inmos

IMS D7214 IBM/NEC PC ANSI C Toolset delivery manual

INMOS Limited

72 TDS 227 00

August 1990

INMOS is a member of the SGS-THOMSON Microelectronics Group.

The C compiler implementation was developed from the Perihelion Software "C" Compiler and the Codemist Norcroft "C" Compiler.

INMOS document number: 72 TDS 227 00

Contents

1 Introduction

This manual provides installation instructions for the IMS D7214 ANSI C toolset for the IBM PC (and compatibles) and the NEC PC. This delivery manual deals with PC specific parts of the toolset. In addition instructions for testing the release are given.

1.1 Layout of this manual

Chapter 1 Introduction: (this chapter) summarises the contents of the release and describes its prerequisites.

Chapter 2 Installing the release: provides installation instructions for this release.

Chapter 3 Confidence testing: contains a simple procedure to check that the installation has been done correctly.

Appendix A Distribution kit: contains a list of the components of the release.

Appendix B Debugger function keys: shows the positions of the debugger and simulator function keys on the IBM and NEC PC keyboard.

1.2 Prerequisites for running the toolset (IBM PC)

In order to use the ANSI C toolset you will require:

- An IBM PC, PC/XT or PC/AT (or compatible).
- DOS version 3.0 or later.
- About 7 Mbytes of free disk space (although you need not install the entire release — see section 2.1).
- An IMS B004, B008 (or similar) transputer board with an IMS T800 or T414 (Rev B or later) transputer.

Note : For interactive debugging an additional 32 bit transputer is required.

1.3 Prerequisites for running the toolset (NEC PC)

In order to use the ANSI C toolset you will require:

- An NEC PC.
- DOS version 3.0 or later.
- About 7 Mbytes of free disk space (although you need not install the entire release — see section 2.1).
- An IMS B010 (or similar) transputer board with an IMS T800 or T414 (Rev B or later) transputer.

Note : For interactive debugging an additional 32 bit transputer is required.

1.4 Contents of this release

The D7214 ANSI C toolset release consists of:

- A set of eleven 360 Kbyte 5.25 inch floppy disks and five 720K 3.5 inch floppy disks.
- The '*ANSI C toolset delivery manual*' (this document).
- The '*ANSI C toolset user manual*'.
- The '*ANSI C toolset reference manual*'.
- The '*ANSI C toolset handbook*'.

1.5 Compatibility with previous releases

This release is not object compatible with previous INMOS products, including all previous alpha releases of the TCOFF toolset and the 3L compiler. All modules will require recompilation.

1.6 PC Hosted Tools

Two versions of some of the tools are supplied; transputer bootable and PC hosted executable. It is up to the user to decide which versions to use. Details of how to set up the system to use either the transputer or PC versions of the tools are given in section 2.2.2.

2 Installing the release

This release of the IMS D7214 ANSI C toolset comes on eleven 360K 5.25 inch floppy disks and 5 720K 3.5 inch floppy disks. The installation procedure is the same whether you install from the 360K disks or the 720K disks. You will require about 7 Mbytes of free space to install the entire release. Depending on the use you make of the toolset it may not be necessary to install the entire release.

2.1 Installation

To install the release first insert Disk 1 in your floppy disk drive. Next run the batch file, **install.bat**, on Disk 1, giving as parameters the drive letter of the floppy disk drive and the drive on which the toolset will be installed.

For example, if your floppy disk drive is A, and the drive on which you want the toolset installed is C, type:

```
    a:install a c
```

You will then be given information on how to proceed with the installation. You will be asked a number of questions to which you can answer 'yes' by typing 'Y' (or 'y'), or answer 'no' by typing 'N' (or 'n').

The installation procedure will ask whether you are using a NEC PC machine. If you answer 'Y' to this question then the correct server for the NEC will be installed together with some support files.

Answering 'N' to this question will result in the installation procedure asking whether you are using a B004 or B008. Answering appropriately causes the correct server to be installed.

The installation procedure also includes a feature to allow the C runtime libraries to be built at installation time. This allows the smallest possible library to be installed. The questions asked take the form:

```
    Do you want T225 support ?
```

Answering 'Y' to this particular question causes modules compiled specifically for the T225 to be included in the library.

A description of the different processor types and classes can be found in chapter 5 of the *'ANSI C toolset user manual'*.

During the installation new directories are created. If a directory with the same name already exists then an error will be reported. Do not worry if this happens

— the installation will not be affected, but will use the directory of that name anyway.

If the installation has been successfull the following messages will be displayed at the end of installation:

```
INSTALLATION COMPLETE

YOU MAY DELETE INSTALL2.BAT
```

`INSTALL2.BAT` is the batch file which installed the release. It is no longer required and so may be deleted.

