SOFTCOM TR 8403-13

(NASA-CR-171549) SPAR IMPROVED STRUCTURAL-FLUID DYNAMIC ANALYSIS CAPABILITY Final Report (Softcom Systems, Inc.) 98 p HC A05/MF A01 CSCL 20D

N85-31449

Unclas
G3/34 21879

SPAR IMPROVED STRUCTURAL/FLUID DYNAMIC ANALYSIS CAPABILITY

FINAL REPORT

CONTRACT NAS8-35772

17 JULY 1985

Prepared for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
GEORGE C. MARSHALL SPACE FLIGHT CENTER
MARSHALL SPACE FLIGHT CENTER, AL 35812

by

M. L. PEARSON

SOFTCOM SYSTEMS, INC.
908 MEMORIAL PARKWAY, NW
HUNTSVILLE, AL 35801

FOREWORD

This final report presents the results of work performed under Contract NAS8-35772 for the National Aeronautics and Space Administration, George C. Marshall Space Flight Center, Huntsville, Alabama. This work was performed by Softcom Systems, Inc., Huntsville, Alabama.

The period of performance for this study was from July 1984 to July 1985. The MSFC Contracting Officer's Representative for this study was Larry A. Kiefling, ED22.

SUMMARY

This report contains the results of a study whose objective was to improve the operation of the SPAR computer code by improving efficiency, user features, and documentation. Additional capability was added to the SPAR arithmetic utility system, including trigonometric functions, numerical integration, interpolation, and matrix combinations. Improvements were made in the EIG processor. A processor was created to compute and store principal stresses in table-format data sets. An additional capability was developed and incorporated into the plot processor which permits plotting directly from table-format data sets. Documentation of all these features is provided in the form of updates to the SPAR users manual, Ref. 1.

CONTENTS

CONTENTS (Concluded)

LIST OF TABLES

1. INTRODUCTION

SPAR (Structural Performance Analysis and Redesign) is a widely used general purpose structural analysis finite element code. SPAR has been developed over the past several years under contract to NASA-Marshall Space Flight Center and NASA-Langley Research Center.

Work performed under this contract represents a continuation of recent development done by Lockheed-Huntsville under contract to NASA-MSFC, (Ref. 2 and 3). The objective of this task was to improve the operation of SPAR by improving efficiency, user features, and documentation.

Additional features were added to the SPAR arithmetic utility system, including trigonometric functions, numerical integration, interpolation, and matrix combinations. Improvements were made in the EIG processor. A processor was created to compute and store principal stresses in table-format data sets. An additional capability was developed and incorporated into the plot processor which permits plotting directly from table-format data sets. The program was also converted to Ascii Fortran.

These updates and additions to the SPAR program were incorporated into a new production version of the code referred to as System Level 16. One significant change in the operation of this version from earlier versions involves the core reset command. The value of the core requested now refers to the actual data space instead of the total value of instructions and data.

This report presents the results of this contract effort. Following this Introduction, Sections 2 through 5 describe the updates and additional capability added to the SPAR processors. Section 6 describes the conversion of the code to Ascii Fortran. Section 7 contains SPAR program file documentation including a listing of all routines with their version designation. A separate listing is provided containing routines added or modified for Level 16.

User documentation is provided in the form of update pages to the SPAR Reference Manual (Ref. 1). These update pages are included in this report as an attachment to Appendix A.

2. AUS PROCESSOR UPDATES

2.1 GENERAL

The SPAR Arithmetic Utility System (AUS) consists of a collection of subprocessors which perform a variety of matrix and other utility functions. Considerable additional capability was added to AUS during this study.

The form of the general arithmetic operation command was expanded from the existing form:

$$Z= \text{Oper}(c_1\ X_1,\ c_2\ X_2,\ \text{---}), \qquad \text{(old)}$$

to allow the following form, except where specifically indicated otherwise:

$$\text{lib}\ Z= c\ \text{Oper}(c_1\ X_1,\ c_2\ X_2,\ \text{---}), \qquad \text{(new)}$$

where lib is a destination library which defaults to the library designated by the last OUTLIB command (or 1 if no OUTLIB command has been given), and c is a floating point constant which defaults to 1.0. Both lib and/or c may be omitted unconditionally.

