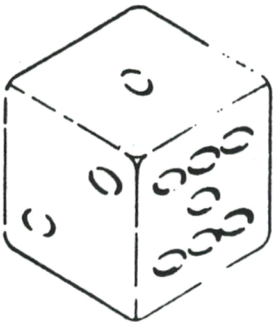

D I C E C O N T R O L P R O G R A M



dux(uk)

MICROPROCESSOR DEVELOPMENT EQUIPMENT

D u x (U K)
(0420) 63724

C O N T E N T S

	Page
1. Contents of the release diskette	1
2. Installation	3
2.1 Procedure	3
2.2 Drivers	4
2.3 Environment	5
2.4 Connection	6
2.5 Usage	7
3. Commands	9
3.1 CD	9
3.2 DIR	10
3.3 DELETE	10
3.4 TYPE	10
3.5 SLEEP	11
3.6 SUBMIT or ZB	11
3.7 RUN	12
3.8 PRINT	12
3.9 Q	13
3.10 LOGIC	13
3.11 HIST	13

4.	Utility programs	15
4.1	DUMP	15
4.2	EXE2HEX	15
4.3	PCONV68K	15
4.4	PCONVZ80	16
4.5	PCONV85	16
4.6	PCONV48	16
4.7	DSPOMF	16
4.8	OMF2HEX	17
4.9	AVZ80SYM	17
4.10	PCONV68	17
5.	Usage considerations	18
6.	Command definitions	19
7.	Known problems	22
A P P E N D I X		23
1.	AVZ80SYM	23
2.	DSPOMF	24
3.	DUMP	25
4.	EXE2HEX	26
5.	ICONV51	28
6.	ICONV63	29

7.	OMF2HEX	30
8.	PCONV09	32
9.	PCONV48	33
10.	PCONV65	34
11.	PCONV68K	35
12.	PCONV85	36
13.	PCONVZ80	38
14.	SCONV48	40
15.	STOHEX	41

NDCP RELEASE 1.1

1. Contents of the release diskette.

=====

DCP release consists of a single floppy disk in IBM XT format.
It contains the following files:

dcp.exe	The dcp program
dcp.doc	This document
auxdrv.sys	Communications port driver
auxdrv2.sys	Communications port driver for port 2
mc68000.sys	Configuration information 68000 processor
mc6809.sys	Configuration information 6809 processor
i8086.sys	Configuration information 8086 processor
i8085.sys	Configuration information i8085 processor
i8051.sys	Configuration information i8051 processor
i8048.sys	Configuration information i8048 processor
h6301.sys	Configuration information 6301 processor
z80.sys	Configuration information z80 processor
config.sys	Example DOS config.sys file
mc68000.cdf	Command definitions 68000 processor
mc6809.cdf	Command definitions 6809 processor
i8086.cdf	Command definitions 8086 processor
i8085.cdf	Command definitions i8085 processor
i8051.cdf	Command definitions i8051 processor
i8048.cdf	Command definitions i8048 processor
h6301.cdf	Command definitions 6301 processor
z80.cdf	Command definitions z80 processor
i8086.hlp	Help file for 8086 processor
i8085.hlp	Help file for 8085 processor
z80.hlp	Help file for z80 processor

autoexec.bat	Example startup job batch file
dump.exe	File dump utility
hist.exe	Histogram display utility
logic.exe	Logic display utility
pconv68k.exe	Paragon absolute format conversion utility
pconv68k.doc	Documentation for pconv68k
pconv68.exe	Paragon absolute format conversion utility
pconv68.doc	Documentation for pconv68k
pconvz80.exe	Paragon absolute format conversion utility
pconvz80.doc	Documentation for pconvz80
pconv85.exe	Paragon absolute format conversion utility
pconv85.doc	Documentation for pconv85
pconv48.exe	Paragon absolute format conversion utility
pconv48.doc	Documentation for pconv48
pconv09.exe	Paragon absolute format conversion utility
pconv09.doc	Documentation for pconv09
iconv63.exe	IAR conversion utility
iconv63.doc	Documentation for iconv63
exe2hex.exe	.exe file to hex file conversion utility
exe2hex.doc	Documentation for exe2hex
omf2hex.exe	Intel absolute object to hex-86 converter
omf2hex.doc	Documentation for omf2hex
dspomf.exe	Intel object module format display utility
dspomf.doc	Documentation for dspomf
demo68k.ice	Example DCP submit file for 68000 processor
demo86.ice	Example DCP submit file for 8086 processor
demo85.ice	Example DCP submit file for 8085 processor
demo48.ice	Example DCP submit file for 8048 processor
demoz80.ice	Example DCP submit file for z80 processor
*.fls	Various file lists
install.bat	Installation script

2. Installation.

=====

2.1 Procedure.

DCP must be installed on version 2.0 or later DOS. It will not run on any earlier release.

