AD-A035 097 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
NASTRAN ON THE TEXAS INSTRUMENTS ADVANCED SCIENTIFIC COMPUTER: --ETC(U)
NOV 76 M E GOLDEN, M M HURWITZ

UNCLASSIFIED DTNSRDC-76-0092 NL

1 OF 1
AD
A035097

END
DATE
FILMED
3-77

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER (14)
DTNSRDC [redacted]-76-0092

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle) (6)
NASTRAN on the Texas Instruments Advanced Scientific Computer: Feasibility Study and Experimentation.

5. TYPE OF REPORT & PERIOD COVERED (9)
Final rept.

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s) (10)
Michael E. Golden [redacted] Myles M. Hurwitz

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
David W. Taylor Naval Ship R&D Center
Bethesda, Maryland 20084

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (16) (17)
62721, F21241, XF21241028
Work Unit 1804-005

11. CONTROLLING OFFICE NAME AND ADDRESS
(12) 23p.

12. REPORT DATE (11)
NOV[redacted]976

13. NUMBER OF PAGES
23

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

NASTRAN
Computer Networks
NALCON
Texas Instruments Advanced Scientific Computer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Texas Instruments Advanced Scientific Computer is a very powerful, very fast, large central memory computer for scientific applications programs. Large programs such as NASTRAN, which perform extensive numerical processing of large matrices, should require significantly less central processor time when run on such a computer. A feasibility study was carried out which addressed the possibility of running NASTRAN in a split mode on the Texas Instruments Advanced Scientific Computer at the Naval Research Laboratory and over

(Continued on reverse side)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387 682 mt

(Block 20 continued)

on the CDC 6000 at DTNSRDC, and included experiments on the TIASC computer.

The major problem area found by the feasibility study involves transfer of data between the two computers, since the availability of a reliable, cheap, and fast means of transfer of very large amounts of data is not a certainty. If this problem can be overcome, the advantages of such a split mode operation, with respect to cost and resources, could be significant. The transfer of data will be accomplished by the Navy Laboratory Computer Network (NALCON), which will tie together the computers of ten Navy laboratories. A file transfer protocol which would allow binary data transfer between these two computers, thereby reducing the amount of data processing required, is recommended as part of NALCON. In addition, the standard NASTRAN matrix decomposition code is not cost effective on the Texas Instruments Advanced Scientific Computer. The matrix decomposition code and the input/output code associated with it should be replaced to make a split mode operation cost effective.

TABLE OF CONTENTS

LIST OF TABLES

ABSTRACT

The Texas Instruments Advanced Scientific Computer is a very powerful, very fast, large central memory computer for scientific applications programs. Large programs such as NASTRAN, which perform extensive numerical processing of large matrices, should require significantly less central processor time when run on such a computer. A feasibility study was carried out which addressed the possibility of running NASTRAN in a split mode on the Texas Instruments Advanced Scientific Computer at the Naval Research Laboratory and on the CDC 6000 at DTNSRDC, and included experiments on the TIASC computer.

The major problem area found by the feasibility study involves transfer of data between the two computers, since the availability of a reliable, cheap, and fast means of transfer of very large amounts of data is not a certainty. If this problem can be overcome, the advantages of such a split mode operation, with respect to cost and resources, could be significant. The transfer of data will be accomplished by the Navy Laboratory Computer Network (NALCON), which will tie together the computers of ten Navy laboratories. A file transfer protocol which would allow binary data transfer between these two computers, thereby reducing the amount of data processing required, is recommended as part of NALCON. In addition, the standard NASTRAN matrix decomposition code is not cost effective on the Texas Instruments Advanced Scientific Computer. The matrix decomposition code and the input/output code associated with it should be replaced to make a split mode operation cost effective.

INTRODUCTION

The Naval Research Laboratory (NRL) is presently performing acceptance tests on their very powerful, very fast, Texas Instruments Advanced Scientific Computer (TIASC). This computer derives its increased speed principally from a pipeline processing concept and is expected to have a capability similar to the IBM 360/91 but with a speed enhancement of approximately four. The TIASC at NRL currently has a central memory of four million bytes, or one million 32-bit words. Both central memory size and speed are expected to be increased in the future In comparison, the CDC 6000s at DTNSRDC have a maximum central memory of 131,072 60-bit words.

