

Parsytec Toolset Supplement
Documentation V3.2

© Parsytec Industriesysteme GmbH

July 1993

Please read this first,
before doing any installation

Parsytec reserves the right to make changes in specifications at any time and without notice. The information furnished by Parsytec in this publication is believed to be accurate, however no responsibility is assumed for its use, nor for any infringement of patents or rights of third parties resulting from its use. No licence is granted under any patents, trademarks or other rights of Parsytec.

This manual is Copyright ©1993 by Parsytec GmbH

All rights reserved. This document may not, in whole or in part, be copied, photocopied, reproduced or reduced to any electronic medium or machine readable form without prior consent, in writing, from Parsytec Industriesysteme GmbH, D-52072 Aachen, Roermonder Strasse 197, Germany.

Transputer is a trademark of the INMOS Limited.

MS-DOS is a registered trademark of Microsoft Corp.

IBM, PC-AT are trademarks of International Business Machines Corp.

inmos and occam are trademarks of INMOS Limited.

The use of registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Contents

1	Introduction	3
2	The Parsytec Reset Scheme	5
2.1	The implementation of the Parsytec Reset Scheme	7
2.2	The 'linkboot' library	8
2.3	The host file server 'iserver'	12
2.4	The debugger 'idebug'	12
2.5	The skip loader 'iskip'	12
3	Installation of the Parsytec Toolset Supplement V3.2	13
3.1	Installation on a PC	13
3.2	Installation on a Sun 4	15
3.2.1	Installation of the 3rd Generation Toolset Supplement	15
3.2.2	Installation of the Inquest Toolset Supplement	19
4	Supplement confidence check	23
4.1	ANSI C Toolset examples	23
4.1.1	A single processor application	24
4.1.2	A single processor application with skip loader	26
4.1.3	A two processor application	26
4.2	Occam 2 Toolset examples	30

4.2.1	A single processor application	30
4.2.2	A single processor application with skip loader	32
4.2.3	A two processor application	32
5	Notes on recompiling existing toolset applications	35
6	Using the 3rd generation Toolset on machines with multiple link access	37

List of Figures

2.1	Parsytec - UniLink	6
2.2	Reset Logic	7
2.3	Reset - Analyse Cycle	8
4.1	Example two processor configuration	23
4.2	Example two processor configuration	30

This manual frequently mentions parts of the original Inmos set of documentation, which are the following:

<u>Title of documentation</u>	<u>Document number</u>
IMS D7314A IBM 386 PC ANSI C Toolset delivery manual	72-TDS-351-01
IMS D4314A Sun 4 ANSI C Toolset delivery manual	72-TDS-350-01
IMS D7314A IBM-PC ANSI C Toolset Release Notes	72-TDS-349-02
ANSI C Toolset Reference Manual	72-TDS-346-01
IMS D7305A IBM 386 PC Occam 2 Toolset delivery manual	72-TDS-389-01
IMS D4305A Sun 4 Occam 2 Toolset delivery manual	72-TDS-364-01
IMS D4305A Sun 4 IMS D6305A VAX/VMS IMS D7305A IBM 386 PC Occam 2 Toolset Release Notes	72-TDS-365-02
Occam 2 Toolset Reference Manual	72-TDS-367-01
IMS D4300A Sun-4 Inquest Toolset delivery manual	72-TDS-374-00
Inquest Toolset Debugger Tutorial	72-TDS-373-00
Inquest Toolset User Manual	72-TDS-372-00
Documentation Basic Transputer System Software for Sun (shipped as part of Parsytec host adapter like BBK-S4 or VMTM.)	

Chapter 1

Introduction

The Parsytec Toolset Supplement V3.2¹ contains everything necessary to use the 3rd generation Toolsets and the Inquest Toolset together with Parsytec transputer hardware products. It is basically the support of the various Parsytec host interfaces and the Parsytec nearest neighbour reset scheme.

This documentation describes the Parsytec nearest-neighbour reset scheme and its implications on the program development. Furthermore the installation of the Parsytec Toolset Supplement will be explained and a step-by-step confidence test will be given to ensure that everything works.

An additional chapter describes the improvement on the 3rd generation Toolset compared to the Dx214 or Dx205 releases. It is intended especially to those users who like to recompile existing applications with the new version of Toolset, as e.g. the behaviour of the configurer tool ('icconf') has been drastically changed to support virtual channel routing. A number of smaller modifications are also described which one should know when adapting to the new Toolset.

¹This documentation relates to the Inmos "3rd Generation Toolset" (ANSI-C Toolset Dx314 and the Occam Toolset Dx305) for the T2/T4/T8 transputer generation and the Inquest Toolset Dx300.

Chapter 2

The Parsytec Reset Scheme

For fault tolerant systems it is important that each node can be reset individually. Within the Parsytec reset scheme there is no global reset, but every processor can be reset at any time by its neighbour.

