

Customer Technical Update

Issue 1

This document covers technical information, errata , and potential problems in INMOS transputer family products (ie T414,T212,M212,C011,C012,boards and software) that have come to light since the appropriate documentation was issued.

1	T800/T414 compatibility... points to note
2	T414 Halt on Error Flag
3	T800 Rev A buglist
4	IMS T800 Revision A errata sheet
4.1	Hardware problems
4.1.1	External memory interface configuration of IMS T800
4.1.2	IMS T800 Link engine
4.1.3	Internal RAM
4.2	Software problems
4.2.1	Installation on IBM PC
4.2.2	INT to REAL
4.2.3	Arithmetic features
4.2.4	Configurer
5	Differences rev A to rev B T800
6	Evaluation boards and ITEMS
6.1	IMS B001
6.1.1	Series damping resistors
6.2	IMS B001, B002 and B004: link buffering
6.3	IMS B002
6.3.1	Coding switch
6.3.2	RS232 level changer
6.3.3	Grounding power supply
6.3.4	Subsystem
6.4	IMS B003
6.4.1	Undershoot on RAMs
6.4.2	Bulk decoupling
6.4.3	Schmitts on reset
6.5	IMS B004
6.5.1	Occasional inability to download programs
6.5.2	The original PC
6.5.3	Power supply decoupling
6.6	IMS B006
6.6.1	RS 232 Terminal driver Software
6.7	IMS B007
6.7.1	Use with IMS B004
6.7.2	Test software changes between T414A and T414B transputers
6.7.3	Windows demonstration program
6.8	ITEM
6.8.1	Hardwired link connections
7	Upgrade paths T414A to T414B to T800
7.1	IMS B004 Boards
7.2	Other Boards
7.3	ITEM 4000
8	Board Maintenance and Repair

9	IMS D700B - known software problems
9.1	Installation
9.2	Occam2 language
	9.2.1 Configuration language
	9.2.2 Implemented features not documented
	9.2.3 Documented features not implemented
	9.2.4 Bugs in compilation environment
9.3	Occam2 compiler
9.4	Editor, user interface and general utilities.
9.5	DOS level utilities and other DOS problem
9.6	Programming interface
9.7	Other problems
10	Technical Notes Available from Inmos and distributors
11	OCCAM User Group
12	INMOS Technical Support Locations
	12.1 United States of America
	12.2 Europe
	12.2.1 UK
	12.2.2 France
	12.2.3 Germany
	12.3 Japan

The T800-20 is pin compatible with the T414B-20, because the speed select pins, being held to ground, select the 20 MHz speed on the T800. A -17 T800 will not run, and a faster than -20 option will actually run at 20MHz.

The T800 should not be plugged into a T414-15/17 socket without suitable changes to the memory configuration, as the faster cycle will probably not match the memory speed.

Exchanging a T800 for a T414A requires the same additional changes as an A-B upgrade, ie PLL capacitor polarity.

Note that the T800 will not run all T414 object code... it is source compatible.

Note the different treatment of the Halt on Error Flag... see next entry in CTU.

Important future info

Note that while the T800m Preview only shows two **ProcSpeedSelect** pins, in the Rev B T800, there will be three, the third being pin D2 on the PGA. Engineers should be advised to bring this out to a switch or jumper together with the other two.

Note that the Halt on Error flag is cleared every time the T414 switches from low to high priority. It regains its old value when the low priority process is resumed.

This means that if the T414 commits an error in high priority, it WILL NOT HALT. This will cause problems with [LOCATE ERROR] on the TDS until the compiler can be adapted.

The T414 datasheet describes accurately the operation of the flags, while the manual does not give any details.

In the T800, the flag retains its value in the same way as the error flag.

4.1 Hardware problems

4.1.1 External memory interface configuration of IMS T800

An error in the shift path means that internal preset configurations between MemAD9 to MemAD31 cannot be set with any reliability. External configurations are not affected, and so the B001-3 is not affected.

4.1.2 IMS T800 Link engine

A hard output channel cannot be used in parallel with any other hard channel, (input or output channel) at a given level of priority. Input channels can be used in parallel with other input channels, and one output channel can be used in parallel with other input channels if the output channel is at a different level of priority from all the input channels.