The TIASC computer should significantly reduce computational cost for the extensive numerical processing of large matrices often encountered by computer programs such as NASTRAN. Therefore, a feasibility study was undertaken to explore the possibility of taking advantage of this new computational facility by sharing a NASTRAN job between DTNSRDC's CDC 6000 computer and the TIASC, which, along with the computers at eight other Navy laboratories, will be tied together through the Navy Laboratory Computer Network (NALCON). The major mathematical computations would be performed on the TIASC, and the pre- and postprocessing for those computations would be handled by the CDC 6000. If such a procedure proved theoretically feasible, experimentation with the TIASC capabilities would follow.

Part I of this report documents the feasibility of running NASTRAN in such a split mode. Part II contains the results of the experimentation on the TIASC. It details central processor (CP) time comparisons, problems encountered, and recommendations both for NASTRAN and for other FORTRAN computer codes on the TIASC.

PART I: FEASIBILITY STUDY

by

Myles M. Hurwitz

Specifically addressed were the problems, costs, advantages, etc. of starting a NASTRAN analysis on the CDC 6000 at DTNSRDC, transferring large matrices to the ASC for "number-crunching" processing, transferring the results back to the CDC computer, and restarting the NASTRAN analysis. The processing may include such operations as matrix decomposition or eigenvalue extraction.

The following questions were considered. The answers given, with underlying assumptions, are based on work performed through 30 June 1976.

Q. The data transfer and cost-effectiveness aspects of the entire operation become meaningless unless the required matrices can be extracted easily from the program, the results inserted back into the program, and the program restarted at the breakpoint. Is this possible?

A. Yes, because of NASTRAN's checkpoint/restart, ALTER, and user tape capabilities. The program can be started with its checkpoint facility operational and can be terminated just after the required matrices are transferred to user tapes using the ALTER capability. After the required operations on these matrices have been completed by the ASC, the results may be stored on user tapes, and NASTRAN may be restarted just after the previous termination point. (The equivalent NASTRAN operations are skipped and the ASC results are read in by the ALTER facility.) NASTRAN will then execute to completion normally.

Q. Can the data be transferred from the CDC 6000 to the ASC?

A. Yes, but a number of operations may have to be performed to effect a meaningful transfer. As presently planned, a file transfer capability will be implemented on the Navy Laboratory Computer Network (NALCON), to which DTNSRDC's CDC 6000 computers and NRL's TIASC computer will be connected. Such a capability is scheduled after terminal access becomes operational on NALCON. (Remote job entry (RJE) will be a third capability.) A major question, however, concerns the type of file that can be transferred. There should be no problem in transferring ASCII files, but transferring binary files, as one would like to do both as a convenience and as a cost-saver, will not be possible unless an acceptable file transfer protocol (FTP) is implemented. The present outlook for such an FTP is not optimistic. If it is not implemented, the NASTRAN binary information, stored on the user tapes, will have to be converted to ASCII format. Some preliminary tests indicate that converting one million numbers will use approximately 90 CPU seconds on the CDC 6000. The one million figure was obtained by assuming a square matrix of order 10,000 and one percent dense, i.e., only one percent of the one hundred million numbers are non-zero. This percentage is typical of large NASTRAN matrices resulting from a standard structural analysis. Unfortunately, such a conversion will have to be performed four times, twice on each machine, once binary to ASCII, once ASCII to binary.

Q. How long will it take to transfer the data and how much will it cost?
A. If we assume that the standard wide-band telephone line will be the mode of communication, a dedicated 50,000-bit/second line will require approximately 1200 seconds to transmit one million numbers, sixty bits per number. Unfortunately, since a sixty-bit binary number may have to be converted to seventy-two ASCII bits, and, since a dedicated telephone line is not probable, the time for transmission may increase to as much as 3600 seconds, or one hour, assuming an effective transmission rate of 20,000 bits per second as suggested by ARPANET studies. These figures are for one-way transmission only. Naturally, the results have to be transferred back to DTNSRDC. The cost of transmitting the data is presently unknown. The cost of using NALCON is currently expected to involve a surcharge over standard computer costs. However, with telephone line charges increasing rapidly, especially in comparison to CPU rates, a surcharge may not be the final method of distributing costs.

If any aspect of the entire operation could cause the operation to be considered unfeasible, it would be the cost and/or time of transmission rather than any technological aspect of the transfer. One possibility which could overcome these problems is microwave transfer rather than telephone line transfer. However, the use of such a system would have to be significant to justify the large initial cost of implementing it.

