TEK MICROCOMPUTER DEVELOPMENT PRODUCTS DIVISION

VOL. IV, ISSUE NO. 1
JUNE, 1985

USER GROUP NEWS

1

1 TEK/IBM PC MICROPROCESSOR SUPPORT S/W DEVELOPMENT TOOLS FOR 8-16 BIT μP

DESIGN AUTOMATION GROUP

Tektronix®
COMMITTED TO EXCELLENCE

PRODUCT INFORMATION SECTION

SA TOOLS NOW AVAILABLE ON VMS V4.1

SA Tools has been upgraded to operate on VAX/VMS V4.1. The release tape supports both VMS V3.4 and V4.1. Only VMS V4.1 supports VAX clusters. SA Tools on VMS V4.1 will support a much larger SA Spec, (about 20 levels) than on V3.4, because of the longer VMS file names. This new release is provided automatically to customers under warranty and Software Subscription Service. (See related article on page 9).

Rodney Bell Product Line Manager

~ ~

MDP OFFERS SOFTWARE FOR VMS V4.X OPERATING SYSTEM

To support customers transitioning to the VMS V4 operating system, all MDP VAX/VMS software products will include two versions of the program on the 9-track mag tape option. In addition to the Product Release compiled on VMS V3.4, the product will include the program recompiled on VMS V4.1. If your software is under warranty or you are a member of the MDP Software Subscription Service, you will receive the software automatically. If you have the TU-58 media of our VMS software and need to update to VMS V4, contact your local Tektronix representative for assistance.

Later this year, when most customers have updated to V4, our products will be shipped with only V4 support on both the 9-track mag tape and the TU-58 cassettes.

The installation procedure included with the software will install the appropriate version for your operating system.

Marilyn Hanson MDP Marketing

~ ~

C-LANGUAGE 8086/8088 AND 80186/80188 SOFTWARE

C LANDS 8086/80186, a complete LANguage Development System (LANDS) package is now orderable for the 8086/8088 and 80186/80188 family on DEC VAX/VMS 4.X and UNIX 4.2 operating systems.

C LANDS consists of:

- Language Directed Editor for C - C Cross Compiler for the 8086/186 family - ICS (Integration Control System) - C Debug

The C Language Directed Editor is a screen-oriented editor that understands HLL. It contains many time saving features. One feature is that syntax errors are brought to the programmer's attention during the edit session, not during compilation.

The compiler translates statements written in the C programming language into executable object code for the 8086/88 and 80186/88 microprocessors. The object modules produced are in Tektronix object format and are compatible with the assember and linker.

For Floating Point support, the compiler includes both direct calls for the 8087 numeric data processor and software libraries. Double precision (IEEE) 64-bit format is supported with the 8087 coprocessor.

The compiler includes a C Integration Control System (ICS), specifically designed to automate the integration of the software written in C with the hardware specification of the prototype. The ICS program uses a list of hardware and software configuration parameters contained in the prototype description (ICS source file), which is provided by the user. ICS generates the interface code and linker command file to provide memory configuration, interrupt handling, and interrupt and program initialization.

C Debug is a real-time symbolic debugging tool that allows the programmer to use C-language constructs to control program execution, display and modify variables, and record and display information.

CLANDS is a complete software package. It includes the assembler, Language Directed Editor, compiler with ICS, and debug.

Literature is available through your local MDP sales representative.

Bobbie Kegley Product Program Manager

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

68XXX FASTER CHIPS

We are now shipping the 68008 probe (8300P38) with 10MHz parts and the 68010 probe (8300P40) with 12.5MHz. The 68000A (8300P39) probe already includes the 12.5MHz part. Probes shipped with the faster chip will have a serial number B03XXXX. Our emulator (8300E36) fully supports the faster chip.

Prior to this time, Motorola was not shipping quantity parts of the faster chip.

To upgrade existing probes, customers can replace the part with the faster chip.

Marilyn Hanson MDP Marketing

8085A-2 SUPPORT BECOMES STANDARD

Effective immediately, the 8085 Emulator and Probe will support a 10MHz clock for 5MHz operation. The 8085A-2 Microprocessor divides the applied clock by two, thus a crystal frequency of 10MHz results in a 5MHz CPU operation. This support has been provided by Option 15 in the past, but now it is a standard feature. **Option 15 is no longer orderable.** Customers wishing to upgrade their existing 8085A emulators and probes can use the Field Service Upgrade Kit.

PRODUCT	DESCRIPTION	AVAILABILITY
8300E06	8085A-2 Emulator and Control S/W	Stock
8300P06	8085A-2 Prototype Control Probe	Stock
V8085	8085A-2 Support	5 Weeks
856140B	8085A-2 Support	5 Weeks
020-1254-00	Field Service Upgrade	Contact Cust. Serv.

These parts are no longer available:

8300E06	8085A Emulator Option 15
8300P06	8085A Probe Option 15

John Owens Marketing Information Manager

ULTRIX SUPPORT

Ultrix is DEC's UNIX operating system. Our UNIX 4.2 software runs on Ultrix. DEC is selling, and supporting Ultrix on their complete line of VAXs, including MicroVAX.

Customers wanting VAX/UNIX will be attracted to the operating system being supported by DEC. TEK products are now available for UNIX 4.2.

NOTE:

We had a report that loading software from TU-58's under Ultrix can cause some problems. We will be offering the 9-track mag tape only, (not TU-58's) for new UNIX 4.2 products.

Marilyn Hanson MDP Marketing

ENHANCED Z80 AND Z80 CMOS EMULATION SUPPORT

Z80 CMOS Emulation and Probe support for the 8540 Integration Unit is now available. Additionally, a new version of the Z80 Emulator and Probe will replace the existing Z80 Emulator and Probe. The current has been depleted, thus, the new Emulator and Probe is now shipping.

NOMENCLATURE PRODUCT	DESCRIPTION	AVAILABILITY
8300E04	Z80 Emulator Includes Firmware	2 Weeks
8300P04	Z80 Probe	2 Weeks
8300P49	Z80 CMOS Probe requires the enhanced 8300E04 emulator *	2 Weeks
VZ80 OPT 12	Replaces 8300P04 with 8300P49 for CMOS Spt.	2 Weeks
856140A OPT 12	Replaces 8300P04 with 8300P49 for CMOS Spt.	2 Weeks
020-0790-07	8540 to 8550 Conversion Kit	CCS

* Note, The 8300P49 probe is supported by the 8300E04 Emulator with serial numbers B05XXX and above. The 8300P49 is not supported for use with Emulators with Serial Numbers B000000 to B049999.

ENHANCEMENTS

Active clock signal enhancements have been added to the Mobile Microprocessor Board. A jumper has been added to optionally ignore prototype power failure. A jumper has been added to optionally ignore prototype clock failure. A jumper has been added to optionally inhibit writes to the prototype when MREQ is always enabled to the prototype, and the emulator is in internal operation.

While most applications do not require these enhancements, they have been added to the new Z80 Emulator and Probe to support those users that previously required an emulator modification.

USER APPLICATIONS

The Toshiba CMOS Z80 is a 4MHz CMOS CPU that will be used in many application areas. Zilog now also produces a CMOS part. Some of the factors in selecting the Z80 or Z80 CMOS are:

FOR CMOS Z80

A need for a lower power consumption product
A need for noise immunity or radiation hardening
Availability of CMOS peripheral components
Availability of a proven architecture in CMOS

FOR Z80 AND CMOS Z80

A need to minimize design costs by retaining existing code in a new design
Availability of code generation tools
Availability of applications software

The Z80 chip is being incorporated into many new designs today. The designer base that has used and understands the Z80 is large, and where the application fits the capabilities of the Z80, the Z80 will continue to be used for years to come.

PRODUCT POSITIONING

The CMOS Z80 Emulator and Probe Utilizes the majority of the new Z80 Emulator and Probe design, thus, the CMOS Emulator and Probe is a well proven product.

No upgrade kits will be available for existing Emulators and Probes with Serial Numbers B000000 to B049999.

John Owens Marketing Information Manager

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

8540 EEPROM PATCH UPDATE, LEVEL 68

The intent of this article is to publish the current 8540 EEPROM patch information. In this article only the rompatches will be published. For additional information regarding rompatches refer to the 8540 System Users Manual, (part no. 070-3939-00), and the 8540 INTEGRATION UNIT EEPROM Patch Instruction Sheet (part no. 070-4287-06). Information regarding the rompatch command can be found in the Command Dictionary section of your 8540 System Users Manual.

The following is a complete list of the current 8540 EEPROM Patches.

```
rompatch -i 02fd0 000000 0 0
rompatch -i 02fd1 000000 0 1
rompatch 09a3f 1 1a5 /140100/2 68
rompatch 052a 2 41d /DEFLT/EX[] 02
rompatch 0e523 3 429 /DEFLT/EX[] 00
rompatch 04a03 4 18b /KERNL/INIT 0418
rompatch 0a6f2 5 28 /KERNL/INIT 3f2b4fc0
rompatch 02066 6 34f /KERNL/INIT 060107cf860117
rompatch 05240 7 190 /KERNL/GO 3bba
rompatch 09251 8 1ab /KERNL/GO 3bba
rompatch 0c145 9 0c /KERNL/DEVDB 9595b5b5
rompatch 0e33f 0a 39 /KERNL/DEVDB 80
rompatch 02688 0b 0d /KERNL/PCB.NMLO 3e
rompatch 02771 0c 06 /KERNL/PCB.TYPE 00
rompatch 0a84f 0d 30 /KERNL/START 95
rompatch 0cdb9 0e 3c /KERNL/RCMINT c7
rompatch 0d29a 0f 98 /KERNL/INIT 1a
rompatch 09126 10 24c /DEFLT/CONFIG[] 3f2490
rompatch 0a6bc 11 490 /DEFLT/CONFIG[] 20cc84980c8451175795
rompatch 0cef5 12 79 /KERNL/RESPTR 7f
rompatch 0102c 13 0 /DEFLT/DI[] 2306
rompatch 0334e 14 306 /DEFLT/DI[] 0623070b17000623071117181f2203
rompatch 0d641 15 0 /138200/3 2800
rompatch 05028 16 800 /138200/3 062807051700062807 0b17181f2330
rompatch 05761 17 0A4A /DEFLT/ROMPATCH[] 3f2dab
rompatch 09a4d 18 0DAB /DEFLT/ROMPATCH[] cc0b83cc0b840c0d8617
```

```
rompatch 0630a 19 21 /KERNL/DEVDB 03
rompatch 0d342 1a 5f0 /KERNL/EMUSVC 3635ff
rompatch 0c6d3 1b 11e2 /137800/1 42
rompatch 02346 1c 2d /KERNL/DEVDB 40
rompatch 0c80b 1d 51 /KERNL/DEVINT 1b
rompatch 026cf 1e 1748 /137800/1 30
rompatch 0f6cd 1f 1086 /137800/1 05
rompatch 09e2b 20 1747 /137800/1 c0c0c0c0c0c0
rompatch 08add 21 0e5 /KERNL/FH85B 38e8
rompatch 08996 22 166 /KERNL/INIT c0c0c00daa05cc0a08c0c0c0
rompatch 092ee 23 18d /KERNL/INIT 0e
rompatch 08e79 24 195 /KERNL/INIT 0a062a07fa1b04062b07023f29e90471cc0a0704a3
rompatch 05fa5 25 1aa /KERNL/INIT cc0a081f2b56c0c0c01824
rompatch 0be60 26 1de /KERNL/INIT 01017540
rompatch 0be85 27 356 /KERNL/INIT 04290504cdea0504000511cdea050406098945f061
rompatch 0ea55 28 36b /KERNL/INIT c884d4ee1b021fbf3f29f854cef4081f29b3
rompatch 0c5fc 29 5 /KERNL/CMDINT 31
rompatch 0ff51 2a 3d /KERNL/DEVHAN e6001808
rompatch 0fa01 2b 51 /KERNL/DEVHAN e6011808
rompatch 052e5 2c 100 /KERNL/QUESUB 0a
rompatch 03728 2d 3c /KERNL/ROMINT 2dc7
rompatch 04ae9 2e 280 /KERNL/GO 1f3fe9
rompatch 0ee30 2f 6ca /KERNL/GO cc9c14063f07f117070100001f3ba2
rompatch 04936 30 0b6 /DEFLT/CONFIG[] 10
rompatch 0145b 31 3bb /DEFLT/CONFIG[] 1f249ac01f249ec0
rompatch 0da57 32 49a /DEFLT/CONFIG[] 041e1b02042bc877
rompatch 04d76 33 4a2 /DEFLT/CONFIG[] 0c844318053f2286
rompatch 03217 34 4aa /DEFLT/CONFIG[] 1b033f22b208681f23e4
rompatch 02740 35 1c4 /KERNL/DEVINT 3fdc
rompatch 0d928 36 1fdc /KERNL/DEVINT 3f234f1f2227
rompatch 08cad 37 1adb /137800/1 030dfa12ec1a109c1b350605
rompatch 0aec0 38 1b05 /137800/1 07
rompatch 0c6c1 39 1b1f /137800/1 06
rompatch 0bde1 3a 1b36 /137800/1 060dfa12c12006000707
rompatch 07454 3b 0658 /137800/1 1f1bee
rompatch 09cf4 3c 1bee /137800/1 04c0d4d004ffd4f3d4f820d498d4d2d4a21f065b
rompatch 0d590 3d 0f10 /136500/0 05
rompatch 0e85a 3e 1004 /173601/0 05
rompatch 0101d 3f 102e /173601/0 05
rompatch 08a6e 40 040 /KERNL/FH85A 35d6
rompatch 0a79d 41 314 /KERNL/INTSRV 87201f3f39
rompatch 0dcb3 42 1f39 /KERNL/INTSRV 1a0404011b030c8738cc873a17
rompatch 06eb9 43 292 /138600/0 3f381e
rompatch 0f190 44 1816 /138600/0 6508cd018c17
rompatch 0cebe 45 5fa /138600/0 4f
rompatch 046b2 46 184 /138600/0 0f
rompatch 0cdba 47 12d1 /135300/0 1f3630
rompatch 04e22 48 0fc2 /135300/0 1f3610
rompatch 05a9c 49 1628 /135300/0 0508c9110990c0f52018040502
rompatch 09195 4a 1635 /135300/0 c9060904e11f32dc0033c4
rompatch 01cd2 4b 1608 /135300/0 0c004544086c14619c33510c01cb
rompatch 0ebac 4c 1616 /135300/0 44216c00959c334b1f316c
rompatch 0637d 4d 01fb /139000/0 1f397b
rompatch 0e2cf 4e 026e /139000/0 0c1987
rompatch 0ce47 4f 1973 /139000/0 cd028b0c0051cc19871f2206
rompatch 03f72 50 29e /138600/0 8bc0c0c0c0c0c0c0
rompatch 03f41 51 1071 /138801/0 03
rompatch 0a743 52 10d4 /138801/0 02
rompatch 0d6ec 53 067c /DEFLT/ROMPATCH[] 95
rompatch 04699 54 06a2 /DEFLT/ROMPATCH[] c0c00401cc0c0a1b43
rompatch 09077 55 00f4 /DEFLT/ROMPATCH[] 1f2db6
rompatch 06422 56 0db6 /DEFLT/ROMPATCH[] 9c2aec0c0d89ec0d929c2aec1f20f7
rompatch 061a2 57 44 /160700/1 1f2c62
rompatch 0916b 58 146 /160700/1 3f2fabc0
rompatch 0accc 59 3ab /160700/1 44030d8e4a45f061cc8e4ad4ee17
```

```
rompatch 065f6 5a 44 /160800/1 1f2c62
rompatch 0a40b 5b 146 /160800/1 3f2fabc0
rompatch 0d6cd 5c 3ab /160800/1 44030d8e4a45f061cc8e4ad4ee17
rompatch 0ccd7 5d 1d3e /136000/0 1f9fd6
rompatch 0f0f6 5e 1cd5 /146500/0 1f9fd6
rompatch 018d6 5f 0ee2 /136500/0 c0c0
rompatch 05e2b 60 1cb8 /136500/0 1f3fc5
rompatch 04b0c 61 1f40 /136500/0 0ffc98e56a9c3d40de3d57
rompatch 02de9 62 1450 /220100/0 1f35f0
rompatch 0718d 63 1567 /220100/0 053615040ff4b9e56a9c34d8de34e3
rompatch 057c8 64 19c8 /136301/0 1f3f5a
rompatch 0c1e2 65 1ed5 /136301/0 0ffa33e56a9c3a50de3a5b
rompatch 08bcc 66 1c70 /173601/0 1f3f19
rompatch 092e6 67 1e94 /173601/0 0ffcdbe56a9c3cf8de3d0f
rompatch 019b  68 815 /173601/0 0a54
```