Insert the installation disk in the a: or b: drive of your machine, and select that as the current drive. Change directory to the root of the selected drive.

For example if you insert the disk in the a: drive, enter

```
a:
chdir 1/3
```

Run the install batch file to install DCP on the disk of your choice. Install requires 2 parameters, the dice type and disk drive name.

Dice type must be one of the following values:

mc68000	for Motorola 68000 support
mc6809	for Motorola 6809 support
i8086	for Intel 8086 support
i8085	for Intel 8085 support
i8048	for Intel 8048 support
h6301	for Hitachi 6301 processor support
z80	for Zilog z80 support

For example, to install 8085 support on your c: drive, enter

```
install 8085 c:
```


The installation copies all the files required to support your chosen DICE type. It renames the existing autoexec.bat file on the target disk to autoexec.sav, and the existing config.sys file to config.sav. If there are any entries in these files, you will need to merge the files using a text editor.

If the directory $\frac{1}{3}$ dcp already exists, a diagnostic message 'Unable to create directory' will be displayed. If either of the files $\frac{1}{3}$ config.sys or $\frac{1}{3}$ autoexec.bat exist then 'Duplicate file name or File not found' will be displayed. These diagnostics may be ignored.

DCP may be re-installed any number of times in this way (for example to add support for additional processors). The contents of the original config.sys and autoexec.bat are preserved in config.sav and autoexec.sav respectively. The contents of the config.sys and autoexec.bat files will reflect the last processor type installed.

2.2 Drivers.

In order to run, DCP must have an installed device driver to handle the communications. To tell DOS to load the device driver, there must be a file called CONFIG.SYS in the root directory of the boot disk. This file must contain the following statement, which cause DOS to load the required device drivers.

```
DEVICE=AUXDRV.SYS
```

If you ran the installation job, it copies auxdrv.sys and config.sys to the root directory of your chosen disk.

If you wish to connect the Dice to serial port two, then this line should be edited to read:

```
DEVICE=AUXDRV.SYS 2
```

If AUXDRV is not installed, DCP will exit with an error message "Can't open ICE comms".

The CONFIG.SYS file may also contain optional entries to set the maximum number of files that may be open simultaneously, and to define the number of disk buffers to be allocated. The file CONFIG.SYS on the distribution diskette is an example of the required file, and may be used or amended as required.

2.3 Environment.

DCP supports various DICE processor types, and needs to be told which is currently connected. This is done by entering the line

```
set dcp=<processor>
```

where <processor> may be one of mc68000, mc6809, z80, i8085, i8086, h6301 or i8048.

If this is not done, DCP will exit with a message 'Processor type not set'.

Two files must be read when DCP is started. These are the appropriate .SYS and .CDF files for the specific processor type. As these files may not be in the current directory when DCP is started, it needs to be told where to find them. This is done by entering the line

```
set dcppath=c:½dcp
```

where `c:\dcp` is the full directory specification required to locate the files. If `dcppath` is not set, then `dcp` will not function correctly.

The `dcppath` path is also used to search for command files (see section 6 below). If a command file cannot be found in the current directory, then the `dcppath` is searched.

If you install DCP with the supplied `install.bat` file it copies a suitable `autoexec.bat` to the root directory of your chosen disk. This file may be edited to perform any other startup functions that you require.

2.4 Connection.

DCP incorporates software protection. As part of the distribution package, you will have received a 'DATAKEY' device, which must be plugged into your computer in order for DCP to run. The DATAKEY must be plugged into the serial port that you wish to use for DICE communications (see the installation section above), and the DICE cable is plugged into the other side.