Q. Are there any other problems with the transfer of the data?

A. Yes, but they are probably relatively minor. Two examples are: (1) the single-precision 60-bit CDC word will have to be converted to two 32-bit ASC words, i.e., one double-precision word; (2) NASTRAN packs its matrices in such a way that zeroes are not stored. If the ASC routines performing the "number-crunching" do not recognize such packing, the matrices will have to be unpacked, operated on, and packed again before the information can be transferred back to the CDC computer.

Q. Once the data have been transferred to the ASC, what software will be available to operate on the data?

A. Texas Instruments is providing the ASC with a Scientific Program Library (SPL) which should contain standard matrix operations, e.g., solution of linear systems, eigenvalue extraction, etc. However, the exact composition of the SPL is still undecided. Presumably the routines in the SPL will take very good advantage of the ASC's pipeline processor, which gives the ASC its very high speed compared with third generation computers. What is unknown at present is whether or not these routines will contain some of the options which are standard in NASTRAN, e.g., decomposition with spill, unsymmetric decomposition, several methods of eigenvalue extraction, complex decomposition, etc. If the SPL does not contain some of these facilities, the appropriate routines could be lifted from NASTRAN with some relatively modest modification. However, such routines may not take good advantage of the pipeline processor without a major overhaul.

Q. How do costs for such operations on the CDC 6000 and the TIASC compare?

A. Before the costs of computation for the two machines can be compared, a number of assumptions must be made. We will assume that, at the lowest priority on the DTNSRDC CDC 6000, a job which uses 140K central memory (which would mean only a relatively small NASTRAN matrix if spilling operations are to be avoided) will cost approximately 12 cents per CPU second or $432 per CPU hour. (The actual cost will be higher due to charges for IO and other miscellaneous items.) On the NRL TIASC, the anticipated lowest priority rate is $1000 per CPU hour. Charges for other items, if any, e.g., central memory, IO, etc., are not yet formulated.

To complete the set of assumptions, we must consider speed comparisons. The anticipated speed of the ASC is four times that of the IBM 360/91. According to NASTRAN time estimates for the decomposition of a real, symmetric matrix, with no spill, the IBM 360/91 is eight times faster than a CDC 6600, which, in turn, is three times faster than the CDC 6400. (Charge times on all DTNSRDC's 6000 computers are in terms of seconds on the CDC 6400.) Therefore, under this set of assumptions, for the type of matrix decomposition described (which is probably the type most commonly used), the TIASC is approximately 96 times faster than the DTNSRDC CDC 6000 computers. With the assumed charge rate, the ASC will be one-fortieth as expensive as the CDC 6000. Note carefully, however, that:

(1) The CDC rate is based on minimum NASTRAN central memory. For very large matrices, the full 300K central memory is probably required, which increases the rate to $576 per CPU hour.

(2) The ASC rate assumes no charge for central memory. The equivalent CDC rate would be $288 per CPU hour, which brings the cost factor down to 28, in favor of the ASC.

(3) These comparisons do not include the costs for converting a matrix on the CDC 6000, with its 60-bit word in NASTRAN packed format, to an ASC 64-bit, double-precision word. Nor do they include binary to ASCII conversions, transferring the data, unpacking the matrix, and then reversing the procedure to bring the results back to DTNSRDC.

(4) It is anticipated that later upgrading of the ASC will increase the speed factor of the ASC to the range of 7 to 10 times that of the IBM 360/91.

Therefore, while the ASC is, in theory, a much faster, much less expensive computer for the types of calculations considered, the final cost comparisons, in practice, are still unknown.

Q. Are there any other advantages to transferring such operations to the ASC?

A. Yes. Because of the ASC's increased speed and a central memory approximately four times larger than that of the CDC 6000, operations on a very large matrix, which would tie up an entire CDC machine for long periods of time, could be performed in just a few seconds on the ASC. (Differences in word size are taken into account; the ASC has one million 32-bit words vs. CDC's 131,000 60-bit words.) Therefore, freeing the computer resources at DTNSRDC for other work could be an important advantage.

Q. As a user, what must I do to run a NASTRAN problem as contemplated?

A. Under the worst possible circumstances, the following steps would be required:

(1) Run NASTRAN on CDC with its checkpoint and user file capabilities operational and terminate execution prior to the time-consuming operation.

(2) Convert each CDC 60-bit binary word on the user file to two 32-bit ASC binary words.