NB An effect of this bug is that the link test program in the B001-3 ROM is likely to fail. The links have been tested individually.

4.1.3 Internal RAM

The on-chip RAM cell suffered some process experiments on this early silicon, and the effect is a slight data retention problem due to leakage. A cell may fail after 7 seconds at 70 degrees C ambient, but will be good for periods in excess of one minute at room temperature.

The on-chip RAM can be disabled on the B001-3 by setting coding switch 8 to OFF.

4.2 Software problems

4.2.1 Installation on IBM PC

The installation instructions on page 2 of the OCCAM COMPILER RELEASE for PC/XT or AT asks you in paragraph 3 to delete the old versions of the compiler and configurer. Before you do so you need to type:

```
cd c:\tds2dir
attrib -r *.*
```

and then the files can be deleted as instructed in paragraph 3.

4.2.2 INT to REAL

INT to REAL conversions must not be used in expressions, only on their own:

ie;

```
temp := REAL32 ROUND a
b := temp + b
```

NOT;

```
b := (REAL32 ROUND a) + b
```

4.2.3 Arithmetic features

$-0 * -0$ in Rounding Modes RP & RM does not give the correct zero.

$INF + NAN$ does not always return a NAN , in some cases an INF is returned.

$0 * d$ does not always produce zero (where d is a denormalized number)

$d * d$ does not always produce zero (or the minimum denorm in rounding modes RM or RP).

4.2 Software problems

4.2.4 Configurer

The configurer may not deal with programs with small workspace requirements correctly.

The floating point error flag and rounding mode is not initialised by the configurer.

In addition to correcting bugs, please note that the Product Preview pinout describes Rev A. Pin D2 of the PGA is marked as Hold to Ground, and becomes **ProcSpeedSelect2** on the Rev B.

6.1 IMS B001

6.1.1 Series damping resistors

When the IMS B001 was designed, the designers were very concerned about the possibility of latch-up caused by input signals to the transputer undershooting below GND. To prevent signal transitions from undershooting, a number of 100 ohm resistor packs were used. These have proved to be unnecessary and have not been used on any of the subsequent evaluation boards. They are even less necessary with the IMS T800, whose latch-up characteristics have been substantially improved over the IMS T414.

6.2 IMS B001, B002 and B004: link buffering

The buffering used on the links of the B001, B002 and B004 introduced skew which may exceed the link skew specification at 20 Mbps. The buffering was inserted partly to protect the transputer from damage from static (ESD); it has since been found that protection diodes give adequate protection. Even the diodes are probably not necessary with the IMS T800 as the ESD performance of the IMS T800 has been substantially improved over the IMS T414.

6.3 IMS B002

6.3.1 Coding switch

There is an error on page 14 of the User manual; the board boots from link when switch 6 of the coding switch near to the edge connector is set to OFF. Full details of the coding switch settings are as follows:

Switch	Signal	OFF	ON	Standard
1	Link0Special		Standard	ON
2	Link123Special		Standard	ON
3	DisablePLL			OFF
4				ON
5				ON
6	Boot source	Link	ROM	ON
7				ON
8	Parity	See below		ON

Switch 3 must be OFF and switches 5 and 7 must be ON.

When the Parity switch is ON, a parity error is reported on the LEDs, but does not generate an error on UpError; when the switch is OFF a parity error is propagated to UpError.

6.3.2 RS232 level changer

Some of the RS232 level changers suffer from crosstalk between the two channels, and between input RS232 and output RS232. This was caused by a design change by the supplier, who has provided replacement level changers for faulty ones.

6.3.3 Grounding power supply

We have found it necessary, particularly with the B002, to connect 0V from the power supply to a Ground or safety earth. It is good practice to make such a connection at a single point in a system.

6.3.4 Subsystem

The B002 can be used at the top of a hierarchy to control a subsystem of other transputers through its subsystem socket. The B002 subsystem logic can not be used, however, if the B002 is in the middle of a hierarchy. The way this subsystem is addressed, moreover, is different from all the other evaluation boards.