The DCP checks that the DATAKEY is present when it starts up, and if it is not found will exit with a message:

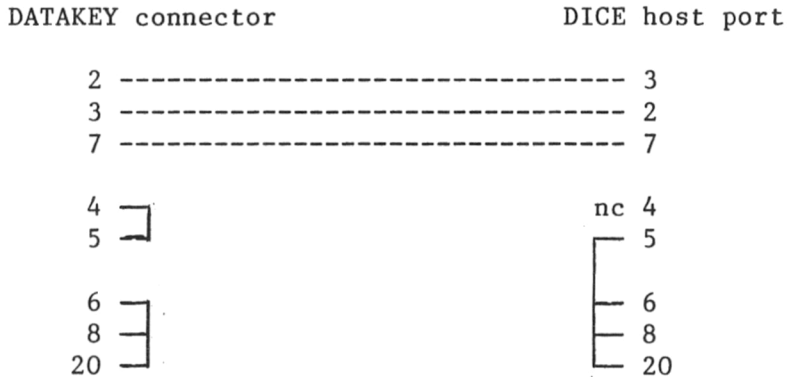
DCP: Illegal copy. Contact your distributor.

DCP does not change the comms port setting when it executes. The baud rate, number of stop bits, etc. should be set using the `MODE` command before running DCP. To set the baud rate to 9600 baud, the following command should be used:

```
MODE COM1:9600,n,8|2
```

This may be included in your `autoexec.bat` if required.

The DICE host port should be attached to the first serial port on your PC with a suitable cable. It is recommended that a cable be made up as follows:



It has been found that this cable works with most DICE/PC combinations. In case of difficulty, check with your system documentation that pin 2 is transmit data. If not, connections 2 and 3 should be reversed in the above diagram.

DICE must be configured for the master mode of operations. In some units this requires a switch setting. Check your DICE manual for details.

2.5 Usage.

In use, DCP functions almost exactly as described in the relevant DICE manual.

One exception is in stopping screen scroll. The dice manuals state that the space bar may be used to stop screen scroll. DCP uses the MSDOS convention of CONTROL-S to stop and start screen scroll. The space bar may be used on some emulators, but its behaviour is not guaranteed.

The other exception is command abortion. DCP consistently uses the ESCAPE key as the command abort key. Other command abort keys (such as carriage return) may not always perform the desired function.

3. Commands.

=====

DCP is designed to work closely with, and enhance the performance of, the DICE range of in-circuit emulators. Most of the commands are those detailed in your DICE reference manual. The following sections describe the additions and differences to the commands in that manual.

Note that in the following examples the directory delimiter character is shown as '/'. For MSDOS systems, this is true only if your config.sys file contains the entry

```
SWITCHAR=-
```

If your system does not accept / as a directory delimiter, substitute $\frac{1}{3}$ in the examples below.

3.1 CD.

To take full advantage of the DOS hierarchical file system it is strongly recommended that directories be created into which related files are grouped. DCP allows display and changing of the current directory with the CD command.

CD with no parameters displays the current directory.
CD <path> changes the directory to path. Path may be a full or relative path.

Example:

```
CD /  
CD ../mydir
```

3.2 DIR.

The DIR command displays the contents of a directory. The usual DOS wildcard characters ? and * may be used.

Example:

```
DIR /usr/src
DIR *.asm
```

3.3 DELETE.

The DEL command deletes the specified file. Wild-card deletes are not supported.

Example:

```
DELETE myfile
DELETE /source/afile.asm
```

3.4 TYPE.

The TYPE command types the contents of a file.

Example:

```
TYPE myfile.asm
```

3.5 SLEEP.

The SLEEP command pauses execution for a number of seconds. It is useful for pausing the execution of a submit file, so that the screen may be viewed.

Example:

```
SLEEP 10
```

3.6 SUBMIT or ZB.

The SUBMIT command submits a file containing a number of commands for batch processing. This is useful when a series of commands needs to be executed frequently. The format of this command is:

```
SUBMIT file p1 p2 .... pn
```

where file is the name of the file to be submitted. If no extent is given, the extent '.ice' is appended. p1 ... pn are the parameters required for the file. These parameters are substituted for strings \$1 ... \$9 in the submit file. Submit files may be nested to a maximum depth of 10. (This assumes that DOS is configured to have that many files open. See section 2.2 above).

Thus if a submit file called TEST.ICE contained the line

```
d $1,$2
```

and was invoked with the command

```
SUBMIT test 0,100
```

DICE would execute the command d 0,100

3.7 RUN.

The run command passes the line entered for processing by the DOS command processor. Thus any DOS command or batch file may be executed from within DCP. When the command finishes, control is returned to DCP.

Example:

```
RUN edlin myfile.asm
```

This causes the DOS editor edlin to be invoked to edit the file myfile.asm.

3.8 PRINT.

Print allows you to specify echo of data sent to the screen on the printer or to a file. Print with no parameters toggles the current print mode; if it was on, it is turned off and vice versa. If a file or device name is given to print, then that is opened as the print device.