The combination of data and reset link is called *UniLink*. Such a *UniLink* combines the signals for link communication and system control. Electrically these are the signals *LinkIn*, *LinkOut*, *ResetIn* and *ResetOut*, which conform to the **RS-422** definition. The reset lines are software controlled.

In contrast to the Parsytec reset Scheme, the INMOS scheme uses the signals *LinkIn*, *LinkOut*, *ResetIn*, *AnalyseIn* and *ErrorOut*. On Parsytec transputer nodes, every incoming reset results in an analyse-reset cycle. Therefore the iserver option 'sa' is included in the 'sr' option. The iserver option 'se' (test error flag) is not supported on Parsytec hardware.

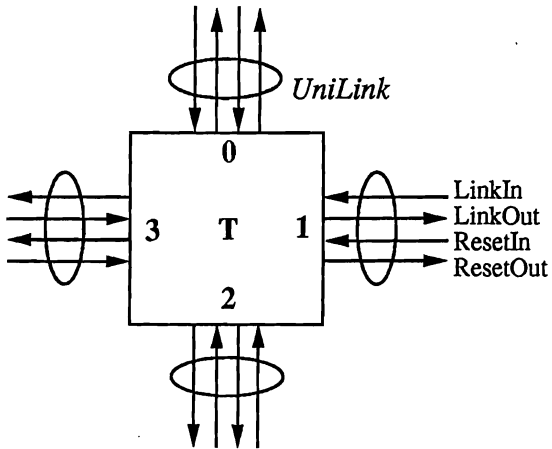


Figure 2.1: Parsytec - UniLink

The Parsytec reset scheme is reflected in the following Toolset objects:

- The library 'linkboot.lib'
- The host file server 'iserver'
- The debugger 'idebug'
- The skip loader 'iskip'

2.1 The implementation of the Parsytec Reset Scheme

This section describes how the Parsytec Reset Scheme is implemented on all Parsytec transputer nodes¹. Figure 2.2 shows only two of the four *UniLinks*. The data links (*LinkIn*, *LinkOut*) are connected directly to the transputer (e.g. T8). All Reset links (*ResetIn*, *ResetOut*) are collected at a *Reset Logic*.

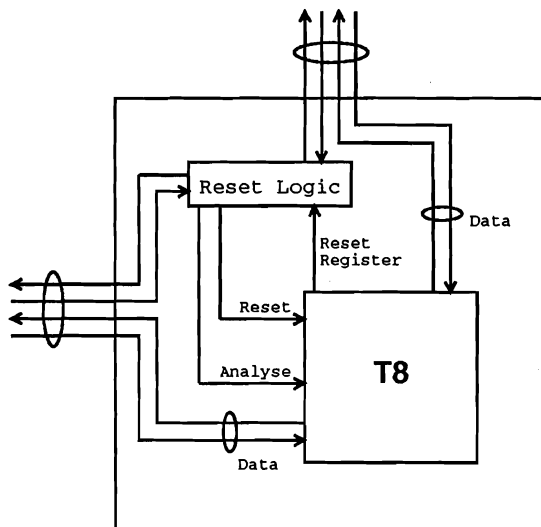


Figure 2.2: Reset Logic

The *Reset Logic* detects an incoming reset signal (see *Link-ResetIn* in figure 2.3), which has to be active high for $t_1 \geq 128\mu\text{s}$. The rising edge of the *Link-ResetIn* signal causes the *Reset Logic* to set the *Transputer-Analyse Signal*. After a delay of $200\text{ns} \leq t_2 \leq 6,4\mu\text{s}$ (depending on board), the logic activates the *Transputer-Reset* signal. *Transputer-Reset* is switched to low by the falling edge of *Link-ResetIn* and, with a delay of $t_3=200\text{ ns}$, *Transputer-Analyse* is also deactivated.

To reset its neighbour node, a transputer writes into its reset

¹except on the MTM 2-10 and VMEbus boards, where analyse is not asserted.

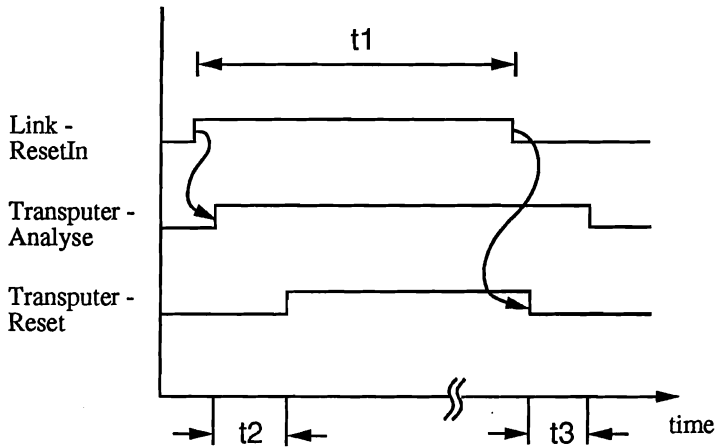


Figure 2.3: Reset - Analyse Cycle

register which is detected by the *Reset Logic*. The Occam or C functions used are listed in Chapter 2.2. The logic activates the corresponding *ResetOut* signal.