6.4 IMS B003

6.4 IMS B003

6.4.1 Undershoot on RAMs

With the small number of RAMs driven by the address multiplexors, and the short length of circuit traces (about 10 cm) the designers of the IMS B003 felt it would not be necessary to series terminate the multiplexed address signals to the RAMs. In practice we found that there was up to 2.5 V of undershoot. This was not observed to cause malfunction of the RAM but is clearly outside the RAM's specification. Experiments were carried out with various TTL families; the 74F241 and 74AC241 were found to give excessive undershoot, 74HCT241s were too slow. The optimum combination of undershoot and delay was given by the 74ALS241, which was better in both respects than the 74LS241; the ALS241 had an undershoot of approximately 0.5 V.

All B003s built now use ALS241s. On all designs since the B003, we have used series resistors.

6.4.2 Bulk decoupling

The earliest B003 boards had several microfarads of ceramic decoupling, but we found it necessary to add about 40 microfarads of bulk decoupling.

6.4.3 Schmitts on reset

The Up/Down logic for Reset, Analyse and Error has been changed to HCT to give full 5 V signals and schmitt triggers to improve the noise margins.

6.5 IMS B004

The IMS B004 is frequently used to run the transputer development system, for which the RevB version of the IMS T414 is both faster and more appropriate. An upgrade kit is available to upgrade a board from the RevA version of the T414 to revB. This kit, the IMS B901 costs approximately the same as a transputer, and includes the transputer, a PGA extractor tool and instructions on how to change the transputer.

Also included in the B901 are optional modifications which overcome:

- 1 A problem which manifests itself as an occasional inability to download programs into the B004
- 2 Incompatibility with the original version of the IBM PC
- 3 Incompatibility with a low cost clone

The B004 has now been used on a wide variety of IBM PC's and clones, so for most users, most of the time, the modifications are not necessary.

6.5.1 Occasional inability to download programs

When the B004 itself is reset, the reset signal is not passed on to the target network controlled by the B004's Subsystem pins. The result of this is that:

A board connected to the Subsystem pins is not reset when the TDS is loaded onto the B004

The target system can continue sending data through links to the transputer on the B004 while the B004 itself is being reset. It occasionally happens therefore that the B004 'boots' itself from the target system instead, and consequently cannot boot from the IBM PC. A message will be output **Can't download transputer system**, which is correct, but puzzling.

Normally this second problem can be overcome by trying to boot the B004 again, but sometimes it needs several attempts and one is inclined to switch the whole system off and start again. Alternatively one can use the reset switches on the target network boards to reset the system, after which the B004 should boot satisfactorily.

6.6 IMS B006

A modification to the B004 to fix this problem is included in the B901.

6.5.2 The original PC

The original twin floppy 8088 IBM PC does not decode its I/O space in the same way as the later versions of the PC and the B004 uses an address (Hex 150) which is invalid on the original PC.

The PAL which is supplied with the upgrade kit offers an alternative address, which is the address allocated to the prototyping board (Hex 300).

6.5.3 Power supply decoupling

One of the more recent low cost clones went into power supply current limit when the B004 started to run programs. The problem was easily overcome by adding a large electrolytic capacitor to the B004.

6.6 IMS B006

6.6.1 RS 232 Terminal driver Software

The terminal driver supplied with the 1986 beta release of the D701 does not directly support the IMS B006. Similarly, the terminal drivers supplied with the D600 (VAX) OCCAM 1 software.

On the D701, the terminal drivers are supplied in directory \TDS2DIR \EXAMPLE . The major change is to the memory location of the DUART. The constant **periph.base** should be #7000. The address decoding on the circuit board has been arranged so that the ports are 4 bytes apart, despite using a 16 bit transputer. Thus **bpw** remains at 4. In order to prevent confusion, remove the definition of **bpw** and replace all occurrences with 4. Recompile with the target processor set to **T212** and the terminal driver is ready for use.

6.7 IMS B007

6.7.1 Use with IMS B004

When using the B007 with the B004:

the B007 software must be compiled with the occam 2 compiler.

the connections between the B004 and B007 are as follows:

B004 Link 1 to B007 link 1
B004 Subsystem socket to B007 Up socket

6.7.2 Test software changes between T414A and T414B transputers

Due to the timer 'tick' rate being corrected on the Rev B transputer, the B007 test software needs to be changed in one of the tests. In the 'typed occam' fold of the B007 test disk there is a fold marked 'EXE graphics test monitor'; inside this EXE there is a fold marked 'delay' which uses the occam **?AFTER** construct.