Brief descriptions of the subprocessors added to AUS are given in the sections below. Detailed descriptions are given in Appendix A.

2.2 ARITHMETIC FUNCTIONS

The following arithmetic and trigonometric subprocessors were added to AUS: COS, ACOS, SIN, ASIN, TAN, ATAN, ATN2, COSH, SINH, TANH, EXP, ALOG, AL10, ABS, IFIX, FLOAT, POWER, and SRSS.

The form of COS, ACOS, SIN, ASIN, TAN, ATAN, COSH, SINH, TANH, EXP, ALOG, AL10, and ABS is: lib Z= c OPER(cx X).

The form of IFIX is: lib Z= IFIX(cx X).

The form of FLOAT is: lib Z= c FLOAT(X).

The form of ATN2 is: lib Z= c ATN2(cx X, cy Y).

The form of POWER is: lib Z= c POWER(cx X, p),
where $z = c*(cx*x)^p$.

The form of SRSS is: lib Z= c SRSS(cx X, cy Y),
where $z = c*SQRT[(cx*x)^2 + (cy*y)^2]$.

2.3 NUMERICAL INTEGRATION

A numerical integration routine, NUM1, was added to AUS. The general form of NUM1 is:

lib Z= c NUM1(cx X, cy Y),

where X is a single-block data set containing n abscissa values, and Y is a multi-block data set consisting of m blocks containing n ordinate values each. The data set produced consists of one block containing m values derived by straight-line integration.

2.4 INTERPOLATION

A series of interpolation routines, XNT1, XNT2, XNT3, XNT4, were added to the utility system. The form of these routines is:

lib Z= XNT1(XY,A),

where XY is a single-block data set containing n pairs of real numbers, (x_i,y_i), defining a piecewise linear function of X. A contains m real numbers representing abscissa values for which y values are to be determined. The output, Z, contains m ordinate values corresponding to the abscissa values in A.

XNT2 is similar to XNT1 except that straight-line interpolation is performed assuming logarithmic (base 10) x and y. XNT3 assumes linear x and logarithmic y. XNT4 assumes logarithmic x and linear y.

2.5 MATRIX COMBINATIONS

Matrix multiplication routines, CBR, CBD, ACBR, and ACBD, were added to AUS. The form of these routines is:

lib Z= CBR(X,Y),

where X is a multiblock data set representing a rectangular matrix. Y may be single or multiblock. CBR performs the matrix product of X and Y. CBD is used for the special case where Y is a single-block data set representing a diagonal matrix.

ACBR and ACBD perform the same functions as CBR and CBD except that each number in the data set X is replaced by its absolute value before the multiplication takes place.

2.6 CORE REQUIREMENT REDUCTION

AUS normally requires enough central memory to hold at least one block of each data set being operated on. The capability for handling these data set blocks in segments to reduce the core requirement was implemented in AUS during this contract. This is especially important when working with large single-block arrays.

This function is automatic, requiring no user action. When not enough core is available to permit whole blocks to be loaded, the arrays are loaded in segments, the lengths of which are determined by the available memory, and operated on accordingly.

This feature was implemented for the following commands: RECIP, SQUARE, SQRT, COS, ACOS, SIN, ASIN, TAN, ATAN, COSH, SINH, TANH, EXP, ALOG, AL10, ABS, FLOAT, IFIX, SUM, and PRODUCT.

3. EIG PROCESSOR UPDATES

3.1 GENERAL

Several areas of the EIG processor were looked into during this study. Some output format changes were made. The eigenvalue summary printout was modified to include the appropriate heading, i.e., FREQ (HZ) for a vibrational solution, and BUCK FACT for a buckling solution. The format width was also increased to allow space between columns.

