SX-3 SUPERCOMPUTER SYSTEM AND ITS RELATED SOFTWARE

Hiroshi Katayama

Basic Software Development Division
NEC Corporation
Nisshin-cho, Fuchu-city
Tokyo 183, Japan

1. INTRODUCTION

The SX-3 supercomputer series has been developed as the successor to the SX-2 which was the first supercomputer developed by NEC. The SX-3 series consists of eight models ranging from the single processor entry model SX-3/1L to the 4 processor top model SX-3/44. The performance range covers from 680 MFLOPS to 22 GFLOPS. Table 1 shows the system specifications of all eight models.

This paper introduces the SX-3 architecture, operating system, Fortran and tools with some performance data. It also highlights hardware and software features to support parallel processing or multitasking.

2. ARCHITECTURE

The architecture of the SX-3 is based upon that of the SX-2, but it is extended and enhanced.

2.1 System Configuration

Figure 1 shows the maximum system configuration.

The arithmetic processor (AP) is the processor on which user programs run, while the control processor (CP) is the processor which is mainly in charge of I/O operations. The machine cycle of the AP is 2.9 nano second.

The main memory unit (MMU) is shared by all AP's. The extended memory unit (XMU) is also shared by them. Its transfer speed is 2.75 G bytes/sec. It can be used for high-speed disk units, high-speed swapping devices or a large capacity of disk cache.

2.2 Processor Architecture

Figure 2 shows the internal structure of the arithmetic processor. The major extension from the SX-2 is as follows.

1. The vector pipelines are multi-functional, that is, each vector pipeline set consists of two add/shift and two multiply/logical pipelines.

2. The capacity of the vector registers is almost doubled. The size of the instruction buffer is also doubled.

3. Two types of pages are available; 32 K byte pages and 1 M byte pages.

4. To enhance program portability from other systems, the following features are added.

Table 1 System Specifications of the SX-3 and SX-2A

MODEL	SX-3								SX-2A
	44	42	24	22	14	12	11	1L	
Vector Peak Performance	22GFLOPS	11GFLOPS		5.5GFLOPS		2.75 GFLOPS	1.37 GFLOPS	0.68 GFLOPS	1.3 GFLOPS
Number of Arithmetic Processors	4		2		1				1
Vector Registers	144KB	72KB	144KB	72KB	144KB	72KB	36KB	36KB	80KB
Vector Mask Registers	256b x 8	128b x 8	256b x 8	128b x 8	256b x 8	128b x 8	64b x 8	64b x 8	256b x 8
Scalar Registers	64b x 128								
Vector Pipelines	16	8	16	8	16	8	4	2	16
Cache Memory	64KB								
Main Memory [Max]	2048MB			1024MB			512MB		1024MB
Extended Memory	16GB							8GB	8GB
I/O Channels [Max]	256							64	64
I/O Transfer Rate	1GB/sec							192MB/s	192MB/s

SX-2A is the name of the newest model of SX-2.