Kevin King Applications Engineer

~ ~

8540 ROM CHECKSUMS

The intent of this article is to describe an easy means of verifying control firmware used with the 8540 Integration Unit.

Each of the proms used in the 8540 contains a checksum. One utility of the ROM-resident diagnostics is a routine that calculates the ROMs checksum and compares it to the checksum contained in the ROM. This routine when selected will display the ROMs part number and the checksum. If the checksum stored in the ROM and the calculated checksums do not match an error is displayed. However, if a part were labeled incorrectly or the diagnostics didn't detect that the ROM was installed, no error would be generated. A detailed description on how to select the ROM-resident checksum routine and a list of ROM part numbers with checksums follows.

To select the ROM-resident diagnostics enter the following command "sel diags". Once selected the ROM-resident diagnostics are menu driven. The following are a list of the menu selections required to enter the checksum diagnostic routine.

- 2-SELECT MODE
- 2-MEMORY and MEMORY FUNCTIONS TEST
- 6-SYSTEM ROM
- 1-ROMS ** default **

For the remaining menus select the default options by entering <CR>. Once selected the checksum diagnostic routine will display the part number and checksum of all the OS/40 and option ROMs that are installed.

The following is a list of current OS/40 and option ROMs. The ROMs are grouped by product description/option number.

OS/40 Operating System

160-1366-00 checksum = cdd9
160-1367-00 checksum = a01b
160-1368-00 checksum = eee5
160-1369-00 checksum = cc3c
160-1370-00 checksum = 84eb
160-1371-00 checksum = b22e
160-1372-00 checksum = 0cea

9900/9989 8300E33 01

160-1737-00 checksum = 52dd
160-1735-00 checksum = 1207
160-1734-00 checksum = c401
160-1736-01 checksum = 3a26

8086/88/87 8300P15/16 01,
8300P17/18 01

160-1373-00 checksum = d14e
160-1374-00 checksum = 243a
160-1375-00 checksum = d395
160-1376-00 checksum = 409a
160-1378-01 checksum = 07d2
160-1379-00 checksum = 604d
160-1380-00 checksum = b589

COMM Option 8540F01Z8001

160-1401-01 checksum = ca26
160-1361-00 checksum = 4124
Extended Hex Option 8540F20

160-1873-00 checksum = 2bb2

TTA Option 8540F03

160-1381-00 checksum = c503
160-1382-00 checksum = 1646
160-1388-01 checksum = 3462
160-1607-00 checksum = 82d8
PROM PROG. CONTROLLER, 8550F30 01

160-1487-00 checksum = 1a8b
160-1486-00 checksum = fa86

8080A 8300E01 01

160-1383-00 checksum = 043a
160-1385-00 checksum = a4b8

6800 8300E02 01

160-1391-00 checksum = 878d
160-1393-00 checksum = e056

Z80A 8300E04 01

160-1390-00 checksum = 31fb
160-1395-00 checksum = c8c5

8085 8300E06 01

160-1386-00 checksum = d43c
160-1389-00 checksum = 59ca

1802 8300E09 01

160-1962-00 checksum = 91e2

8048 8300E10 01

160-1396-00 checksum = 071f
160-1397-00 checksum = 04f5
160-1398-00 checksum = 7635

NSC800 8300P37 01

160-2233-00 checksum = fd9e
160-2234-00 checksum = 252d
160-2235-00 checksum = 18b3
160-2236-00 checksum = e579
160-2237-01 checksum = c49c
160-2238-00 checksum = d6df

160-1356-01 checksum = de92
160-1357-01 checksum = 46ac
160-1358-01 checksum = eeaf
160-1611-01 checksum = 0570
160-1612-01 checksum = 456b
160-1359-01 checksum = fa21

8300P20 01

160-1360-00 checksum = 2642

160-1362-00 checksum = e966
160-1608-00 checksum = 032b
160-1363-01 checksum = 42f1

Z8002 8300P22 01

160-1465-00 checksum = 520d
160-1466-00 checksum = c16d
160-1467-00 checksum = e618

160-1363-01 checksum = 42f1

68000 8300P26 01

160-1353-00 checksum = 5d60
160-1364-00 checksum = a206
160-1399-00 checksum = d863
160-1606-00 checksum = e418
160-1400-00 checksum = 6262

6809 8300P28 01

160-1354-00 checksum = aea4
160-1355-00 checksum = 7115
160-1365-00 checksum = 80d7

6801/68120 8300P29/P30 01

160-1822-00 checksum = ae6d
160-1823-00 checksum = 4e26
160-1824-00 checksum = 0efd

68XXX 8300P38/P39/P40 01

160-2197-00 checksum = b614
160-2198-00 checksum = cce1
160-2199-00 checksum = dd4d
160-2200-00 checksum = ff9d
160-2201-00 checksum = 08d9

80186/80188 8300P45/P46 01

160-2520-02 checksum = 60d8
160-2521-02 checksum = 238e
160-2522-02 checksum = c0f7
160-2524-01 checksum = 243c

1750A 8300P47

160-2565-00 checksum = 0681
160-2566-00 checksum = 81e8
160-2567-00 checksum = 1f3f
160-2568-00 checksum = 5f50
160-2681-00 checksum = 5e46

Kevin King Applications Engineer

~ ~

SA TOOLS - AVAILABLE FOR VMS

SA Tools for VMS is now shipping with 3 week delivery.

Enhancements

This excellent product, consistent with VMS style, released on schedule, has new features suggested by UNIX and TNIX users. With enhancements to the VMS version, you can easily decompose and partition your system.

- The DFD editor has *change* commands (e.g., reconnect dataflows, change a data item - like a process to a file), to expand and modify DFDs more easily.
- Child DFDs are created with parent dataflows included, freeing the user from trying to remember and enter them.
- You can show multiple DFDs on 4107/9 or 4115B while editing another DFD or a Mini-Spec or the data dictionary.

VMS Version

SA Tools runs on VAX/VMS Version 3.4 and on later releases of Version 3. As of AP601, it also runs on VMS V4.1. (see related article on page 18).

Terminals

The VMS version of SA Tools makes better use of Tek terminals 4107/9, because it uses another feature: viewports. With viewports you can put several DFDs on the screen at once, and also edit another DFD or (by limiting the dialog area) a Mini-Spec or the Data Dictionary.

Publicity

EDN published an article by Rodney Bell on Sturctured Analysis in their March 21 issue (page 251). It's titled 'Structured Analysis Aids in μC System Design'. Also, **Software News** covered SA Tools in a survey of analysis/design.

Customers

Companies have already purchased SA Tools on VMS for applications in DoD R&D, communications, DoD electronics, DoD contract specifications, oil service electronics, and other areas. 8560 and VAX/UNIX customers of SA Tools are developing applications in information systems, micro-electronics, TV, photography, industrial machinery, AVIONICS, and others.

Orderability

PRODUCT	OPT	DESCRIPTION	DELIVERY
STRUCTA		SA Tools (QRO)	
	1F	for VAX/VMS	(tape)
020-1391-00		Documentation Kit for SA Tools on VMS	2 Weeks

Rodney Bell Product Line Manager

~ ~

SOFTWARE VERSION LIST

The following is a listing of the current MDP software versions. If subscribers have not received their updates through Software Subscription Service updates, please contact your local sales representative. (See **FORWARD** for more details).

Software Name	Version #	Part Number
8086/8088 DIAGS	1.4	04-20-82
8300A01 ASM8085 1Z	4.0A	12-05-82
8300A02 ASM6800 1Z	4.0A	11-18-82
8300A04 ASMZ80 1Z	4.0A	12-05-82
8300A05 ASM9900	4.0A	12-05-82
8300A07 ASM3870	4.0B	12-05-82
8300A09 ASM1802 1Z	4.0A	12-05-82
8300A10 ASM8048 1Z	4.1A	12-05-82
8300A14 ASM6500	4.0A	10-10-80
8300A15 ASM8086/88	4.1B	01-30-81
8300A20 ASMZ8000	4.0	09-17-80
8300A26 ASM68000	4.0	10-17-84
8300A28 ASM6809	4.0	01-28-81
8300B05 ASM9900 1Z	01.04-35	06-22-82
8300B15 ASM8086 1Z	01.18-41	07-08-82
8300B19 ASM8051 1Z	01.09-61	12-05-82
8300B20 ASMZ8K 1Z	01.01-01	12-07-81
8300B26 ASM68K 1Z	01.15-65	01-07-83
8300B28 ASM6809 1Z	01.09-65	01-04-83
8300C01 ACEDIT 1Z	03.00-00	06-07-83
8300D15 8086 SBC	1.0	07-22-81
8300E01 8080A EMULATOR	2.1	08-13-82
8300E02 6800/6802 EMULATOR	2.1	08-13-82
8300E04 Z80A EMULATOR	2.1	07-18-83
8300E05 TMS9900 EMULATOR	1.0	12-10-80
8300E06 8085A EMULATOR	2.2	02-06-84
8300E07 3870/72/F8 EMULATOR	2.1	12-15-82
8300E09 1802 EMULATOR	2.1	10-12-82
8300E10 8048/21/41/22 EMULATOR	2.0	06-15-82
8300E14 6500/01 EMULATOR	1.1	10-09-81
8300E33 9900/89 EMULATOR	2.0	08-05-82
8300E38/40 80186/88 EMULATOR	2.1	10-22-84