2.2 The 'linkboot' library

This library is added to the application code by the code collector 'icollect' whenever the resulting bootable file is to be loaded on a multi-transputer network. The code contained in is used to boot the application program onto the transputer network, i.e. it provides every transputer with the code fragments it should execute according to the configuration file (*.cfs).

Working with Parsytec Transputer hardware a network of transputers is not reset on the whole by using the iserver option 'SR'² as it is the case with Inmos hardware. Instead a pair of reset lines parallel to the transputer data links allow the controlled reset of individual transputers in a network. For that reason the linkboot

²Using Parsytec hardware the iserver option 'SR' causes only the reset of the root transputer, that is the one connected directly to the host.

library is adapted to send a reset signal down one link to an adjacent processor before the code is loaded.

Without the adapted linkboot library from the Parsytec Toolset Supplement a multi-processor application will usually fail to boot (except directly after power-on reset). In this case, the iserver will display an error message indicating that the application code can not be loaded completely.

The function code (see Occam and C listing below) which is used to send a reset signal down any of the four links makes use of a special register available with all Parsytec transputer nodes.

```

-----
--
-- PROC hardreset (VAL INT link.number)
--
-- This Occam process is used to reset the transputer
-- directly connected via the link number passed
-- as parameter, using the nearest neighbour reset
-- for Parsytec hardware. Parameter link.number must
-- be in the range 0...3.
--
-----

```

```

PROC hardreset (VAL INT link.number)
  INT addr.reset :
  -- reset PAL addr. #20000030 for T4/T8
  PLACE addr.reset AT #20000030 :
  -- reset PAL address #7FFF for T2
  -- PLACE addr.reset AT #7FFF :
  TIMER clock :
  VAL INT wait IS 2 :
  -- 2 times 64 microseconds
  SEQ
    addr.reset := 0
    addr.reset := 1
    addr.reset := 2
    addr.reset := 3
    addr.reset := (1 << link.number)
    -- code for link to be reset
    clock ? time
    clock ? AFTER time PLUS wait
    -- wait 128 microseconds
    addr.reset := 0
    -- clear reset

```

```

/*****
*
*   void hardreset (link)
*
*   This C function is used to reset the transputer
*   directly connected via the link number passed
*   as parameter, using the nearest neighbour reset
*   for Parsytec hardware. Parameter link must be
*   in the range 0...3.
*
\*****/
#ifdef T2
/* for T2 transputers */
#define RESET_REGISTER (int *)0x7FFC
#else
/* for T4/T8 transputers */
#define RESET_REGISTER (int *)0x00000C0
#endif

void hardreset (link)
int link;
{
    *RESET_REGISTER = 0;          /* first write */
    *RESET_REGISTER = 1;          /* sequence to */
    *RESET_REGISTER = 2;          /* unlock link */
    *RESET_REGISTER = 3;          /* reset hardware */

    *RESET_REGISTER = (1<<link); /* set the reset line */

    /* ICC call to wait 128 us (low priority) */
    ProcWait(2);

    *RESET_REGISTER = 0;          /* clear reset */
}

```

2.3 The host file server 'iserver'

The iserver works slightly different on Parsytec hardware than it is described in the Toolset Reference Manual (chapter 13. iserver):

- the 'se' option is not supported for Parsytec hardware.
- the 'sr' option resets only the transputer directly connected to the host.
- the 'sa' option is included in the 'sr' option, because every reset results in a analyse-reset cycle.

2.4 The debugger 'idebug'

Starting the debugger on Parsytec hardware the option 'ap' (assert non-INMOS subsystem analyse) must always be used. For more information, please refer to the Toolset Reference Manual (chapter 4. idebug).

2.5 The skip loader 'iskip'

Using the skip loader on Parsytec hardware, the option 'rp' (reset alternative subsystems) is required. The Toolset Reference Manual (chapter 15. iskip) describes the skip loader in detail.

After an 'iskip' command the 'iserver' must not be started with the 'sr' switch, otherwise the root transputer will be reset and the 'iskip' has no effect anymore.

Note: The 'sr' switch is included in the 'sb' switch. As the 'sr' switch must not be used after an 'iskip' command, the 'sb' switch cannot be used in that case (The iserver has to be started with the 'ss' and 'sc' options instead).