Example:

```
PRINT listfile      starts listing to file LISTFILE
PRINT               closes the file and stops printer echo
```

Dice output contains the backspace character, which may not be handled correctly by some printers.

3.9 Q.

Q quits dcp and returns to DOS. Confirmation is solicited before dcp exits.

3.10 LOGIC.

LOGIC provides a logic waveform display of the external trace data bits. It reads this information from the trace buffer, and displays it on the PC screen. To use this facility, first collect the required data in the trace buffer. Then enter the command

LOGIC

There are no parameters. While DCP is reading the trace data from DICE, it displays the count of samples read. The default is to read the whole of the trace buffer. This may be terminated at any point by pressing <ESC>, to display the sampled collected so far.

To terminate the logic display and return to DCP, press <ESC>.

3.11 HIST.

HIST provides a histogramming facility, which may be useful in determining where performance bottlenecks are in a system.

HIST samples the program counter at regular intervals and, if in range of a number of 'buckets', increments the appropriate bucket counter. This may be used to build up a profile of where the machine is spending most of its time.

To invoke the histogram facility, enter the command

HIST

Histogram will then prompt for a number of values:

Number of buckets (2-32)? determines the number of bucket counters to be allocated, and the top address to be considered 'in range'. Any address that is in between the base address and base address+(number of buckets*size) is considered in range, and one of the counters is incremented. If it is out of range, none of the bucket counters is incremented. The number of buckets should be entered as a decimal number.

Bucket size (10-800H bytes)? determines the granularity of the information to be obtained. Use a large bucket size initially to determine the broad areas to be examined more closely, then reduce the size to obtain more detailed information about specific areas. The bucket size is entered as a hex number.

Base address? determines the lowest address to be considered in range, and is the address of bucket 0. The base address is entered as a hex number.

When these values have been given, DCP displays all buckets and their associated counters while it is collecting data. The screen is periodically updated to show the current counts.

Sampling will continue until the <ESC> key is pressed. DCP will then prompt whether to display the results in histogram form.

4. Utility programs.

=====

The DCP package contains several utility programs which may prove useful during development. These are not required for DCP operation, and are run from DCP as external commands (see section 6 below). Additional utility commands may be added to DCP in a similar manner.

Each utility is described separately in an accompanying ".doc" file. A brief summary is given below. (Refer to APPENDIX - page 23.)

4.1 DUMP.

Dump provides a hexadecimal dump with ascii interpretation of a file.

4.2 EXE2HEX.

The exe2hex utility converts a DOS .exe and .map file to an Intel hex file and Dice format symbol file.

4.3 PCONV68K.

Pconv68k is a utility program that takes the output of the Paragon LOD68K linker .abs module and produces a Motorola s-record .s file and Dice format symbol table.

4.4 PCONVZ80.

Pconvz80 is a utility program that takes the output of the Paragon LODz80 linker .abs module and produces an Intel hex file and Dice format symbol table.

4.5 PCONV85.

Pconv85 is a utility program that takes the output of the Paragon LOD85 linker .abs module and produces an Intel hex file and Dice format symbol table.

4.6 PCONV48.

Pconv48 is a utility program that takes the output of the Paragon ASM48 assembler .obj module and produces an Intel hex file and Dice format symbol table.

4.7 DSPOMF.

Dspomf displays Intel 8086 object module format files.

4.8 OMF2HEX.

Omf2hex processes Intel absolute format object modules to produce a Hex file and symbol file suitable for down-loading to dice-86

4.9 AVZ8OSYM.

Avz80sym processes the output list file from the Avocet z-80 assembler to produce a symbol file suitable for loading by DCP.

4.10 PCONV68.

Pconv68 is a utility program that takes the output of the Paragon LOD68 linker .abs module and produces Dice format download file and symbol table.

5. Usage considerations.

=====

Commands are entered through the keyboard in the usual way. Editing of the command may be accomplished by using the backspace character to delete the character before the cursor, or control-x to delete the entire input line.

During the course of processing by DICE, any characters received are displayed on the console. Most commands may be aborted by pressing the escape key. If for any reason DICE fails to respond, pressing the escape key still returns DCP to the command input mode, although no prompt will be displayed. Any of the locally processed commands including quit may be processed under these conditions.

If an error occurs during the processing of a submit file, the user is asked if the submit file should be closed. A single y or n key may be pressed; any other results in the bell being sounded.