The statement **tick ? AFTER (now + 4400000)**

should be changed to **tick ? AFTER (now PLUS 44000)**

and the EXE recompiled using the occam 2 compiler.

6.7.3 Windows demonstration program

One of the demonstrations supplied with the B007 board has an error. The windows are supposed to move, but when the demonstration is compiled on the IBM PC, the windows do not move. The cause of the problem is uninitialized variables, and to fix it, find the fold:

EXE graphics demo monitor in this you will eventually find

SC B007.demo in this you will eventually find

... MAIN in this you will eventually find

... set windows in this you will eventually find

... initialise window placements

and into this add:-

```
change.cx := 2  
change.cy := 2  
change.ox := 2  
change.oy := 2
```

6.8 ITEM

6.8.1 Hardwired link connections

Within the ITEM, users may find it useful to wire-wrap connections between boards. If this is done, care must be taken to ensure a reasonably constant 100 ohm impedance connection for the links. This is most easily done by wire-wrapping with twisted pairs.

The following sections outline some hardware possibilities. Note that to use the T800 effectively, a compiler that supports it is required. (IMS D701 BETATWO , March 87 or later).

7.1 IMS B004 Boards

The board included in the D701 Development System was originally shipped with Revision A 12 MHz T414 transputers. Customers wishing to upgrade to a 15 MHz Rev B part can buy an upgrade kit that includes the T414 transputer, a PAL, a Capacitor, an Extraction tool and full instructions.

The part number is IMS D901.

7.2 Other Boards

IMS B003-2, and B003-3 boards can be upgraded from T414B to T800 by simple buying T800-20 parts and plugging them in.

7.3 ITEM 4000

From time to time INMOS offer special prices on ITEMS equipped with T414 transputers, to be upgraded to T800s later. For details, contact Microcomputer Marketing at INMOS.

Inmos Evaluation Boards will be repaired after the warranty has expired for a fixed charge of 500 pounds. The boards will be replaced by the original vendor (ie distributor usually), for a 500 pound charge. Inmos will ship a replacement board to the distributor on receipt of the faulty board, invoicing him appropriately.

This section is a checklist of known problems in this beta1 release software, dated May/June/July 1986. Most of the problems are bugs that will be cured in the later releases. Some are design errors or documentation inadequacies that will be considered for improvement in due course.

It is important for all readers who come across problems which are not in this list to report them to Michael Poole, Software support, INMOS, Bristol, preferably on a Beta release software problem report form.

9.1 Installation

Versions of DOS earlier than 3.0 do not permit a path name in a command call. If such a DOS is in use the call of `server9` in `tds2.bat` should be amended to:

```
server9 -b c:\tds2dir\boot.b4 %1 %2
```

this will locate the server by virtue of the current search path including `\tds2dir`.

9.2 Occam2 language

9.2.1 Configuration language

The language used at configuration level in an occam 2 transputer network program is not the full occam 2 language. The restrictions are of two kinds documented in different places.

The SC containing code for one target processor must contain a single `PROC` whose parameters may only be `VAL INT`, or `CHAN` (no arrays, no bytes, ...). `CHANS` may not have protocols. This is documented in the User Manual Section 7.1. Violations of these restrictions currently lead to a TDS crash (**Transputer system error flag set**)

The configuration code calling these SCs must conform to the restrictions documented in the Occam implementation manual section 1.3. This language is currently processed by a variant of the occam 1 compiler front end and so the error messages produced need to be interpreted in an occam 1 sense. Note that the expressions used in this language cannot contain type conversion operators or any other operators that have no occam 1 counterpart.

Configuration language does not permit a channel to be declared of which only one end is `PLACED`. Such channels (e.g those to link adaptors) must be declared and placed within the SC code for the relevant processor.

The channel parameters of an `EXE` or `UTIL` must be declared without protocols. If protocols are declared on these channel then the compilation will fail at the link stage.

9.2.2 Implemented features not documented

The implementation supports specifications in front of `IF` and `ALT` guards. i.e. the definition of *scope* should be extended to include *choice* and *alternative*. The implementation, however, has a bug which causes rejection of such specifications in front of `ALTs`.