The installation procedure creates a directory called `\ICTOOLS`. All the programs necessary to install the toolset are copied to this directory. All the components of the toolset itself are copied into sub-directories of `\ICTOOLS`, as shown in the following table:

Directory	**Contents**
`\ictools\itools`	The transputer bootable tools.
`\ictools\tools`	The PC hosted tools.
`\ictools\libs`	The toolset libraries and include files.
`\ictools\examples`	Examples directory.
`\ictools\examples\simple`	Simple example sources.
`\ictools\examples\debugger`	Debugger example sources.
`\ictools\examples\imakef`	Imakef example sources.
`\ictools\examples\config`	Configurer example sources.
`\ictools\examples\config\b008`	Configurer example B008 configuration files.
`\ictools\iserver`	The `iserver` executables.
`\ictools\source`	Source code.
`\ictools\source\iserver`	Server sources (see '*ANSI C toolset user manual*'.)
`\ictools\source\imakef`	Imakef sources.
`\ictools\iterms`	Example iterm files and driver program.
`\ictools\nec`	NEC PC support files (if installed).

The release installation procedure installs everything onto the hard disk. Certain parts of the toolset release may be removed from the hard disk if disk space is a problem. The following table indicates which parts of the release are essential for its correct operation.

Component	Necessary
ITOOLS	yes
TOOLS	yes †
LIBS	yes
ITERMS	yes
ISERVER	yes
SOURCE	no
EXAMPLES	no
NEC (if installed)	yes

† If you are happy to use just the transputer based tools then the TOOLS directory may be deleted. Conversely if you prefer to use the hosted tools then the following files may be deleted from the ITOOLS directory.

```
icconf.exe
icconf.btl
icollect.exe
icollect.btl
icvemit.exe
icvemit.btl
icvlink.exe
icvlink.btl
iemit.exe
iemit.btl
ieprom.exe
ieprom.btl
ilibr.exe
ilibr.btl
ilink.exe
ilink.btl
ilist.exe
ilist.btl
imakef.exe
imakef.btl
```

Do NOT delete the ITOOLS directory as it contains some tools for which hosted equivalents do not exist.

If you have enough disk space then it may be advantageous to leave the contents of the ITOOLS directory unchanged as it is possible that the hosted versions of the tools may run out of memory on very large jobs. In these circumstances the transputer versions of the tools can be used as they generally have more memory available.

Having installed the toolset there are a number of environment variables to be set up before you can use any of the tools.

2.2 Setting up the toolset for use

This section explains how to set up the environment necessary to use the toolset. It describes the basic changes to the system configuration file CONFIG.SYS which you should make before you attempt to use the toolset and shows how to set up the necessary environment variables.

2.2.1 Setting the FILES variable

The FILES command in your system configuration file CONFIG.SYS should be changed to specify 20 simultaneously open files. For example:

```
        FILES=20
```

Note: Any other file handling software used on the system (such as PC-NFS) should also be reset to accept 20 simultaneously open files.

2.2.2 Setting the correct PATH

You should add the DOS commands given below to your autoexec.bat file so that they will be set up whenever you switch on your PC.

If you wish to use the transputer based tools then only add the directories \ICTOOLS\ISERVER and \ICTOOLS\ITOOLS to your DOS path. To do this use the DOS PATH command.

If hosted tools are required add ICTOOLS\TOOLS before ICTOOLS\ITOOLS.

For example to set your path to your system commands and then the toolset (on drive C), type:

```
        PATH=C:\DOS;C:\ICTOOLS\ISERVER;C:\ICTOOLS\ITOOLS
```

The above command sets up the path to find the transputer based tools only.

2.2.3 Setting an alternative ISERVER

Alternative iservers can be used by defining the iserver to be used in the ISERVER environment variable. If the ISERVER environment variable is defined on the system then the iserver is referenced by the environment variable rather than from the ICTOOLS\ISERVER directory.

e.g.

To use a server called MYSERVER.EXE from your \BIN directory on drive C, use the following definition:

```
set ISERVER=C:\BIN\MYSERVER.EXE
```

If `ISERVER` is not defined on the system then the DOS path, set up using the `PATH` command, is used to find the server.

2.2.4 Setting the board memory size

Before you can use any tool which runs on your transputer evaluation board you must set up an environment variable, `IBOARDSIZE`, giving the size of the memory on the board (in bytes). To do this use the DOS `set` command. For example, to set a board size to 2 Mbytes type:

```
set IBOARDSIZE=#200000
```

You may give either a decimal or hexadecimal (preceded by '#') number. On keyboards without '#', the '$' character can be used instead. Leading and trailing spaces are prohibited.

If `IBOARDSIZE` is specified incorrectly, for example as a character, string or with leading or trailing spaces, the system defaults to a board size of 0 (zero) and the program cannot be run. If `IBOARDSIZE` is explicitly set to a very small value a similar error may occur.