(3) Convert each ASC word from binary to ASCII.

(4) Instruct NALCON to transmit the ASCII file to the ASC.

(5) On the ASC, convert each ASCII word back to binary format.

(6) Expand matrix from NASTRAN packed format to unpacked format, and then to format required by ASC "number-crunching" program.

(7) Execute ASC program and store results on ASC file.

(8) Convert the matrices on the ASC file to NASTRAN packed format.

(9) Convert binary words to ASCII format.

(10) Instruct NALCON to transmit the ASCII file back to the CDC.

(11) Convert each ASCII word to binary format.

(12) Convert each pair of 32-bit ASC words to one 60-bit CDC word and store on NASTRAN-type user file.

(13) Restart NASTRAN, with the user file as input, starting after the time-consuming operation.

Q. Can these steps be automated?

A. Possibly. With CDC's BEGIN/REVERT capability, a set of control cards can be catalogued and called into the control card record. Therefore, a complete set of control cards could be set up for users and called in by an individual user with one or two cards. However, this method assumes that steps 4-10 above, those steps which involve NALCON's file transfer and RJE capabilities, can be performed from the CDC control card and/or input records. It is not presently known whether this procedure will be possible.

Q. What other alternatives exist to accomplish the type of operations discussed?

A. The time, cost, and reliability of transferring very large matrices over telephone lines through the network are probably, at present, the major problem areas which must be addressed before such a transfer can become a reality. Therefore, if the reasons for such data transfers are primarily lower cost and/or freeing DTNSRDC computer resources, rather than technological advances in computer sharing through networks, then two other alternatives are immediately obvious:

(1) Because of the relative proximity of DTNSRDC and NRL, a computer tape could be brought to NRL for processing, and the results could also be stored on tape and returned.

(2) The complete NASTRAN program could be converted to operate on the ASC, and input data and output results could be transferred through the NALCON RJE capability.

Q. What is involved in converting NASTRAN to the ASC?

A. One of the major possible stumbling blocks in converting NASTRAN to a new computer is the lack of a linkage editor with which to assemble NASTRAN object decks into an executable program. This problem existed on the CDC 6000 and on the Honeywell 6000. Happily, only minor problems may be expected on the ASC, which has an IBM-type linkage editor. Therefore, conversion to the ASC would probably require approximately one man-year of work plus required education in such ASC areas as the FORTRAN compiler, ASC assembly language, and linkage editor. If, however, this assumption concerning the linkage editor is incorrect, a significantly higher level of effort would be required. The resulting program would probably contain a number of operations in FORTRAN rather than in machine-dependent assembly language as they exist in the CDC, IBM, and UNIVAC versions. NASTRAN's I/O package, called GINO, is the primary example. A major unknown, however, is the efficiency of the overall execution of NASTRAN on the ASC as compared to its efficiency on the CDC 6000 since most of the program's operations are not of the "number-crunching" variety. NASTRAN's major mathematical routines, however, could be replaced by similar routines from the SPL, hopefully with relatively modest modification, to increase overall speed.

Any effort to convert NASTRAN to the ASC will be greatly facilitated by NALCON's RJE capability and could be greatly hindered by the usual problems encountered with a new computational facility.

SUMMARY AND CONCLUSIONS

The transfer of NASTRAN data from DTNSRDC's CDC 6000 to NRL's TIASC for lengthy numerical computations and the return transfer of the results have been studied with regard to procedures, problems, costs, advantages, and alternatives. While a significant amount of processing may have to be performed on the data to accomplish such a transfer, the transfer does seem feasible. At present, however, the major problem area seems to be the transfer itself, since the availability of a reliable, cheap, and fast means of transfer of very large amounts of data is not a certainty. If this potential bottleneck can be overcome, the advantage, with respect to cost and resources, could be significant.

If there is one recommendation to be made from the study, it is to attempt to implement a file transfer protocol between the TIASC and CDC 6000 which would allow binary data transfer. Such a protocol would significantly reduce the amount of data processing required.

PART II: EXPERIMENTATION

by

Michael E. Golden

The major task addressed was that of obtaining central processor (CP) times for NASTRAN's real, symmetric matrix decomposition. As this portion of NASTRAN is usually the most CP-bound portion of the program for large problems, it was hoped that CP time for NASTRAN on the TIASC would be significantly less due to its pipeline processing capability. As will be shown, this was not the case. A decrease was obtained, but it was only enough to make the TIASC just marginally cost effective for the standard NASTRAN program.