The bell is also sounded if any character that is not recognised is pressed. For example if a control character is entered during command input, or if DICE is processing a command and a key which is not an abort key is pressed the bell will sound.

6. Command definitions.

=====

DCP v3.14 provides limited command configuration and addition abilities.

The files of the form <type>.CDF contain the command definitions for processor type <type>. The files are ASCII files and may be examined or changed using any source editor. Each line in the file defines a command name and the action to be taken. Each line is in three fields, separated by a space. The first field defines the command name. The second field defines the command number and the third field the parameters.

Example:

```
DUMP 10 %A %B
```

DUMP is the first field and defines the command name.
10 is the second field and defines the internal command number.
%A %B is the third field and defines the parameters. The parameters are constructed from the parameters typed into the command line. %A refers to the command name itself, %B ... %J refer to parameters 1-9.

The following is a list of internal command numbers.

FUNCTION	NUMBER
QUIT	0
SLEEP	1
DIR	2
CD	3
TYPE	4

FUNCTION	NUMBER
UPLOAD	5
DOWNLOAD	6
SYMBOL LOAD	7
PRINT ECHO	8
DEL	9
EXTERNAL COMMAND	10
HELP	11
SUBMIT COMMAND	12
SUBMIT FILE	13
INTERNAL SYMBOL LOAD	14
INTERNAL SYMBOL LIST	15
INTERNAL SYMBOL CLEAR	16
DEFINE INTERNAL SYMBOL	17
SAVE INTERNAL SYMBOLS	18
SYMBOLIC DISPLAY TOGGLE	19
Z80/8085 GO COMMAND	20
68000 TS COMMAND	21
LOGIC ANALISER	21
HISTOGRAM	22
68000 TJ, TM & T COMMAND	23

Of the most interest are the EXTERNAL COMMAND and SUBMIT COMMAND functions. With these, new commands may be introduced to provide additional functions to DCP.

RUN EXTERNAL formats a command line and passes it to the DOS command processor for execution. For example, to include the command EDLIN, add the following line to your <processor>.CDF file:

```
EDLIN 10 %A %B
```

Edlin may then be invoked by typing

```
EDLIN myfile.
```

Use of this facility assumes that you have sufficient memory to load both the requested program and DCP simultaneously.

SUBMIT FILE allows a command to automatically submit a file for processing. The file that is submitted is the command name with '.ice' suffixed. For example to define a command load, which loads a hex file and its associated symbol file, include in the CDF file the line

```
LOAD 13 %A %B
```

Create a submit file called LOAD.ICE which contains the following:

```
R $1.HEX      (or the appropriate file suffix)
R* $1.SYM     (or the appropriate symbol read command for the
              DICE type)
```

This may then be invoked by

```
LOAD file
```

to download the object and symbols.

The order in which new commands are added to the file is important. Commands are added alphabetically, with the LONGEST FIRST. This is because DCP searches forward through the commands and will stop at the FIRST MATCH.

Thus, if there were two commands S and SUBMIT, if S were placed before SUBMIT then entering the command SUBMIT would always match the S, and SUBMIT could not be reached.

7. Known problems.

=====

There is a problem with some manufacturers' implementation of ANSI.SYS. The problem occurs because the driver falsely indicates that there has been a key pressed, and DCP attempts to read it. If DCP appears to function erratically, or requires that a key be pressed before displaying the next line from DICE, you should remove the line DEVICE=ANSI.SYS from your config.sys file.

The first time SLEEP is used it does not delay for the specified time.

A P P E N D I X

=====

1. AVZ80SYM

NAME

avz80sym

SYNOPSIS

avz80sym file

DESCRIPTION

Avz80sym extracts the symbol table information from the AVOCET z80 cross-assembler listing file and writes it to a file in a form suitable to be read by DCP. The listing file may contain the assembler source; avz80sym scans for the symbol table.

file is the name of the AVOCET listing file. If no extent is given, it defaults to ".prn". The output file name is derived from the input file name, with an extent ".sym".

2. DSPOMF

NAME

dspomf

SYNOPSIS

dspomf obj-file

DESCRIPTION

DSPOMF is a utility which displays Intel object module file contents. It may be useful in debugging link or locate errors, or for finding out group and segment name information from modules for which source code is not available. Record types are displayed with their standard Intel designation, together with pertinent information in the record.

obj-file is a required parameter. It is the name of an Intel 8086 object or library file. If no extent is given, the default extent .obj is assumed. To access a file with no extension, obj-file should have a trailing period.