Note: that setting very small board sizes may cause some tools to hang. This is an important point to remember when developing software for the T2. Remember to reset the boardsize after testing the software as some of the tools will hang if run with the small value of `IBOARDSIZE` required for the T2.

2.2.5 Setting a file system search path

To enable the tools to find libraries and include files you must set up an environment variable called `ISEARCH`. This environment variable normally will give the standard library and include file directory (`\ICTOOLS\LIBS\`) and any user directories as required.

Note: that unlike the DOS path you must add the closing backslash, '\', to a directory name.

Directories may be separated by a space or a semi-colon. For example to set up `ISEARCH` to point to the standard include files and libraries and to a user directory called \MYDIR type the following DOS command:

```
        set ISEARCH=C:\ICTOOLS\LIBS\;C:\MYDIR\
```

2.2.6 Setting root memory size for idebug

The amount of memory on the root transputer must be defined for idebug, using the environment variable IDEBUGSIZE. This variable is set up in the same way as IBOARDSIZE (see section 2.2.4) and should be set to the available memory. Leading and trailing spaces are prohibited.

The debugger requires at least 400K of memory on the root transputer to operate correctly although 1 Mbyte or more is recommended.

2.2.7 Setting an alternative board address

The default PC bus address used by the iserver for locating the transputer board is 150_{16} for the IBM PC and $D0_{16}$ for the NEC PC. If your transputer board resides at a different address in the PC bus you should set up the environment variable TRANSPUTER which gives the address of the board. The address must be given in hexadecimal.

For example, if your transputer board is at address 200_{16}, use the following command:

```
        set TRANSPUTER=200
```

You can also use the iserver 'SL' option to override the address specified by TRANSPUTER.

2.2.8 Special ITERM support for the debugger and simulator

If you are using a NEC PC 9800 series computer then you must set your ITERM environment variable to use necpc.itm and you should ignore the rest of this section.

In order to use the debugger or simulator you will need to use a screen device driver that can recognise ANSI *escape sequences.* You should use either ANSI.SYS which is supplied with DOS, or BANSI.SYS which is supplied with this toolset release.

The normal ANSI.SYS supplied with a PC does not support features such as *insert line* and *delete line.* Without such features, the debugger has to redraw the whole screen whenever it scrolls. In order to overcome such limitations this release includes a replacement for **ANSI.SYS** which is known as **BANSI.SYS**. This is compatible with **ANSI.SYS**; it merely provides extra functionality.

Once installed it may be used with tools other than the debugger or simulator.

In order to install **BANSI.SYS** you will need to place the following in your **CONFIG.SYS** file (if you prefer to install **ANSI.SYS** you must add the appropriate DEVICE line for it) :-

```
DEVICE=C:\ICTOOLS\ITERMS\BANSI.SYS
```

This line should replace a similar line that references **ANSI.SYS** (if it was present).

You will need to re-boot the PC in order for the **BANSI.SYS** device driver to be installed.

You should then set your **ITERM** to use **pcbansi.itm** if you have installed **BANSI.SYS** and **pcansi.itm** if you have installed **ANSI.SYS**.

```
set ITERM=C:\ICTOOLS\ITERMS\PCBANSI.ITM
```

2.3 Driver program errors

The transputer based tools are executed through a driver program which itself generates error messages. For example:

```
Fatal-driver- unable to execute 'icc', Arg list too long
```

In this example the messages indicates that the DOS limit on the length of the command line has been exceeded.

Driver errors are generated for limitations or errors such as a command line too long, denial of read/write access to a file, and file or directory not found.

2.4 Transputer error flag

The driver programs for the transputer hosted tools (except idebug and idump) monitor the error flag as the tool executes in order to catch any internal errors of the tool should they occur. If your hardware is configured as a down system (wired *down*, see the '*ANSI C toolset user manual*') and consists of more than one transputer, the driver programs may be fooled into thinking the tool has set the error flag if the error flag on one of the extra processors is already set when the tool is executed. In order to overcome this problem, you should run a network check program, such as **ispy**, or boot a (dummy) program that uses all of the processors in the network.

The **ispy** program is provided as part of the board support software for INMOS *i*q systems products. These products are available separately through your local INMOS distributor.

Note: that once cleared, an error flag on a transputer will only become set again if you execute an erroneous program on the transputer or you power on the transputer again.

2.5 Running idebug and isim on NEC machines

On the NEC PC the commands that invoke the debugger and the simulator must be prefixed with the letter 'n'. For example:

```
nisim

nidebug
```

These commands ensure that the NEC keyboard and screen are initialised for use with the **NECPC.ITM** ITERM file. When the tool finishes the keyboard is restored to normal DOS mappings.

Note: that when performing a shell escape from the **iserver** the keyboard will not be remapped for DOS (i.e. it will still be mapped for the tool).