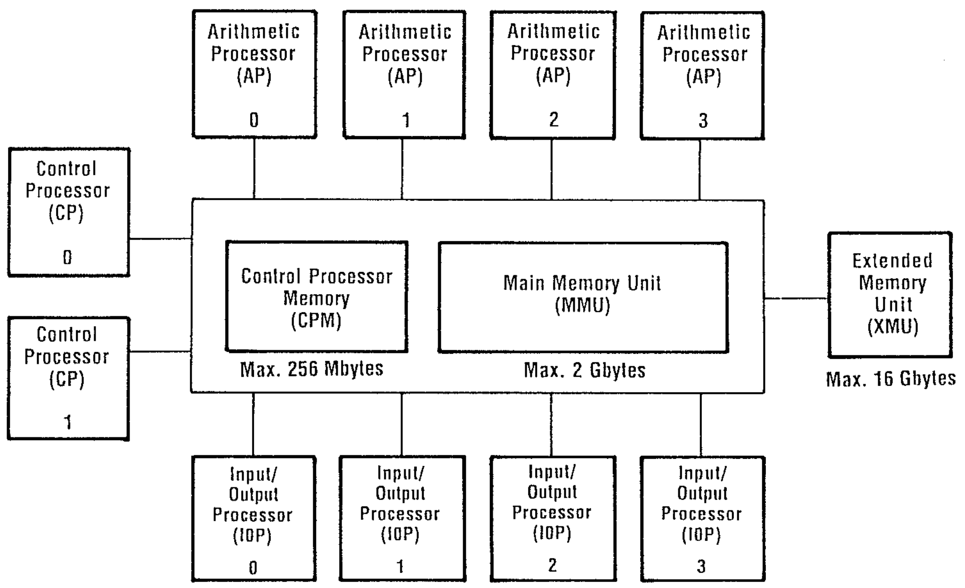

Arithmetic Processor (AP) 0
Arithmetic Processor (AP) 1
Arithmetic Processor (AP) 2
Arithmetic Processor (AP) 3
Control Processor (CP) 0
Control Processor (CP) 1
Control Processor Memory (CPM)
Max. 256 Mbytes
Main Memory Unit (MMU)
Max. 2 Gbytes
Extended Memory Unit (XMU)
Max. 16 Gbytes
Input/ Output Processor (IOP) 0
Input/ Output Processor (IOP) 1
Input/ Output Processor (IOP) 2
Input/ Output Processor (IOP) 3

Figure 1 System Configuration of the SX-3

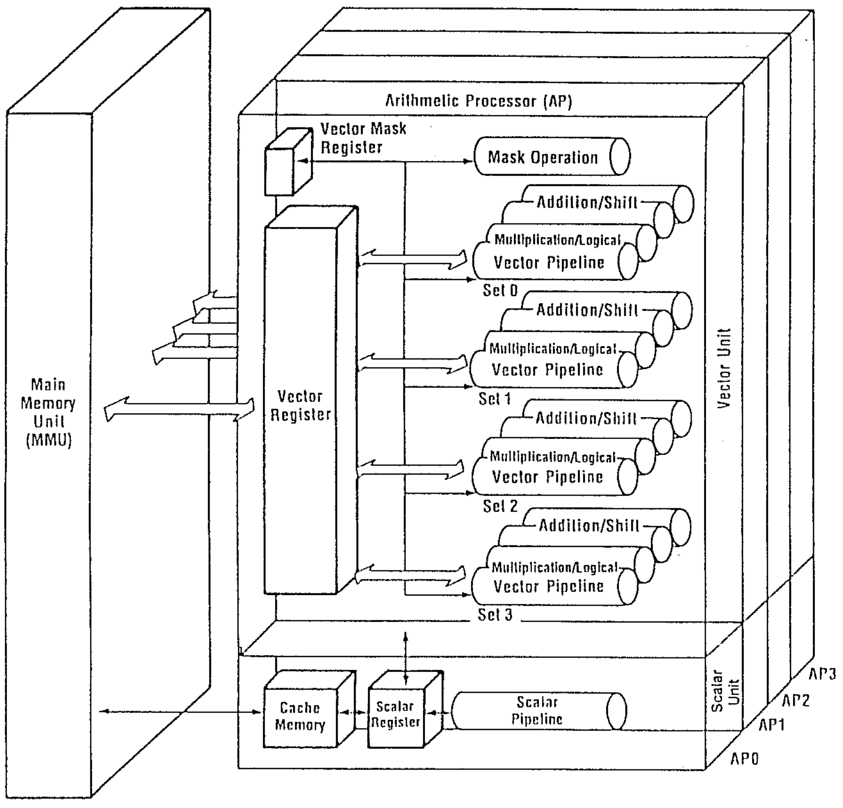


Figure 2 Internal Structure of the Arithmetic Processor

- Extended floating-point data format with wider exponent
- 64 bit binary operations

5. Features added to the SX-3 include minor but important ones for program debugging and tuning such as:

   - Interruption by accessing the specified memory address
   - Registers to count up the number of floating point operations or to accumulate loss time caused by bank conflict by vector load/store instructions.

3. SOFTWARE

The SX-3 supports two operating systems; SUPER-UX and SXOS. The SXOS is the proprietary operating system developed for the SX-2. The SUPER-UX is the UNIX-based operating system newly developed for the SX-3.

The languages supported for the SX-3 are Fortran, C, Pascal and assembler. Several utilities are available as program debugging and tuning tools.

This section presents the major features of the SUPER-UX, Fortran and tools. The multitasking feature will be presented in the later section.

3.1 SUPER-UX Operating System

The SUPER-UX is developed on the base of AT&T System V R3.1 with many extensions and enhancements to meet supercomputer requirements. It also incorporates many BSD 4.3 functions. The major areas which are extended and enhanced are as follows.

