Transputer Toolset Introduction

Version 91.1
12/1/91

Logical Systems
P.O. Box 1702
Corvallis, OR 97339 USA
(503) 753-9051
FAX: (503) 753-4629

Welcome aboard! This document is the starting point for learning about the **Transputer Toolset**. First, let's get the paperwork out of the way:

License Agreement

This package is intended for use on only one Transputer software development system at a time, by one user at a time. It is protected by United States copyright laws and various international treaty provisions.

Please note that this software and documentation is shipped outside the U.S. under provisions of the U.S. Export Control Administration regulations for "GTDR" data and may not be transferred to countries for which this general license doesn't apply.

Logical Systems imposes no royalty or licensing restrictions on programs produced using the tools in the **Transputer Toolset** as long as the original "source" form of the programs are not derived from copyrighted software provided by Logical Systems. Please contact Logical Systems for other licensing arrangements should your requirements differ.

Disclaimer

Logical Systems makes no representation or warranties, either expressed or implied, with respect to the adequacy of this documentation or the programs which it describes in regard to merchantability or fitness for any particular purpose or with respect to its adequacy to produce any particular result. The computer programs and documentation are sold "as is", and the entire risk as to quality and performance is with the buyer. In no event shall Logical Systems or their dealers be liable for special, direct, indirect or consequential damages resulting from any defect in the programs or documentation. Some states and countries do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitations and exclusions may not apply to you.

Trademarks

The following trademarked terms are used in the **Transputer Toolset** documentation:

Apple, Macintosh and MPW are trademarks of Apple Computer Corp.
AT+T and UNIX are trademarks of AT+T Bell Laboratories Corp.
Borland and Borland C++ are trademarks of Borland International Corp.
CP/M is a trademark of Digital Research Corp.
IBM, PC XT and PC AT are trademarks of International Business Machines Corp.
Microsoft and MS-DOS are trademarks of Microsoft Corp.
Motorola is a trademark Motorola Corp.
QC and QCX are trademarks of Quality Computer Systems.
SUN is a trademark of SUN Microsystems Corp.
TCX, TASM, TLNK, TLIB, LD-ONE, LD-NET, CIO and TIO are trademarks of Logical Systems.
Transputer is a trademark of Inmos Ltd.
Z80 is a trademark of Zilog Corp.
VAX and VMS are trademarks of Digital Equipment Corp.

Support

We provide 6 months of updates and telephone support for customers who purchase the **Transputer Toolset** directly from Logical Systems. If you purchased the package through another source please consult the additional information your vendor may have provided for specific support policy information.

For those who wish to stay current after the initial upgrade/support period lapses, most vendors offer some form of support contract or upgrade policy. For direct customers of Logical Systems two options are available:

1. A 12 month software maintenance contract available for $395. You get all the updates as they come out and continued telephone support.

2. One-time upgrades. You can purchase an updated copy of the **Transputer Toolset** for $125 on a as-needed basis. The upgrade comes with 2 weeks of telephone support.

Getting Bootstrapped

Whew! Now on to the OTHER important stuff.

As you probably already know, the **Transputer Toolset** is a collection of tools designed to allow you to develop computer software written in "C" and assembly language for the Inmos 16 and 32 bit Transputers. What you are probably unfamiliar with is how to actually use the tools! In general, two levels of description are needed; one describing how the various tools are used together to accomplish the overall task, and another, lower level, description of the details of the individual tools ("strategy" and "tactics" if you will). The "tactics" descriptions are provided by the individual manuals

which are included for the components of the **Transputer Toolset**, the "strategy" aspect is the subject of the remainder of this manual.

The Big Picture

The **Transputer Toolset** functions as a cross development environment with the supplied tools executed on a host system and the resulting application programs run on an attached Transputer array. Tools are provided for both program development and loading. The development tools include:

PP	The "C" preprocessor.
TCX	The "C" compiler.
TASM	The assembler.
TLNK	The linker/locator.
TLIB	The relocatable file librarian.
MAKE	A program maintenance utility.
TCC	The command line driver.