8300G01 PAS8080 1Z	4.03	05-30-84
8300G02 PAS8086 1Z	01.01-07	10-12-82
8300H01 MDL8085 1Z	2.1A	12-05-82
8300H02 MDL6800 1Z	2.0	08-06-80
8300H02 MDL6800 1Z	2.1	10-05-81
8300H02 MDL6800 1Z	2.1A	12-05-82
8300P15/18 8086/88/87 EMULATOR	1.15	11-01-82
8300P20/22 Z8001/2 EMULATOR	1.9	03-30-82
8300P26 68000 EMULATOR	1.20	07-11-83
8300P28 6809 EMULATOR	2.0	03-30-82
8300P29/30 6801/120 EMULATOR	1.1	11-01-82
8300P37 NSC-800 EMULATOR	1.0	11-09-83
8300P38/39/40 68000/08/10 EMULATOR	2.0	08-02-83
8300S01 RT11/50 O.S.	1.00	09-07-82
8540/50/60/U04 INTEL COMM.	1.00	09-17-82
8540/50/60/U04 INTEL COMM.	1.00	09-17-82
8550 GUIDE	1.0	06-15-82
8550 SYSTEM DIAGS	3.0	08-16-84
8550F20 EXTENEDED HEX	1.0	12-10-82
8550F30 PROM PROGRAMER	2.1	03-30-82
8550S01 RTDOS UTILITY	1.00	08-20-82
8550U04 EHEX COMMAND	1.0	09-17-82
8560 LINKER/LSTR/LIBGEN	02.02-00	06-22-82
8560/61 SYSTEM DIAGS	01.03-01	07-11-84
8560B01 ASM8085 1A	02.00-16	05-18-84
8560B02 ASM6800 1A	01.03-18	02-15-82
8560B04 ASMZ80 1A	03.00-16	01-05-84
8560B05 ASM9900 1A	01.04-34	06-04-82
8560B09 ASM1802 1A	01.01-56	11-02-82
8560B10 ASM8048 1A	02.04-08	05-04-83
8560B15 ASM8086 1A	02.04-11	09-23-83
8560B16 ASMZ8K 1A	02.03-09	08-17-83
8560B17 ASM68K 1A	02.02-15	12-20-83
8560B18 ASM6809 1A	02.01-09	05-24-83
8560B19 ASM8051 1A	01.08-56	12-05-82
8560D02 PDB8086 1A	01.11-00	08-16-83
8560D03 PDBZ8K 1A	01.05-00	12-17-82
8560D04 PDB68K 1A	02.03-01	03-14-84
8560F20 PLDEDIT 1A	02.04-00	05-18-84
8560F21 ACEDIT 1A	03.00-00	06-07-83
8560G02 PAS8086 1A *2	02.10-00	06-28-84
8560G03 PASZ8K 1A	01.09-08	01-21-83
8560G04 PAS68K 1A	01.01-14	04-22-83
8560G04 PAS68K 1A	02.08-01	12-12-83
8560U01 TEXT PROCESSING	2.0	06-16-83
8560U02 NATIVE PROGRAMMING	2.0	06-16-83
8560U03 AUXILIARY UTILITIES *2	2.0	06-16-83
8560U04 EHEX COMMAND	3.4	09-17-82
8560U05 UNICOM	1.01	11-21-83
AS8086 1F	02.11-00	11-06-84
ASM1750 1E *3	02.00-15	09-18-84
ASM1750 1F	02.00-15	09-18-84
ASM6809 1B *2	02.00-16	03-04-85
ASM6809 1C	02.00-16	03-05-85

ASM6809 1D *2	01.00-06	08-08-83
ASM6809 1E *2	02.01-09	06-28-84
ASM6809 1F	02.01-09	07-02-84
ASM6809 1G *2	Y01.00-00	11-28-83
ASM6809 1H *2	01.00-06	01-28-85
ASM6809 1J	01.00-06	01-28-85
ASM68K 1B *2	02.04-16	01-10-85
ASM68K 1C	02.04-16	01-10-85
ASM68K 1E *2	02.03-09	06-28-84
ASM68K 1F	Y01.00-00	11-28-83
ASM68K 1F	02.03-09	07-02-84
ASM68K 1G *2	Y01.00-00	11-28-83
ASM68K 1H *2	02.04-16	01-28-85
ASM68K 1J	02.04-16	01-28-85
ASM7720A 1A	01.00-16	03-04-85
ASM7809 1A	01.00-16	07-19-84
ASM7811 1A	01.00-16	07-05-84
ASM78C06 1A	01.00-16	04-13-84
ASM8085 1B *2	02.00-16	03-04-85
ASM8085 1C	02.00-16	03-05-85
ASM8085 1D *2	01.00-06	07-21-83
ASM8085 1E *2	02.01-09	07-10-84
ASM8085 1F	02.01-09	07-11-84
ASM8085 1G *2	Y01.00-00	11-28-83
ASM8085 1H *2	01.00-06	01-28-85
ASM8085 1J	01.00-06	01-28-85
ASM8086 1B *2	02.07-16	01-10-85
ASM8086 1C	01.01-08	12-22-83
ASM8086 1C	02.07-16	01-09-85
ASM8086 1C	02.07-16	01-10-85
ASM8086 1D *2	01.01-08	12-22-83
ASM8086 1E *3	02.07-09	07-02-84
ASM8086 1F	02.07-09	07-03-84
ASM8086 1G *3	Y01.00-00	11-28-83
ASM8086 1H *2	02.07-16	01-28-85
ASM8086 1J	02.07-16	01-28-85
ASMZ80 1B *2	02.00-16	02-14-85
ASMZ80 1C	02.00-16	02-20-85
ASMZ80 1D *2	01.00-06	07-21-83
ASMZ80 1E *2	02.01-09	06-28-84
ASMZ80 1F	02.01-09	07-02-84
ASMZ80 1G	Y01.00-00	11-28-83
ASMZ80 1H *2	01.00-06	01-28-85
ASMZ80 1J	01.00-06	01-28-85
CCC68K 1A	02.09-00	02-25-85
CCC68K 1B *4	02.09-00	02-14-85
CCC68K 1B VOL 2 *2	01.00-00	04-18-84
CCC68K 1C	02.09-00	02-18-85
CCC68K 1D VOL 1	01.00-01	07-19-83
CCC68K 1D VOL 2 *2	01.00-01	07-19-83
CCC68K 1E *6	02.06-00	01-28-85
CCC68K 1E VOL 2 *2	01.01-00	09-28-83
CCC68K 1F	02.06-00	01-28-84
CCC68K 1G VOL 1	01.01-00	11-16-83

CCC68K 1G VOL 2 *2	01.01-00	11-16-83
CCC68K 1H *4	02.05-00	01-28-85
CCC68K 1J	02.05-00	01-28-85
CDB68K 1A	01.08-00	03-05-85
CDB68K 1B	01.09-00	02-25-85
CDB68K 1C	01.04-00	01-11-85
CDB68K 1C	01.09-00	02-25-85
CDB68K 1E	01.10-00	03-14-85
CDB68K 1F	01.10-00	03-14-85
CDB68K 1H	01.04-00	02-25-85
CDB68K 1J	01.04-00	02-25-85
CLDEDIT 1A	01.07-00	05-17-85
CLDEDIT 1B *2	01.04-00	04-08-85
CLDEDIT 1C	01.04-00	04-08-85
CLDEDIT 1E *2	02.01-00	08-09-84
CLDEDIT 1F	02.01-00	08-09-84
CLDEDIT 1H	01.03-00	04-08-84
CLDEDIT 1J	01.03-00	04-08-84
COLORKY 1B *2	1.2	03-26-84
COLORKY 1C	1.2	03-22-84
COLORKY 1F	02.00-01	08-16-84
DDL 1A	2.0	08-09-84
DDL 1B *2	2.0	08-09-84
DDL 1C	2.0	08-09-84
DOS/50 VERSION 1 O/S	1.1A	03-26-82
DOS/50 VERSION 2 OPERATING SYSTEM	2.1A	11-16-82
ICOM40 1B *2	2.0	01-10-85
ICOM40 1C	02.00-00	01-10-85
ICOM40 1D *2	1.0	08-02-83
ICOM40 1E	02.02-00	02-11-85
ICOM40 1F	04.04-00	05-31-85
ICOM40 1G	1.0	11-16-84
ICOM40 1H *2	1.0	12-20-84
ICOM40 1J	1.0	01-09-85
ICOM40 1Y	01.03-00	05-03-85
ICOM40 3C	02.00-00	01-09-85
ICOM40 3D *2	1.0	08-02-83
ICOM40 3E	1.0	11-04-83
ICOM40 3F	1.0	11-04-83
ICOM40 3G	1.0	11-16-83
LAS LINKER/LIBGEN	02.08-00	09-07-82
MDL8085 1A	3.0	02-06-84
MDLZ80 1A	3.0	02-17-84
MDP-FTE FIXED DISC DIAGS	1.00	04-23-83
MICROLAB 1 3870/3872	1.9	01-30-81
MICROLAB 1 MCS48	1.1	01-30-81
MICROLAB MONITOR 6801/120	1.0	10-05-82
PAS68K 1D *5	01.02-00	02-16-84
PAS68K 1E *5	02.03-00	08-30-84
PAS68K 1F	02.03-00	09-04-84
PAS68K 1H *4	01.02-00	01-28-85
PAS68K 1J	01.02-00	01-28-85
PAS8086 1C	02.13-00	02-11-85
PAS8086 1J	02.13-00	02-11-85

PDB68K 1D	02.01-00	01-12-84
PDB68K 1F	02.02-00	04-26-85
PDB68K 1H	02.01-00	01-28-85
PDB68K 1J	02.01-00	01-28-85
PDB68K	02.02-00	04-22-85
PLDEDIT 1B	01.05-00	02-11-85
PLDEDIT 1C	01.05-00	02-11-85
PLDEDIT 1D	01.02-00	09-13-83
PLDEDIT 1E *2	02.00-00	08-09-84
PLDEDIT 1F	02.00-00	08-09-84
PLDEDIT 1G	01.03-00	11-16-83
PLDEDIT 1H	01.05-00	01-28-85
PLDEDIT 1J	01.05-00	01-28-85
PVRX68K 1A	03.00-00	05-03-85
STRUCTA 1A	01.04	08-16-84
STRUCTA 1B *2	01.00	08-09-84
STRUCTA 1C	01.00	08-09-84
STRUCTA 1F	01.01-00	06-05-85
TNIX 1.2+ TO 2.0 UPDATE *3	2.00	06-20-83
TNIX 1.2/1.3 UPDATE	1.00	09-29-82
TNIX 2.0 TO 2.1 UPDATE *3	2.10	04-26-84
TNIX OPERATING SYSTEM *5	2.1	05-01-84
TNIX REVISION	2.1B	03-29-85
TTA/HLP 1A	1.0	01-06-84

Myrl Kwiatkowski MDP Marketing

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

APPLICATIONS SECTION

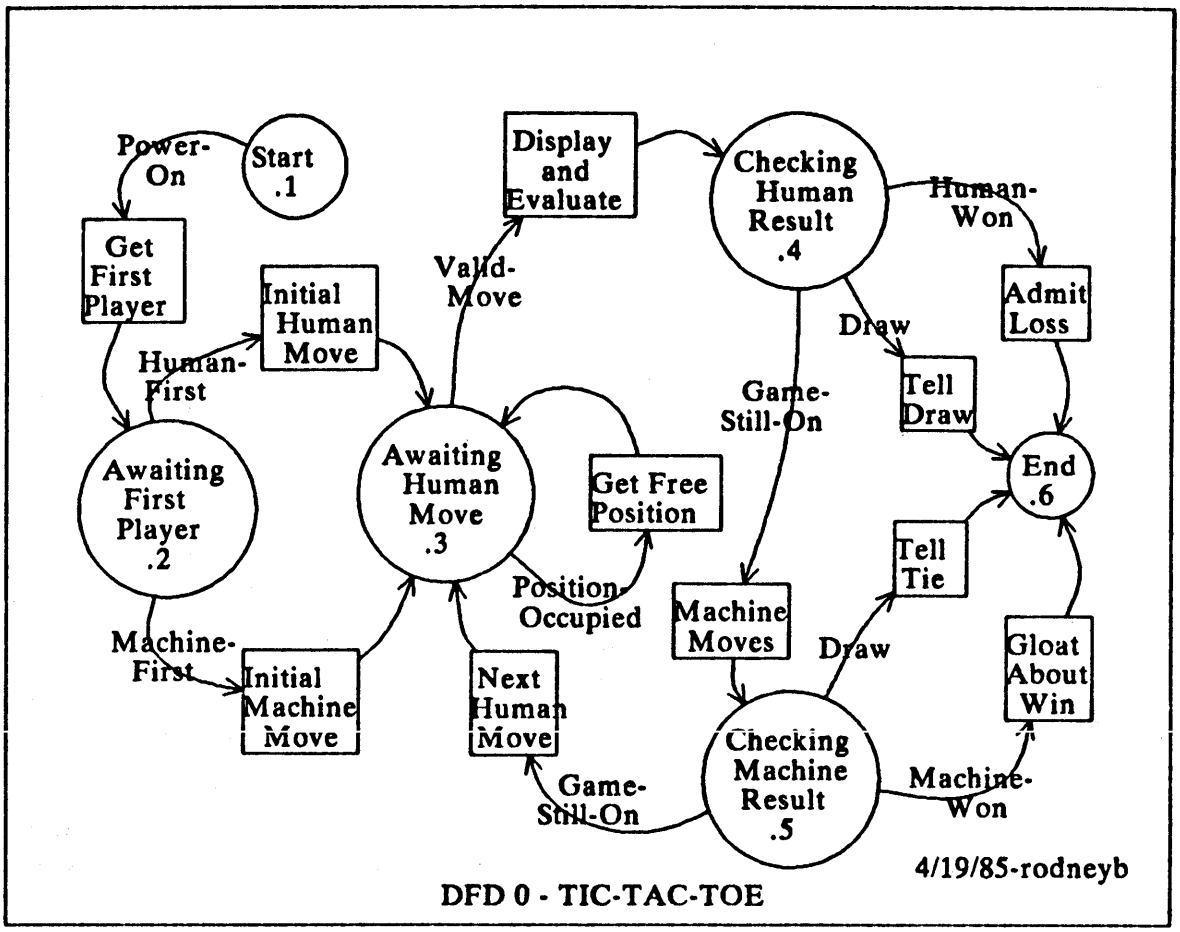
SA TOOLS: USE TO SPECIFY REAL-TIME SYSTEMS

SA Tools is well suited for specifying the data and functional aspects of real-time systems. Tektronix engineers have been doing so for several years. SA Tools can also be adapted for specifying control and timing aspects of real-time systems.