A basic understanding of a TIASC vector instruction is fundamental to the ensuing discussion. A vector instruction performs vector and matrix operations to take maximum advantage of the pipeline processing capability of the TIASC. A vector instruction resembles a nested FORTRAN DO-loop but operates as a single TIASC instruction.[1] For example, the nested loop

```
      DO 10 I=1,10
      DO 10 J=1,10
   10 A(I,J) = B(I,J) + C(I,J)
```

would execute as one vector instruction instead of 100 scalar instructions.

The computer program on which the experimentation was based is the simultaneous linear equation solver extracted from NASTRAN Level 15 but with NASTRAN's General Input/Output (GINO) routines extracted from NASTRAN Level 12. Level 12 GINO, written in FORTRAN, was used to avoid re-programming Level 15 GINO, which was written in computer-dependent assembly language. The major time-consuming portion of the equation solver is the matrix decomposition.

---

1 "Glossary of Terms and Partial List of Acronyms", Texas Instruments Incorporated, Publication Number 930200-1, March 1976, p. 37.

TIMING RESULTS

The fast compile, slow execute (FX) FORTRAN compiler was used first for experimentation, for debugging purposes and to obtain a fully executable program. However, due to TIASC system software problems, it was used for most of the project time period. These problems will be discussed later in this report. The execution time for NASTRAN's real, symmetric matrix decomposition on the TIASC using the FX compiler was roughly 80% of that for the same problem on a CDC 6400.

After satisfactory results were obtained with the FX compiler, the slow compile, fast execute (NX) FORTRAN compiler was used. More system software problems, to be discussed later, were encountered. The compiler, given standard NASTRAN source code, could produce no vector instructions; and NASTRAN therefore could not use the pipeline processing capability. The lack of vector instructions was due to the type of NASTRAN FORTRAN code presented to the compiler. Unfortunately, precise restrictions must be met in order for the NX compiler to vectorize FORTRAN source statements. Standard NASTRAN source code did not meet all these restrictions. The execution time for NASTRAN matrix decomposition using the NX compiler with no vectorization was approximately 30% of that for the same problem on a CDC 6400.

When the NASTRAN source code (the inner loop of subroutine SDCOMP, which contains the code most important with respect to CP time) was modified, vectorization was obtained using the NX compiler. Execution time dropped by approximately 20% over the non-vectorized NX version. The final execution time for NASTRAN matrix decomposition on the TIASC was therefore about 25% of that for the same problem on a CDC 6400.

The costs given in the following two tables concern only CP charges. They are provided only as a rough estimate of cost to the user and do not incorporate such chargeable items as central memory usage, costs for printing output, etc. CP cost of the TIASC at NRL is currently $720 per CP hour but will change to $1000 per CP hour in the near future. The cost of the CDC 6400 runs as given is for priority P2, at $270 per CP hour.

Table 1 summarizes the CP times and costs for NASTRAN on the TIASC and for the same problem run on a CDC 6400. The problem involved the solution of a set of simultaneous linear equations of order 1500, with a symmetric coefficient matrix with a semi-bandwidth of 300. Double precision computation was used on the TIASC; single precision computation was used on the CDC 6400. CDC single precision has a 48-bit fraction, and TIASC double-precision has a 56-bit fraction. However, double precision operations on the TIASC take no longer than single precision operations.

TABLE 1 - NASTRAN MATRIX DECOMPOSITION
TIMES AND COSTS

CDC	TIASC		
6400	FX	NX, no vectorization	NX with vectorization
47.4	38.6	14.4	12.3
$3.56	$7.72	$2.88	$2.46

DTNSRDC Code 184 has been using the TIASC experimentally for non-NASTRAN programs to obtain CP comparisons. With suitable source code modification, vectorization was obtained. CP cost reductions have been on the order of 35-40 to 1 when compared to DTNSRDC's CDC 6400, making the TIASC very cost effective for straight-forward, vectorized FORTRAN routines of the "number-crunching" variety. Since the TIASC is advertised as approximately four times faster than the IBM 360/91, experiments were also performed on the IBM 360/91 at Johns Hopkins University Applied Physics Laboratory. The current cost to DTNSRDC users for that IBM 360/91 is $1000 per CP hour plus surcharge.