- File system and I/O processing
- Batch processing
- Scheduling
- Security
- Accounting

- Operation management
- Multiprocessor and multitasking support
- Networking

In the later release, the SUPER-UX will be enhanced to comply System V R4.1. System V R4.1 is viewed as fully compliant with IEEE 1003.1 (POSIX).

In the following sections, the file system will be explained as an example of major enhanced features from the standard UNIX.

3.1.1 File System of the SUPER-UX

The file system of the standard UNIX has many advantages. However, as a file system for a supercomputer which requires a large volume of data files and high-speed data transfer, it has several serious deficiencies .

In the SUPER-UX, the second file system has been designed and built to remove these deficiencies of the UNIX file system.

In addition, the concept of a "virtual volume" is introduced to form a flexible, large-scale file system.

3.1.2 Virtual Volume

A virtual volume is a logical volume which can be composed of several physical volumes of the same or different types. It brings the following advantages.

- A file can extend over multiple volumes because a virtual volume can be composed of multiple physical volumes.
- Disk striping can be realized by constituting a virtual volume from several disks.
- A file which exceeds XMU free space can be allocated by constituting a virtual volume from XMU and disks.

3.1.3 IAS

The file system of the SUPER-UX has a layered structure as shown in Figure 3. The upper is a file management system layer and the lower is a physical driver layer.

The physical driver layer is named the IAS (Intelligent I/O Accelerator System). The virtual volume driver is the main part of the IAS. It converts an access request to a virtual volume into a physical access request to a XMU driver or disk driver by using the "vdevsw" switch. Disk striping is implemented in the virtual volume driver. The IAS can use XMU as disk cache to speed up disk I/O.

3.1.4 SFS

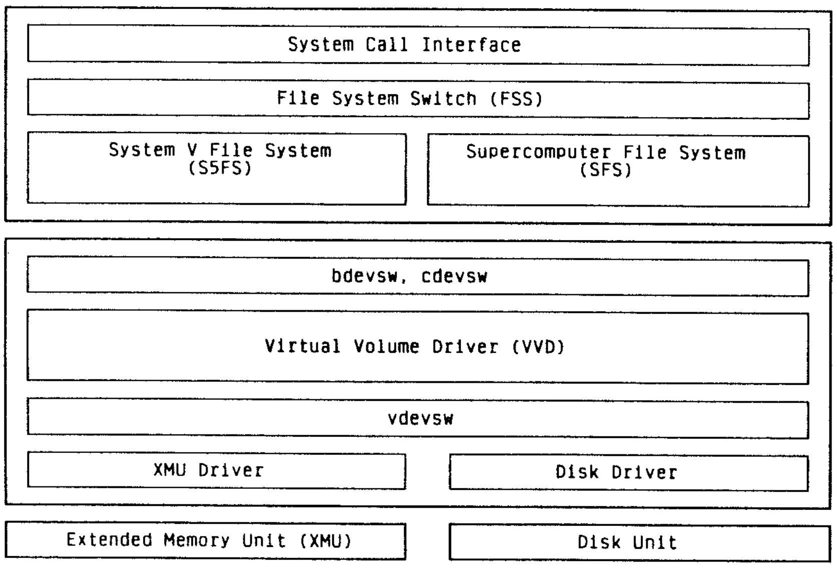
The file management system layer is composed of the FSS (File System Switch), S5FS (System V File System) and SFS (Supercomputing File System). The SFS is the file system developed for the SUPER-UX, while the S5FS is the original file system of the UNIX.

A file system is selected from SFS and S5FS when it is defined by a mkfs command. A SFS file system can be mounted on S5FS file system and vice versa. Files are compatible in both file systems.

In the SFS, data are transferred without using buffer cache. This speeds up transfer of large data because extra data move and buffer flush can be avoided.

To make file allocation and access more efficient, a "cluster" is defined in addition to a block. It is the largest group of blocks which can be allocated contiguously. Its size is specified by users. The I-nodes in the SFS are managed in clusters instead of blocks. Table 2 shows addressing capabilities in case of 64 K byte cluster and 256 K byte cluster.