The syntactic abbreviations using the semicolon to represent sequences of inputs and outputs have been removed from the definition of the language but not from the implementation. They will be removed from a future implementation.

A special string escape character `*L` which may be used to represent the length of a string in occam 1 style, is implemented but not documented. This facility is provided merely to allow automatically converted programs to run without modification, and may not be a permanent feature of the language.

There is no discussion of lexical matters such as comments, continuation lines, structure of names, etc. To a first approximation these aspects are the same as in proto-occam.

9.3 Occam2 compiler

The document fails to state that **MINUS** can be used as a monadic operator.

9.2.3 Documented features not implemented

These shortcomings are listed in section 1.5 of the Occam implementation manual.

In addition:

The implementation of code generation for T2 (16-bit) transputers does not support extended arithmetic types (including explicit **INT16**).

The T2 compiler contains an erroneous implementation of the undocumented predefined procedures **UNPACKSN** and **ROUNDSN**.

9.2.4 Bugs in compilation environment

Channel and variable usage checking does not work reliably. Many programs which should be accepted are rejected with an error numbered in the range 1400 - 1599.

The run-time locate utility does not work on the B004 running the TDS.

The configuration utilities do not pulse reset or analyse on the subsystem connector. It may be necessary to reset a transputer network by some other means. A suggested method (in the absence of a hardware reset button) is to write a program using the procedures supplied on the B004 test disk.

The loader will deadlock if no correctly reset transputer is connected to the link being used for loading. It ought to time out.

CHECK on a **.TOP** fold crashes the system

COMPILE fails on **PROGRAM** foldset containing **CONFIG INFO**.

The boot path fold in the **CONFIG INFO** fold uses physical processor numbers (counting the processors as they are encountered) rather than logical ones (as introduced by the programmer in the **PROCESSOR** construct) as used in the accompanying folds.

The **CONFIG INFO** utility does not delete a pre-existing **CONFIG INFO** fold in a foldset to which it is applied.

The **CONFIG INFO** utility's error messages are wrongly prefixed **Extraction error:**

EXTRACT on a voidset gives the misleading message **Cannot create code fold**.

LOAD NETWORK does not flush the input link channel before loading a transputer network program. If the program starts up by sending data up this link then such data may be preceded by rubbish, with unpredictable consequences. It is recommended that communication with a running network program across such a link always starts with a suitable exchange of messages to ensure both ends are ready.

The loader can load incorrect code, which will result either in code not being run or a run-time crash, in the following circumstances:

A separately compiled unit to be loaded is so small (code + workspace) that it overlaps the loader. The configuration requires this one **SC** to be loaded onto more than one processor at different positions, e.g. because of different numbers of soft channels being declared.

9.3 Occam2 compiler

The operator **MINUS** should only be allowed between operands of integer type. The compiler allows it between other types with unpredictable consequences.

9.3 Occam2 compiler

Usage checking in many cases will reject as invalid many programs which are valid. If problems of this kind are encountered then usage checking should be left switched off.

Alias checking is not implemented. This is inadequately discussed in the Occam implementation manual. The compiler can thus not be relied on to check against some dangerous uses of the abbreviation and retyping features of the language.

The checking of **RETYPE** specifications is inadequate. Some invalid specifications will not be rejected - with undefined consequences. It is also possible that some valid uses will be rejected.

The compiler does not generate code to set transputer error when a **STOP** process is executed.

Arrays whose size when rounded up to a number of words is a number of bytes greater than **MAXINT** cannot be declared (even for **PLACEMENT**).

If arrays on a transputer are **PLACED** after other variables in the same scope have been declared, the compiler will generate code which may mishandle workspace and cause unpredictable run-time corruption.

The code generated from a **PRI PAR** construct does not force the transputer to return to low priority after both branches have completed.

The checker or compiler can deadlock if it fails to open a file.

If a parallel construct contains only declarations of separately compiled procedures, and no main process, the compiler will terminate with **Compiler error: 2216**.

There is an implementation limit of 10 dimensions in an array declaration. Violations of this limit will not give meaningful error messages.