Table 2 gives CP time in seconds and cost for the NRL TIASC, the DTNSRDC CDC 6400, and the APL IBM 360/91. The non-NASTRAN problem was the solution of a set of 300 linear equations in 300 unknowns by the method of Gaussian elimination. The matrix concerned was real, full, and non-symmetric. All results given in this table are for single precision computations.

TABLE 2 - MATRIX INVERSION TIMES AND COSTS

CDC 6400			IBM 360/91	TIASC	
OPT=0	OPT=1	OPT=2	H Compiler	NX Compiler	
335.8 $25.19	142.5 $10.69	135.4 $10.16	8.8 $2.45	27.1 $5.42	unmodified subroutine
			10.3 $2.86	3.8 $0.76	modified subroutine for vectorization

The IBM 360/91 column in Table 2 indicates that the TIASC derives its speed primarily from the execution of vector instructions. Most computer programs do not vectorize immediately on the TIASC, so usually additional coding must be added or existing code changed to achieve vectorization. The changes/additions generally take the form of DO-loop simplification. The simplification adds more scalar code to the program, and the program should therefore consume more CP time. However, the vectorization achieved more than compensates for the code added, resulting in a net reduction in CP time on the TIASC.

Unfortunately, this does not happen on a scalar machine such as the IBM 360/91. The addition of new code increases total job CP time in all cases, as Table 2 shows. In general, a vectorized program which runs most efficiently on the TIASC will not be the most cost effective tool for computation on non-TIASC hardware. Any program written for the TIASC will, in general, be a special purpose program to take advantage of vector instructions and will contain code which on any other computer would be considered highly inefficient.

Also note that, for this problem, the fastest TIASC time was only 2.3 times faster than the fastest IBM 360/91 time, rather than the advertised factor of 4.

PROBLEMS

A secondary task of this experimental work was to determine the general reliability of the TIASC at this time. Many problems, of varying importance, were encountered and will be discussed in order of increasing importance. Anticipated problems associated with total conversion of NASTRAN to the TIASC will also be discussed.

The first problem, although very minor, indicates the current state of the TIASC operating system (OS) and system software. The problem is the loss of permanent files. On one occasion, the master NASTRAN source file was destroyed, apparently by the OS. Only the existence of an earlier backup file prevented a total catastrophe. In any event, one week was lost in returning the file to its prior condition.

In addition, an interactive system store command to a permanent file followed immediately by a system crash will totally destroy the file. The severity of this event can be minimized to some extent by an alternate store command, but the possibility of loss still exists.

A second problem involves a minor difference in the FORTRAN code expected by the TIASC compiler from that of IBM, CDC, and UNIVAC computers. Alternate returns from subroutines must be preceded by an "&" character, not a "$" character as in IBM and UNIVAC FORTRAN. As alternate returns appear profusely throughout NASTRAN, a line by line search of the source code is necessitated. Although the search can be automated through a string manipulation language such as SNOBOL, the changes must be made before NASTRAN will compile properly. The TIASC character set is standard EBCDIC.

Another problem lies in the failure of the FORTRAN BACKSPACE command to work properly due to a bug in the I/O portion of the FORTRAN library. The problem can be eliminated by replacing the BACKSPACE command with a REWIND and a series of READ commands, or it can be overcome by using TIASC machine dependent code as described in recommendation two at the end of this report.

Without a doubt, the FORTRAN compilers present the most serious problem encountered to date. Most of the time for experimentation was spent in "kludging" the NASTRAN source code to get around compiler bugs. Virtually no routine was left untouched. The bugs include:

(1) faulty linkage to FORTRAN statement numbers referenced via alternate returns if the alternate return was the only reference to that statement number in the routine;

(2) failure to detect FORTRAN errors such as unbalanced parentheses and CDC call statements with alternate "RETURNS" phrases left in, and then execution of this faulty code (FX compiler only);

(3) failure to pass externals properly as subroutine argument parameters; and

(4) generation of faulty DO-loops code: storing by the compiler of a value in a register, destroying that value, and then the later assumption by the compiler that the value still existed in the register for use.

Regardless of the problems encountered, however, the NRL and TI personnel were extremely helpful throughout the experimentation phase. All the compiler bugs could be removed by source code modification, but problems with the compiler do exist and will exist for the foreseeable future. All prospective users of the TIASC should be aware of this problem.

Curiously, the other non-NASTRAN programs run on the TIASC have experienced no compiler errors. This may be due to straightforward, simplistic code or other factors. Further investigation would seem to be in order.