3.1.5 Performance of the File System



System Call Interface
File System Switch (FSS)
System V File System (S5FS)
Supercomputer File System (SFS)
bdevsw, cdevsw
Virtual Volume Driver (VVD)
vdevsw
XMU Driver
Disk Driver
Extended Memory Unit (XMU)
Disk Unit

Figure 3 Structure of the SUPER-UX File System

Table 2 Addressing Capabilities of SFS files

	16blocks/cluster	32blocks/cluster
direct	640 KB	2.5 MB
single indirect	512.6 MB	8.0 GB
double direct	4.0 TB	256.0 TB
triple direct	2^55 B	2^63 B

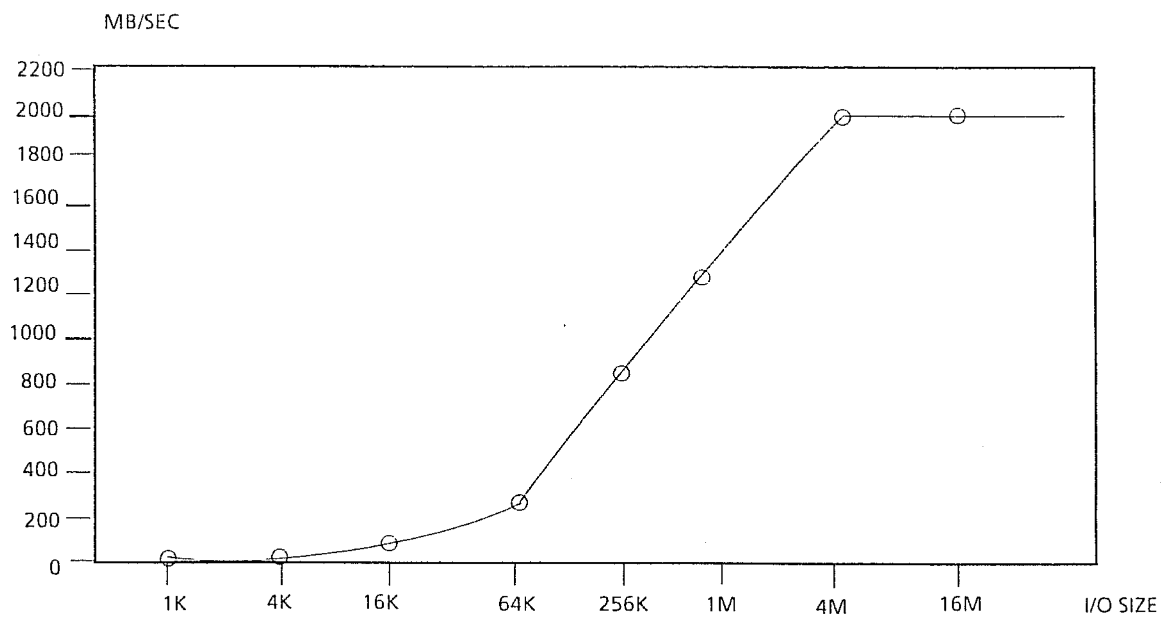


Figure 4 XMU I/O Performance

Figure 4 shows the I/O performance when the SFS is used for files in XMU.

3.2 Languages and Utilities

The main languages under the SUPER-UX are FORTRAN77/SX and C/SX which are the names of Fortran and C of the SX-3. Many utilities are available as useful tools for these languages. The examples are dbx and ANALYZER/SX.

3.2.1 FORTRAN77/SX

In designing the FORTRAN77/SX, we set three objectives; performance, ease of use and portability. These objectives are achieved by its vectorization, optimization and multitasking features, standard-conforming language with many extensions, and other features.

(1) Language

The language is based on the ANSI/ISO FORTRAN 77 and has many extensions such as symbolic names of up to 31 characters, DO WHILE and END DO statements, and INCLUDE statement.

The 3F functions of the standard UNIX and the POSIX Fortran Bindings (P1003.9) which is under standardization by IEEE are also supported.

(2) Automatic vectorization feature

Table 3 summarizes automatic vectorization feature of the FORTRAN77/SX.

Loop unrolling makes it possible to use the vector pipelines fully even if few operations are included in loops. Unrolling with the index of an outer loop is also possible.

Conditional vectorization is the feature to generate both scalar and vector code for a loop. Selection is done at run time.

Figure 5 shows some examples of vectorized loop. They are taken from LINPACK and LFK (Livermore Fortran Kernels) benchmark programs. "V", "X", "W" and "C" on the left side show that the loop is vectorized, interchanged and vectorized, collapsed and vectorized, or conditionally vectorized.

(3) Features to enhance portability from other systems

The SUPER-UX supports the CONVERTER/SX which is a conversion utility of Fortran source programs and data files from other systems. To supplement it, some dialects of other Fortran systems are directly supported in FORTRAN77/SX language. In addition, the following features are supported.