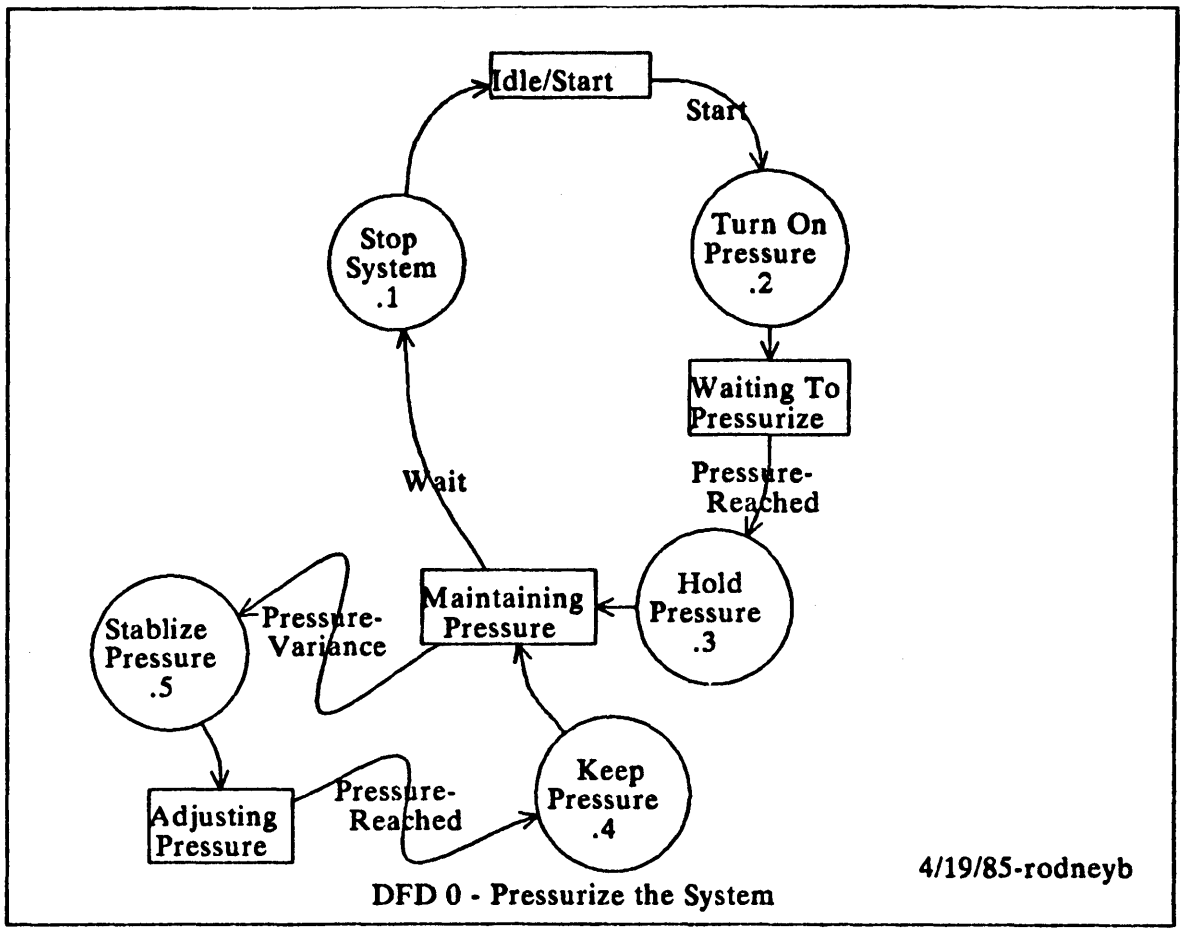
Integrated methods to specify both the data and control aspects of a system are just now being developed. MDP is developing support for such methods through relationships with customers and training companies.

One successful real-time method was developed by Lear Seigler Instruments and is currently used by Boeing Commercial Airplane Computers. It has been proven on a successful project, the Flight Management Computer System for the Boeing 737. Contact Rodney Bell, MDP Marketing for additional information.

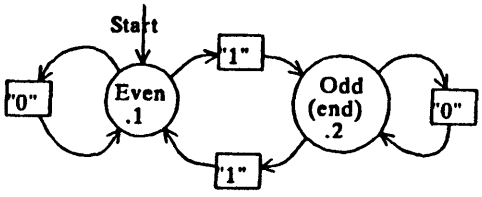
SA Tools can be adapted for analyzing real-time systems through the open-ended quality of SA Tools and adapting the DFD editor to do control diagrams. Since SA Tools use ASCII files of published format, customers can insert comments on real-time aspects and add their own tools to analyze this information. With certain conventions, the DFD editor can draw state transition diagrams or finite state machines.



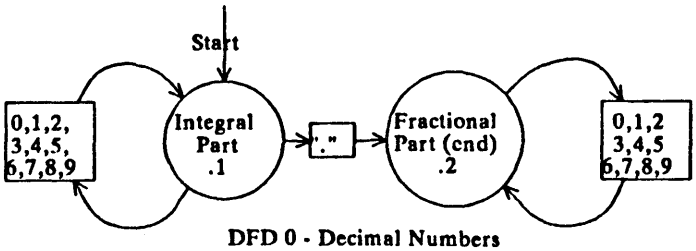
State Transition Diagrams. This diagram represents control states and the transitions between them. A new state is reached when a certain input occurs. During the transition, some action may also be driven. Two conventions are possible. In the first example, a TIC-TAC-TOE machine, states are represented by circles and actions by rectangles.
On the next page, a System Pressurizer states are represented by rectangles and actions by circles. In both, the transition and input are represented by labeled, directed lines connecting the states.



Finite State Machines. This graphical model is like the Control diagram, except there are no actions represented. The conventions used in the two following examples, a Parity Checker and a Decimal Number Acceptor, use circles for states, directed lines for transitions, and rectangles for inputs. Final states are denoted by a parenthetical "end" remark in the state name.



DFD 0 - Parity Checker



DFD 0 - Decimal Numbers

Rodney Bell Product Line Manager

GET THE BEST USE OF SA TOOLS

Hosts, Terminals, & Copiers

How does the new 4105A enhance the utility of SA Tools?

The 4105A supports new monochrome copy devices: the HP Thinkjet, the Tek 4644, the Epson FX-80, and Centronics-compatible copiers. The 4105A also supports the Tek 4691 and 4692 ink-jet color copiers. This increases your options with respect to cost, quality, and speed of copiers. There is a field upgrade kit also.

Can I obtain the source for SA Tools for a reasonable price?

Yes, we now offer two license agreements to suit both large and small users. A new "Progressive Rights" agreement offers an inexpensive pay-as-you-go arrangement for users with few systems. A conventional "Unrestricted Use" agreement offers site license for users with many systems for a single fee.

With the Progressive Rights agreement, a relatively low initial fee licenses use of the source on one system. Then for fees similar to those for standard SA Tools products the user can license a derived binary or another copy of the source for use on designated systems.

The source for SA Tools can be ported to another UNIX BSD4.2 system with a few days effort. MDP will offer support for such users by separate agreement with availability of source updates and consulting fees similar to AE fees. With these source agreements, you can now have SA Tools on systems besides 856X and VAX.

How can I use an Imagen laser printer with the 856X and SA Tools?

Imagen sells laser printers which can be used to copy DFDs. They offer high speed and high quality monochrome copies. Contact Imagen at (415) 960-0714 (California) for the name of the nearest distributor.

Plots of data flow diagrams may be obtained on an Imagen 8/300 laser printer connected to an 8560. Additional software from Imagen is not required. The following procedures are necessary to obtain the plots.

The Imagen 8/300 should be connected to one of the printer ports of the 8560. For this example let us assume the Imagen is connected to printer port 1. An additional terminal must be connected to the Imagen (at 9600 baud) to serve as a console for the Imagen. This is necessary to properly configure the Imagen to work with the 8560. The following steps should be performed:

1. Power up the Imagen

2. When the Imagen is powered up, the console will display a message asking if a new configuration is desired. The user has a few seconds to hit a key on the console to indicate that a new configuration is desired.

3. At this point the file name for the configuration file must be entered. Enter "configurator".

4. A system menu will be displayed allowing the user to configure the Imagen. Enter "8/300" to define the laser printer model.
5. Select serial byte stream communications.
6. Set baud rate to 2400 (baud rate of printer port).
7. Set simple printer emulation "off".
8. Set the EOF character to "0x04".
9. Set quoting character to "0x02".
10. Set 8-bit data mode "off".
11. Set Ignore any non-printing characters "off".
12. Set XON/XOFF flow control "on".

The above procedure only has to be performed once. Subsequent use of the Imagen will use the settings established above.

A plot of a data flow diagram is produced using the showsa command. For example,

```
TERM=4014 showsa dfd1 >dfd.plot
```

The file dfd.plot will contain a plot of dfd1 with 4014 style commands. The Imagen can process 4014 commands. A few characters must precede the dfd.plot file and an EOF character must follow the file. The file header must contain the following:

```
@document(language tektronix)
```

An end of line is NOT allowed between the above characters and the first character from the dfd.plot file. This header may be created using the cat command as follows:

```
cat >header
@document(language tektronix) ctrl-D ctrl-D
```

NOTE: The last line must not contain blanks, they were used above for clarity.

A trailer containing an EOF (i.e., - 0x04) must be attached to the plot file. The trailer also cannot contain an end of line. The trailer may be created as follows:

```
cat >temp
# ctrl-D ctrl-D
```

This will create a file with a # character and nothing else. The # character may then be changed to an EOF using the tr command, e.g., -

```
tr '#' ' 04' <temp >trailer
```

The cat command is now used to output the plot information to the Imagen laser printer. The print spooler command lpr may not be used as this command will insert other characters in the stream of characters. The additional characters will cause the Imagen controller to go haywire. An example of a cat command would be:

```
cat header dfd.plot trailer >/dev/lp1
```

A full page plot will be obtained using this technique. Obviously, if more than one user is using the 8560, they must coordinate use of the Imagen because a print spooler is not being used. If a job header is desired along with the dfd plot, then the header should be created as shown below:

@document(language tektronix,jobheader on)

Imagen does have a spooler (i.e., - ipr) which can be used with their laser printer. Tektronix, however, is not distributing this software and does not have an agreement with Imagen for the distribution of the software. At present, the need does not appear to warrant this distribution arrangement.

How can I use an Imagen laser printer with VAX-UNIX?

There are two ways. Use the driver software (**ipr**) provided by Imagen and the Tek 4010/14

$ TERM=4014 showsa dfd0 | ipr -Ltektronix -Djobheader

Or use the AT&T Documentors Workbench product (**pltroff and ditroff**) and the SA Tools routine **dfdtoplot** that converts DFDs to UNIX plot format:

$ dfdtoplot dfd0| pltroff | ditroff -ms

Are there any known bugs in SA Tools for VMS V3.4?

Yes. SA Tools for VMS V3.4 won't run on VAX clusters. There is another problem with defining USER and TERM logical names. If these aren't defined a core dump will result. To prevent the dump, the system should define both USER and TERM as logical names automatically or the system manager should inform each person using the tools to define them in their LOGIN.COM files. This is fixed in the VMS 4.0 version of SA Tools.

What is the advantage of the 4692 copier?

With the appropriate SACAP entry, you can have multiple DFDs on the screen at once.

The 4692 is a Tek designed and built ink-jet copier that you can attach four terminals to. It is more reliable than the 5695 and produces better quality copies. With an additional rasterizer, one can copy DFDs with much higher resolution.

What new features does SA Tools for VMS offer?

With enhancements to the VMS version, you can more easily decompose and partition your system.

- The DFD editor has *change* commands (e.g., reconnect dataflows, change a data item - like a process to a file), to expand and modify DFDs more easily.

- Child DFDs are created with parent dataflows included, freeing the user from trying to remember and enter them.
- You can show multiple DFDs on 4107/9 or 4115B while editing another DFD or a Mini-Spec or the data dictionary.
- Evaluate Data Dictionary identify undefined component names of definitions and fix adds those as TO-BE-DEFINED's.

DFD Editing

How can I see more than one DFD at a time on the screen?

By using SA Tools on VMS with 4107/9 or 4115B. DEFINE TERM as 4107v and use SA/EDIT or SA/SHOW with multiple files in the invocation. This procedure uses the four viewports set up in the SACAP entry for 4107v. Also, the visit command will use these viewports to edit the children DFDs. You can create other viewport arrangements by defining the VP parameter of a local SACAP.

What is an easy way to get all the data flows from the parent DFD drawn onto the child DFD?

Using SA Tools on VMS, simply visit the undefined child process and indicate that it is to be a DFD. The DFD editor automatically draws in all the parent dataflows connected to a single unnamed process on the new child DFD. This eliminates remembering or copying down the names of parent dataflows when you're drawing the child DFD.