3. DUMP

NAME

dump

SYNOPSIS

dump file [offset]

DESCRIPTION

Dump is a utility which displays file contents in hex and ascii. It is executed with two parameters; the file name and (optionally) the offset within the file at which to start the dump. The offset is in hex. If not given, it defaults to 0.

Examples:

```
dump demo.obj
```

```
dump  $\frac{1}{3}$ usr $\frac{1}{3}$ src $\frac{1}{3}$ binfile 1000
```

DIAGNOSTCS

If the offset is too large, dump prints an error message and reports the value of end of file.

dump complains if it cannot find the file.

4. EXE2HEX.DOC

NAME

exe2hex

SYNOPSIS

```
exe2hex [-v] exe-file [sym-file [hex-file]]
```

DESCRIPTION

Exe2hex processes a .exe and .map file produced by the Microsoft link utility to produce an Intel Hex file and symbol file suitable for loading to DICE-86. Exe2hex will allow segment re-ordering and absolute placement of segment classes.

Exe2hex reads the map file to ascertain the names and addresses of all classes in the .exe file. It then requests a relocation address for each class. This should be given as a hex paragraph number.

-v is the optional verbose flag. If set, exe2hex prints details of its operation.

exe-file is a required parameter. It defines the names of the .exe and .map files to process.

sym-file is an optional parameter. It defines the name of the output symbol file. If not given, it is assumed the same as exe-file. The default extent is .sym.

hex-file is an optional parameter. It defines the name of the output hex file. It may be given only if sym-file is given. If not given, it is assumed the same as exe-file. The default extent is .h86.

Exe2hex is best suited to assembly language developments, where the number of different classes is generally low, and under programmer control.

If a class CODE is found, associated labels will be assigned to DICE-86 LABELS symbol type. DATA class is assigned to VARIABLES symbol type. Any other class is assigned to NUMBERS symbol type. DICE-86 symbolic interface is best suited to small model of computation, in which there are two active segments. LABEL symbols are defined in terms of offset from the current cs, VARIABLES in terms of the offset from ds, es and ss (which should be the same).

For a map file to be produced, the name of the map file must be specified to the link utility (the default is nul). Exe2hex requires that the map file has the same name as the .exe file, with extension .map.

For symbols to be included, it is necessary to specify the /map switch to the linker. The default is to exclude symbols from the map file. See your linker manual for more details of these options.

DIAGNOSTICS

Exe2hex prints a warning if it cannot place a relocation item in any class.

BUGS

It is sometimes not possible to resolve the relocation constant if the classes are not paragraph aligned. It is recommended that, in assembly language developments, segment align type is PARA (the default) or PAGE.

5. ICONV51

DUX symbol conversion utility =====

The ICONV51 symbol conversion utility will convert the symbolic output format file of the IAR Systems ICC8051 cross compiler linker, XLINK, into a symbol (SYM) file for downloading to the DUX DICE 8051 in circuit emulators.

The process for creating a pair (HEX and SYM) of downloadable files for the DICE is as follows:

- 1) Create source file(s) and compile and assemble with the compiler and assembler:
 ICC8051 <C_source_files>
 A8051 <Asm_source_files>
- 2) Create a link control file which uses at least the following options for output:
 SET-NEW-FORMAT INTEL-STANDARD
 DUMP <program>.HEX
 SET-NEW-FORMAT SYMBOLIC
 DUMP <program>.SYP
 And link the modules using XLINK.
- 3) Create a SYMbol file from the SYP file using the ICONV51 utility (do not include any extension for <program>, .SYP is assumed for the input file, and .SYM is used for the output file):
 ICONV51 <program>

The files <program>.HEX and <program>.SYM are now ready to download to the DICE.

6. ICONV63

DUX symbol conversion utility =====

The ICONV63 symbol conversion utility will convert the symbolic output format file of the IAR Systems ICC6301 cross compiler linker, XLINK, into a symbol (SYM) file for downloading to the DUX DICE 6301 in circuit emulators.

The process for creating a pair (HEX and SYM) of downloadable files for the DICE is as follows:

- 1) Create source file(s) and compile and assemble with the compiler and assembler:
 ICC6301 <C_source_files>
 A6801 <Asm_source_files>
- 2) Create a link control file which uses at least the following options for output:
 SET-NEW-FORMAT INTEL-STANDARD
 DUMP <program>.HEX
 SET-NEW-FORMAT SYMBOLIC
 DUMP <program>.SYP
 And link the modules using XLINK.
- 3) Create a SYMBOL file from the SYP file using the ICONV63 utility (do not include any extension for <program>, .SYP is assumed for the input file, and .SYM is used for the output file):
 ICONV63 <program>

The files <program>.HEX and <program>.SYM are now ready to download to the DICE.