2.6 Environment space

The PC may not have enough environment space by default. This may need to be increased in order to run the toolset.

All versions of DOS allow the environment space to be increased to a maximum of 32 Kbytes, with varying degrees of difficulty. For the commands or procedures to use on your system consult the user documentation for the specific version of DOS you are using.

For DOS versions 3.2 and later the **SHELL** command in the **config.sys** file can be used to set up an environment size when the PC is booted. For example:

```
SHELL=command.com /e:1024 /p
```

This example gives the name of the DOS command processor, sets the environment space to 1024 bytes and makes this version of the command processor permanently resident.

In DOS version 3.3 and later the command called COMMAND can be used to increase the environment space. For example:

```
COMMAND /e:1024 /p
```

This has a similar effect to the **SHELL** command example but is invoked from DOS.

Earlier versions of DOS require the command processor (**command.com**) to be patched. Microsoft provide a utility **SETENV** that will do this automatically.

2.7 Server interrupts

It is possible to interrupt the server, go to DOS to issue DOS commands, and subsequently return to the server. This has the effect of temporarily halting the server. The program continues to run until access to the server is required.

To interrupt the server, use the following procedure. Remember to enable the BREAK key first.

Use CTRL-BREAK (in preference to CTRL-C) in order to interrupt the program. Type ‘S’ at the prompt, which enters a new DOS command processor. DOS commands can now be executed as necessary.

In order to return to the server type ‘exit’. This quits the DOS command processor and restarts the **iserver**.

When in DOS do not invoke any tool or program that runs on the transputer board, or the program running in the background will be corrupted.

The ability to interrupt the server relies on the existence of either a DOS environment variable COMSPEC or a DOS command file COMMAND.COM in order to recall DOS.

3 Confidence testing

This chapter describes a short procedure which may be followed to check that installation has been done correctly.

1 Set the current disk to the same disk as the compiler has been installed on. For example, if the compiler has been installed in directory `C:\ICTOOLS`, do this:

```
D>c:

C>
```

2 Set the current directory to a convenient directory for doing this test. For example:

```
C>cd \mine

C>
```

3 Copy the example `hello.c` file to the current directory:

```
C>copy \ictools\examples\simple\hello.c
        1 File(s) copied

C>
```

4 Compile the example for the TA processor class (this will work on all 32 bit processors, alternatively, replace the `/ta` with the relevant option for your particular processor type):

```
C>icc hello /ta

C>
```

If, instead of the `C>` prompt, the computer outputs the following, or something similar —

```
Error - iserver - protocol error ...
```

— it is likely that there has been some error in setting up the transputer board. In particular, please check that the wire links, accessible from the back of the PC, have been correctly installed. The transputer board's documentation should help with this.

It may also be necessary to change the default address at which the transputer is assumed to be in the PC's bus as the compiler plus the other tools are loaded at the address 150_{16}. For example, if your transputer board is at address 200_{16} then it will be necessary to add the option '`/sl 200`' to the command line when running the tools (or set the environment variable `TRANSPUTER` to 200).

5 Link the resulting binary file with the necessary parts of the run-time library (note that if a different option to `/ta` was used on the compiler command line then the same option should replace `/ta` on the linker command line) :

```
C>ilink hello.tco /f startup.lnk /ta

C>icollect hello.lku /t
```

6 Finally, the program can be run:

```
C>iserver /sb hello.btl

Hello World

C>
```

The output '`Hello World`' comes from the `hello.c` example program. If it does not appear, we recommend that the installation procedure should be carefully repeated, and the confidence test procedure followed again.

Appendices

A Distribution kit

This appendix lists the files which make up the distribution kit for this version of the ANSI C Toolset. Each filename is accompanied by a short description of the file's function.

A.1 Directory `\ictools\itools`

`icc.btl`	C compiler code
`icc.exe`	C compiler driver program
`ilink.btl`	linker code
`ilink.exe`	linker driver program
`ilibr.btl`	librarian code
`ilibr.exe`	librarian driver program
`icollect.btl`	collector code
`icollect.exe`	collector driver program
`ilist.btl`	lister code
`ilist.exe`	lister driver program
`icconf.btl`	configurer code
`icconf.exe`	configurer driver program
`idebug.btl`	debugger code
`idebug.exe`	debugger driver program
`ieprom.btl`	eprom file constructor code
`ieprom.exe`	eprom file constructor driver program
`isim.btl`	simulator code
`isim.exe`	simulator driver program
`icvemit.btl`	memory description format converter code
`icvemit.exe`	memory description format converter driver program
`iemit.btl`	external memory interface program code
`iemit.exe`	external memory interface program driver program
`imakef.btl`	makefile generator code
`imakef.exe`	makefile generator driver program
`idump.btl`	core dumper code
`idump.exe`	core dumper driver program
`iskip.btl`	skip loader code
`iskip.exe`	skip loader driver program
`icvlink.btl`	format converter code
`icvlink.exe`	format converter driver program

The following will also be present if the installation took place for the NEC.