I know how to move the elements of a DFD and change their names, but how can I change what they are or what they connect?

When using SA Tools on VMS, use the new change commands of the DFD editor. Swap (~) will reverse the direction of the dataflow on which the cross-hair is positioned; it also changes the level of a file (one- or two- lines). Using the control key while pressing D or O allows you to change the destination or origin of the dataflow on which the cross-hair is positioned. Using the control key while pressing P, F, E, or B allows you to change to; (Process, File, External, or Boundary Point), whatever data item on which the cross-hair is positioned.

What is the maximum number of processes a DFD will hold?

On the 856X, about 90 total processes, externals, dataflows, files, and boundary points. On VAX-UNIX, 255 processes is the limit. On VAX-VMS, 4096 processes is the limit. We've experimented with DFDs containing 500 and 255 dataflows for prospects that want to use the DFD editor to model networks. The major impact is that the editor takes longer to initially draw the DFD and a little to longer respond to changes.

How many levels of DFDs are possible with SA Tools on VMS?

Within a directory on VMS3.X, five (5) levels plus the top one, O.DFD. On VMS 4.X, it's nineteen (19) levels. These limites are due to the VMS limit on the length of file names, 9 characters before the prefix on VMs 3.X, and 39 on VMS 4.X. You can extend these limits by creating sub-directories to contain sub-trees of the DFD hierarchy. Review the technique 'Partitioning the Specification' in the 'Mastering the Tools' section of the manual.

How can I change the title of 0.DFD?

By editing the file with a text editor.

Can I traverse and view the DFD hierarchy while protecting the DFDs from change?

You can't absolutely. The visit command of the DFD editor requires write permission for the file it is working on. You must exercise care not to make any changes while in the editor. Alternatively, use the show function on a series of files, and on VMS use the viewport feature to display them simultaneously.

How can I remove or change the date-author field in a DFD copy?

Edit the DFD with a text editor to change those lines, then use the show function to make copies.

How can I position the dataflow names on the connecting line?

By positioning route points. The dataflow name goes on the centermost route point when there is an odd number, and between the two middle points when there is an even number.

How can I make a large DFD?

Use a 4107/9 or 411X/412X terminal. Define TERM as 4107s (a special SACAP entry with samll text size), or create your own SACAP entries with small text. Use the terminal's zoom/pan features to edit or view.

Why does the DFD editor limit the number of data flows and processes on a DFD?

It DOESN'T. This is only a warning that the DFD is more complex than the guidelines of DeMarco's Structured Analysis suggest.

How can I use the swap (~) command of the DFD editor when connected to the VAX-VMS by cu?

According to the cu man page for UNIX V7, a line containing the first two characters "~ ~ " will send one tilde, "~ ", followed by the remaining characters on the line. This implies that to send a tilde while using the SA editor through cu, you must type a tilde before the tilde gin report is sent.

To do this on non-VMS systems, you must cancel the gin (hit the Cancel, shifted Setup, key), type a tilde, go into setup mode and place the terminal in local mode (i.e., local yes), exit setup mode, type ESC CTRL-Z (which should turn on the cross-hairs), enter setup mode, get out of local mode, exit setup mode and then type the tilde.

On VMS the above also works, but there is a better way. You can change your SACAP entry for the terminal that you are using to use gin input type 1, along with changing your initialization string to set the EOL string to be null. What this does is disable the terminal from sending a carriage return at the end of gin reports. The tilde is only significant if it is the first character on the line. Since the terminal no longer ends the gin report with a carraige return, the only time that you can't use the tilde command is after responding to a prompt and ending it with a return.

There is a side effect to the method on VMS. After invoking the editor, you must wait a second, hit a return, wait another second, and hit another return. This is to terminate two reports sent from the terminal.

Mini-Specs and Data Dictionary

How can SA Tools be used for more comprehensive data dictionary analysis?

SA Tools includes the primitive tools needed for most analysis of and reports on the data of a system. The user or Tektronix Application Engineer (on consulting time) can produce more complex analysis/reporting tools using these primitives, the UNIX shell or VMS DCL, and text processing packages. SA Tools offers a Data Dictionary in ASCII form that can be accessed by custom tools, rather than an inaccessible proprietary database.

How can I find out whether the components of a data definition are defined? How can I update the Data Dictionary to include definitions of components?

On VMS, simply use the evaluate and fix commands in the Data Dictionary. On UNIX and TNIX, you must do this by inspection.

Can I specify a DCL file as the text editor for SA Tools on VMS? Can I specify different ones for MS and DD editing?

Yes and no, the text editor can be a DCL (job control language for VMS) file. This might be useful, for example, to enter the MS or DD in a standard form. While you cannot specify a different text editor for MS and DD, the DCL file could make the distinction and call the right one.

How can I define different classes of definitions in the Data Dictionary?

Use the comment fields to denote whether the definition is of a dataflow, file, or a component. Process "definitions" are contained in the Mini-Spec.

Can there be Mini-Specs at other levels of the specification?

Yes, but this is not standard SA methodology. Give the MS the same number as the DFD. Evaluating that MS will compare it to the parent DFD; fixing it will also. Using visit from the parent will take you to the DFD. Use these Mini-Specs to describe or comment on the activity in the associated DFD.

How can I show data type information in the Data Dictionary?

Use literals, as in "0..255" or "a | b | ... | z" or "{red, green, blue}".

Compatibility with Other Products

Will SA Tools work with version control products like RCS, and build products like make?

Yes. Simply extract and store the SA Spec files from the version control system before or after using the SA Tools on them. However, don't expect the visit command to do this. Keep in mind, such intermixed use of version control and SA Tools is not supported.

How will SA Tools work through the 8540?

Connect as usual. Set the 410X terminal's XMTL parameter to 4800 instead of the factory setting of 19200. Otherwise, the 8540 will occasionally miss an incoming character when editing DFDs and the editor will hang-up.

Why does ColorKey+ sometimes run out of memory on an 8560? How can I avoid this?

The use of too many scripts can take up almost all the available memory in the 64K allocated. Build another CK+ script and leave out some of the tools. A good idea is to have several overlapping scripts. For example, use SA Tools and language tools in one script, language and debugging tools in another.

Will SA Tools work with the new 412X family of terminals?

Yes. The 412X family uses 4115B features.

What Tek terminal emulators are available on the IBM/PC?

The below listed emulators let one use SA Tools with a PC acting as a terminal attached to a VAX. MDP has evaluated one from Grafpoint, seen another from MicroPlot, and know of several others.

Vendor	Product	US Price	Limits	Evaluated
Grafpoint 408-249-7951	TGRAF-05 for 4105	$395	no function keys dialog with graphics	yes, with 3 graphic boards
MicroPlot 614-882-4786	PC-PLOT for 4010	$95	?	observed
FTG 714-995-3900	EMU-TEK for 40XX for 4105	$395	?	no available soon
CompuView 313-996-1299	VGRAPH for 4010	$120	?	no

Must I install SA as a global DCL verb on VMS?

We recommend it. The alternative of setting up a local DCL verb doesn't support spawned jobs. Thus, DD evaluate and fix tools won't work; nor will ColorKey+.

How can I move SA files from 8560 to VAX/VMS?

The following steps are suggested for porting SA files developed on an 8560/TNIX system to a VAX/VMS system.

1 An RS232 line must be connected between the 8560 and the VAX.

2 KERMIT, a program for copying files between host computers, must be installed on both the 8560 and the VAX. KERMIT is available in source and 856X binary in MDP User Group Library Volume 4 and 5.

3 On the 8560 all file names must be renamed to be compatible with VMS. 8560 DFD or MS names longer than 12 characters cannot be converted to equivalent VMS names, due to limit on VMS3.X file name length. The DD can easily be renamed with the command "ln dd DD.LIS". The shell script can be used to create correct VMS names from 8560 DFD and MS names. The VMS names are created as links to the 8560 files.

```
for i in dfd* ms*
do
        x='echo $i | sed -e 's/[.]/X/g' -e 's/dfd)/1.DFD/' -e 's/ms)/1.MS/'
        echo "$i - > $x"
        ln $i $x

done
```

4 Use KERMIT to copy all the VMS-named files from the 8560 to the VAX. The VMS files should all be text files (variable record length).

5 On the VAX the data dictionary can be converted into its correct form by the command SA/PARSEDD/INPUT=DD.LIS/OUTPUT=SA.DD.

How much load does SA Tools place on the VAX-VMS?

Many user's VAXs are already heavily used and they're concerned how SA Tools will impact their operations.

A rough guideline is that the DFD editor is like a text editor working on files of a few KBytes (probably a little more processing, a little less data transfer, and about as interactive).

The DD tools are like an compiler working on a source file of length similar to the Dictionary and 'include' files of number and length similar to the DFDs. The DFD/MS evaluate and fix tools are similar to file conversion utilities; they are fast and have little I/O.

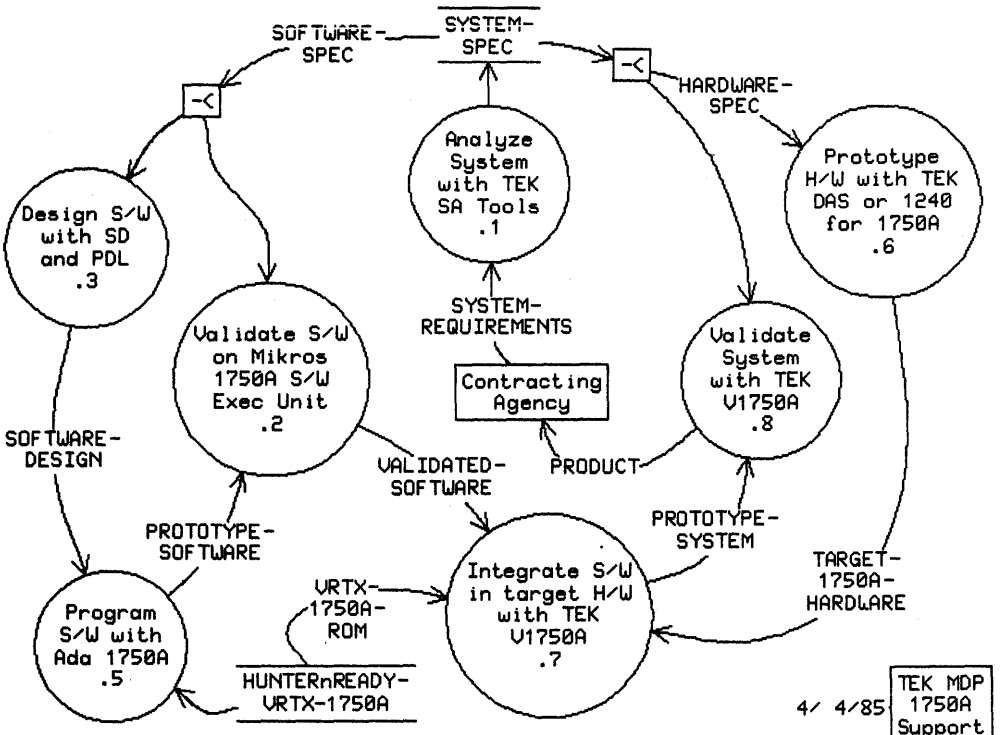
Will SA Tools work with DEC terminals and copiers?

Yes, but not as well as with Tek terminals and copiers. DEC's VT240/1 graphics terminal and LN01S laser printer have Tek 4010/14 modes. When set up in these Tek-compatible modes the VT240 and LN01S let you draw and copy DFDs. While Tek peripherals have the advantages of the 41XX and 469X features, the DEC peripherals will work.

One difference with the VT240 is that moves/deletes are done by putting an "X" on the old item until the screen is redrawn. ColorKey+ may work by using the VT240 in VT100 mode except for DFD editing in 4010/14 mode. Use the 4010 or 4014 SACAP entry and switch between VT100, and 4010/14 modes in the initialization and termination strings of the SACAP entry. With any Tek-compatible peripheral, try it with SA Tools before your purchase.

What is the connection between SA Tools and MDP's 1750 Support?

MDP offers a broad range of support for 1750 as depicted in the following diagram from a 5695. SA Tools is used at the start to analyze requirements and specify the 1750 system to be designed.



DFD 0 - Develop 1750 Application with MDP Products

Rodney Bell Product Line Manager

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

CDB COMMAND OPTIONS

The CDB command offers two options to decrease time of invocation. If a user needs to interrupt a debugging session, but does not intend to recompile and change the load module, the -P and -S options on the command line will be of great interest. For example, if the program resides in the emulator, having been previously downloaded, the -S specifies that only the symbols from the load module be loaded into the debugger. The object code is not downloaded to the emulator.

-P on the invocation line specifies that a permanent symbol table will be used. The name of the symbol table is the name of the load module with .SM appended. If there is no file .SM in your directory, the debugger will create one from the load module. The next time you invoke CDB, the .SM file will be used unless the load module has a more recent modification date, (see CDB User's Manual page 3-2 for more information).

Future versions of PASCAL Debug will also have these features added.