If a separately compiled procedure is declared other than in front of a constructor (**SEQ, PAR**, etc) the compiler will stop on the process after its scope with an indentation error.

There is a bug in ordering comparisons (>, etc) between **REAL64** values. The result is determined by the relative values of the least significant words only, rather than by the full 64-bit value.

If a formal parameter is a channel and the corresponding actual parameter is a value the misleading error message "expected assignable parameter" is generated.

The compiler can give spurious errors claiming incompatible array sizes when constructing 3-dimensional arrays using abbreviations of 2-dimensional arrays, or arrays with different sizes in all 3 dimensions.

The compiler wrongly accepts **VAL** abbreviations of channels.

If the last line of a **PROC** body is a declaration, a **COMPILER ERROR** is signalled.

If both operands of **AFTER** are constant the compiler evaluates the expression using - and so can generate overflow and stop the TDS.

If the same **INT64** variable is used as an operand and as the result in an assignment such as **x := x * 10 (INT64)** a rubbish result will be computed if any component exceeds 32 bits.

Bitwise operations on **INT16** quantities on the T4 can set error as the truncation of the answer to **INT16** can wrongly appear to be a range violation. These problems will be avoided if explicit conversions to and from **INT** type are always coded in such expressions.

The value **MOSTPOS INT64** is wrongly computed by the compiler. Use **BITNOT (MOSTNEG INT64)** instead.

The checker accepts **channel ! procname**, and this construction crashes the compiler.

9.4 Editor, user interface and general utilities.

If a PAR contains a specification, but no processes, the compiler terminates with error 2216.

The code **chan ! z :: b**, where z is defined by an abbreviation **z IS a[i] :**, checks and compiles but crashes at run time. **chan ! a[i] :: b** is rejected by the compiler.

An array segment such as **[a FROM 2 FOR 3]** where the base is a constant will be incorrectly compiled as if the base was zero when used on the right hand side of an array assignment.

9.4 Editor, user interface and general utilities.

The HELP key is inadequately documented. Any key will cause return to the editor's window. The system version number and date at the foot of this display are the definitive identity of a particular software release.

The searcher can enter non-text folds, which can then be "edited"!

If the searcher matches text off the screen in a long line, then the cursor must be moved manually off that line before starting a subsequent search.

The searcher can stop for lack of buffer space when moving from one filed fold to a succeeding one. A subsequent press of the SEARCH key will usually cause the search to resume successfully.

If the searcher fails to open a fold, it displays the "cannot open" message and then immediately overwrites it with "... not found". The user will get the wrong impression in such cases.

If a run time error occurs in the code of the TDS (e.g user filer) while a user program is running, and the user attempts to locate the source of the error with LOCATE ERROR it will locate to the first code line of the last SC in the program.

LOCATE ERROR can deadlock if the fold manager runs out of space while it is searching for the SC responsible for the error.

If a run time error results from a programmed stop in a library routine (eg invalid parameter to a mathematical function), the error message will give an offset within the library, but there is no way this number can be related to a particular library routine.

The move buffer file **TOPLEVEL.MOV** survives the end of a session and can lead to difficulties later.

9.5 DOS level utilities and other DOS problem

If the name used on folds to be filed includes a dot then the following characters must be an approved lower case filename extension. If not, the consequences are unpredictable.

The TDSCOPY utility should be used with extreme caution as it does not check for pre-existing files with the same name as new ones being copied.

The TDSLIST utility can fail because of shortage of fold-manager space.

The TDSCON utility cannot be used if the DOS text file being converted includes TAB characters.

Some versions of DOS on some hardware are such that the way screen handling has been done at the bottom of the screen leads to rubbish being displayed and the cursor going to an unexpected position. The problem only occurs in the bottom 2 lines of the screen with some editing operations which manipulate whole lines. The use of the REFRESH key will recover the correct view. The problem is not present on the IBM PC-AT using DOS 3.0.

The PC screen is not accessed through ANSI.SYS and so the various escape sequences to drive colour displays, etc are not available.

9.6 Programming interface

Server9 hangs (and the PC needs to be rebooted) if transputer error flag is set while the server has a file stream open for writing.