An alternate core utilization table was added to the EIG printout which tabulates the core required versus the number of vectors which may be held in core at one time and the number of passes required to process all vectors. This provides the user with information which may be used to determine possible core resets for minimizing I/O activity for a particular problem.

3.2 REDUNDANT VECTORS

One of the problems which occurs in EIG periodically is the appearance of a dependent (or redundant) system vector in the Rayleigh-Ritz procedure producing a negative determinant in the Cholesky reduction process. This causes an error condition resulting in termination of the EIG execution.

A procedure was developed and implemented under this contract which automatically discards the redundant vector and continues processing. No user interaction is required.

4. PSR PROCESSOR

A processor was developed and incorporated into SPAR which reads multiblock, table-format stress data sets and computes and stores principal stresses in similar data sets. This processor, PSR, is applicable to two-dimensional element types E31-E33 and E41-E43, and three-dimensional solid element types S41-S81.

The order of stress quantities in the data sets produced by PSR for 2-d element types are as follows: 1) ANG, 2) MAX PS, 3), MIN PS, 4) MAX SHR, and 5) SEFF, effective stress.

The order of stress quantities for 3-d solid element types are as follows: 1) NS1, 2) NS2, 3) NS3, 4) SS1, 5) SS2, 6) SS3, 7) ONS, and 8) OSS, octahedral shear stress.

The PSR processor is executed as follows:

```
@XQT PSR
 "etype"
   or
 "etype"   n3   n4
```

Examples:

```
@XQT PSR
 E43                (reads    ES    E43   mask   mask)
                    (creates  PSTR  E43   n3     n4   )
       or

 E33   1   2        (reads    ES    E33   1       2   )
                    (creates  PSTR  E33   1       2   )
```

5. PLOT PROCESSOR UPDATES

5.1 GENERAL

Several items involving the plot processors were incorporated into SPAR Level 16. The improvements made to the Tektronix version of PLTB (PLTB/TEK) during the previous contract (Ref. 3) were included, as was the laminate stress display capability also developed during the previous effort. Update pages to the SPAR Reference Manual describing these features are included in Appendix A.

The PLTB (FR-80) and the PLTB/TEK (Tektronix) routines were consolidated into a single file (RP) for Level 16. The routines which are configured for operation on the Tektronix were given "TEK" version designations for both the symbolic and relocatable elements. The @Map symbolics for both PLTB and PLTB/TEK were updated to include the relocatable version names where necessary to avoid ambiguity.

5.2 TABLE-FORMAT DATA SETS

The capability for plotting from table-format stress data sets was developed and incorporated into both PLTB and PLTB/TEK. This permits the user to plot any type stress (or any other quantity) which is contained in a table-format data set. The data set may contain either one value per element, assumed to be at the center, values at each of the nodes, or values at each of the nodes plus the center of the element (NNODES+1 values). Stress displays may be created for 2-node, 3-node, or 4-node elements. Depending on the number of values contained in the data set per element, stresses will be displayed at the center of the element, at the corners, or both, accordingly.

The data sets may be created in AUS and must have names of the following form:

"name1" "etype" iset ng

where,

"name1" is any name supplied by the user, which may describe the quantity contained in the data set,

"etype" is a valid 2-d element type (E21,E23,E24,E31, E32,E33,E41,E42,E43,E44),

iset is supplied by the user and may correspond to a load set designation, and

ng is the element group number to which the data set corresponds.

Example: ES E43 1 1

A separate data set must be constructed for each group of each element type which is to be plotted. Descriptive information for frame labelling purposes may be placed in a data set named:

TABL TITL iset mask

where iset refers to the iset value in the stress data set names. If such a data set is present, the contents (up to 60 characters) will be displayed at the top of the plot frame.