Marilyn Hanson MDP Marketing

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

CLDE: PARSE COMMAND EXTENSIONS

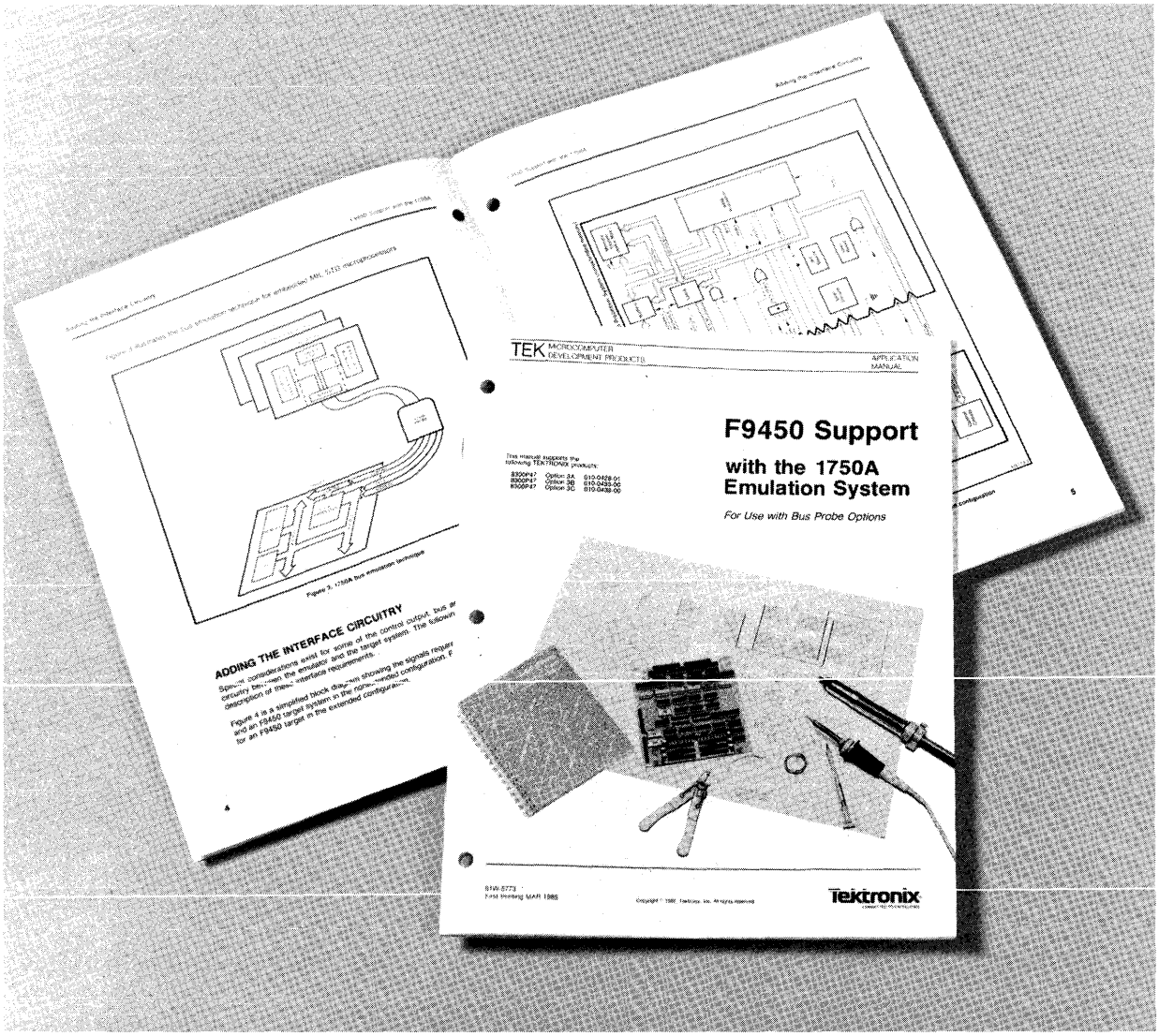
The CLDE Editor includes a couple of extensions that are not available in the PASCAL Editor. The PARSE command actually includes some preprocessing capabilities. #DEFINE statements are expanded, and the expanded syntax checked. Also, if you have #INCLUDE file in your source file, the PARSE command will parse the "include file" along with the C source. That means, if you have #DEFINEs in the "include file", it will also expand those "defines".

#INCLUDE <stdio.h> is used in many programs for standard I/O support. CLDE will parse that file if it is included. Therefore, the pathname must be defined or an error message will be given when you use the PARSE command. $TEKPATH must be set up, as well as the uP variable and the LANDS variable.

Marilyn Hanson MDP Marketing

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

F9450 APPLICATION MANUAL



A new Application Manual is available that shows how to develop a "bus interface" for the 1750A emulator in an F9450 application. The Bus probing technique is used when the socket is not available to accommodate the in-circuit probe. (Note: A separate Application Manual for the *F9450 In-Circuit Probe* will be available in July 1985).

The new Application Manual includes interface considerations, example circuit diagrams and verification, and checkout procedures. This Application Manual is free upon request from your local Tektronix sales representative. Ask for literature #61W-5773.

Bobbie Kegley MDP Product Program Manager

CUSTOMER SHARES 8560 INTEGRATION EXPERIENCE

Steiger Tractor, Inc., (Fargo, North Dakota), shares their integration experience in implementing the Motorola HDS200 into the 856X environment. The following letter is indicative of their success.

TEKTRONIX 856X/MOTOROLA HDS200

Desirable mircocomputer families for low-cost, control-oriented systems are the Motorola 6805 and 6804 families. However, currently no Tektronix 8540-based emulation support is available for these single-chip microcomputers.

We have successfully implemented at least an interim solution utilizing the Motorola HDS200 Hardware Development System, which is termed "host- and terminal-independent" by Motorola. It possesses separate host and terminal connection ports, much like a Tektronix 8540.

Our precise interface is: Tektronix 8561 - RS232 (9600 baud) - Motorola HDS200 - RS232 (9600 baud) - Tektronix 4105 terminal. After terminal baud-rate setup, the Motorola HDS200 is restarted in a local mode. The Tektronix 856X is then available for direct communications and down/up-loading by entering a Transparent mode. All RS232 connections are made with standard 856X-type (9-conductor) RS232 cables.

The only peculiarities of the interface that we must allow for are: 1) The Motorola HDS200's Host port's (ACIA) Stop-Transmission handshaking character must be changed from ASCII "ETB" (Hex 17) to ASCII "DC3" (Hex 13), utilizing the interactive HDS200 command "PF" (*P*ort *F*ormat). The importance of this is only noticed during down/up-loading operations; 2) The Host-port connector should be connected after entering the "TM" (*T*ransparent *M*ode) command at the HDS200, and disconnected before powering down the HDS200, since the HDS200 can otherwise confuse the 856X "getty" process; 3) Tektronix-loadable modules must be converted to Motorola S-records using the TNIX command "ehex" before downloading to the HDS200.

Otherwise, we utilize all TNIX resources on the 856X directly, including the Keyshell, at 9600 baud in the HDS200's Transparent mode.

The Motorola HDS200 is essentially a modular hardware development system consisting of system-executive hardware and firmware, with optional emulators, personality firmware and EPROM-version programmers for members of the Motorola 6805 and 6804 single-chip microcomputer families.

Emulation capabilities include standard start, halt, trace and breakpoints, with some chaining of breakpoints possible, although certainly not to the extent of the Tektronix 8540-resident Trigger-Trace Analyzer. Additional features encompass down/up-load (S-records), one-line assembler, disassembly and executive functions.

We have augmented this emulation with the Tektroninx 1240 Logic Analyzer, to assist in viewing and debugging I/O operations in real-time. The logic analysis is invaluable in a single-chip microcomputer development environment, where software under development is controlling serial peripherals and bitwise I/O, much different in nature than hardware-controlled bus timing typical of multi-chip microcomputer designs.

Tim Wolf Steiger Tractor, Inc.

~ ~

COMPARISONS OF TEKTRONIX C/KERNIGHAN & RITCHIE C/ANSI

The following is intented to answer your most often asked questions regarding the language our CLANDS compiler accepts. It is a comparison of Tektronix C, Kernighan & Ritchie C, and the current draft (Feb 11, 1985) of the proposed ANSI standard. This list is not complete, but shows major differences.

Question:

Does Tektronix support the new proposed ANSI C standard?

Question:

What flavor of C does Tek support?

Answer:

Tek implements the C language according to the Kernighan & Ritchie book, "The C Programming Language", with some added extensions. We include a copy of K&R with our compilers.

The Kernighan & Ritchie "standard" is the only thing close to an accepted standard at this time. The K&R book is fairly old (1978), and lacks a description of some of the newer features that C users are used to. Time has also shown some inconsistencies and omissions in the documentation.

The American National Standards Committee is charged with the task of trying to standardize the C language. They list their primary goal as trying to "codify existing practice". Several drafts of a "proposed" standard have been released for "informal" public comments. Tek can not implement the proposed ANSI standard because it is not yet in its final form. The standard is due to be released for formal public comment in March 1986. For now, "Readers are requested not to specify or claim conformance to [the ANSI proposal], which may be reproduced only for purposes relevant to the standardization process". The proposed ANSI standard is also dealing with standardizing the libraries that work with a C compiler (this would include I/O routines, math routines, signal handling, etc.). In the K&R philosophy, the libraries are not part of a compiler. No attempt has been made to describe any differences in this area.

* Label usage.
  K&R, ANSI, and TEK all agree that "the only use of a label is as a target of a goto". However, many compilers (pcc, for example) allow the user to take the address of a label.

```
main()
{
    int *lp;
lab:  lp = &lab;  /* Illegal in Tek C */
}
```

* Asm statement.
  K&R: Reserves the asm word, but doesn't describe its behavior.
  ANSI: Describes it as a common extension to the C language.
  TEK: Implements the asm command at the statement level.

```
asm ("   clr a0");      /* Illegal in tek C */
main ()
{
  asm ("   clr a0");   /* ok */
}
```

* Switch selectors.
  K&R: Switch selector must evaluate to an int.
  ANSI: Switch selector must be of integral type.
  (Types of char and int, of all sizes, signed and unsigned, and

enumerations are called integral types.)
TEK: Switch selector must evaluate to an int.

```
main ()
{
  int i;
  long l;
  unsigned int ui;
  enum processor {intel, motorola, national, zilog} enumty;

  switch (l)        /* Illegal in Tek C */
  switch (ui)       /* Illegal in Tek C */
  switch (i)        /* ok */
  switch (enumty)   /* ok */
  switch ((int)l)   /* ok */
  switch ((int)ui)  /* ok */
}
```

* Name space of identifiers.
K&R: 2 name spaces: struct/union members and tags,
other identifiers (declarations, labels).
ANSI: 4 name spaces: label names,
tags of structs/unions/enums,
members of structs/unions,
other identifiers (declarations, enum constants)
TEK: 3 name spaces: tags of structs/unions,
members of structs/unions,
other identifiers (declarations, enum constants,
labels)

```
main ()
{
  /* Three distinct identifiers for "a". */
  struct a { int a; } a;

  /* Illegal in Tek C (label a clashes with variable a). */
a: ;
}
```

* Integral widening conventions.
The coercions that happen in expressions will either preserve the unsignedness of an operand, or preserve the value of an operand.

K&R: Undefined,
ANSI: The ANSI committee is currently undecided on this issue,
TEK: The unsigned attribute will be preserved in upward coercions.

```
main ()  /* assume a target machine with 16 bit int */
{
  long result;
  long lng = 1;
  unsigned short us = 0xffff;

  /* This addition: */
  result = us + lng;                    /* result = 0x00010000 */

  /* would act like this in the "preserve unsigned attribute" case */
  result = (long)((unsigned int)us) + lng;  /* result = 0x00010000 */

  /* and like this in the "preserve value" case. */
  result = (long)((int)us) + lng;           /* result = 0x00000000 */
}
```

* Scope of externals.
An external function identifier could have either file scope (known

within the entire file), or block scope (known only within this block).

K&R: File scope.
ANSI: The ANSI committee is currently undecided on this issue.
TEK: File scope.

```
long int *l;
main ()
{
  extern long *func();
  l = func();
}

proc ()
{
  l = func();  /* Meaningful in Tek C */
}
```

* Extending the use of the unsigned type modifier to char, short, and long variables.
K&R: Defines only unsigned int;
ANSI: Allows unsigned char, short, int, long.
TEK: Allows unsigned char, short, int, long.

```
unsigned a;
unsigned int b;
unsigned char c;
unsigned short d;
unsigned long e;
```

* Structures:
K&R: Disallows structure/structure assignment,
Disallows passing structures as arguments,
Disallows functions that return structures.
ANSI: Allows structure/structure assignment.
Allows passing structures as arguments,
Allows functions that return structures.
TEK: Allows structure/structure assignment,
Allows passing structures as arguments,
Allows functions that return structures.

```
struct x { double a;
           double b; } first;

struct x second;

main ()
{
  extern struct x change_structure ();
  first = second;
  second = change_structure (first);
}
```

* The type void has been included to allow the specification that a function does not return a value (i.e. is a procedure). This allows a programmer to be more precise in specifying the correct type of a function.