- Floating point data format selection from conventional and extended format
- Word size selection from 32 bits and 64 bits
- Input/output of binary files prepared for workstations

3. 2. 2 ANALYZER/SX

The ANALYZER/SX is a tool to analyze dynamic characteristics of Fortran programs. It gives users the following information about their programs.

- Vector ratio, average loop length
- Timing and call frequency of each subprogram
- MOPS (Million Operations Per Second) and MFLOPS information
- Severity of loss time caused by bank conflict
- Vectorization status
- Execution frequency of each executable statement

Figure 6 shows output listing examples of the ANALYZER/SX.

Table 3 Summary of Automatic Vectorization Feature

BASIC VECTORIZATION		
HIGHLIGHTS	Loops Including	
	IF Statements	
	Vector Subscripts	A(IX(I))=B(IX(I))+C(IX(I))
	Macro Operations	
	Summation	S=S+A(I)
	Inner Product	S=S+A(I)*B(I)
	Product	S=S*A(I)
	Iteration	A(I+1)=X*A(I)+B(I)
	Find Max/Min	Finding Max/Min Value and/or Its Index
	Search	Table Lookup
	Compress/Expand	Compression/Expansion of Matrix
EXTENDED VECTORIZATION		
FOR HIGHER VECTOR RATIO	Loops Including	
	Unvectorizable Statements	⇨ Partial Vectorization
	Data Dependencies	⇨ Statement Interchange Introduction of Work Vectors
	Subprogram Calls	⇨ Inline Expansion
	Unknown Dependencies	⇨ Conditional Vectorization
	Nested Loops	⇨ Vectorization of Outer Loops
FOR EFFICIENT VECTORIZATION	Short Loops with Constant Iteration	⇨ Loop Expansion
	Small Loops with Few Operations	⇨ Loop Unrolling
	Tightly-nested Loops	⇨ Loop Collapsing Loop Interchange
	Loops with Variable Length	⇨ Conditional Vectorization

```
                                do 195 j=1,64
5----------->                   !  do 195 k=1,8
! X--------->                   !  !  b(j,k)=1.ØØØ25
! !                             !  !  c(j,k)=1.ØØØ25
! !                             !  !  h(j,k)=1.ØØØ25
5-X----------           195 continue


                                do 2Ø5 j=1,4
*----------->                   !  do 2Ø5 k=1,512
! W--------->                   !  !  p(j,k)=1.ØØØ25
*-W----------           2Ø5 continue


                                DO 4Ø3 L= 1,MC
V----------->                   !  M= (LR+LR)*(L-1)
!                               !  DO 4Ø1 J= 1,2
! 1Ø-------->                   !  !  DO 4Ø1 K= 1,LR
! ! V------->                   !  !  !  M= M+1
! ! !                           !  !  !  S= REAL(K)
! ! !                           !  !  !  PLAN(M)= R*((S + FW)/S)
! 1ØV--------           4Ø1     !  !  !  ZONE(M)= K+K
V------------           4Ø3 CONTINUE


                                DO 1Ø I = 1,N
C----------->                   !  DY(IY) = DY(IY) + DA*DX(IX)
!                               !  IX = IX + INCX
!                               !  IY = IY + INCY
C------------            1Ø CONTINUE
```

Figure 5 Examples of Vectorized Loops

4. PERFORMANCE DATA

Table 4 shows some performance data of the SX-3/14 for several famous benchmark programs, with performance data of the SX-2.

5. MULTITASKING SUPPORT

Two types of multitasking are supported; macrotasking and microtasking. Macrotasking is a multitasking of a subroutine level and is useful for top-down task partitioning, while microtasking is a multitasking of a loop or a statement level and is useful for bottom-up task partitioning.

This section explains how multitasking is supported by hardware, SUPER-UX kernel, Fortran and utilities in the SX-3.

5.1 Hardware Support

To support multiprocessor and multitasking, the SX-3 hardware has the communication register of 2 K bytes. Several instructions including test-and-set instructions are provided for the register.

Hardware also provides some features to support debugging of multitasked programs. The most important one is instantaneous interruption to other AP's when an exception has occurred or a special instruction has been executed on an AP.

5.2 SUPER-UX Kernel Support

To support multitasking, a "task" which is a light weight process is introduced in the SUPER-UX. Both macrotask and microtask are implemented as tasks.