9.6 Programming interface

The terminal handling interface package **term.p** has some minor shortcomings. The most serious of these are that keyboard input wrongly translates ASCII codes DEL and TAB to internal codes and that cursor tracking on the PC screen does not handle ***c** and ***n** properly.

The facilities for reading and writing DOS format text files are not yet fully implemented. The commands **uf.open.text.read** and **uf.open.text.write** should not be used.

The way keyboard mode is changed from raw to cooked and vice-versa will be different in the product (to allow characters already typed ahead in the wrong mode to be flushed).

If the **uf.read.fold.attr** command is used when the current item is not a fold the user filer will STOP, crashing the TDS.

If a file is read in data stream input mode, records in folds whose type is not text should not be seen. In the present implementation such records are seen and will have leading indentation space characters put on to them if the fold is indented relative to the base fold. This can lead to loss of data at the end of records which are at or near the maximum size of 512 bytes.

The command **fsc.read.file.id** should return **fsd.error** if the fold is not filed. In fact it returns **fsd.fileid** with name length zero.

If a user filer operation is called when the current item is not a fold the error message (-1205) wrongly says "Fold already in use ...".

9.7 Other problems

The instructions in the Getting Started section of the manual concerning the copying of the on-line tutorial should read:

```
C> cd \tds2dir
C> tdscopy tdstutor.top \temp
```

The on-line tutorial contains some minor errors which derive from its history in other versions of the software. In due course it will be re-extended to include introduction to utilities, compilation, etc, etc.

When a program which has converted the keyboard from cooked mode into raw mode terminates, the system is unusable, as a necessary internally generated switch back to cooked mode does not occur. If the program itself converts the keyboard back to cooked mode before terminating, there is no problem.

The occam 1 to occam 2 converter program (GO12) does not handle configuration level code such as **PROCESSOR** and **PLACE**.

GO12 will produce fold comment lines with rubbish in them if a filed fold is converted whose fold comment does not contain a '.' or a space. Such lines may cause screen corruption when displayed, but are otherwise innocuous.

In some conditions GO12 can crash during the input of its option letters if the ? is used to attempt to abort the run.

The B004 board may have upto 2 megabytes of RAM. The current version of the TDS does not use the second megabyte. However programs using the whole board may be created using the configuration system.

Early B004 boards have Rev A T414s on them. All the problems of this pre-qual silicon are potentially relevant to users of this software. Users must consult the errata sheet provided with the B004 board. The predefined

PROC FRACMUL is not available on the T414A.

The mathematical library contains errors (concerning the use of FRACMUL) which can lead to transputer error being set in some situations where it should not. A separate note discusses ways these problems can be avoided.

The procedure **REAL64.write** can crash.

The procedure **REAL32.write** writes the number in the wrong format.

Document title	Available
Transputer Ref Man (Oct 86)	Available
IMS T800 Product Preview	Available
IMS T414 Engineering Datasheet	Available
IMS C004 Datasheet	April 87
IMS T800 Product data	March 87
IMS M212 Product data	March 87
Occam 2 Tutorial DP+Occam 2 DEF	Available
Occam 2 Ref Man	April 87
Transputer Instruction Set and Compiler Writers Guide	Available

Technical Notes :

0	Transputer based navigation system	Available
1	Extraordinary use of links	Available
2	Testing embedded systems	Available
3	Getting started with the TDS	Available
4	TDS EPROM programming	Available
5	Program design for concurrent systems	Available
6	IMS T800 Architecture	Available
7	Exploiting concurrency; a ray tracing example	Available
8	IMS B010 NEC add-in board	March 87
9	Designing with the T414 and T800 memory interface	March 87
10	IMS B003 design of multi-transputer board	Available
11	IMS B004 IBM PC add-in board	Available
12	IMS B007 A transputer based graphics board	Available
13	Transputer networks using the IMS B003	Available
14	IMS B006 A single board computer using T212	Available
15	IMS B005 Design of a disk controller board	Available
16	Developing occam programs for the transputer	Available
17	Maximising performance	April 87
18	Connecting Inmos Links	April 87
19	Dual In-line Transputer Modules	April 87
20	Communicating processes and occam	Available
21	The transputer implementation of occam	Available
22	Communicating process computers	Available
23	Compiling occam into silicon	Available
24	Exploring multiple transputer arrays	Available
25	Occam 2: aspects of the language and its development	
26	T800 Graphics instructions	April 87

Also under preparation :

The Philosophy of occam
Guide to configuration

The following Oxford PRG monographs are also available :

The laws of occam programming
Exploiting parallelism in the graphics pipeline
The pursuit of deadlock freedom
Formal methods applied to a Floating Point number system

An OCCAM User Group exists in the UK, Germany and in the USA, and there is interest in forming one in France.

