Parallel Fortran Version 2.1.3 Release Note

3L Ltd.

December 5, 1990

1 Introduction

This Release Note accompanies version 2.1.3 of 3L Parallel Fortran for the Inmos transputer, and outlines the changes in Parallel Fortran since version 2.1.0. The most important changes are:

- Full support for the interactive source-level debugger, Tbug.
- New `DOUBLE COMPLEX` and `BYTE` data types.
- Static variables are now initialised by default to zero.
- The master task of a processor farm may "broadcast" a message to all the worker tasks.
- The linker has been completely rewritten.
- Application files output by the configurer will now contain only one copy of each task.
- The configurer supports sub-networks within the larger main network.
- Standard Fortran I/O may be done from any task by using the multiplexer task.
- New components include a `worm` program.

- The **afserver** program has been amended so that Fortran I/O is done is a record-oriented way, to help when porting the software to non-IBM PC hosts.

The Release Note should be read in conjunction with the *User Guide* distributed with V2.1.3. References in the text in square brackets "[...]" are to chapters and sections in this Guide.

2 Installation Procedure

[Chapter 1] This version of Parallel Fortran is supported by a new interactive installation program. This ensures that installation will be possible for all versions of DOS.

3 The Compiler

3.1 Language Enhancements

Long source lines [8.3.1] If a program is compiled with the new **/R** switch, the maximum line length is increased to 132 characters.

Exclamation mark designates a comment [8.3.1.3] A '**!**' either in column 1 or in column 7 or beyond indicates that the rest of the line is to be treated as a comment.

Debug comments [8.3.1.3] A '**D**' in column 1 designates a debug comment. Such a line is treated as a comment except when the compiler switch '**/D**' is used, in which case the line is compiled as usual.

Escape sequences in character constants [9.2.1.11] Character constants may now include certain sequences starting with '\' in order to include non-printable characters in the constant.

Subscripts may be real [9.2.4] An array subscript may be a real expression.

DOUBLE COMPLEX data type [9.3.5] The DOUBLE COMPLEX data type includes new specific intrinsic functions and extensions to some generic intrinsics [Appendix E]. Mixed expressions including DOUBLE PRECISION and COMPLEX elements are now permitted [11.1.6].

BYTE data type [9.3.5] A BYTE variable can take integer values in the range -128 to $+127$.

VIRTUAL statement [10.2.2] This statement, supplied for compatibility with other compilers, is equivalent to the DIMENSION statement.

Initialisation of numeric variables [10.3.1.5] This may now be done using character constants.

Output of NaN and infinity [15.3.1.2, 15.3.1.3] The special IEEE values NaN and $\pm\infty$ are output as a series of '?' characters.

/Q edit descriptor [15.3.1.16] This edit descriptor may be used in a format to discover the number of remaining characters in the current record.

Default field widths [15.3.3] This facility enables an edit descriptor to be used in a format without specifying a field width. The field width assumed depends on the type of the element being transferred.

ACCEPT and TYPE statement [16.3.1.1] These statements perform input and output on the standard input and output units.

Multiple backspacing allowed [16.3.2.3] Previous versions of Parallel Fortran restricted the number of times that the BACKSPACE command could be applied to a sequential file. Now it is permitted to BACKSPACE a file right to the beginning.

Namelist-directed I/O [16.6] This new facility enables groups of variables to be output and input, together with their names, with a single simple I/O statement.

Preconnections and default filenames [16.8.1.2] Units other than unit 5 and unit 6 are now preconnected to files with names of the form FORT*nnn*.DAT, where *nnn* is the unit number. This means

that an **OPEN** statement which does not include a **FILE=** specifier will open a file with this form of name.

New I/O specifiers [16.8.2] The **OPEN** statement may now include the **NAME**, **RECORDSIZE** and **READONLY** specifiers.

ENCODE and DECODE [Appendix D.1] These statements provide an alternative method for performing internal I/O.

DEFINE FILE statement [Appendix D.2] This statement is an alternative method of opening unformatted direct files.

Alternative method of record selection [Appendix D.3] Records in direct files may now be selected using a '' character followed by an integer expression, as well as by using the standard **REC=** specifier.

FIND statement [Appendix D.4] **FIND** is an alternative method of positioning unformatted direct files.

New trigonometric intrinsic functions [Appendix E] These new trigonometric functions accept arguments expressed in degrees.

3.2 Initialisation of variables

[3.5] Previously the values of any Fortran variables not explicitly initialised were undefined when a program started. Static variables are now initialised to zero by default. This includes all arrays, variables in **COMMON** blocks and any variable which appears in a **SAVE** statement. Other variables are held on the stack, and these are not initialised to zero; however, if this is important, the compiler can be made to hold all variables in static by using the **/S** compiler switch. This facility will only work if the linker distributed with this version of Fortran is used.