All of these tools are completely independent of the particular Transputer-to-host interface being used. The loading and execution tools include:

LD-ONE	The single Transputer loader.
LD-NET	The "network" loader for arrays of Transputers.
CIO	The host I/O server for "C" I/O on the Transputer.
TIO	A simple "example" server for standard I/O.

These tools are dependent on the actual Transputer interface being used and are supplied configured to run with a Inmos IMS B004/B008 compatible board plugged into a IBM PC XT or AT compatible system running MS-DOS (version 2.1 or higher). If you plan on running these tools with another interface (or operating system/computer type), you will probably have to make some changes to the link driver (or perhaps supply a new one). Please contact Logical Systems for information about driver availability for other interfaces (see the "I/O DRIVER INTERFACE SPECIFICATION MANUAL" for more information).

Installing and Configuring the Tools

The actual details of the installation procedure are dependent on whether a source or binary-only version of the package is being installed. Additional sources of variation include the particular host operating system environment being used and what technique, if any, your vendor has chosen for automatic software installation. More information on the particular installation procedure to be used is commonly found in the "read.me" file on the first disk (you should probably browse through it after reading this introduction). In any event, the following general configuration issues are required for most environments:

1. Wherever the executable versions of the tools are installed you must ensure that your shell/operating system can find them. Under MS-DOS, for example, this means that you must make sure that the appropriate directory pathname is part of the string assigned to the "PATH" environment variable.

2. You should create and assign an environment variable named "TMP" to the pathname of a directory to be used by the various tools for temporary files. Ideally this directory will be located on the fastest mass storage available since many of the tools use intermediate files heavily. Note that you may still override this environment variable on a case-by-case basis using a command line option to the individual tools. If neither the "TMP" variable nor the command line option is used with a particular tool the current directory will be used.

3. You should tell the preprocessor (PP), where to find the system include files to use when compiling Transputer programs. This may be specified by setting an environment variable called "PPINC" equal to the pathname to the directory which the include files will be installed in. See the PP manual for more information about this variable and what the defaults are under various host environments if this variable isn't used.

4. You should inform the linker (TLNK), where it can find the system libraries to use when linking Transputer programs. This may be specified by setting an environment variable named "TLIB" to the pathname of the directory where the libraries are installed. Note that if you don't set this environment variable you must either explicitly specify the pathname or have the appropriate library in the current directory during compilation. If you are only compiling programs for one particular type of Transputer (and floating point model), you do not need to install all the provided libraries. See the TLNK documentation and the "read.me" file on the first disk for more information.

5. You may need to tell the host-transputer interface software driver how it should communicate with the particular Transputer hardware you are using. Whether this is required depends on your system configuration and the capabilities of the driver software you are using. If necessary, this is accomplished by setting an environment variable named "LINKNAME" to the specific string defined in the software driver documentation ("I/O DRIVER INTERFACE SPECIFICATION MANUAL").

6. You may wish to create an environment variable named "TCC" to configure the TCC command line driver for your specific environment and application. See the "TCC COMMAND LINE DRIVER USER GUIDE" for additional information.

Compiling and Running a "C" Program

The following steps are necessary to compile and run a "C" program on one or more Transputers:

First, PP is run on the source code to handle "C" preprocessing functions. Then, TCX is run to convert the code to Transputer assembly language.

Next, TASM is run to assemble the code. At this point the code may be optionally included in a library using TLIB or linked into a executable file using TLNK (or both).

Finally, the code is executed using the LD-ONE loader (for the case of a single Transputer), or the LD-NET loader (if more than one Transputer is to be loaded). In either event, since the program probably does runtime I/O, the CIO "C" host I/O server will be required.

Note that the individual PP/TCX/TASM/TLNK steps outlined above are typically automated by using the TCC command line driver. TCC may be used to process "C", assembly language, "preprocessed" assembly language, or object files. One or more of the resulting object files may also be linked into a single executable. TCC may be used as a standalone interface to the various tools, or it may be used in conjunction with the MAKE program for even greater flexibility.

Assembling and Running an Assembly Language Program

The following steps are necessary to assemble and run an assembly language program on one or more Transputers:

If the assembly language program is using PP to enhance the capabilities of TASM, then it is first run through PP to handle file inclusion, macro's, etc.