7. OMF2HEX

NAME

omf2hex

SYNOPSIS

```
omf2hex obj-file [hex-file [sym-file]]
```

The square brackets denote optional parameters, and should not be entered.

DESCRIPTION

OMF2HEX is a utility which processes an Intel absolute object module, produced by LOCATE86, to produce an Intel hex file and symbol file suitable for down loading to DICE-86.

obj-file is a required parameter. It is the name of the Intel absolute object file to be processed. The default extent is .obj.

hex-file is an optional parameter. It is the name of the Intel hex file to produce. If not given, it is assumed to be the same as objfile. The default extent is .h86

sym-file is an optional parameter. It is the name of the symbol file to output. It can only be given if hex-file is specified. If not given, it is assumed to be the same as objfile. The default extent is .sym

OMF2HEX gathers symbols from LOCSYM, BLKDEF and DEBSYM records. It does not check for duplicate symbols, and if these occur they will be duplicated in the output symbol file. Symbols defined in DGROUP are assigned to the DICE VARIABLES class, those in CGROUP to LABELS.

RESTRICTIONS

OMF2HEX is designed to be compatible with DICE handling of small model of computation only. In this mode there can be at most two groups of up to 64k bytes. Conventionally these are called CGROUP for the code group, and DGROUP for the data group. All procedure definitions should be defined as near for compatibility. Any symbol is referenced by its offset from the group only. If a symbol is defined which is not in any group, a warning is printed, and the symbol is placed in the NUMBERS group.

DIAGNOSTICS

OMF2HEX will complain and exit if it finds any relocatable or fix-up records in the object.

Symbols must be defined in a group for the reasons described above. Symbols not in any group will elicit a warning.

BUGS

None known.

8. PCONV09

NAME

pconv09

SYNOPSIS

pconv09 asb-file [hex-file [sym-file]]

DESCRIPTION

pconv09 processes the output of the PARAGON lod68 loader to produce separate Intel hex and symbol files. The format of the hex file is suitable for downloading to the DICE in-circuit emulator. The symbol file may be loaded by DCP to provide for symbolic display and entry to DCP.

abs-file is a required parameter. It is the name of the PARAGON object file produced by lod68. There is a default file extent of .abs associated with this parameter.

hex-file is an optional parameter. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .hex associated with this parameter.

sym-file is an optional parameter. It can only be given if hex-file is given. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .sym associated with this parameter.

In order to produce a symbol table, the symbols must be present in abs-file. In order to achieve this, you should use the LIST B control in your source file and LIST D,S control for the link. For further information on these controls, consult your PARAGON assembler manual.

9. PCONV48

NAME

pconv48

SYNOPSIS

```
pconv48 asb-file [hex-file [sym-file]]
```

DESCRIPTION

pconv48 processes the output of the PARAGON asm48 assembler to produce separate Intel hex and symbol files. The format of the hex file is suitable for downloading to the DICE-48 in-circuit emulator. The symbol file may be loaded by DCP to provide for symbolic display and entry to DCP.

abs-file is a required parameter. It is the name of the PARAGON object file produced by asm48. There is a default file extent of .obj associated with this parameter.

hex-file is an optional parameter. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .hex associated with this parameter.

sym-file is an optional parameter. It can only be given if hex-file is given. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .sym associated with this parameter.

In order to produce a symbol table, the symbols must be present in abs-file. In order to achieve this, you should use the /debug and /symbols switches as documented in your PARAGON assembler manual.

10. PCONV65

NAME

pconv65

SYNOPSIS

pconv65 asb-file [hex-file [sym-file]]

DESCRIPTION

pconv65 processes the output of the PARAGON lod65 loader to produce separate Intel hex and symbol files. The format of the hex file is suitable for downloading to the DICE-6502 in-circuit emulator. The symbol file may be loaded by DCP to provide for symbolic display and entry to DCP.

abs-file is a required parameter. It is the name of the PARAGON object file produced by lod65. There is a default file extent of .abs associated with this parameter.

hex-file is an optional parameter. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .hex associated with this parameter.

sym-file is an optional parameter. It can only be given if hex-file is given. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .sym associated with this parameter.

