12 APRIL 1988
UPDATED 9 NOVEMBER 1988

NUMERICAL ELECTROMAGNETICS COMPUTATION USING THE INMOS T800 TRANSPUTER ON AN OLIVETTI M24 PERSONAL COMPUTER

M.ENG PROGRESS SUMMARY

J.J. LE ROUX
ANTENNA AND MICROWAVE GROUP
DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
UNIVERSITY OF STELLENBOSCH, STELLENBOSCH, 7600
REPUBLIC OF SOUTH AFRICA

STUDY LEADER : Prof. J.H. CLOETE

INTRODUCTION : As is known to most engineers and scientists interested in computational electromagnetics analysis, G.J. Burke and A.J. Poggio developed a code called NEC2 at the Lawrence Livermore Laboratory in California [1]. This code was developed using a mainframe computer and was written in FORTRAN. The code has subsequently been used on a VAX785 mainframe at this University.

Recent developments at INMOS in the field of fast 32-bit parallel processors however opened up a new avenue of arithmetic processing using their transputer concept. The first transputer available from INMOS was the T414, and later the T800, with floating point processing.

The natural language of the transputer architecture is OCCAM. However, facilities to use alien languages, like C, PASCAL and FORTRAN on the transputer have recently been made available.

The computer group in the Department of Electrical and Electronic Engineering at the University of Stellenbosch, led by Professor Jan du Plessis, has been exploiting the advantages of the transputer for a variety of high speed computing applications, including real-time control of dynamic systems, and number crunching. In the process they have developed a board which houses a number of T800-processors as well as 2MB on-board static storage. This board plugs directly into any IBM-compatible PC, like in this case, the M24 Olivetti PC with a 20MB hard disk drive. The cost of a transputer board with one T800 processor is about $US2000.

At the suggestion of Professor du Plessis, a study was initiated to investigate the feasibility of running NEC2 on the T800 transputer.

In order to test this idea, a transputer board, containing a single T800, was plugged into an Olivetti M24 PC, with the following additional hardware : 2 x 360kB floppy drives; 1 x 20MB hard drive; 1 x 8087 co-processor; and 640kB of RAM.

PROCEDURE TO TRANSFER NEC2 TO T800 TRANSPUTER : After less than 1 month's work, NEC2 was successfully run on the T800 transputer, using the following procedure :

1. Firstly the necessary software for FORTRAN use on the transputer, as supplied by INMOS, was loaded and installed on the hard drive.

2. To load NEC2 onto the PC hard drive, it was found that it would be easier to transport the PC to a mainframe serial port, and transfer the program serially from the VAX785 magnetic storage to the PC hard drive, using a suitable error-free transfer packet.

3. Having transferred the program onto the PC hard disk, the use of NEC2 on the transputer was investigated. The program was renamed NEC2.F77 to comply with INMOS terminology. The logical first step seemed to be to try compiling the program using the transputer's FORTRAN compiler. Obviously there were a lot of errors and warnings, but at least it proved that the compiler accepted the NEC2 package.

4. The nature of the warnings and errors after compilation were the following :

a. The compiler did not accept the REAL*8 statement. This was replaced with the DOUBLE PRECISION statement.

b. Some of the declared constants were too large for single precision format. All these were replaced with double precision format, i.e.

DATA TA/1.745329252E-02 replaced with

DATA TA/1.745329252D-02
^

(The E replaced by a D)

c. The compiler has an alternative protocol for READ/WRITE statements which involves the exclusion of a comma, i.e.