`nidebug.bat`	debugger batch file for NEC
`nisim.bat`	simulator batch file for NEC

A.2 Directory `\ictools\tools`

`icconf.exe`	configurer PC executable
`icollect.exe`	collector PC executable
`icvemit.exe`	memory description format converter PC executable
`icvlink.exe`	format converter PC executable
`iemit.exe`	external memory interface program PC executable
`ieprom.exe`	eprom file constructor PC executable
`ilibr.exe`	librarian PC executable
`ilink.exe`	linker PC executable
`ilist.exe`	lister PC executable
`imakef.exe`	makefile generator PC executable

A.3 Directory `\ictools\libs`

`libc.lib`	C run-time library
`libcred.lib`	reduced C run-time library
`centry.lib`	C entry point library

`linkboot.lib`	boot-from-link bootstrap library
`romboot.lib`	boot-from-ROM bootstrap library
`sysproc.lib`	system process library

`startup.lnk`	C startup linker indirect file
`startrd.lnk`	C startup linker indirect file for reduced library

`setconf.inc`	default configuration startup file
`boards.inc`	INMOS evaluation board definitions
`trams.inc`	INMOS transputer module definitions

`ctype.h`	run-time library header files
`dos.h`	
`string.h`	
`stddef.h`	
`iocntrl.h`	
`errno.h`	
`time.h`	
`semaphor.h`	
`channel.h`	
`assert.h`	
`float.h`	
`limits.h`	
`locale.h`	
`setjmp.h`	
`signal.h`	
`stdarg.h`	
`stdlib.h`	
`process.h`	
`stdio.h`	
`math.h`	
`mathf.h`	
`host.h`	
`stdiored.h`	
`conndx11.h`	
`subsem.h`	
`subchan.h`	
`misc.h`	

A.4 Directory `\ictools\iterms`

`bansi.sys`	screen device driver
`pcansi.itm`	iterm file for use with ANSI.SYS
`pcbansi.itm`	iterm file for use with BANSI.SYS
`necpc.itm`	iterm file for use with an NEC PC
`readme.txt`	information file

A.5 Directory `\ictools\examples\simple`

`hello.c`	'Hello world' program
`parhello.c`	parallel 'Hello world' program

`main.c`	3 modules to demonstrate separate compilation
`hellof.c`	
`worldf.c`	

A.6 Directory `\ictools\examples\debugger`

`abort.c`	forces a C program to halt for postmortem debugging
`abort.mkf`	makefile to build the abort example
`abort.bat`	batch file to build the abort example
`debug.c`	example of use of the debug support functions
`debug.mkf`	makefile to build the debug example
`debug.bat`	batch file to build the debug example
`display.c`	inspect variables example
`display.mkf`	makefile to build the display example
`display.bat`	batch file to build the display example
`facs.c`	parallel example using channels
`facs.mkf`	makefile to build the facs example
`facs.bat`	batch file to build the facs example
`free.c`	cause a runtime error by freeing an invalid pointer
`free.mkf`	makefile to build the free example
`free.bat`	batch file to build the free example
`stack.c`	cause a runtime error by overflowing the stack
`stack.mkf`	makefile to build the stack example
`stack.bat`	batch file to build the stack example
`master.c`	master process - part of cpair example
`mult.c`	multiplier process - part of cpair example
`cpair.cfs`	configuration source file
`cpair.mkf`	makefile to build the cpair example
`cpair.bat`	batch file to build the cpair example
`makefile`	makefile to build all the examples

`readme.txt`	information file

A.7 Directory \ictools\examples\imakef

main.c	3 modules which make up a hello world program
hellof.c	
worldf.c	
hello.lnk	linker command file for the hello world program
master.c	master process - part of multi example
mult.c	multiplier process - part of multi example
master.lnk	linker command file for the master linked unit
mult.lnk	linker command file for the mult linked unit
multi.cfs	configuration source file