The data set plotting is invoked in PLTB or PLTB/TEK with the DISPLAY command as follows:

DISPLAY=TABLe NAME1 "etype" iset

where, NAME1 "etype" iset, refers to the first three names of the data set desired to be plotted. The fourth name, ng, is not required on the DISPLAY command since a plot specification may contain elements from different groups, and the data set corresponding to the group designation of the elements being plotted is read automatically.

6. ASCII FORTRAN CONVERSION

6.1 DESCRIPTION

The SPAR code was converted to Ascii Fortran and compiled with the Ascii (FTN) compiler. The principal features of the conversion approach were as follows:

- o Use the existing assembly language routines.
- o Read card images in Fieldata using the existing card reader logic.
- o Convert alphameric data to Ascii before using in the program.
- o Keep date and time in Fieldata since their format requires 6 characters and changing this would affect the data set storage allocation.
- o Store data set names in Fieldata.
- o Store type 4 (alphameric) data sets in Fieldata.

This conversion strategy provides many benefits. Some key aspects are listed below:

- o Full data set compatibility is retained between the Ascii version of SPAR and the Fortran V version. This means that existing SPAR and EAL libraries can be read with the Ascii version and vice versa.
- o Most of the code changes involving Ascii/Fieldata conversion are confined to a few routines.
- o The impact on the SPAR data set structure and storage allocation is minimal, meaning no broad code changes are needed that would require extensive checkout.

6.2 RESULTS

Several test cases were executed with no problems. The FTN compilations were made using the "Z" and "E" options so that fully optimized code would be produced. CPU times, however, were found to be significantly longer for the Ascii version compared with the Fortran V (FOR) version.

Table 6-1 shows run time comparisons between the Fortran V and Ascii SPAR versions for a 1001-node plate problem. As seen from the table the CPU time relationship varies from processor to processor, with the largest increase occurring in INV.

Examinations of the code produced by the FOR and FTN compilers for the key routine in the INV processor show this to be the result of much less efficient code generated by the FTN compiler. Since the I/O times are essentially identical for the two versions, the SUP times show a much smaller increase. The numerical results agreed to 8 places in all cases.

Table 6-1

SPAR ASCII/FORTRAN V RUN TIME COMPARISON

PLATE PROBLEM 25X40 GRID (1001 NODES)

	FOR	FTN	%
TAB	3.233	3.119	-3.5
ELD	2.139	1.855	-13.3
TOPO	15.180	17.488	15.2
E	2.597	2.787	7.3
EKS	67.919	78.687	15.9
M	32.057	32.682	1.9
K	11.866	12.256	3.3
INV	144.963	199.235	37.4
EIG	160.998	195.015	21.1
Total CPU	441.005	543.182	23.2
Total SUPs	1547.35	1655.84	7.0

7. PROGRAM FILE DOCUMENTATION

This section contains information on the SPAR program file contents, processor file requirements, and subroutine/processor cross reference data for System Level 16.

Table 7-1 lists the SPAR routines which were modified for System Level 16.

Table 7-2 lists each SPAR program file by number (order in which it resides on the tape), along with its file name and the processor main programs it contains.

Table 7-3 lists the SPAR processors in alphabetical order along with the file containing the MAP symbolic element, the file containing the main program, and other files (if any) required for collecting (@MAP'ing) that processor.

Table 7-4 lists the SPAR routines in alphabetical order by subroutine name. Main programs are listed by processor name preceded by MP, e.g. MPAUS. For each routine, the name of the file containing the routine, names of the symbolic and relocatable elements, and the processors which use the routine are listed.