K&R: Not defined.
ANSI: Allowed.
TEK: Allowed.

```
void proc (x)
  int x;
```

```
{
}

int another (x)
  int x;
{
  return x + 27;
}

main ()
{
   (void) proc (3);       /* ok */
   (void) another (3);    /* ok */
   i = another (2) + i;   /* ok */
   i = proc (2) + i;      /* Illegal */
}
```

* Simple enumerated types have been included.
  K&R: Added later (not in the K&R book),
  ANSI: Allowed.
  TEK: Allowed.

```
main ()
{
  enum language {pascal, c, fortran, forth=7, lisp};
  enum language tek_supported;

  tek_supported = pascal;
  tek_supported++;
}
```

* Identifiers may be of arbitrary length.
  K&R: only first 8 chars are significant.
  ANSI: auto and static variables significant within 31 char,
      external variables, non-static functions, global vars are
      restricted by assemblers, linkers, other language related
      utilities; required to support up to six chars.
  TEK: 14 for all identifiers.

```
main ()
{
  int a_long_identifier;
  /* next line will cause error: redeclaration of "a_long_identif" */
  int a_long_identification;
}
```

There are many features described in the ANSI PROPOSED STANDARD that will be different in Tek C and K&R C. This section will point out some of the differences, and explain why we made the choice.

* Strings are not writable in the ANSI standard.
  K&R: Allows modification of string literal.
  K&R: Disallows modification of string literal.
  TEK: Allows modification of string literal.

```
main () /* legal Tek C program, illegal in ANSI Proposed C */
{
  char *p = "hello world";
  *(p+3) = 'p';
  *(p+4) = '!';
  /*  p points at the string "help! world"  */
}
```

* Macros: concatenate strings by eliding a comment.
  K&R: says nothing.
  ANSI: defines a new operator ##.
  TEK: Allows it.

```
#define concat(a,b) a/**/b          /* Works in Tek C */
#define concat(a,b) a ## b          /* Works in Proposed ANSI C */
```

* "Function prototypes". Allows type checking of function arguments.
  K&R: not implemented.
  ANSI: defined.
  TEK: not implemented.

```
int proto1 (long, char *, short);  /* This line not legal in Tek C */
double proto2 (void);              /* This line not legal in Tek C */
unsigned int oldfunc();

main ()
{
  int abc;
  char *cp;
  short xyz;
  double dbl;

  abc = proto1 (abc, cp, xyz);  /* abc is widened to a long  */
  dbl = proto2 (abc);           /* will cause an error in the
                                   proposed ANSI standard.   */
  ui = oldfunc (1, 23, xyz);    /* no type checking          */
}
```

* The "const", "signed", and "volatile" type modifiers.
  The const type modifier can be used for ROMable data and structured constants. The signed modifier will guarantee that the compiler will view the type of the identifier as signed. (The Tek C Compiler will always treat a char declaration as signed, therefore, the signed keyword is unnecessary). The volatile modifier indicates to the compiler that this object may be modified in ways unknown to the compiler. (For example, in a memory mapped I/O system, a read or write of a location may have some special side effects.)

  K&R: not implemented.
  ANSI: defined.
  TEK: not implemented.

```
main ()
{
  const short var = 12;                  /* not legal in Tek C */
  const int pow2 [7] = {1, 2, 4, 8, 16};  /* not legal in Tek C */

  signed char sc;                        /* not legal in Tek C */
  unsigned char uc;
  char c;

  volatile char iomem;                   /* not legal in Tek C */
  int abc;

  abc = var + pow2 [3];   /* appropriate use of constants */
  var = 3;                /* invalid in proposed ANSI C */
  pow2 [4] ++;            /* invalid in proposed ANSI C */

  /* assume 8 bit char, two's complement arithmetic */
  uc = '377';
  sc = '377';
  c  = '377';
```

```
    abc = uc;     /* abc = 255 */
    abc = sc;     /* abc = -1  */
    abc = c;      /* implementation dependent (abc = 255 or -1) */

    /* If iomem wasn't declared volatile, an ANSI C compiler would be
       free to throw out the first assignment to iomem.  */
    iomem = 0;
    iomem = 1;
}
```

Brian Kennedy MDP Engineering

~ ~

PRODUCT PERFORMANCE SECTION

MDP BUG BASE

The following Product Performance Reports (PPR) have been added to or changed in our data base since the last issue of User Group News. A PPR number has been added after each PPR title for you to reference in corresponding with your local Application Engineer. We will keep you informed about the progress toward the solutions. We will also try to provide a "work-around" immediately.

Stephen Wood Software Applications Manager

~ ~

VT100 RUBOUT KEY DOESN'T WORK IN LDE (5006)

Configuration

8561 running TNIX 2.1 VT100 or compatible terminal running LDE

Description

The VT100 RUBOUT/DEL key will not erase the character above the cursor. LDECONFIG contains the code DEL (<ctl>?) for the delete function. This code is trapped by TNIX I/O and not passed to LDE for action. This problem also occurs in TNIX 2.0 and earlier.

Work Around

Reassign the keypad <ENTER> key to perform the character delete function in LDE.

LONG LINES MESS UP LDE DISPLAY (5007)

Configuration

856X running TNIX 2.1 VT100 or compatible terminal running LDE

Description

When lines in a file which contain greater than 80 characters are being edited, the screen display becomes garbled as the lines are being scrolled left and right. This problem is most noticeable if characters are added to the end of a line of text, causing it to extend past the end of the screen, and then the cursor is moved back and forth along the line. The cause of the problem is the assignment of the for VT100k, VT100, etc. This file has only become available with TNIX 2.1. Users of /etc/termcap do not experience this problem because there is no ':am:' entry for these terminals.

Work Around

Remove the ':am:' entry from /tek/lib/termcap.

INFINITE LOOP IN PASCAL 8086 (5011)

Configuration

856X running TNIX 2.1b, with 11/73 PASCAL 8086 V02.10-00

Description

Customer has a PASCAL source which when compiled with 'pas -ldv' generates 'internal error - core dumped' during Phase 2. An innocent-looking line receives the error 'CGEN: 828(c) Compiler internal error'. If the '-s' (suppress optimization) switch is used, there is no problem, and the program compiles correctly. A customer reported that on his 856X an infinite loop occurs in 'o86' (he may have an 11/23).

Work Around

Use the '-s' switch to supress optimization.

1750A REQUIRES EVEN NUMBER OF BYTES IN BLOCKS (5013)

Configuration

1750A Assembler 8560 under TNIX 2.1

Description

Assemble a 1750A source file with the following contents:

```
SECTION BUFFER          BLOCK   131          END
```

Then invoke "link -c cfile", where cfile contains:

```
-d
-r
-O objectfile
```

```
-o /tmp/garbage
-L sec=BUFFER base 7f7d
```

The linker produces the following:

link:110(E) No memory allocated to BUFFER in logical memory NONAME&

If the base of BUFFER is changed to 7f7c, the link proceeds without errors, and appears to be correct. The problem here seems to be that the BLOCK contains an odd number of bytes, and the 1750 is a word-oriented machine.

Work Around

Use BLOCKs containing an even number of bytes.

PASCAL MAY GENERATE TOO MANY BYTES (5014)

Configuration

PASCAL 8086 Compiler V02.10-00 TNIX 2.1b

Description

When you declare a variable public or absolute by an ORIGIN statement, the data section or absolute section has one more byte than it should. If you look at the pseudo-assembly language, the block statement generates the correct number of bytes. If you do a lstr on the object module, the size of the data or absolute module is one more than seems correct. We came across this problem because the linker could not link in a section due to no space. There are three examples in Marketing.

1) test86.ps - is a subrange type of address which should generate 2 bytes.
2) reg86.ps - is an integer type complied with -i option which should generate 2 bytes
3) array86.ps - is an array of integer 1..10 compiled with -i and should generate 20 bytes.

Compile any of these modules pas -ldvsia file.ps >file.pl then lstr -nosv file.po and look at D.MODE_ONE (or in case 1 and 2 A.MODE_ONE).

7811 ADDRESS DIRECTIVE PROBLEM (5016)

Configuration

7720 Assembler V01.00-16 856X running TNIX 2.1b

Description

The assembler does not generate code for the 'ADDRESS' directive. No error message is given. The following example demonstrates the problem:

```
      org   400h
jan   ascii 'JAN', 0
```

```
feb   ascii 'FEB', 0
months    address     0000, jan, feb
      end
```

Work Around

A workaround is to replace the ADDRESS directive with a series of WORD directives.

PAS8086 LARGE PACKED ARRAY PROBLEM (5017)

Configuration

PAS8086 V02.10-00 8560 running TNIX 2.1A

Description

A program containing a packed array larger than 8K is compiled. When executing, a routine 'BITLIQQ' is called upon accessing an element of the packed array. BITLIQQ does not return to the calling routine, but instead to somewhere else in memory. There is no problem with non-packed arrays. The customer is using large data, small code, and no optimization. It seems to be a problem with large code also.

Work Around

There is a workaround for packed arrays of constant length bit fields: redefine the array as a non-packed array of packed arrays which are less than 8K. This requires double indexing in the program, but it will work. There is no easy workaround for packed arrays of records.

1750 ASSEMBLER INCONSISTENCY (5018)

Configuration

856X running TNIX 2.1b

Description

There is an inconsistency in the MIL-STD ASM. According to MIL-STD 1750A, the displacement field for a base-relative instruction is a positive 8-bit value (0..255). The MIL-STD ASM erroneously interprets this field as an 8-bit two-s complement offset.

Work Around

In the MIL-STD ASM, to specify a displacement > 127, a negative value must be used. The IEEE ASM works correctly. The VMS MIL-STD ASM (V02.00-15) has the same problem.

ASC LISTING LEFT IN

Configuration

TNIX 2.1A CCC68K Option 1A ... Version 2.05-00

Description

After running the TCC compiler, a file named "" may be left in the users current working directory. The file contains the asc listing and is produced when the listing is not asked for. To remove the file, type rm '""'<cr>.

68000 EMULATOR ADDRESS STROBE PROBLEM (4061)

Configuration

8300P38/39/40 68000/08/10 Emulator V2.0

Description

The problem occurs in em 1 or 2 when the HALT line to the 68000 is brought logically true. Referring to 68000-A service manual, diagram #7, I will trace through the sequence of events. The customers UHALT is brought into U5030B, then is clocked through U4040A, U3050D, and U4040B. At this point, it is gated with AS(H), and will become A/D TRIST(L). What occurs is that in certain instructions, such as an autovectored interrupt, the UHALT signal has time to be clocked through before the end of the instruction. Between the second last memory access, which is a stack push, and the last access, which is another stack push, the 68000 returns AS(H) to the deserted state. This allows our probe to then tristate the address and data buffers on diagram #3. The 68000 then starts it's last access, which asserts AS(H), and our probe places AS(H) directly out to the probe tip. At this point, our probe still has the address line tristated until AS(H) has time to propagate through U4020B, U4020D, and diagram #3 U3110D. If the customer chooses to use the falling edge of address strobe to strobe in the address to his memory circuit, he will find our lines still tristated.

Work Around

In this case a workaround was easy, we added an addition delay in the customers circuit, holding off strobing the address for 10 nsecs.

CAN'T START KSH THRU 8540 (4036)

Configuration

410X connected to 8540 to 8560 running <= TNIX 2.1A

Description

The .profile created by keyshell has a menu section that queries the 410X series terminal to find out if it is a 4105, 4107, 4109. If an 8540 is connected between the terminal and the 8560, the escape codes sequences that are sent by the 8560 to the 410X do not pass thru the 8540. Because the information never gets back to the 8560, the menu items never get set up.

Work Around

Select option 4 ("other"), and specify 4105 at that time This problem has been fixed in TNIX 2.1b.

BREAKPOINT AFTER SC #0 FAILS (5027)

Configuration

PDBZ8K Version 1.05-00 Z8002 Emulator 8540 8562 TNIX 2.1A

Description

If the user places a hardware breakpoint after a SC #0 instruction, the emulator fails to execute the SC instruction. The customer has not tried S/W breakpoints in PDB since he has experienced many problems with them. The problem exists whether or not he uses PDB. Other information: This is an unmodified Z8002 running at 6MHz. It happens on several (4) emulators. Problem does not exist if trace is on or if no breakpoints exist around the SC #0 instruction. This is occurring in mode 0, no prototype connected. Field Service has checked at least one of the emulators exhibiting the problem, and all is well according to them.

SETKSH INFINITE LOOP (5028)

Configuration

TNIX 2.1A PLAN68K Option 1A CLAN68K Option 1A

Description

When running setksh, and specifying both the C and PASCAL 68000 compilers, debuggers, and LDEs, setksh goes into an infinite error loop when trying to determine what the default LDE will be. The problem occurs when /tek/ksh/lib/ getlde attempts to check the string entered for the default lde.

Work Around

Edit line 115 of this file to change the parameters to grep from "swi" to "shy". ALL works fine once the change is made.

PDB INCORRECTLY HANDLES CONSTANTS (5030)

Configuration

All Released PDB Versions

Description

You may not display individual elements of a constant array. Attempting to use a subscript with a constant array causes the following message to be displayed:

pdb: invalid array index

Work Around

You may display the entire array by using its name without a subscript. All future versions will allow subscripts.

80186 PCB INITIALIZATION MACRO PROBLEM (5031)

Configuration

TNIX 2.1 4.2bsd UNIX 4.1bsd UNIX 3.X VMS

Description

The ICS macros that initialize the 80186 peripheral control block (PCB) do not function properly. A test case hung when it attempted to access the PCB. The bug is located in the PCB_IO_Init$ and PCB_MEM_Init$ ICS macros. Marketing has new versions of these macros. The correction will be made to the 4.X VMS version when it is released. No mods to the current 8086 products are planned at this time.