A.8 Directory \ictools\examples\config

hello.bat	batch file to build hello example
hello.cfs	configuration file for hello example
hello.mkf	makefile for hello example
hello2.bat	batch file to build two processor hello example
hello2.cfs	configuration file for two processor hello example
hello2.mkf	makefile for two processor hello example
hellop.c	hello world process code
hostmult.c	iserver multiplexor process code
hostpass.c	example host node process code
multpass.c	example remote node process code
pipe.bat	batch file to build pipe line example
pipe.cfs	configuration file for pipe line example
pipe.mkf	makefile for pipe line example
ring.bat	batch file to build ring example
ring.cfs	configuration file for ring example
ring.mkf	makefile for ring example
setup.inc	default configuration include file
square.bat	batch file to build square array example
square.cfs	configuration file for square array example
square.mkf	makefile for square array example
square1.bat	batch file to build one processor square array example
square1.cfs	configuration file for one processor square array example
square1.mkf	makefile for one processor square array example
tree.bat	batch file to build tree example

tree.cfs	configuration file for tree example
tree.mkf	makefile for tree example
types.bat	batch file to build configuration types example
types.c	configuration types process code
types.cfs	configuration file for configuration types example
types.mkf	makefile for configuration types example
worldp.c	world hello process code
zigzag.bat	batch file to build pipeline on a square array example
zigzag.cfs	configuration file for pipeline on a square array example
zigzag.mkf	makefile for pipeline on a square array example
readme.txt	information file

A.9 Directory \ictools\examples\config\b008

b008	B008 hardwire definition for MMS
ring5	ring softwire definition for MMS
square5	square softwire definition for MMS
tree5	tree softwire definition for MMS

A.10 Directory \ictools\iserver

isvrb04.exe	host file server and loader program (IBM PC version for B004)
isvrb08.exe	host file server and loader program (IBM PC version for B008)
isernec.exe	host file server and loader program (NEC PC version)
iserver.exe	the server which was selected at installation time

A.11 Directory \ictools\source\iserver

Contains the sources for the host file server. The sources exist for building the file server to communicate with the following transputer board products:

IMS B004	(INMOS Ltd)
IMS B008	(INMOS Ltd)
IMS B010	(INMOS Ltd)
IMS B011	(INMOS Ltd)
IMS B014	(INMOS Ltd)
IMS B015	(INMOS Ltd)
IMS B016	(INMOS Ltd)
C3 QT0	(Caplin Cybernetics Corp.)

Makefiles are supplied for building the **iserver** for all the different board types supported. Source for using the file server under the Helios operating system (Perihelion Software Ltd) is also included.

A.12 Directory \ictools\source\imakef

Contains the sources for the makefile generator. Makefiles exist for building the makefile generator for the PC and transputer.

A.13 Directory \ictools\nec

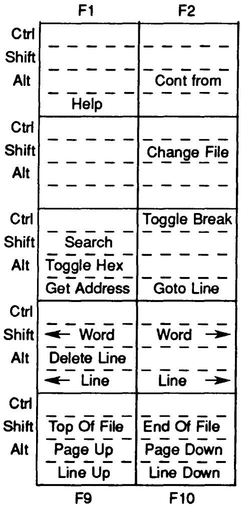
This directory is only present when the toolset is installed for the NEC machine.

doskeys.ld	remap NEC keyboard for DOS
doskeys.tbl	remap NEC keyboard for DOS
necini25.lis	disable status line on NEC screen
tds3keys.ld	map NEC keyboard for nidebug/nisim
tds3keys.tbl	map NEC keyboard for nidebug/nisim
nidebug.bat	debugger batch file for NEC - also copied to itools
nisim.bat	simulator batch file for NEC - also copied to itools

B Debugger function keys

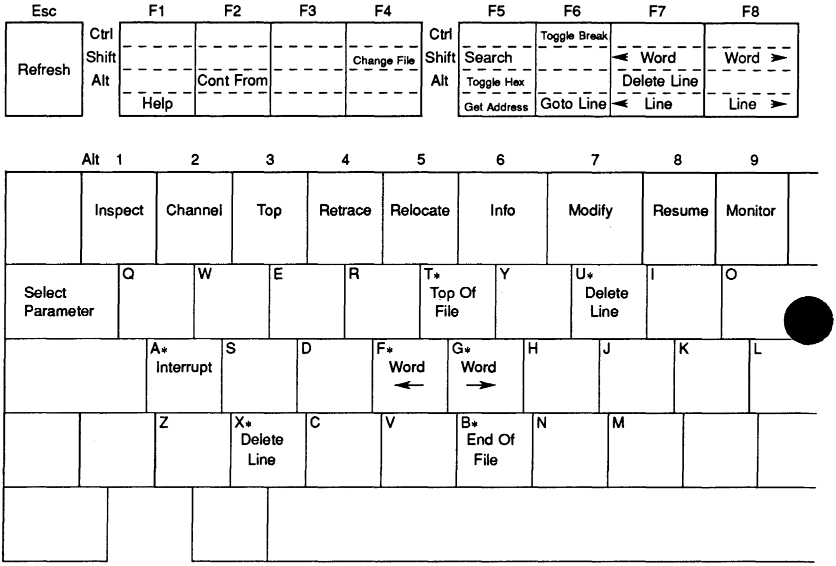
This appendix gives the keyboard assignments for the debugger symbolic functions for both the IBM PC and compatibles (`PCANSI.ITM` and `PCBANSI.ITM`), and the NEC PC (`NECPC.ITM`). Some of the keys are applicable to the simulator aswell.