In scheduling of microtasked program, "family scheduling" is introduced. Tasks which belong to the same microtasking group are treated as a family by the scheduler and controlled so that they are scheduled as close in time as possible.

```
SUMMARY LIST

*------------------------------------------------------------------------*
I ENVIRONMENT                                                            I
I       ANALYZER/SX REVISION    : REV.010                                I
I       ANALYZED DATA           :                                        I
I                                                                        I
I PROGRAM TOTAL INFORMATION                                              I
I       EXECUTION ( CPU ) TIME =   0 :  0 '  4 "   9    (    4009 MSEC)  I
I       INSTRUCTION COUNT      =          136839917                      I
I       VECTOR OPERATION RATIO =  98.61%                                 I
I       MOPS                   =        1445.982                         I
I       MFLOPS                 =         547.947                         I
I       BANK CONFLICT          : NONE                                    I
I                                                                        I
I OVERHEAD INFORMATION                                                   I
I       SUBROUTINE CALL FACTOR =  22.35 MICRO SEC                        I
*------------------------------------------------------------------------*

                                                      V.OP.                                                                    BANK
ATR. PROGRAM   FREQUENCY      TIME(   %  )   AV.TIME  RATIO  AV.V.LEN.    INSTR.   V.INSTR.  V.ELEMENT    MOPS    MFLOPS  CONF.

-->  ANFXRE            1         2(  0.06)        2   0.75      19.0      47855         19        361    19.97      0.42
FNC  PCGVZ            14         0(  0.01)        0   0.00       0.0       7028          0          0    26.69      0.48
SUB  PBCL           1420        76(  1.92)        0  97.94     255.7    2827600     443400  113372800  1505.65      0.11
SUB  ZKZGFG            1        14(  0.37)       14   0.00       0.0     558390          0          0    37.95     14.15
SUB  ZKZ             100       178(  4.45)        1  99.32     256.0    9642500    3494400  894566400  5044.89   2349.23
SUB  SSGGFG            1        39(  0.99)       39  94.81     256.0    1153061      76800   19660800   522.27    173.65
SUB  PSSGQ1          201       346(  8.65)        1  99.42     255.0    6486624    2609591  665515492  1931.12    661.75
SUB  PSSGQ2          201       497( 12.40)        2  98.92     127.8   14230324    5939091  759127460  1542.93    527.08
SUB  PUBGFG            1        38(  0.95)       38   0.00       0.0    1328099          0          0    34.77     12.65
SUB  PUYFXL          200       318(  7.94)        1  98.07     251.0   16321400    2748400  689848400  2208.17    784.31
SUB  OGEGFG            1        84(  2.12)       84   0.00       0.0    3015323          0          0    35.48     12.39
SUB  OGEVK           600      1522( 37.96)        2  97.32      28.0   53704200   30343800  849626400   573.55    208.45
SUB  TZGGFG            1         0(  0.01)        0   0.08       5.0      12663          2         10    32.89     13.35
SUB  TZGEL             2       299(  7.46)      149  98.51     158.5   11476400    3384332  536334840  1820.23    753.67
SUB  RZVGFG            1        43(  1.08)       43   0.11      85.4    1946941         24       2050    44.84     17.98
SUB  RZVG             10       215(  5.38)       21  99.64     245.4    4006750    2122390  520937500  2422.16   1098.41
SUB  ICRGFG            1        50(  1.27)       50   0.00       0.0    1554359          0          0    30.55     13.40
SUB  ICRAGN          400       279(  6.97)        0  99.51     128.0    8520400    5220800  668363200  2402.31    881.02
```

```
ELN     ILN  EXECUTION COST(%)   LOOP            FORTRAN STATEMENT

000980   1        2                                     SUBROUTINE RCRTF( A,N,IA,EPS,IT,IND )
000990   2                                              REAL           A,EPS,TOL,AMAX,GMAX,W,WI,WS1,WS2
001000   3                                              INTEGER  IA,IND,IT,IPIVOT,N,I,J,K,M,MI,M2
001010   4                                              DIMENSION  A( IA,N ),IT( N )
001020   5        2(  0.00)                             IND=0
001030   6        2(  0.00)                             IF( IA.LT.N ) GO TO 100
001040   7        2(  0.00)                             IF( N.LE.0)  GO TO 110
001050   8        2(  0.00)                             IF( N.LE.1 )  GO TO 120
001060   9        2(  0.00)                             AMAX=0.0
001070  10        2(  0.00)                             DO 20 I=1,N
001080  11      150(  0.04)   I----------->             I  DO 10 J=1,N
001090  12    12500(  1.55)   I V--------->             I  I  WS1=ABS(A(I,J))
001100  13    12500(  0.78)   I I                       I  I  IF( WS1.GT.AMAX)  AMAX=WS1
001110  14    12500(  1.94)   I V----------          10 I  CONTINUE
001120  15      150(  0.01)   I------------          20 CONTINUE
001130  16        2(  0.00)                             TOL=EPS*AMAX
001140  17        2(  0.00)                             WRITE(6,3000) TOL
001150  18                                         3000 FORMAT(1H ///20X,4HTOL=,E15.6)
001160  19        2(  0.00)                             DO 90 M=1,N
001170  20      150(  0.01)   3----------->             I  IF( M.LE.1 ) GO TO 50
001180  21      148(  0.01)   I                         I  MI=M-1
001190  22      148(  0.05)   I                         I  DO 40 I=M,N
001191  23                    I                         I  I  *VDIR NODEP(A)
001200  24     6175(  1.92)   I 4--------->             I  I  DO 30 K=1,MI
001210  25   187475( 34.90)   I I V------->             I  I  I  A(I,M)=A(I,M)-A(I,K)*A(K,M)
001220  26   187475(  5.82)   I I V--------          30 I  I  CONTINUE
001230  27     6175(  0.57)   I 4----------          40 I  CONTINUE
001240  28      148(  0.02)   I                         I  CALL OVERFL(J)
001250  29      148(  0.01)   I                         I  IF( J.EQ.1) GO TO 130
001260  30      150(  0.03)   I                      50 I  GMAX=ABS(A(M,M))
001270  31      150(  0.01)   I                         I  IPIVOT=M
001280  32      150(  0.01)   I                         I  M2=M+1
001290  33      150(  0.01)   I                         I  IF( M.GE.N) GO TO 65
001300  34      148(  0.06)   I                         I  DO 60 I=M2,N
001310  35     6175(  0.77)   I V--------->             I  I  WI=ABS(A(I,M))
001320  36     6175(  0.38)   I I                       I  I  IF(WI.LE.GMAX) GO TO 60
001330  37        0(  0.00)   I I                       I  I  GMAX=WI
001340  38        0(  0.00)   I I                       I  I  IPIVOT=I
001350  39     6175(  0.57)   I V----------          60 I  CONTINUE
001360  40      150(  0.01)   I                      65 I  IF( GMAX.LE.TOL) GO TO 140
001370  41      150(  0.01)   I                         I  IF( IPIVOT.EQ.M ) GO TO 75
001380  42        0(  0.00)   I                         I  DO 70  I=1,N
001390  43        0(  0.00)   I V--------->             I  I  W=A(M,I)
001400  44        0(  0.00)   I I                       I  I  A(M,I)=A(IPIVOT,I)
001410  45        0(  0.00)   I I                       I  I  A(IPIVOT,I)=W
001420  46        0(  0.00)   I V----------          70 I  CONTINUE
```