In order to produce a symbol table, the symbols must be present in abs-file. In order to achieve this, you should use the LIST B control in your source file and LIST D,S control for the link. For further information on these controls, consult your PARAGON assembler manual.

11. PCONV68K

NAME

pconv68k

SYNOPSIS

pconv68k asb-file [srec-file [sym-file]]

DESCRIPTION

pconv68k processes the output of the PARAGON lod68k loader to produce separate S-record and symbol files. The format of these files is suitable for downloading to the DICE-68k in-circuit emulator.

abs-file is a required parameter. It is the name of the PARAGON absolute file produced by lod68k. There is a default file extent of .abs associated with this parameter.

srec-file is an optional parameter. If given, it specifies the name of the s-record file to be produced. If not given, it is the same as abs-file. There is a default extent of .s associated with this parameter.

sym-file is an optional parameter. It can only be given if srec-file is given. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .sym associated with this parameter.

In order to produce a symbol table, the symbols must be present in abs-file. Consult your PARAGON compiler, assembler and loader manuals for details of how to do this.

12. PCONV85

NAME

pconv85

SYNOPSIS

pconv85 asb-file [hex-file [sym-file]]

DESCRIPTION

pconv85 processes the output of the PARAGON lod85 loader to produce separate Intel hex and symbol files. The format of the hex file is suitable for downloading to the DICE-85 in-circuit emulator. The symbol file may be loaded by DCP to provide for symbolic display and entry to DCP.

abs-file is a required parameter. It is the name of the PARAGON absolute file produced by lod85. There is a default file extent of .abs associated with this parameter.

hex-file is an optional parameter. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .hex associated with this parameter.

sym-file is an optional parameter. It can only be given if hex-file is given. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .sym associated with this parameter.

In order to produce a symbol table, the symbols must be present in abs-file. This may be achieved by observing the following:

1. Include the source line statement

```
$DEBUG
```

in your assembly source program. This causes the local symbol table to be included in the assembler output object module.

2. Assemble your source to produce a .obj file.

3. Use the command file method of specifying loader directives. Include the following line in the loader command file

```
list s
```

This causes the loader to write local symbols to the .abs file.

13. PCONVZ80

NAME

pconvz80

SYNOPSIS

pconvz80 asb-file [hex-file [sym-file]]

DESCRIPTION

pconvz80 processes the output of the PARAGON lodz80 loader to produce separate Intel hex and symbol files. The format of the hex file is suitable for downloading to the DICE-z80 in-circuit emulator. The symbol file may be loaded by DCP to provide for symbolic display and entry to DCP.

abs-file is a required parameter. It is the name of the PARAGON absolute file produced by lodz80. There is a default file extent of .abs associated with this parameter.

hex-file is an optional parameter. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .hex associated with this parameter.

sym-file is an optional parameter. It can only be given if hex-file is given. If given, it specifies the name of the hex file to be produced. If not given, it is the same as abs-file. There is a default extent of .sym associated with this parameter.

In order to produce a symbol table, the symbols must be present in abs-file. This may be achieved by observing the following:

1. Include the source line statement

```
list    B
```

in your assembly source program. This causes the local symbol table to be included in the assembler output object module.

2. Assemble your source to produce a .obj file.

3. Use the command file method of specifying loader directives. Include the following line in the loader command file

```
list    S
```

This causes the loader to write local symbols to the .abs file.

14. SCONV48

DUX symbol conversion utility =====

The SCONV48 symbol conversion utility will convert the MAP output file of the 2500AD Software 8748 cross assembler linker, version 4.00 and later, into a symbol (SYM) file for downloading to the DUX DICE 8048 in circuit emulators.

The process for creating a pair (HEX and SYM) of downloadable files for the DICE is as follows:

- 1) Create source file(s) and assemble with the X8748 assembler:
X8748 <source_files>
- 2) Create a link control file which uses at least the H and D output options - if you want local symbols use the M option as well:
LINK <link_control_file>
- 3) Create a SYMbol file from the MAP file using the SCONV48 utility (do not include any extension for <program>, .MAP is assumed on input and a .SYM file is output):
SCONV48 <program>

The files <program>.HEX and <program>.SYM are now ready to download to the DICE.

15. STOHEX

S record to HEX conversion utility

=====

This utility will convert a file of Motorola S records into Intel standard HEX format.

The utility expects a parameter which is the name of the source file minus its extension (an extension of .LD is assumed), and produces a HEX file of the same name with the extension .HEX:

```
stohex <filename>
```

will take the S record file <filename>.LD and produce the HEX file <filename>.HEX.