B.1 IBM PC LH-keypad

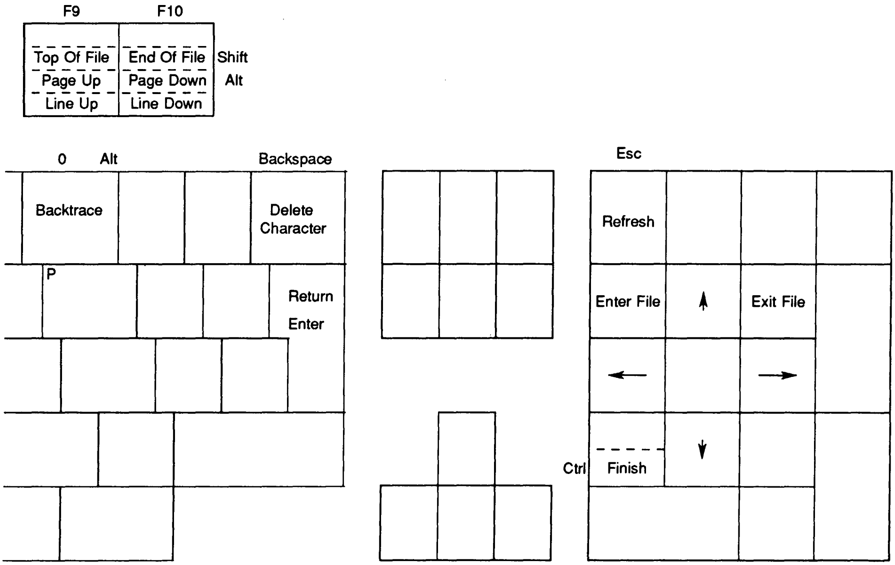


	F1	F2
Ctrl		
Shift		
Alt		Cont from
	Help	
Ctrl		
Shift		Change File
Alt		
Ctrl		Toggle Break
Shift	Search	
Alt	Toggle Hex	
	Get Address	Goto Line
Ctrl		
Shift	← Word	Word →
Alt	Delete Line	
	← Line	Line →
Ctrl		
Shift	Top Of File	End Of File
Alt	Page Up	Page Down
	Line Up	Line Down
	F9	F10