FUTURE PROBLEMS

Attempts to convert NASTRAN completely to the TIASC will immediately encounter two problems. One is relatively minor and can be easily solved: that of the BLOCK DATA subprogram as it exists on the TIASC. Only one such subprogram is allowed; however, NASTRAN has many. Therefore, all NASTRAN BLOCK DATA subprograms will have to be merged into other subroutines or made into dummy subroutines. The second problem, which may ultimately render a total conversion process nearly impossible, concerns the placement of FORTRAN COMMON statements in the load module via the linkage editor. The TIASC linkage editor does not permit a programmer to explicitly reference COMMON blocks for arbitrary placement in an overlaid program tree structure except to move them up into the tree into low order core. As a result, NASTRAN open core COMMON blocks cannot easily be placed at the end of a particular tree branch in high order core at the end of the machine instructions.

This problem may be solvable, since the loader loads COMMON blocks in each tree branch at the end of the first subprogram in which they are encountered, and in the order specified in the subprogram. To solve the problem, an open core COMMON block must appear in only one subprogram, and that subprogram must be at the end of its appropriate branch in the overlay structure. This solution would be usable only through major NASTRAN source code modification.

If in NASTRAN two tree branches use the same open core COMMON block for passing information from one to the other, this solution is useless. The only possible solution would be to modify the TIASC linkage editor to handle COMMON block placement, rather than to modify NASTRAN. Based on earlier attempts by Ford Motor Company to modify the Honeywell H6000 linkage editor for NASTRAN, conversion of the TIASC linkage editor for NASTRAN would probably take in excess of two years. After a reliable modified linkage editor was obtained, the NASTRAN conversion would still lie ahead.

RECOMMENDATIONS

The experiments performed indicate that the standard NASTRAN program, as a whole, will vectorize only marginally, and that any program on the TIASC which does not vectorize to a significant degree will be only marginally cost effective. NASTRAN was originally written so as not to restrict the size of problems by using arrays of fixed dimensions. As a result, its large arrays are passed through argument lists or placed in COMMON statements to be overlaid at the end of source coding in a tree structure. These arrays typically are given a dimension of one, and their true size is driven by NASTRAN's open core philosophy. Unfortunately, the TIASC compiler must have absolute dimensions on all arrays at compile time in order to generate vector instructions. The result is that:

(1) all DIMENSION and COMMON statements must be recoded to provide true array sizes, or

(2) all loops using these arrays must be broken into their simplest components.

Otherwise, vectorization on the TIASC does not occur.

The work described here has led to several recommendations for any future work with NASTRAN on the TIASC. The first recommendation is a direct consequence of the CP comparisons obtained thus far and the fact that NASTRAN will vectorize only marginally. The standard NASTRAN matrix decomposition code, as well as other time consuming numerical computations, e.g., eigenvalue extraction, should be replaced. Although the complex and non-symmetric real equation solvers were not tested, they generally follow the form of the real symmetric solver tested. These routines should be replaced with a set of routines which will handle large, sparse, banded matrices in-core with a separate set of routines for matrices too large to fit in core. Such code either already exists at DTNSRDC, developed by the Computation and Mathematics Department, or can easily be adapted from existing code. This task should take no longer than three man-months, and the result would be a code which utilizes the TIASC pipeline capability. With a CP speed ratio of 40 to 1 as evidenced by the non-NASTRAN studies carried out, the TIASC is a cost effective tool for solving large systems of equations if total or partial vectorization of the source code is achieved.

Secondly, the level 12 NASTRAN GINO FORTRAN code currently in use with NASTRAN on the TIASC should be replaced with machine-dependent code, either by direct disk I/O via the S$DIO utility or by BSAM assembly language macros. CP costs associated with I/O will drop one-third to one-half. This task should take no longer than 3 to 6 man-months to accomplish.

Prior experience indicates that a full NASTRAN conversion should take two to three man-years. However, the problems cited earlier in this report could result in a much longer period of time, possibly in excess of five years, for a total NASTRAN conversion to the TIASC.

A final general recommendation can be made. Any groups envisioning use of the TIASC for their programs should run extensive benchmarks on their own computers. Upon conversion of the programs to the TIASC, all program paths should be tested exhaustively. Conversion is entirely possible, but users should be prepared to encounter problems with the FORTRAN compiler, and to perform major code revisions to obtain pipeline processing or remove unforeseen system problems.