Figure 6 Examples of the ANALYZER/SX Output Listings

Table 4 Performance Data of the SX-3

PROGRAM	SX-3/14	SX-2A

LFK24 Scalar
Vector(A.M.)
Vector(H.M.)
	34.4 MFLOPS
	552.7
	68.1
		15.0 MFLOPS
		212.4
		25.5

LINPACK 100 TPP	219.5 MFLOPS 3897.0	43.0 MFLOPS 885.0
Matrix Multiply (1,024)	5085.6 MFLOPS	1285.1 MFLOPS

To reduce overhead in synchronization or exclusion control in macrotasking, the task scheduler is implemented. It is a light scheduler integrated into the same process as user programs. It is also used in microtasking for task control.

5.3 FORTRAN77/SX

The FORTRAN77/SX supports macrotasking through library calls shown in Table 5. Microtasking is basically supported through automatic parallelization, but explicit parallelization is also possible by using parallelization directives. Table 6 shows the major primitives used in microtasking with directive parameters.

5.4 dbx

The SUPER-UX supports both sdb and dbx as symbolic interactive debuggers, but the main debugger is dbx. Dbx is enhanced to support debugging of multitasked programs. The major enhancements are as follows.

- Specify task to be debugged
- Display task lists
- Display history trace
- Limit the number of processors

5.5 Tools for Tuning of Multitasked Programs

To support program tuning in parallelization as well as vectorization, the SX-3 provides two tools; ANALYZER-P/SX and PARALLELIZER/SX.

