rd Party SW	42
8086 Pascal	22, 49-50	Running Software on the 8560	44
8086 Pascal Debug	48	SCCS - 3rd Party SW	43
8086 Pascal Dedug	33	Section Assignment Errors	48
8086 Pascal, FORTRAN, & Assembly - 3rd Party SW	41	Segmented Addresses on Linker	10
8560 Distribution Media Information	44	Software Product References	43
Aceconfig	34	Software Referral Catalog Corrections	39
Ada - 3rd Party SW	41	Software Referral Catalog New Products	40
Assembler	17	Software Referral Catalog Update	39
Auxiliary Utilities Ver 2	2	Software Submittal	35
BASTOC - 3rd Party SW	41	Software Subscription Service	45
C 16032 - 3rd Party SW	41	TMS 320 Assembler - 3rd Party SW	42
C 6809 on 8560 - 3rd Party SW	41	TNIX	16, 45
C 8086 + Debug - 3rd Party SW	42	Tektronix Software Referral Statement	37
C Executive - 3rd Party SW	42	Text Processing Ver 2	2
C Library Attributes	47	Third Party Software Information	37
C Math Libraries	17	Third Party Software MDL Compatibility	38
C Programs	11	UNIX Applications Software - 3rd Party SW	43
CT8500	11	Updating Third Party Software Information	38
Class C Install Procedure	47	User Group Library	2
Comm 60 - 3rd Party SW	42	Users Group News Subscription	35
Cvt	16	Using the Product Referral Tables	38
Disk Backup	7	VAX Support	1
Errata	34	VAX/8560/8540 Configuration	10
Fast Man	11	VRTX x80 - 3rd Party SW	42
File Dump	11	Vc (spreadsheet) - 3rd Party SW	43
GPIO Controller	4	Viewcomp (spreadsheet) - 3rd Party SW	43
GPIO Option	7	Z8000 Assembler	46
Getty Bug	45	Z8000 Emulator	34
Global Error Messages	47	Z8000 Pascal	49
Identifier Truncation	50	Z8001 Pascal	22
Immediate Data	10	<i>Ascii Text Extractor</i>	55
Including Math Libraries	17	<i>Ascii</i>	54
Jovial - 3rd Party SW	41	<i>Ascitable</i>	54
LOGIN/OUT Times for a User	11	<i>Assembler File Extension</i>	54
Link	10	<i>At Status</i>	54
Linker	17	<i>Calendar Printer</i>	59
MDP Software Referral Service	37	<i>Communications</i>	60
MDP Third Party Software Benefits	37	<i>Converter</i>	54
MTOS 68000 - 3rd Party SW	42	<i>Debug Session Restore</i>	60
Manuals	8	<i>Debug Utility</i>	57, 60-61
Mode Select Error	48	<i>Decimal Converter</i>	55
Native Programming Package	4	<i>Fast Man Command</i>	56
Native Programming Ver 2	2	<i>File Dump</i>	55
New Application Software	42	<i>Hex Converter</i>	56
New Real Time Operating System Products	42	<i>IOC Form</i>	57
New Third Party Language Software	40	<i>Intel MDS Symbol Lister</i>	59
Nex (screen editor) - 3rd Party SW	43	<i>Intel VIP Utility</i>	57
Nroff	19	<i>Linker Listing</i>	56
		<i>Log Simulator</i>	57
		<i>Login Stats</i>	53

<i>Mail To All Users</i>	58
<i>Mailing Label Printer</i>	59
<i>Mailto</i>	58
<i>More Paging Utility</i>	62
<i>Nroff Utility</i>	57
<i>Octal Converter</i>	59
<i>Patch Utility</i>	60
<i>Personal Commands Directory</i>	61
<i>Personal Commands Help</i>	56
<i>Phone Call Logger</i>	58
<i>Prolog Prom Programmer</i>	60
<i>Repeated Command Script</i>	55
<i>Spooler Queue Status</i>	60
<i>Symbolic Debug Loader</i>	57
<i>Tek Hex File Split</i>	62
<i>Telephone Number Search</i>	61
<i>Trace All Filter</i>	61-62
<i>Utility</i>	53-62
<i>Z80 Debug Session Save</i>	61
<i>Z80 Emulator</i>	60

Italic Index entries denote library articles

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

Copyright © 1983, Tektronix, Inc. All rights reserved. Printed in U.S.A. Tektronix products are covered by U.S. and foreign patents, issued and pending. Information in this publication supersedes that in all previously published material. Specification and price change privileges reserved. TEKTRONIX, TEK, SCOPE-MOBILE, TELEQUIPMENT, and  are registered trademarks. For further information, contact: Tektronix, Inc., P.O. Box 500, Beaverton, OR 97077. Phone: (503) 627-7111; TWX 910-467-8708; TLX: 15-1754; Cable: TEKTRONIX. Subsidiaries and distributors worldwide.

[REDACTED]

IMPORTANT NOTICE

Please complete this card IF:

1. The address label is incorrect, or
2. You are not the person who should receive this newsletter, or
3. You wish to add names to our mail list.

For identification purposes, please place the number on the upper right of your current mailing label here: _____

add to list change address remove from list

Company name _____

Company address _____

internal address or department

street & number or P.O. Box

city, state, zip

country

User Name _____

Title _____

Do you want the newsletter label to specify the:

title only name only both in future
newsletter mailings
date _____

[REDACTED]

Return to:

Tektronix, Inc.
Microcomputer Development Products
Attn: Circulation Manager
P.O. Box 4600, D.S. 92-635
Beaverton, OR 97075

[REDACTED]

[REDACTED]

IMPORTANT NOTICE

Please complete this card IF:

1. The address label is incorrect, or
2. You are not the person who should receive this newsletter, or
3. You wish to add names to our mail list.

For identification purposes, please place the number on the upper right of your current mailing label here: _____

add to list change address remove from list

Company name _____

Company address _____

internal address or department

street & number or P.O. Box

city, state, zip

country

User Name _____

Title _____

Do you want the newsletter label to specify the:

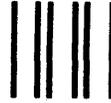
title only name only both in future
newsletter mailings
date _____

[REDACTED]

Return to:

Tektronix, Inc.
Microcomputer Development Products
Attn: Circulation Manager
P.O. Box 4600, D.S. 92-635
Beaverton, OR 97075

[REDACTED]



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.1 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Tektronix, Inc.
Microcomputer Development Products
Attn: Circulation Manager
P.O. Box 4600, D.S. 92-635
Beaverton, OR 97075



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.1 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Tektronix, Inc.
Microcomputer Development Products
Attn: Circulation Manager
P.O. Box 4600, D.S. 92-635
Beaverton, OR 97075