KSH CAN'T FIND VT100K TERMCAP (5032)

Configuration

VAX/VMS ColorKey+ V02.00-01 (9/6/84)

Description

After building a custom keyshell script or using the default, selection of a VT100 terminal causes ksh to report "ksh: no termcap for vt100k" when it is run. There seems to be an entry for vt100k in [tek8540.lib]termcap.vms. The problem is that a 'define TERM vt100k' under VMS maps vt100k to VT100K (lower to upper case).

Work Around

Use 'define TERM "vt100k"' after ksh reports the error to re-run ksh. A permanent solution would be to put an uppercase VT100K in the proper place in TERMCAP.VMS.

PASCAL ICS CAN'T FIND LINKER (5033)

Configuration

8560 TNIX V2.1 & 11/73 68000 PASCAL ICS V2.03-01 Linker V2.10-00 (2.1A revision disk installed)

Description

When running ICS with the "-vk" options a failure occurs when ICS tries to invoke the linker. The error is "pas.sys: cannot execute /bin/link". Link is not in /bin, it's in /tek/bin. The PATH variable is set correctly. ICS tries to invoke the linker with an absolute pathname of /bin/link, and it has been moved to /tek/bin/link with the 2.1 upgrade.

Work Around

Do an "ln" of /tek/bin/link to /bin/link.

STRUCTA DOESN'T CREATE MS CORRECTLY (5036)

Configuration

8560 with J11 and 1Mbyte, TNIX Version 2.1A STRUCTA Version 01.04

Description

With SA Tools a file must begin with certain header information. If the file is created without this header information, it will not be fixable with 'fixsa'. This is documented in the manual.

Work Around

Either add the required header informaton before running 'fixsa' or use 'fixsa' in the first place.

ASC REQUIRES FULL PATH NAME INVOCATION (5039)

Configuration

8560 with J11 and 1 MByte Memory TNIX Version 2.1A CCC68K Version 02.05-00

Description

When invoking the modified ASC Assembler, if the complete path name '/tek/c/68000/asc . . .' is not used, the assembler returns an ASC 27 error message even though TEKPATH, LANDS, and uP are properly set. If the full path name is used, ASC operates correctly.

Work Around

Use full path name.

KSH CAN'T FIND CCC68K (5040)

Configuration

8560 with J11 and 1 MByte Memory TNIX Version 2.1A CCC68K Version 02.05-00

Description

With the scripts supplied with CCC68K, setksh cannot detect the presence of the C compiler. Correcting the /tek/ksh/lib/kshoptions file to allow the C compiler and ICS fixes the problem. Version 02.09-00 corrects this.

PAS68K PACKED STRUCTURE PROBLEM (5041)

Configuration

PAS68K V02.03-00 VMS V3.7

Description

The compiler produces incorrect code when dealing with a packed data structure comprised of two other packed structures. If the size of the first "sub-structure" is an odd number of bytes, and you refer to a byte-aligned element of 8 or 16 bits size in the second "sub-structure", the compiler produces code which only affects the 1 or 2 lower bits of the element instead of the whole element.

The compiler produces correct code if the length of the "sub-structures" are an even number of bytes, or if the "main structure" is not packed.

Apparently the routine STUL2QQ (which is called to do the store) is passed the number of bytes to move, and should be passed the number of bits.

Work Around:

Pad the "sub-structure" to an even number of bytes.

TLINK GETS

Configuration

VMS V3.7

Description

During a link the message "EOF confusion 1" occurs on the screen. The object produced seems to be OK. The problem is that the linker was built with a now obsolete version of the interface library. Rebuilding the linker with ULOSI will correct the problem. The object module is correct because the linker detects and fixes the "EOF confusion".

VMS FILE PROTECTION (5043)

Configuration

VMS 3.X

Description

Files created by many of MDP's VMS products have protection (RWD,RWD,RWD,RWD), no matter what default protection user has set. User may want created files to have this current default protection.

Work Around

Define a symbol which will set the protections correctly when invoked. Future versions - (V04) will use the default protection of user.

ASM CAN DISABLE SYSTEM ERROR MESSAGE TEXT (5044)

Configuration

ASM68K V02.03-09 VMS V3.7

Description

If an assembly is aborted with control-y, it is possible to disable the text portion of future system error messages. This is because in asm.com, "set message/notext" is used, and there is no trap for control-y. This is also the case for tpas.com, and maybe others.

Work Around

It can be easily fixed by adding "on control_y then set message/text" to the beginning of the com file.

PAS68K DISABLES VMS ERROR MSG FIELDS (5045)

Configuration

PAS68K V02.03-00 VMS V3.7

Description

The tpas.com file turns off the "FACILITY", "SEVERITY" and "IDENTIFY" fields of the system error messages, and never turns them back on.

Work Around

Define a symbol to manually re-enable the fields.

PAS68K INCORRECTLY HANDLES PACKED RECORDS (5041)

Configuration

PAS68K V02.03-00 VMS V3.7

Description

The compiler produces incorrect code when dealing with a subrange data type of three bytes in a packed record, if the subrange data type is located at an even address. The compiler generates a MOVE.L instruction, thus affecting 4 bytes. Marketing has an example.

Work Around

Pad the record with a dummy byte in order to force an odd address for the subrange data type.

SYNTAX ERROR IN ASM DEBUG SCRIPT (5062)

Configuration

VAX/VMS V3.6 ColorKey+ V2.00-01

Description

When a custom KSH script is built and ASM debug options are specified, a syntax error is put into the scripts. The error is in main.s under The two lines after the 'f1;absolute; {' line should end in a single right paren followed by a comma. The next line should not begin with a comma.

Work Around

After building main.s, edit it and make the above changes.

ASM6809 FORWARD REFERENCE PROBLEM (5063)

Configuration

ASM6809 V02.00-16 UNIX V4.2

Description

When the operand to an 'LEAS' instruction is a forward reference to an equated value, the assembler generates label phase errors for all subsequent labels. The problem seems to be that during pass 1, the assembler assumes the operand will be an address and generates the long form of the instruction. Then during pass 2 it knows that the operand is an equate and generates the short form, resulting in the phase errors.

Work Arounds

1) Move the equate so that there is no forward reference.
2) Change the "equ" to "byte", to make the operand an address.
3) Use an immediate value instead of a label.

PASCAL ORIGIN VARIABLES ARE NOT PUBLIC (5065)

Configuration

PAS8086 V02.10-00 TNIX 2.1b

Description

The PASCAL Language Reference states on page 3-15 that a variable declared with ORIGIN is automaticly given the PUBLIC attribute. This is not true. Declaring a variable both ORIGIN and PUBLIC causes an error.

Work Around

Declare the variable with ORIGIN in all modules. This will not require any additional storage.

CCC68K ICS INVOCATION ERROR (5066)

Configuration

CCC68K (VMS) V02.06-00 ColorKey+ (VMS) V02.00-01

Description

ColorKey+ expects ICS to be invoked with "@tek8540_device:[tek8540.bin]ics.com", but the C-compiler comes with an ics.exe, which must be invoked with "$tek8540_device:[tek8540.bin]ics.exe". When installng KSH and CCC68K, and pushing the "ICS" softkey, the message "can't find ..." is printed. This is not a problem with the Pascal compilers. This will be fixed in the interim Y03 releases of the VMS pascal compilers and ColorKey+. All compilers will use an ics.exe in the future.

Work Around

Modify tekset.com to define

PASCAL ORIGIN-ED POINTER PROBLEM (5071)

Configuration

PAS68K 1A V02.02-01 TNIX 2.1 on 11/73

Description

If the following PASCAL program is compiled without optimization, all is well. However, if optimization is turned on, bad code is generated.

```
program bug;
{$nilck-}
type word = -32768..32767;
var  addr1 [origin 16#F00000]: ^word;
     addr2 [origin 16#F00004]: ^word;
     i: integer;
begin
  i := addr1^ + addr2^;
end.
```

The optimizer thinks both addr1 and addr2 point to the same place.

Work Around

Use the '-s' switch to suppress optimization.

BLANK LINES CAUSE ISCP ERRORS (5072)

Configuration

ICS68K V2.05-00 ICSP68K V2.02-00 VAX/VMS 3.7

Description

When running ICSP with the blank comment lines in the .IS file, ICSP reports a file format error and tells the user to run ICS to determine what the error is. ICS reports no errors. By removing the empty comment lines between memory declarations the error is not reported. The manual does not specify that there cannot be comment lines between memory declarations.

Work Around

Avoid putting blank comment lines in your icsp source file.

LDECONFIG COMMAND ERROR (5073)

Configuration

VAX 11/750 Ultrix 1.0 (4.2BSD) Ldeconfig from PLDE

Description

Invocation of the ldeconfig command supplied on the PLDE distribution tape, results in the message "Bus error - core dumped."

Work Around

Use the ldeconfig command supplied with CLDE.

SRDY ASSERTION ERROR ON MSC BOARD (5074)

Configuration

856X MSC disk controller board

Description

The MSC board uses a 80186 and the SRDY control line to insert wait states at appropriate times. It seems that SRDY is asserted too soon which results in a short read (no wait state inserted).

Work Around

Installing a delay line to delay SRDY by 40ns corrects the problem. Some evaluation was done by manufacturing and no negative effects were encountered.

TPP/VMS INCLUDE FILES (5076)

Configuration

CCC68K V02.06-00 (VMS) VAX 11/780 VMS 3.06

Description

When compiling a file that uses "includes" of header files (file.h), TPP (preprocessor) will complain "TPP: CANNOT FIND INCLUDE FILE" if that include file is not in variable-length record VMS file format (i.e., if it is in 512 byte record format from file transfer). The error message is very misleading and can cause confusion and lost time in finding the problem.

Work Around

Make sure your include file is in variable-length format.

PRODUCT PERFORMANCE REPORT

PRODUCT NOMENCLATURE AND SERIAL NUMBER: Enter the product description and/or order name and serial number, i.e., "ACEDIT OPT. 1A, B010101"

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

IS THIS SOFTWARE COVERED BY SOFTWARE SUBSCRIPTION SERVICE?

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

REPORTED BY:

Customer Name _____ Date _____

Company Name _____ Title _____

Company Address _____

Internal Address/Dept. _____

City _____ State _____ Zip Code _____

Area Code _____ Tel. No. _____ Ext. _____

Send To:
MDP Technical Support Manager
Tektronix, Inc.
Del. Station 92-635
P.O. Box 4600
Beaverton, Oregon 97075-6400

USER GROUP LIBRARY ABSTRACTS

DASSETUP-SENDS AND RECEIVES DAS SETUP INFO

Utility — MUGL TNIX Vol VII
856X — C

Abstract

Program to send and receive DAS setup data. Uses following buffer full signals - cts on receive and xonxof on send. Link /dev/DAS to /dev/ttyx before using. DAS port must be set up as non-login port. Currently setup for 4800 baud operation. Written for TNIX version 2.0 using <tiop.h> for comm handler

Author: John Simmons — Newport News Office

GED-GRAPHIC EDITOR PROGRAM FOR 4105 TERMINAL

Graphics Editor — MUGL TNIX Vol VII
856X — C

Abstract

Syntax: ged [-f] <file> Action: Without the -f flag, the editor is invoked interactively on the named file. Editing is performed by positioning the graphic cursor and hitting an 'action' key (h for help). When invoked with the -f flag, the editor runs non-interactive, reading the commands found in <file> and reproducing the graphic output on stdout. This can be redirected to a file to give a 'cat'able version.

Author: Paul Kaye - Tek UK

H29CFG-SUPORT FILES FOR ZENITH H29 TERMINAL

Termcap Entry
856X

MUGL TNIX Vol VII
Ascii Text

Abstract

Contains support files for the Zenith H29 terminal. Included are entries for /etc/termcap and lde config. Also included is a modified 'more' command which is required to use 'more' with the H29.

Author: Stan England - Huntsville Field Office

SWITCHSYS-CHOOSE FROM TWO L.SYS FILES

System Utility
856X

MUGL TNIX Vol VII
Shell Script

Abstract

Allows superuser to have alternate L.sys files to control the times uucico can run to remote systems. Useful if user is using a dial out modem to transfer information over the UNIX network and has placed restrictions on when uucico can use the modem. (Usually to take advantage of lower phone rates.) SWITCHSYS allows you to have an alternate L.sys which can allow calls at other times or all the time for transfers that have to occur immediately.

Author: Ed Marks - Boston Field Office

---

dumb: Unknown terminal type Visual needs addressible cursor or upline capability

---

ZSCAN-DOWNLOAD TO ZILOG ZSCAN BOX

Communications Utility
856X

MUGL TNIX Vol VII
C

Abstract

This program provides download capability to a Zilog Zscan box.

INDEX OF VOLUME 3

USER GROUP NEWS

1

For further information, contact:

U.S.A., Asia, Australia, Central & South America, Japan
Tektronix, Inc.
P.O. Box 1700
Beaverton, Oregon 97075-6400
For additional literature, or the address and phone number of the Tektronix Sales office nearest you, contact:
Phone: (800) 547-1512
Oregon only: (800) 452-1877
TWX: (910) 467-8708
TLX: 151754
Cable: TEKWSGT

Europe, Africa, Middle East
Tektronix Europe B.V.
European Headquarters
Postbox 827
1180 AV Amstelveen
The Netherlands
Phone: (20) 471146
TLX: 18312 – 18328

Canada
Tektronix Canada Inc.
P.O. Box 6500
Barrie, Ontario L4M 4V3
Phone: (705) 737-2700
TLX: 06-875672
Cable: TEKANADA

Tektronix®
COMMITTED TO EXCELLENCE