INITIAL DISTRIBUTION

Copies:

1 USA Aberdeen Proving Ground
  1 Code AMXBR-VL-REB/E. Quigley

1 USA Picatinny Arsenal
  1 Code SARPA-MI-E/B. Nagel

1 USA Watervliet Arsenal
  1 Code SARWV-RR-AM/P. O'Hara

1 CNO
  1 Code OP-91/R.R. Jeske

6 NRL
  1 Texas Instruments/P. Anderson
  1 Code 4200/A. Bligh
  1 Code 4221/H. Brock
  1 Code 2312/R. Perlut
  1 Code 4221/G. Perez
  1 Code 7043/L. Turner

1 DNL

2 USNA
  1 Dept Math
  1 Lib

2 NAVPGSCOL
  1 Code 0211/D.G. Williams
  1 Lib

1 NAVWARCOL

1 NROTC & NAVADMINU MIT

1 NAVAIRDEVCEN
  1 Code 85/H. Tremblay

2 NELC
  1 Code 5000/A. Beutel
  1 Code 5020/J. Goertz

1 NAVUSEACEN San Diego
  1 Code 453/L. Maudlin

1 NAVWPNSCEN
  1 Code 303/L. Lakin

1 NCSL
  1 Code 760/C. Callahan

2 NSWC White Oak
  1 Code WA-21/R.J. Edwards
  1 Code WA-53/N. McElroy

Copies:

1 NSWC Dahlgren
  1 Code DK-60/C. Lamonica

2 NUSC NPT
  1 Code 3701/C. Curtis
  1 Code 3701/R. Messier

5 NUSC NLONLAB
  4 Code EM/A. Carlson
    J. Frye
    R. Kasper
    J. Patel
  1 Code PA-4/K. Jones

1 NAVSHIPYD BREM/Lib

1 NAVSHIPYD CHASN/Lib

1 NAVSHIPYD MARE/Lib

1 NAVSHIPYD NORVA/Lib

1 NAVSHIPYD PEARL/Lib

1 NAVSHIPYD PHILA/Lib

1 NAVSHIPYD PTSMH/Lib

12 DDC

2 AFFDL
  1 Code FBR/J. Johnson
  1 ASIAC/G. Negaard

1 NASA Ames Res Cen
  1 M.S. 213-4/M. Chargin

2 NASA Goddard SFC
  1 M.S. 720.2/J. Mason
  1 M.S. 720.2/R. Mitchell

1 NASA Langley Res Cen
  1 M.S. 253B/D. Weidman

2 NASA Marshall SFC
  1 Comp. Lab./B. Harrison
  1 Code AAD/R. McComas

1 Anamet Laboratories, Inc.
  100 Industrial Way
  San Carlos, CA 94070
  1 C. Johnson

INITIAL DISTRIBUTION

Copies:

1 Exxon Production Research
P.O. Box 2189
Houston, Texas 77001
1 A. Potvin (N.W. 450)

1 Sperry Support Services
716 Arcadia Circle
Huntsville, Alabama 35801
1 R. Schmitz

1 US Steel Research Center
Monroeville, PA 15146
1 G.J. Hutchins

CENTER DISTRIBUTION

Copies:		
1	1100	Ellsworth W M
1	1500	Cummins W E
1	1700	Murray W W
1	1720	Krenzke M A
1	1720.1	Kiernan T J
1	1720.2	Hom K
1	1720.3	Jones R F
1	1730	Stavovy A B
1	1730.5	Adamchak J C
1	1740	Hansen I S
1	1800	Gleissner G H
2	1802	Frenkiel F
		Theilheimer F
1	1804	Avrunin L
1	1805	Cuthill E H
1	1809.3	Harris D
1	1830	
1	1840	Lugt H J
2	1843	Schot J W
		Haas M E
8	1844	Dhir S K
		Everstine G C
		Gignac D A
		Henderson F M
		Matula P
		McKee J M
		Schroeder E A
		Zarda P R
20	1844	Golden M E
20	1844	Hurwitz M M
1	1850	Corin T
1	1856	Skall M W
1	1860	Sulit R A
1	1880	Camara A W
1	1890	Gray G R
1	1890.1	Taylor N M
2	1892.1	Strickland J D
		Good S E
1	1892.2	Sommer D V
1	1892.3	Minor L R
1	1900	Sevik M M
1	1960	Feit D
30	5214.1	Reports Distribution
1	5221	Library (C)
1	5222	Library (A)