Table 7-1

SPAR SYSTEM LEVEL 16 ROUTINES

Name	File	Symbolic	Relocatable	Processors	
CBRD	R9	CBRD/16	CBRD/M	AUS	(new)
CLVT	R9	CLVT/16	CLVT/M	EIG	
DPCHOL	R9	DPCHOL/16	DPCHOL/M	EIG	
EIGEX	R9	EIGEX/16	EIGEX/M	EIG	
EIGLD	R9	EIGLD/16	EIGLD/M	EIG	
KEXP	R2	DYNEXP/16	DYNEXP/M	*All	
LDPLTB	RP	LDPLTB/16	LDPLTB/M	PLTB	
MPAUS	R9	MPAUS/16	MPAUS/M	AUS	
MPDCU	R2	MPDCU/16	MPDCU/M	DCU	
MPPLTB	RP	MPPLTB/16	MPPLTB/M	PLTB	
MPPSR	R9	MPPSR/16	MPPSR/M	PSR	(new)
NODVAL	RP	NODVAL/16	NODVAL/M	PLTB	(new)
NUM1	R9	NUM1/16	NUM1/M	AUS	(new)
P3CALC	R9	P3CALC/16	P3CALC/M	PSR	(new)
PCALC	R9	PCALC/16	PCALC/M	PSR	(new)
POWER	R9	POWER/16	POWER/M	AUS	(new)
PWR	R9	PWR/16	PWR/M	AUS	
REDUCE	R9	REDUCE/16	REDUCE/M	EIG	(new)
RSET	R2	RSET/16	RSET/M	*All	
SDPLAY	RP	SD02/16	SD02/M	PLTB	
SEGMPY	R9	SEGMPY/16	SEGMPY/M	AUS	(new)
SLABL	RP	SD05/16	SD05/M	PLTB	
SPROD	R9	SPROD/16	SPROD/M	AUS	
SSUM	R9	SSUM/16	SSUM/M	AUS	
SVV	R9	SVV/16	SVV/M	AUS	
XATN2	R9	XATN2/16	XATN2/M	AUS	(new)
XNTI	R9	XNTI/16	XNTI/M	AUS	(new)
XPRNT	R9	XPRNT/16	XPRNT/M	EIG	

Table 7-2

SPAR PROGRAM FILE CONTENTS

File Seq #	File Name	Processors Contained (main programs)					
1	SPAR16						
2	R2	DCU					
3	R3	TAB	ELD				
4	R4	TOPO	PAMAP	PKMAP	STRP		
5	R5	E	EKS	PRTE	MN		
6	R6	K	KG	INV	M	PS	FSM
7	R7	EQNF	SSOL	VPRT	DR		
8	R8	GSF	PSF				
9	R9	AUS	EIG	PSR			
10	RA	SSBT	STRP	SYN			
11	RC	CEIG					
12	RJ						
13	RM	SM					
14	RP	PLTA	PLTB	PXY	PLTB/TEK		
15	T1						
16	T2	TGEO					
17	T3	SSTA					
18	T4	TRTA					

Table 7-3

SPAR PROCESSOR CROSS REFERENCE

Processor	File Containing Map Symbolic	File Containing Main Program	Other Files Required
AUS	R2	R9	R7,R8
CEIG	R2	RC	R7,R9,RJ
DCU	R2	R2	
DR	R2	R7	
E	R2	R5	R7,RJ
EIG	R2	R9	R7
EKS	R2	R5	RJ
ELD	R2	R3	
EQNF	R2	R7	R6,R8
FSM	R2	R6	
GSF	R2	R8	R7
INV	R2	R6	
K	R2	R6	RJ
KG	R2	R6	
M	R2	R6	RJ
MN	R2	R5	R7,RJ
PAMAP	R2	R4	
PKMAP	R2	R4	
PLTA	R2	RP	R5
PLTB	R2	RP	R7,R8
PRTE	R2	R5	
PS	R2	R6	
PSF	R2	R8	
PSR	R2	R9	
PXY	R2	RP	
SM	R2	RM	R5,R6,RC,RJ
SSBT	R2	RA	R8
SSOL	R2	R7	R9
SSTA	T3	T3	R2,T1
STRP	R2	RA	R4
SYN	R2	RA	
TAB	R2	R3	R5,R7
TGEO	T2	T2	R2,T1
TOPO	R2	R4	
TRTA	T4	T4	R2,T1
VPRT	R2	R7	