WRITE (6,*),'WHAT TO BE WRITTEN ON SCREEN' becomes

WRITE (6,*) 'WHAT TO BE WRITTEN ON SCREEN'
^

(Comma removed)

d. The OPEN statement does not allow the RECORDTYPE and CARRIAGECONTROL options, as found on the VAX785.

e. The compiler does not accept the use of Hollerith constants. All these were replaced with CHARACTER declarations and the H-characters in the DATA-statements were removed. Use was also made of type conversions, i.e. conversion from CHARACTER to INTEGER and vice versa, to bypass the Hollerith constraints.

f. Line 66 of subroutine COUPLE was changed to the following form :

```
FORMAT (2(1X,I4,1X,I4,1X,I5,2X),45H**ERROR** COUP.....
```

(inclusion of a comma after the first closing bracket)

g. The transputer clock procedure, ICLOCK, produces a time value (seconds) in an integer format, as opposed to the VAX785 real format. All the time printout FORMAT statements were modified to accommodate the integer format.

5. After all the above errors and warnings were rectified, it was possible to compile the program, but the compiler failed to generate the binary code, NEC2.BIN. (It was not necessary to debug the warnings, as they were, in fact, just warnings. They would not have prevented the creation of an executable file.) After checking the diagnostic printout and consulting with Mr. Pieter Bakkes, one of the lecturers involved with the transputer project, the conclusion was drawn that the source code, NEC2.F77, was too large for compilation (690kB). It was thus decided to divide the whole program into 17 separate segments and compile each segment separately, after which all segments were linked together to create the executable file NEC2.B4. This proved to be successful, and a file NEC2.B4 of size 340kB was created.

6. The file NEC2.B4 was subsequently executed, and at first everything seemed to indicate that NEC2 was running. However, about a quarter of the way through the run execution terminated. After a lengthy investigation inside the source code using several WRITE statements to follow the progress of the program, it became clear from the INMOS literature that only 2kB of on-chip (T800) storage is used for the stack. To use external storage for the stack, a NEC2.EXE file has to be created (by copying and renaming the link file, LINKT.EXE to NEC2.EXE) and executed. This proved to be successful, and NEC2 ran for the first time from beginning to end, using the 12-element log-periodic antenna example in the NEC2-users manual [1].

Note that because this particular example employs 78 segments, it would mean a matrix size of 6084 elements, which is more than the maximum of 4000 possible elements budgeted for in the original NEC2-version. The dimensions of the variables CM and IRESRV were changed to 10000 to avoid the use of external file storage. Provision still has to be made for the opening of files for disk storage.

RESULTS : The 12-element log-periodic antenna example in the NEC2 users manual [1] was used as a benchmark.

The results obtained from analysing the 12-element log-periodic antenna,using NEC2, on the T800 transputer were compared with those published in the NEC2-users manual [1], as well as those obtained from microVAX and VAX785 executions. The results agreed excellently in all three cases.

The runtime comparisons are as follows [2]:

Transputer T800 using static storage 32s

(In the original report, the T800 execution time was stated as 42s. However, a memory clock jumper on the transputer board was incorrectly set for the type of memory used, hence the radical improvement from 42s to 32s.)

microVAX using core storage 47s

VAX785 using magnetic tape storage
(with 11 other users on the VAX) 40s

" " (with no other users) 34s

VAX785 using core storage
(with 2 other users on the VAX) 18s

RECENT DEVELOPMENTS : The Department recently acquired the new INMOS Parallel FORTRAN compiler for use on the T800-transputer. This compiler was originally developed for use with several transputers in a parallel environment, with the added advantage of being able to write software in FORTRAN for parallel processes. The feasibility of compiling NEC2 with this compiler was investigated.

It was found that it was possible to compile the entire NEC2 source code, unsegmented, using this compiler. The execution time with this compiler is, however, about 3 seconds slower than the above benchmark, using the same log-periodic antenna (i.e. ± 35s). Other advantages for use with NEC2 could not be detected.

Additionally, the Group recently acquired the NEEDS package, as distributed by the Applied Computational Electromagnetic Society. This consists of an excellent pre-processor, IGUANA, for creating, editing and plotting the required NEC2 data cards for a given structure, accessing NEC2 or MININEC for execution and a graphics package, GRAPS, for plotting of the results. This package also includes SOMNEC, a FORTRAN program which creates an interpolation table for use by NEC2, for the cases where a non-ideal earth surface is used. This program was immediately transferred to the hard disk, debugged and successfully compiled on the transputer.