Chapter 3

Installation of the Parsytec Toolset Supplement V3.2

3.1 Installation on a PC

Before installing the Parsytec Toolset Supplement V3.2, you should have installed the Inmos Toolset D7314 or D7305 on your PC according to the appropriate Inmos 'Toolset Delivery Manual'. Please note the prerequisites for running the Toolset as described there in chapter 1.2 (Prerequisites for running the toolset). For the following description it is assumed that the Inmos Toolset has been installed in the default directory (C:\D7314A for ANSI C, resp. C:\D7305A for Occam).

The supplement software for IBM 386 PC (or compatible) running MS-DOS 5.0 (or later) is delivered on one standard 3.5 inch double sided high density 1.4 MByte disk. This disk contains the root directories D7314A (C Toolset) and D7305A (Occam Toolset), including the supplement subdirectory PARSYTEC.SUP which contains

- the changed linkboot library (see chapter 2.2), which is necessary to support both reset schemes used by INMOS and Parsytec hardware,
- the modified linkboot library sources
- additional connection databases (see Toolset Reference Manual, 13.8 The connection database) to support Parsytec PC host adapter hardware.

in the corresponding Toolset subdirectories. The disk can be copied to the harddisk using the standard DOS XCOPY command (Provided that 'B' is your floppy disk device):

C Toolset users have to type in the command line

```
XCOPY B:\D7314A C:\D7314A /S /V
```

Occam Toolset users have to type in the command line

```
XCOPY B:\D7305A C:\D7305A /S /V
```

To complete the installation change to the new PARSYTEC.SUP directory on the harddisk

```
C:
cd \D7314A\PARSYTEC.SUP
resp.
cd \D7305A\PARSYTEC.SUP
```

and start the *install* batch to copy the new linkboot library and the new connection databases into the corresponding Inmos Toolset subdirectories.

NOTE: The linkboot library of the Inmos Toolset will be overwritten. Generally this will not generate any problems, because the new library supports as well Parsytec as Inmos hardware.

```
install
```

The new connection databases are now added into the Toolset subdirectory 'CONNECT':

bbkpc.dat:

```
|BBKPC |T|localhost|#150|b004|||Passive Bus Bridge for PC|
```

tpmpc.dat

```
|TPMPC |T|localhost|#150|b004|||Transputer module for PC|
```

mtmpc.dat

```
|MTMPC |T|localhost|#150|b004|||Multi Transputer Module PC|
```

All these PC adapters are compatible to the 'b004' adapter. The device names are default names and the devices are accessed at address #150. If one has used some other name or address, the connection database file must be changed. Please refer to the hardware documentation to find out the board address and how to change it, if necessary.

E.g. if you have a BBK-PC passive bus bridge installed in your PC, you should set the iserver environment variables as follows:

```
set ICONDB=C:\D7314A\CONNECT\BBKPC.DAT  
resp.  
set ICONDB=C:\D7305A\CONNECT\BBKPC.DAT  
  
set TRANSPUTER=BBKPC
```

3.2 Installation on a Sun 4

3.2.1 Installation of the 3rd Generation Toolset Supplement

Before installing the Parsytec Toolset Supplement V3.2, you should have installed the Inmos Toolset D4314 or D4305 on your Sun 4 according to the appropriate Inmos 'Toolset Delivery Manual'. Please note the prerequisites for running the Toolset as described there in chapter 1.2 (Prerequisites for running the toolset). For the following description it is assumed that the Basic Software (see Basic Transputer System Software Documentation) is installed and that "\$D4314A" refers to the directory where the Inmos ANSI C

Toolset has been installed ("D4305A" for the Occam Toolset).

The Parsytec Toolset Supplement V3.2 software for SUN-UNIX computers is distributed on one 3.25" floppy disk. To install this software on your system you must have superuser permission. After the installation every normal user may use Toolset.

The following installation guide covers the C Toolset and the Occam Toolset installation on SUN 4.

You may use this notation in the following commands by setting an environment variable of the same name, for example:

```
setenv Toolset D4314A
resp.
setenv Toolset D4305A
```

This works only if the environment variable D4314A *resp.* D4305A is already set from the installation of the INMOS release.

Be very careful during the installation of Toolset. Though UNIX is a user friendly and save operating system it is very easy for the superuser to destroy a great amount of software.

Contact your local system manager for further help before starting dangerous experiments!

Step by step software installation instructions :

- Get superuser permission. As mentioned above contact the system manager and be very carefully.
- Create a new user entry with user number 94 named **toolset**. This is required only for the reason that every file is owned by somebody. All the files on the disk are owned by this dummy user. This user will never login. If it is not possible to use 94 as user number on your system you may use another number. The entry in the passwd file should look like this:

```
toolset:*:94:25:Toolset system:/usr:/bin/csh
```


- Add the toolset user to the group `vmebus`. On a machine not connected to a network the only action is to change the following line in the file `/etc/group` to:

```
vmebus:*:25:toolset
```

Otherwise the system manager knows how to add the Toolset user.