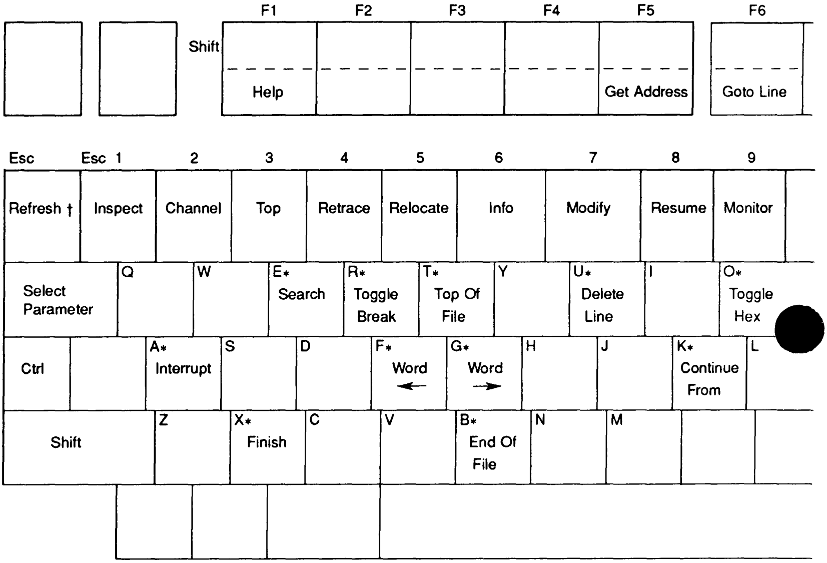
B.2 IBM PC main keyboard



* Ctrl + key

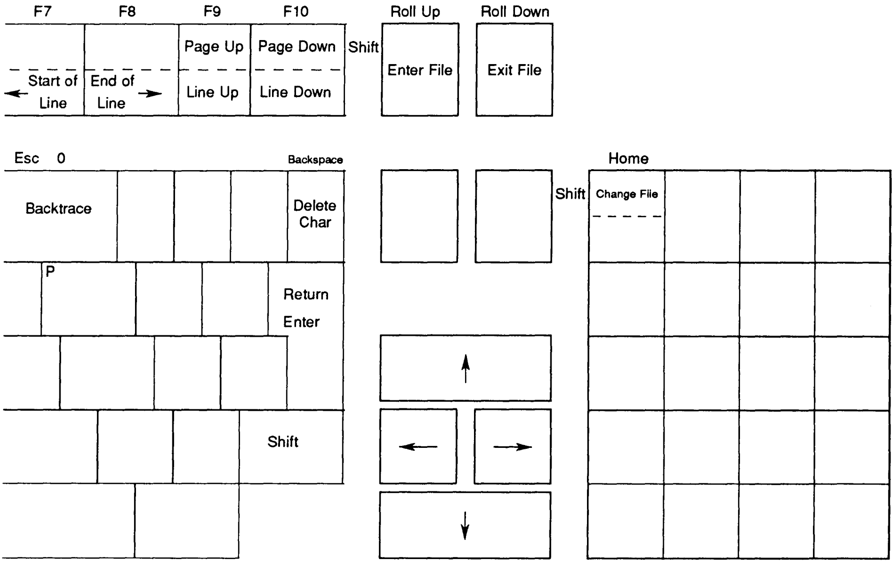

F9
F10
Top Of File
End Of File
Shift
Page Up
Page Down
Alt
Line Up
Line Down
0
Alt
Backspace
Backtrace
Delete Character
P
Return
Enter
Esc
Refresh
Enter File
Exit File
Ctrl
Finish

B.3 NEC PC keyboard layout



* Ctrl + key

† Refresh = Esc Esc


F7
F8
F9
F10
Roll Up
Roll Down
Start of Line
End of Line
Page Up
Line Up
Page Down
Line Down
Shift
Enter File
Exit File
Esc
0
Backspace
Backtrace
Delete Char
Home
Shift
Change File
P
Return
Enter
Shift

Worldwide Headquarters

INMOS Limited
1000 Aztec West
Almondsbury
Bristol BS12 4SQ
UNITED KINGDOM
Telephone (0454) 616616
Fax (0454) 617910

Worldwide Business Centres

USA

INMOS Business Centre
Headquarters (USA)
SGS-THOMSON Microelectronics Inc.
2225 Executive Circle
PO Box 16000
Colorado Springs
Colorado 80935-6000
Telephone (719) 630 4000
Fax (719) 630 4325

SGS-THOMSON Microelectronics Inc.
Sales and Marketing Headquarters (USA)
1000 East Bell Road
Phoenix
Arizona 85022
Telephone (602) 867 6100
Fax (602) 867 6102

INMOS Business Centre
SGS-THOMSON Microelectronics Inc.
Lincoln North
55 Old Bedford Road
Lincoln
Massachusetts 01773
Telephone (617) 259 0300
Fax (617) 259 4420

INMOS Business Centre
SGS-THOMSON Microelectronics Inc.
9861 Broken Land Parkway
Suite 320
Columbia
Maryland 21045
Telephone (301) 995 6952
Fax (301) 290 7047

INMOS Business Centre
SGS-THOMSON Microelectronics Inc.
200 East Sandpointe
Suite 650
Santa Ana
California 92707
Telephone (714) 957 6018
Fax (714) 957 3281

INMOS Business Centre
SGS-THOMSON Microelectronics Inc.
2620 Augustine Drive
Suite 100
Santa Clara
California 95054
Telephone (408) 727 7771
Fax (408) 727 1458

INMOS Business Centre
SGS-THOMSON Microelectronics Inc.
1310 Electronics Drive
Carrollton
Texas 75006
Telephone (214) 466 8844
Fax (214) 466 7352

ASIA PACIFIC

Japan

INMOS Business Centre
SGS-THOMSON Microelectronics K.K.
Nisseki Takanawa Building, 4th Floor
18–10 Takanawa 2-chome
Minato-ku
Tokyo 108
Telephone (03) 280 4125
Fax (03) 280 4131

Singapore

INMOS Business Centre
SGS-THOMSON Microelectronics Pte Ltd.
28 Ang Mo Kio Industrial Park 2
Singapore 2056
Telephone (65) 482 14 11
Fax (65) 482 02 40

EUROPE

United Kingdom

INMOS Business Centre
SGS-THOMSON Microelectronics Ltd.
Planar House
Parkway Globe Park
Marlow
Bucks SL7 1YL
Telephone (0628) 890 800
Fax (0628) 890 391

France

INMOS Business Centre
SGS-THOMSON Microelectronics SA
7 Avenue Gallieni
BP 93
94253 Gentilly Cedex
Telephone (1) 47 40 75 75
FAX (1) 47 40 79 27

West Germany

INMOS Business Centre
SGS-THOMSON Microelectronics GmbH
Bretonischer Ring 4
8011 Grasbrunn
Telephone (089) 46 00 60
Fax (089) 46 00 61 40

Italy

INMOS Business Centre
SGS-THOMSON Microelectronics SpA
V.le Milanofiori
Strada 4
Palazzo A/4/A
20090 Assago (MI)
Telephone (2) 89213 1
Fax (2) 8250449