Table 7-4

SPAR SUBROUTINE CROSS REFERENCE

Name	File	Symbolic	Relocatable	Processors
ABA	R3	F1T09/15	F1T09/M	TAB
ABC	R3	F1T11/9	F1T11/M	TAB
ABD	R3	F1T12/7	F1T12/M	TAB
ACON	R3	F1T15/10	F1T15/M	TAB
ADDF36	R7	ADDF36/12	ADDF36/M	EQNF MN
ADDH	R6	ADDH/3	ADDH/M	K SM
ADSK	R3	F1T10/10	F1T10/M	TAB
ADVANC	RP	ADVANC/15	ADVANC/M	PXY
AFEX	R6	AFEX/6	AFEX/M	INV
AFGO	R6	AFGO/7	AFGO/M	INV
AFLD	R6	AFLD/9	AFLD/M	INV
AGEN	RJ	AGEN/9	AGEN/M	EKS SM
AJREF	R3	F1T06/9	F1T06/M	TAB
ALEIJ	R7	ALEIJ/9	ALEIJ/M	EQNF
ALFCNT	RP	ALFCNT/15	ALFCNT/M	PXY
ALIO	R2	ALIO/9	ALIO/M	DR
AMAT	R3	F1T02/11B	F1T02/M	TAB
AMREF	R3	F1T07/10	F1T07/M	TAB
ANSW	R3	F1T03/7	F1T03/M	TAB
AQ	R3	F1T04/10	F1T04/M	TAB
AQJJT	R3	AQJJT/7	AQJJT/M	TAB
ARGS	R2	ARGS/9	ARGS/M	AUS DR
ARL2	R3	F1T08/7	F1T08/M	TAB
ARMASS	R3	F1T18/12	F1T18/M	TAB
ASA	R3	F1T13/11	F1T13/M	TAB
ASB	R3	F1T14/7	F1T14/M	TAB
ASET2	RC	ASET2/12	ASET2/M	CEIG
ASG	R2	ASG/3TRK	ASG/M	*All
ASKEX	R6	ASKEX/12	ASKEX/M	K
ASKGEX	R6	ASKGEX/13	ASKGEX/M	KG
ASKGGO	R6	ASKGGO/9	ASKGGO/M	KG
ASKGO	R6	ASKGO/7	ASKGO/M	K
ASMEX	R6	ASMEX/15	ASMEX/M	M
ASMGO	R6	ASMGO/7	ASMGO/M	M
ASMQJ	R3	ASMQJ/7	ASMQJ/M	TAB
ATD	RJ	ATD/9	ATD/M	EKS SM
ATEXT	R3	F1T01/10	F1T01/M	TAB
BACKSL	R7	BACKSL/8	BACKSL/M	AUS SSOL EIG CEIG
BARKG	R6	BARKG/7	BARKG/M	KG