- Now read in the contents of the **Parsytec Toolset Supplement** disk for Inmos Toolset D4305 and D4314 into the Toolset directory. Let us assume `/dev/fd0` is your floppy disk device:

```
cd $Toolset          # set working directory
                    # to Toolset root
bar xvpf /dev/fd0   # read in floppy
```

- change to the supplement directory and build a new iserver (If you don't have the default configuration of the *Basic Transputer System Software* you may have to edit the makefile in the current directory).

```
cd parsytec.supplement
make iserver
```

Note: Because of a new iserver interface the `link.o` file of the *Basic Transputer System Software* cannot be used to build a new iserver. The `parsylink.o` file (located in the supplement directory `linkio/src` of the iserver source) is used instead of the `link.o` file. The main difference is, that the interface functions got the prefix "Parsy" (e.g. the `OpenLink()` function is now called `ParsyOpenLink()`). With the exception of the above mentioned `parsylink.o`, all other files from the *Basic Transputer System Software* are still used.

- Change the owner/group-number of all supplement files with the following command, assuming owner `toolset` and group `vmebus`:

```
/etc/chown -R toolset.vmebus $Toolset
```

- Complete the installation by typing in the command line (Note: this will overwrite the linkboot library and the iserver of the Inmos Toolset):

```
make install
```

The Parsytec Toolset Supplement provides a new connection database ('parsytec.dat') in the Toolset subdirectory 'connect', prepared for the use of Parsytec host adapter hardware.

The 5th column of a connection database file (e.g. see `parsytec.dat` file below) contains the *linkdev* (type of device providing user link). The *linkdev* 'parsytec' must be used for all Parsytec bus bridges (Sbus or VMEbus). The basic software of the bus bridges, which has to be linked to the tools with direct hardware access, covers the different hardware. In any case the links are consecutively numbered beginning with 0. The available link *Ids* may be displayed with the 'links' command. Only those *Id*'s can be used in the 4th column of a connection database file.

The file 'parsytec.dat' contains the following text lines:

parsytec.dat:

```
|any |T|localhost|0|parsytec|||BBK-S4 with T805 network|
|0 |T|localhost|0|parsytec|||BBK-S4 with T805 network|
|any |T|localhost|1|parsytec|||BBK-S4 with T805 network|
|1 |T|localhost|1|parsytec|||BBK-S4 with T805 network|
|any |T|localhost|2|parsytec|||BBK-S4 with T805 network|
|2 |T|localhost|2|parsytec|||BBK-S4 with T805 network|
|any |T|localhost|3|parsytec|||BBK-S4 with T805 network|
|3 |T|localhost|3|parsytec|||BBK-S4 with T805 network|
```

E.g. if you have Sun 4 system (Sbus or VMEbus) with a Parsytec host adapter module installed you should set ICONDB as follows:

```
setenv ICONDB $Toolset/connect/parsytec.dat
```

In contrast to the iserver of the previous Toolset, you now have to specify a so called *capability*, which describes the link entry. Every link can be assigned to any number of capabilities e.g. to describe the type of Transputer network it is connected with (see the file parytec.dat above).

In order to use a specific link with the above listed connection database, you can set the environment variable TRANSPUTER as follows:

```
setenv TRANSPUTER 0
```

In case that all link entries are equivalent (i.e. they are connected to the same type of network) you can also use the following setting:

```
setenv TRANSPUTER any
```

For more information on connection database files please refer to the 'Toolset Reference Manual: 13.7 Connecting transputer to computer networks'.

3.2.2 Installation of the Inquest Toolset Supplement

Before installing the Parsytec Toolset Supplement for the Inquest Toolset, you should have installed the Inmos 3rd Generation Toolset D4314 resp. D4305 and the Inmos Inquest Toolset (D4300) on your Sun4 according to the appropriate Inmos Toolset 'Delivery Manual'. Please note the prerequisites for running the Toolsets as described there in chapter 1.2. For the following description it is assumed that the Basic Software (see Basic Transputer System Software Documentation) is installed and that "\$D4300A" refers to the directory where the Inmos Inquest Toolset has been installed.

You may use this notation in the following commands by setting

an environment variable of the same name, for example:

```
setenv Inquest $D4300A
```

This works only if the environment variable \$D4300A is already set from the installation of the INMOS release.

Step by step software installation instructions :

- Get superuser permission. For further help please contact the system manager and be very carefully.
- Now read in the contents of the **Parsytec Toolset Supplement** disk for Inmos Inquest Toolset D4300 into the Inquest directory. Let us assume /dev/fd0 is your floppy disk device:

```
cd $Inquest          # set working directory
                    # to Inquest root
bar xvpf /dev/fd0    # read in floppy
```

- change to the supplement directory and build new Inquest tools (If you don't have the default configuration of the *Basic Transputer System Software* you may have to edit the makefile in the current directory).

```
cd parsytec.supplement
make tools
```

Note: Because of a new transport service interface the *link.o* file of the *Basic Transputer System Software* cannot be used to build new tools. The *parsylink.o* file (located in the supplement *extend/source* directory) is used instead of the *link.o* file. The main difference is, that the interface functions got the prefix "Parsy" (e.g. the `OpenLink()` function is now called `ParsyOpenLink()`). With the exception of the above mentioned *parsylink.o*, all other files from the *Basic Transputer System Software* are still used.