5.5.1 ANALYZER-P/SX

The ANALYZER-P/SX analyzes static and dynamic characteristics of Fortran programs. Static analysis information gathered by the ANALYZER-P/SX is as follows.

Table 5 Macrotasking Library Routines

GROUP	FORM	FUNCTION
TASK CONTROL	CALL PTFORK (tid,tparam, subname[,arg-list])	Create a subtask and invoke <u>subname</u>
	CALL PTJOIN (tid-list)	Wait for completion of subtasks
	CALL PTPARAM (tparam)	Get a parameter passed to the subtask

	I = IPTSTAT (tid)
	I = IPTID ( )
	I = IPTNAP ( )
		Get status of the task
		Get the task identifier
		Get the number of available AP's

LOCK CONTROL	CALL PLLOCK (lkid)	Set the lock or wait for the lock cleared
	CALL PLUNLOCK (lkid)	Clear the lock
	CALL PLASGN (lkid[,n])	Allocate a lock
	CALL PLFREE (lkid)	Free the lock

	I = IPLSTAT (lkid)
	I = IPLSTATL (lkid)
		Get status of the lock (and lock)

EVENT CONTROL	CALL PEPOST (evid)	Post the event
	CALL PEWAIT (evid)	Wait for the event posted
	CALL PECLEAR (evid)	Initialize the event
	CALL PEASGN (evid[,n])	Allocate an event
	CALL PEFREE (evid)	Free the event
	I = IPESTAT (evid)	Get status of the event
BARRIER CONTROL	CALL PBASGN (barid, ntasks[,n])	Allocate a barrier and set the number of tasks necessary to break it
	CALL PBSYNC (barid)	Wait until all the other tasks join
	CALL PBFREE (barid)	Free the barrier
	I = IPBSTAT (barid)	Get status of the barrier

Table 6 Microtasking Primitives

PRIMITIVE	FORM	FUNCTION
Parallel Do	PARALLEL DO END DO	Each iteration is executed concurrently
Parallel Case	PARALLEL CASE CASE CASE END CASE	Each case is executed concurrently
Critical Section	CRITICAL SECTION END CRITICAL SECTION	Executed one at a time
Serial Section	SERIAL SECTION END SERIAL SECTION	Executed only once
Barrier	BARRIER	All tasks wait for the last one to arrive

- Task and module structure
- Macrotask library calls and their correspondence
- Inter-procedure data and file reference
- Warning of possible errors such as inconsistency between PTFORK and PTJOIN, PLLOCK and PLUNLOCK, etc.

Dynamic analysis information gathered by the ANALYZER-P/SX is almost the same as that of the ANALYZER/SX, but some information about multitasking is added. The examples are as follows.

- Parallel ratio and parallelization status
- Waiting time for synchronization
- Processor utilization

5.5.2 PARALLELIZER/SX

The PARALLELIZER/SX is an interactive tool which has X-window interface. It accepts queries from users, analyzes the program by using information in the database file output by the ANALYZER-P/SX, and gives the answers to users. It gives users the following information.

- Timing of subprograms, DO loops or blocks of statements
- List of variables which satisfy specified conditions
- Vectorization or parallelization status
- Program structure with timing information
- History trace information
- Multitasking overhead

6. CONCLUSION

Major features of the SX-3 and SUPER-UX are introduced. Though almost all of them are available in the first release, there are some features which will be released in later releases. Multitasking support is included in them.

REFERENCES

Furukatsu, T., T. Watanabe and R. Kondo, 1984: Supercomputer SX System with a Peak Performance of 1.3 GFLOPS and 6 nanosecond Cycle Time. Nikkei Electronics, 356, pp.237-272 (Japanese)

Katayama, H. and M. Tsukagoshi, 1986: FORTRAN and Tuning Utilities aiming at Ease of Use of a Supercomputer. FJCC, pp.1034-1040.

Nishino, H., S. Naka and K. Ikumi, 1989: High Performance File System for Supercomputing Environment. Supercomputing '89.

Watanabe, T., H. Katayama, A. Iwaya, 1986: Introduction of NEC Supercomputer SX System. Supercomputers, Class VI Systems, Hardware and Software, pp.153-167, S. Fernbach (Editor), Elsevier Science Publishers B. V.

Watanabe, T., 1987: Architecture and Performance of NEC Supercomputer. Parallel Computing, 5, pp.247-255.

Watanabe, T., 1990: Advanced Architecture and Technology of the NEC SX-3 Supercomputer. NATO ASI Series, F62, pp.119-128.