Next, the file is assembled by TASM. As in the case of "C" code, the resulting relocatable file may be optionally included in a library using TLIB, or linked into a executable file using TLNK (or both).

Finally, the code is executed using the LD-ONE loader (for the case of a single Transputer), or the LD-NET loader (if more than one Transputer is to be loaded). In either event the selected host I/O server will probably either be TIO (to handle simple console I/O), or some specialized server constructed for a particular application (probably based on the TIO source code).

Again note that the individual PP/TASM/TLNK steps outlined above are typically automated by using the TCC command line driver.

Mixed "C" and Assembly Language Programs

Of course, if the desired application contains both "C" code and assembly language then creating the final executable file will be a combination of the above two processes. If the required assembly language is fairly short (and localized), the use of the "inline assembly" feature of PP and TCX may be a useful alternative to creating completely separate assembly language files.

File Extension Naming Conventions

The tools which make up the **Transputer Toolset** use a variety of file name "default extensions". These serve to distinguish different types of files. Although the types used by a particular tool are documentated in the individual manuals, the following general list may be helpful for those just getting started. Note that these are just the "usual" meanings since you are free to override the defaults:

.c	"C" source file.
.lst	Assembly language list file from TASM.
.map	Linkage map from TLNK.
.mcf	"memory configuration file" for LD-NET.
.nif	"network information file" for LD-NET.
.pal	Assembly language file which requires preprocessing.
.pp	Preprocessed text file written by PP.
.s	Motorola "S" record ROM text file written by LD-NET.
.tal	Assembly language file.
.tld	Executable generated by TLNK.
.tll	Library generated by TLIB.
.trl	Relocatable object file generated by TASM.
.vcm	"virtual channel map" file generated by LD-NET.

Sources of Information

Note that a individual manual exists for each component of the **Transputer Toolset**. An additional manual is provided to cover the file formats used by the various tools.

Another source of vital information is your local Inmos representative! No effort is made in the documentation to describe the architecture or instruction set of the various Transputers. Two publications available from Inmos which are particularly useful:

1. "transputer reference manual"

The hardware reference document for the Transputer. Also includes some architectural information.

2. "The transputer instruction set - a compiler writers' guide"

Contains the definitive description of the Transputer instruction set (a bit theoretical in presentation for some tastes).

Additional information may be gleaned from various magazine articles and other sources such as Inmos seminars, training sessions, etc. If anyone comes across a particularly useful bit of information which should be added to the above list please let us know!

Acknowledgements

As is the case with any major project, the **Transputer Toolset** is the result of the work of many people. Although the complete list of people who helped out is too lengthy to include here, several people deserve special mention (in no particular order):

Jeffrey Mock of Pixar. Early Transputer user who put together what was probably the first parallel library for "C" on the Transputer (which we used as the specification for part of our "C" library). Kindly allowed us to include a copy of his revised paper to help people figure out how to use it...

Gary Oliver of AO Computer Systems. TCX compiler writer. The only known person to keep stranger hours than Kirk Bailey. Worked an incredible number of hours for almost non-existant pay (but think of the glory).

Dave Regan of AO Computer Systems. TCX compiler writer. The fastest "C" coder in the west (or probably north, south, etc.). Worked more hours for less pay than he would like to admit.

Kirk Bailey of Logical Systems. The instigator of the Transputer Toolset. The only known person to keep stranger hours than Gary Oliver. Last seen heading south with a mob of irate programmers in hot pursuit yelling "C++, Fortran, ...".

Scott Lloyd of Alta Technology. Implemented the version of Jeff Mocks' concurrency library which is used in our library (work performed while he was with Computer Systems Architects). Also did the initial conversion of the math functions we use. All of which he accomplished in VERY short order (rumored to understand the theory of time "warps" whereby a monday through friday work week gets compressed into a single thursday afternoon).

Jim Colvin of Quality Computer Systems. Author of the QC and QCX "C" compilers which formed the original basis for TCX. A good guy we've worked with for many years but whom we've never met in the flesh (both parties being loath to minimize our charitable contributions to the phone company).