The UK secretary is

Dr Michael Poole
Software Support
Inmos Ltd
1000 Aztec West
Almondsbury
Bristol BS12 4SQ 0454 616616

The German contact is

Joachim Stender
Brainware GmbH
Kirchgasse 24
6200
Wiesbaden
West Germany 06121/372011

The next meeting of the UK based group is in Grenoble, France in September 1987, in the English language. Speakers are invited to contact:

Dr Traian Muntean
IMAG
Lab de Genie Informatique
38402 St Martin d'Herès
Grenoble
France Tel 76 51 46 00 ext 5217

For details of the USA user group, please contact:

Brad Hartman
Microcomputer Marketing
Inmos Corp
1110 Bayfield Drive
Colorado Springs
Colorado 80935

12.1 United States of America

Suite 206,
9861 Broken Land Parkway
COLUMBIA,
MD 21046, USA
TEL: + 1-301-995-6952
TLX: USA TWX 710-862-2872

8300 Norman Centre DR
MINNEAPOLIS,
MN 55437, USA
TEL: + 1-612-831-5626
TLX: USA WUD 509-995

2 Westborough Business Park (BOSTON)
Suite 3001,
WESTBOROUGH,
MA 01581, USA
TEL: + 1-617-366-4020
TLX: USA TWX 510-601-6099
FAX: + 1-617-366-5265

2620 Augustine Drive
Suite 180
SANTA CLARA,
CA 95054, USA
TEL: + 1-408-727-7771
TLX: USA TWX 910-338-2151

23505 S Crenshaw Blvd
Suite 201,
TORRANCE,
CA 90505, USA
TEL: + 1-213-530-7764
TLX: USA TWX 910-347-7334

14 643 Dallas Parkway,
Suite 730
LB9 DALLAS,
Texas 75240, USA
TEL: + 1-214-4909522
TLX: 910 861 0034

12.2 Europe

12.2 Europe

12.2.1 UK

INMOS Ltd
1000, Aztec West
Almondsbury,
BRISTOL
BS12 4SQ

TEL: 0454 616616
TLX: 444723
FAX: 0454 617910

12.2.2 France

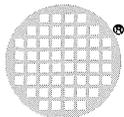
INMOS SARL
Immeuble Monaco,
7 Rue le Corbusier,
Silic 219,
94518 Rungis Cedex,
France
TEL: + 33-14-687-2201

12.2.3 Germany

INMOS GmbH
Danziger Str 2,
8057 Eching
Munich,
W Germany
TEL: + 49 89-319-1028

12.3 Japan

INMOS International (JAPAN)
Room 308
Kowa No 16 Annex,
9-20 Aksamasa 1-chome
Minato-ku,
Tokyo 107,
Japan
TEL: + 81 3-505-2840
FAX: + 81 3-505-2844



INMOS Limited
1000 Aztec West
Almondsbury
Bristol BS12 4SQ
UK
Telephone (0454) 616616
Telex 444723

INMOS Corporation
PO Box 16000
Colorado Springs
CO 80935
USA
Telephone (303) 630 4000
TWX 910 920 4904

INMOS GmbH
Danziger Strasse 2
8057 Eching
Munich
West Germany
Telephone (089) 319 10 28
Telex 522645

INMOS SARL
Immeuble Monaco
7 rue Le Corbusier
SILIC 219
94518 Rungis Cedex
France
Telephone (1) 46.87.22.01
Telex 201222

INMOS International
Room 308 Kowa No. 16 Annex
9-20 Akasaka 1-chome
Minato-ku
Tokyo 107
Japan
Telephone 03-505-2840
Telex J29507 TEI JPN

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

 inmos, IMS, and occam are trade marks of the INMOS Group of Companies.

March 1987

72 TRN 12 400