An obvious next step was to integrate the transputer-NEC2 and -SOMNEC into the NEEDS-facility. The status of this exercise is as follows :

a) It is now possible to access the transputer-NEC2 as well as the transputer-SOMNEC from the NEEDS menu; it involved a simple modification of the path to the transputer directory in the relevant menu-batch files.

b) Unfortunately the card deck output format of IGUANA is not the same as that required by the transputer-NEC2 (possibly to be compatible with the MININEC input format). An interface program was written to reconvert the IGUANA output deck (containing commas between fields) to the format required by the transputer-NEC2 (with spaces between fields, as in the NEC2 users manual [1]). This seemed to be less troublesome than to alter the READ and FORMAT statements in NEC2 in order to accommodate the commas in the IGUANA deck format. This interface program is automatically executed immediately after using IGUANA, just before returning to the main menu.

c) NEEDS contains an updated version of NEC2, with the added facilities of a helix generation card (GH) as well as a plotting card (PL) to facilitate storage of current, near-fields, far-fields or field strengths on disk for later plotting. As it was too much trouble to recommence debugging of this version for compilation on the transputer, it was decided to copy the subroutine HELIX, as well as all statements relating to this subroutine and the PL card to the existing transputer version of NEC2, debugging these additions where necessary.

Initially a problem was experienced in that only a maximum of 7 files could be opened, regardless of the number of OPEN-statements in the program. After initially blaming the transputer compiler, it later transpired that the CONFIG.SYS-file in the MS-DOS root directory did not make provision for enough files to be created. This error was promptly rectified by editing this file.

After the abovementioned modifications and additions were performed, a totally PC-based antenna analysis workstation was obtained, with performance comparable to a mainframe system. The program components which are primarily used, are the following :

IGUANA - this NEEDS-based program is used to to create, edit, view and plot the structure to be analysed. The final structure file is saved as NAME.DAT, where NAME is any user-defined name, and is stored in the transputer directory. This program is executed in the DOS-environment.

SOMNEC - this program is used to create non-ideal earth interpolation tables, which is saved as SOMNEC.DAT, for use by NEC2 where necessary. This program is executed on the transputer.

NEC2 - using the transputer, this program calculates currents, radiation patterns etc. This program makes MININEC obsolete. The output listing is saved as LISTING.DAT.

NECPP - this handy post-processor program in NEEDS searches through the output listing, LISTING.DAT, for the relevant data to be plotted. After finding the data, it is saved in the GRAPS-directory for subsequent plotting. It operates in the DOS-environment.

GRAPS - this is the graphical plotting sub-program in NEEDS, with several options for plotting of the results obtained from the execution of NEC2. It also operates in the DOS-environment.

The only limitation is a maximum segment dimension of 300, but this will be extended by the planned acquisition in the near future of a transputer board containing about 4MB of on-board memory.

CONCLUSIONS : The transputer compares very favourably with the mainframe systems, and can be thought of as the poor man's VAX. In effect microVAX performance can be obtained by using a standard PC as host for the University of Stellenbosch, Department of Electrical and Electronic Engineering T800 transputer board. The total investment is less than $US5000.

ACKNOWLEDGEMENT

The valuable assistance of Mr Pieter Bakkes, Senior Lecturer in the Department, is gratefully acknowledged.

REFERENCES

[1] Burke, G.J. and Poggio, A.J. ; Numerical Electromagnetics Code (NEC) - Method of Moments , 3 Program Description Manuals, NOSC, San Diego, California, USA.

[2] J.J. le Roux et al, Electronics Letters, 4 August 1988 , Vol 24, no 16, pp 991-992