- Change the owner/group-number of all supplement files with the following command, assuming owner `toolset` and group `vmebus`:

```
/etc/chown -R toolset.vmebus $Inquest
```

- Complete the installation by typing in the command line (Note: this will overwrite the tools in the `$Inquest/bin` directory):

```
make install
```

Use the step-by-step tutorial (see: Debugger Tutorial '3 An example debugging session') as **confidence check**.

Note: The transputer bootables which are used by the Inquest tools (*irun*, *iprof* and *imon*), must be recompiled completely on the host where the Inquest Toolset is running on, to make sure that the Inquest tools will find the correct versions of the C resp. Occam Toolset libraries. Otherwise the tools could not work properly.

Due to the Parsytec Reset Scheme, which is not yet supported by some tools of the Inquest Toolset, the skip loader of the 3rd Generation Toolsets must be used instead of the skip loader of the Inquest Toolset. Therefore the command lines containing

```
... -sc skipn.btl ...
```

as described in the User Manual (4.2 The skip loader) must be changed to:

```
... -sb iskip.btl n -rp ...
```

Therefore the ISEARCH environment has to be extended with `$D4314A/tools/` resp. `$D4305A/tools/`.

Because of the same reason, the '`rspy`' works only properly after a total reset of the network.

Chapter 4

Supplement confidence check

This chapter describes several examples for the C and Occam Toolsets, to check the installation.

The tests should be done in a separate directory, which is called *mytest* throughout this chapter.

4.1 ANSI C Toolset examples

The examples within this section may be loaded onto the network shown in figure 4.1.

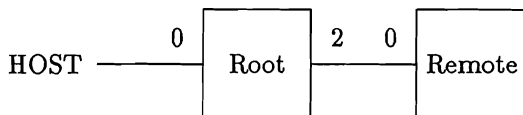


Figure 4.1: Example two processor configuration

4.1.1 A single processor application

This example is built to run on a single transputer, and will check if the iserver works correctly.

Running the example on a PC

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
copy \D7314A\examples\simple\hello.p.c  
copy \D7314A\examples\simple\hello.cfs
```

Compile the C source file:

```
icc hello.c /t8
```

Link the file:

```
ilink hello.tco /f cstartup.lnk /t8
```

To run the example on a T8 transputer, the entry

```
T425 (memory = 1M) Single;
```

in the configuration file 'hello.cfs' has to be changed to:

```
T800 (memory = 1M) Single;
```

Configure and collect the example:

```
icconf hello.cfs  
icollect hello.cfb
```

Downloading the code

```
iserver /sb hello.btl
```

should generate the following output:

```
Hello World
```


Running the example on a Sun

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
cp $D4314A/examples/simple/hello.p.c  
cp $D4314A/examples/simple/hello.cfs
```

Compile the C source file:

```
icc hello.c -t8
```

Link the file:

```
ilink hello.tco -f cstartup.lnk -t8
```

To run the example on a T8 transputer, the entry

```
T425 (memory = 1M) Single;
```

in the configuration file 'hello.cfs' has to be changed to:

```
T800 (memory = 1M) Single;
```

Configure and collect the example:

```
icconf hello.cfs  
icollect hello.cfb
```

Downloading the code

```
iserver -sb hello.btl
```

should generate the following output:

```
Hello World
```

4.1.2 A single processor application with skip loader

Within this section the options are preceded by '/' for MS-DOS based toolsets, the slashes must be replaced with '-' for UNIX-based toolsets.

To check if the skip loader 'iskip' works properly, the bootable code of the previous example may be used.

First load the skip loader on the Root node (see figure 4.1):
(The option 'rp' causes the Remote processor to be reset.)

```
iskip 2 /rp
```

Now download the code to the Remote processor via the skip loader:

Note: in this case the iserver option 'sr' must not be used.

```
iserver /si /ss /sc hello.btl
```

The following output should be generated:

```
Hello World
```

4.1.3 A two processor application

This example runs on two transputers (T800 or T805) and checks if the correct version of the *linkboot* library is installed. If this example runs at least two times, one can be sure, that the installation has been done correctly.