3.3 Compiler Switches

The following switches are new.

`/D` [8.3.1.3] Causes debug comments to be compiled as ordinary statements.

`/PMn` [17.2.4.5] Controls the size of the module number gap left by the compiler in the start-up code to subprograms.

`/R` [8.3.1] Allows the compiler to accept 132-character lines.

`/U` [9.3.1] Causes variables to flagged as errors if their types are not defined by an `IMPLICIT` statement or an explicit type statement.

`/Zi` [17.2.5.1] Causes information about variables to be included in the binary file for the use of Tbug.

`/Zd` [17.2.5.2] This switch currently has no effect, but in later releases will be used to control the output of information about the source program line numbers for the use of Tbug.

3.4 Support for Tbug

[17.2.5] The compiler now fully supports Tbug, 3L's interactive source-level debugger. Two new switches, `/Zi` and `/Zd`, control the output of debugging information. Notice that the version of the linker included in this release must be used.

3.5 Running the compiler

The environmental variable `TMP` [17.1] This can now be used to specify the directory in which the compiler should create its temporary files. Previously, they were always put in the current directory.

MS-DOS system return code [17.1] This is now set to 1 if errors are detected during the compilation. This allows a batch file to check if a compilation has succeeded using `if errorlevel 1`.

Environmental variable TF [17.2.1] If you wish to use the same switches for every compilation, you can enter the switches through the environmental variable **TF**.

Output files [17.2.3] The behaviour of the compiler has been changed in cases where the **/Fb**, **/Fo**, **Fl** or **/Fh** option switches are used, and the specified output file name does not contain an explicit path name, and the source file is not in the current directory. For example:

```
C>t8f ..\pqr /Foxyz
```

Previously the output file **xyz.bin** was created in the same directory as the source file (in this case "**..**"); now the output file will always be created in the current default directory if a path name is omitted from a **/F** option switch. The behaviour when a path name is supplied, e.g., **/Fb\dir\file**, is unchanged.

4 Run-Time Library

The following new subprograms have been added to the run-time library.

F77_NET_BROADCAST [18.2.7] The new member of the **NET** package can be used to send a broadcast message to every worker task in a processor farm.

F77_SEMA_TEST_WAIT [18.2.4] This new member of the **SEMA** package enables the program to test a semaphore without the thread being paused.

Accessing the host's I/O ports [18.2.2] Two new members of the **DOS** package enable this to be done.

F77_GET_COMMAND [18.2.10.4] Enables a program to read the command parameters.

F77_DO_COMMAND [18.2.10.4] Executes a host command.

EXIT [18.2.10.4] Terminate the program and set the MS-DOS result code.

Testing for NaN and infinity [18.2.10.2] Four functions are provided to do this.

Low-level memory access [18.2.10.3] Five functions are provided to do this.

5 The Linker

[Chapter 19] The linker has been entirely rewritten. Amongst the new facilities are:

- Support for the interactive source-level debugger, Tbug.
- Modules can be selected by reference to external names they contain for placing at the beginning of the image, so that they may be placed on on-chip RAM, if available.
- Library files may contain **COMMON** blocks.
- The files in the link list may contain more than one definition of a single external name. In this case, the linker selects the first occurring. In this way, files in a library can be overridden by routines with the same name appearing earlier in the link list.
- The command line has been simplified by permitting space as a connecting character in addition to '+' and assuming filename extensions for the various types of files accessed.

[3.3.1] The batch files which call the linker have also been enhanced to allow more than one object file to be specified. You may also specify linker switches.

6 The General Configurer

[Chapter 26] The General Configurer, `config`, has been improved in several ways.

- If a user declares multiple tasks all of which have the same image file, the contents of that file will appear only once in the application file.
- The number of copies of the loading software included in the application file has been much reduced.
- If the user places more than one identical task on the same processor, they will now share one copy of the code for the task.
- The PROCESSOR statement has a new BOOT attribute. This enables the user to specify sub-networks within a larger main network.
- The PROCESSOR statement also has a new RAM attribute, which forces the loader to assume the specified size.
- Filenames specified in configuration files are now treated as case-significant.

7 Global I/O

[Chapter 6] By using the multiplexer task `filemux` included with Parallel Fortran, standard Fortran I/O can be done from any task in an application.

8 New Components

Apart from the components mentioned above, the following are new:

The `worm` program [Chapter 22] is a utility for exploring transputer networks.

The `stub` task [Chapter 28] can be configured with a user task to enable it to be linked with the full run-time library even though it does not need to perform standard I/O.

The `tnm` and `tunlib` utilities [Chapters 23 and 24] used to be unsupported utilities. They are now officially part of the product.