Running the example on a PC

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
copy \D7314A\examples\icconf\c-cfs\src\hello.c
copy \D7314A\examples\icconf\c-cfs\src\hostmult.c
copy \D7314A\examples\icconf\c-cfs\src\worldp.c
copy \D7314A\examples\icconf\c-cfs\src\hello2.cfs
copy \D7314A\examples\icconf\c-cfs\src\setup.h
copy \D7314A\examples\icconf\hw-cfs\hwtwo.cfs
copy \D7314A\examples\icconf\hw-cfs\hwsetup.h
```

Compile the C source files:

```
icc hello.c /t8
icc hostmult.c /t8
icc worldp.c /t8
```

Link these files:

```
ilink hello.tco /f cstartup.lnk /t8
ilink hostmult.tco /f cstartup.lnk /t8
ilink worldp.tco /f cstartup.lnk /t8
```

To use the same configuration as shown in figure 4.1, the entry

```
connect Root.link[2] to Remote.link[1];
```

in the file 'hwtwo.cfs' has to be changed to:

```
connect Root.link[2] to Remote.link[0];
```

Configuring the example with the option 'G'

```
icconf hello2.cfs /g
```

generates the warning:

```
Warning-icconf- using single hop software virtual links
```

At last, collect the example:

```
icollect hello2.cfb
```

Downloading the code

```
iserver /sr hello2.btl
```

should generate the following output:

```
Hello world from C
World hello from C
Hello world from C
World hello from C
Hello world from C
World hello from C
Hello world from C
World hello from C
Hello world from C
World hello from C
```

Running the example on a Sun

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
cp $D4314A/examples/icconf/c-cfs/src/hello.c
cp $D4314A/examples/icconf/c-cfs/src/hostmult.c
cp $D4314A/examples/icconf/c-cfs/src/worldp.c
cp $D4314A/examples/icconf/c-cfs/src/hello2.cfs
cp $D4314A/examples/icconf/c-cfs/src/setup.h
cp $D4314A/examples/icconf/hw-cfs/hwtwo.cfs
cp $D4314A/examples/icconf/hw-cfs/hwsetup.h
```

Compile the C source files:

```
icc hello.c -t8
icc hostmult.c -t8
icc worldp.c -t8
```

Link these files:

```
ilink hello.tco -f cstartup.lnk -t8
ilink hostmult.tco -f cstartup.lnk -t8
ilink worldp.tco -f cstartup.lnk -t8
```

To use the same configuration as shown in figure 4.1, the entry

```
connect Root.link[2] to Remote.link[1];
```

in the file 'hwtwo.cfs' has to be changed to:

```
connect Root.link[2] to Remote.link[0];
```

Configuring the example with the option 'G'

```
icconf hello2.cfs /g
```

generates the warning:

```
Warning-icconf- using single hop software virtual links
```

At last, collect the example:

```
icollect hello2.cfb
```

Downloading the code

```
iserver -sr hello2.btl
```

should generate the following output:

```
Hello world from C
World hello from C
Hello world from C
World hello from C
Hello world from C
World hello from C
Hello world from C
World hello from C
Hello world from C
World hello from C
```

4.2 Occam 2 Toolset examples

The examples within this section may be loaded onto the network shown in figure 4.2.

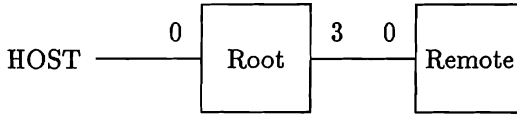


Figure 4.2: Example two processor configuration

4.2.1 A single processor application

This example is built to run on a single transputer, and will check if the iserver works correctly.

Running the example on a PC

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
copy \D7305A\examples\manuals\simple\simple.occ
copy \D7305A\examples\manuals\simple\simple.pgm
```

Compile the occam source file:

```
oc /ta simple.occ
```

Link the file:

```
ilink /ta simple.tco hostio.lib /f occama.lnk
```

Configure and collect the example:

```
occonf simple.pgm
icollect simple.cfb
```

Download the code:

```
iserver /sb simple.btl
```

The program should ask for a name, generating the following output:

```
Please type your name : Occam  
Hello Occam
```

Running the example on a Sun

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
copy $D4305A/examples/manuals/simple/simple.occ  
copy $D4305A/examples/manuals/simple/simple.pgm
```

Compile the occam source file:

```
oc -ta simple.occ
```

Link the file:

```
ilink -ta simple.tco hostio.lib -f occama.lnk
```

Configure and collect the example:

```
oconf simple.pgm  
icollect simple.cfb
```

Download the code:

```
iserver -sb simple.btl
```

The program should ask for a name, generating the following output:

```
Please type your name : Occam  
Hello Occam
```

4.2.2 A single processor application with skip loader

Within this section the options are preceded by '/' for MS-DOS based toolsets, the slashes must be replaced with '-' for UNIX-based toolsets.

First load the skip loader on the Root node (see figure 4.2):
(The option 'rp' causes the Remote processor to be reset.)

```
iskip 3 /rp
```

Now download the code to the Remote processor via the skip loader:

Note: in this case the iserver option 'sr' must not be used.

```
iserver /si /ss /sc simple.btl
```

The program should ask for a name, generating the following output:

```
Please type your name : Occam  
Hello Occam
```

4.2.3 A two processor application

This example runs on two transputers and checks if the correct version of the *linkboot* library is installed. If this example runs at least two times, one can be sure, that the installation has been done correctly.

Running the example on a PC

Set the current directory to *mytest*:

```
cd mytest
```

Copy some files to the current directory:

```
copy \D7305A\examples\manuals\config\myprog.pgm
```



```
copy \D7305A\examples\manuals\config\app.occ
copy \D7305A\examples\manuals\config\buff.occ
```

To build a two processor program, the entry *number.of.processors* in the *myprog.pgm* file must be changed from 1 to 2. The *myprog.pgm* file is prepared for T414 transputers. Using other processor types, the *T414* entries within this file has to be changed.

Use the *imakef* tool to get the corresponding makefile *myprog.mak*:
imakef myprog.btl

Now, the bootable code can be generated easily:
make /f myprog.mak

Downloading the code
iserver /sb myprog.btl

ould generate the following output:
Hello world

Running the example on a Sun

Set the current directory to *mytest*:
cd mytest

Copy some files to the current directory:
copy \$D4305A/examples/manuals/config/myprog.pgm
copy \$D4305A/examples/manuals/config/app.occ
copy \$D4305A/examples/manuals/config/buff.occ

To build a two processor program, the entry *number.of.processors* in the *myprog.pgm* file must be changed from 1 to 2. The *myprog.pgm* file is prepared for T414 transputers. Using other processor types, the *T414* entries within this file has to be changed.

Use the *imakef* tool to get the corresponding makefile *myprog.mak*:
imakef myprog.btl

Now, the bootable code can be generated easily:

```
make -f myprog.mak
```

Downloading the code

```
iserver -sb myprog.btl
```

should generate the following output:

```
Hello world
```

Chapter 5

Notes on recompiling existing toolset applications

The new 3rd generation Toolset introduces some new features to the user, like e.g. the virtual routing configuration, and some general changes compared to the previous 2nd generation releases (Dx214 and Dx205). It is strongly recommended to read the "Toolset Release Notes" for detailed information.

When recompiling existing applications with the 3rd generation Toolset, there are a few things which need the programmers attention:

- New behaviour of the configurer 'iconf' switches: 'NV', 'G', 'GA', 'PRE', 'PRU'
 - 'iconf' will add additional processes for routing next to the user processes, where needed. This will be indicated by one of the two configurer warnings:
using single hop software virtual links
or
using through routed software virtual links

- using one of the options 'G', 'GA', 'PRE', or 'PRU' without option 'NV', starts virtual channel routing processes, even if the used channels are explicitly placed in the *.cfs file.
- the behavior of the 2nd generation Toolset is rebuilt using the switches: 'NV' and 'G'
- New linker indirect files need to be used (see 'User Guide: 3.11.1 ANSI C Toolset'):
 - Non-configured single processor programs (those built with the collector option 'T') must have been linked using the new linker indirect file 'cnonconf.lnk'.
 - For configured programs the code should be linked by using the new linker indirect files 'cstartup.lnk' instead of 'startup.lnk' and 'cstartrd.lnk' instead of 'startrd.lnk'.
- The definition of 'clock_t' in 'time.h' has been changed from unsigned long to unsigned int:


```
typedef unsigned int clock_t
```
- When mixing languages, using the 3rd Generation ANSI C (Dx314) and 2nd Generation Occam (Dx207), virtual channels should NOT be used. Post-mortem debugging is still possible for mixed language programs but interactive debugging cannot be used in general.

Chapter 6

Using the 3rd generation Toolset on machines with multiple link access

The iserver V1.5 which is shipped within the 3rd generation Toolset offers the possibility to claim a certain link entry on machines with multiple link access (like MultiCluster-2 or SuperCluster). This simplifies the development cycle of configuration (using 'saconfig'), loading code and debugging, because a link cannot be accessed by another user between the single steps.

A session is started via:

```
iserver -sl capability -sm
```

The specified link is now reserved until the session is terminated by the 'exit' or 'quit' command. There are some predefined commands, which can be used within the session. To use the tools 'iskip', 'idebug', 'saconfig' etc. the user has to define them depending on the Toolset which has to be used. The definitions should be done in the session manager configuration file, named

in the environment variable ISESSION. The entries for the above mentioned tools could be:

iskip	-sb /usr/local/D4314A/tools/iskip.btl	UNIX
idebug	-sb /usr/local/D4314A/tools/idebug.btl	
saconfig	-sb /export/home/localhost/tools/saconfig.btl	

iskip	/sb c:\d7314a\tools\iskip.btl	DOS
idebug	/sb c:\d7314a\tools\idebug.btl	
saconfig	/sb c:\parsytec\tools\saconfig.btl	

For reference see Toolset Reference Manual: 13.6 'Using the session manager interface'.