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
BEAMKG	R6	BEAMKG/5	BEAMKG/M	KG
BEAMT	R6	BEAMT/5	BEAMT/M	M KG EQNF
BEGIN	RP	BEGIN/15	BEGIN/M	PXY
BFLUSH	RP	BFLUSH/11	BFLUS	PLTB
BFLUSH	RP	BFLUSH/15	BFLUSH/15	PXY
BIGLAB	RP	L6010/V70E	L6010/M	PLTB
BLIO	RA	BLIO/10	BLIO/M	SSBT
BLKDAT	R2	BDAL/7	BDAL/M	*All
BNF	R9	BNF/8	BNF/M	AUS
BTA	R7	BTA/9	BTA/M	DR
BTA1	R7	BTA1/9	BTA1/M	DR
BTA2	R7	BTA2/9	BTA2/M	DR
BTA3	R7	BTA3/9	BTA3/M	DR
BTB	R7	BTB/9	BTB/M	DR
BTX	R7	BTX/9	BTX/M	DR
BW02	RJ	BW02/13	BW02/M	EKS MN
BW03	RJ	BW03/12	BW03/M	EKS MN
CARDSA	R3	CARDSA/11	CARDSA/M	TAB
CBABK2	RC	CBABK2/12	CBABK2/M	CEIG
CBAL	RC	CBAL/12	CBAL/M	CEIG
CBRD	R9	CBRD/16	CBRD/M	AUS
CLVT	R9	CLVT/16	CLVT/M	EIG
CMEXPE	R6	CMEXPE/12	CMEXPE/M	M
COMHES	RC	COMHES/12	COMHES/M	CEIG
COMLR2	RC	COMLR2/12	COMLR2/M	CEIG
COP	R2	COP/6	COP/M	DCU
CORCHK	R2	CORCHK/6	CORCHK/M	DCU
CPUTIM	R2	CPUTIM/1	CPUTIM/M	*All
CRDPLT	RP	L6015/V70I	L6015/M	PLTB
CRRITZ	RC	CRRITZ/12	CRRITZ/M	CEIG
CUBIC	RP	L6011/V70E	L6011/M	PLTB
CXA	RJ	CXA/9	CXA/M	EKS SM
CXTYD	RC	CXTYD/12	CXTYD/M	CEIG
CXTYG	RC	CXTYG/12	CXTYG/M	CEIG
CYLQ	R5	F32A1/1	F32A1/M	TAB E PLTA
DAL	R2	DAL/9	DAL/M	*All
DASKEX	R6	DASKEX/12	DASKEX/M	K
DATIM	R2	DATIM/7	DATIM/M	*All
DAX	R2	DAX/9	DAX/M	DR
DCROW	R8	DCROW/12	DCROW/M	PSF
DECODE	R2	DECODE/1	DECODE/M	TAB INV AUS SYN SSBT VPRT
DEIGEN	R9	DEIGEN/5	DEIGEN/M	EIG
DEL	R2	DEL/7	DEL/M	TOPO DCU
DELOC	RP	DELOC/1	DELOC/M	PLTA
DEVICE	R2	DEVICE/1	DEVICE/M	*All

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
DI1	R9	DI1/9	DI1/M	AUS
DIRCOS	R5	F32A4/1	F32A4/M	E
DIRX	RC	DIRX/12	DIRX/M	CEIG
DMEXPE	R5	DMEXPE/12	DMEXPE/M	E
DMFORM	RM	DMFORM/13	DMFORM/M	SM
DMTEX	RC	DMTEX/14	DMTEX/M	CEIG SM
DMULT	R7	DMULT/8	DMULT/M	AUS SSOL EIG
DMULTX	R7	DMULTX/8	DMULTX/M	AUS SSOL EIG
DOTTED	RP	L6020/V70J	L6020/M	PLTB
DOTV	RP	DOTV/M	DOTV/M	PLTB
DPAFEX	R6	DPAFEX/8	DPAFEX/M	INV
DPC	RM	DPC/12	DPC/M	SM
DPCHOL	R9	DPCHOL/16	DPCHOL/M	EIG
DPN	RM	DPN/12	DPN/M	SM
DPTRN3	R6	DPTRN3/12	DPTRN3/M	K
DPTRN6	R6	DPTRN6/12	DPTRN6/M	K
DPX	RM	DPX/13	DPX/M	SM
DSCALE	RP	GPF04/V70I	GPF04/M	PLTA
DSGO	R7	DSGO/9	DSGO/M	SSOL
DSLD	R7	DSLD/9	DSLD/M	SSOL
DSMUL	RC	DSMUL/12	DSMUL/M	CEIG SM
DSUM	RJ	DSUM/11	DSUM/M	EKS SM MN
DSX	R7	DSX/9	DSX/M	SSOL
DTEX	R7	DTEX/10	DTEX/M	DR
DTX1	R7	DTX1/9	DTX1/M	DR
ECHO	R2	ECHO/8	ECHO/M	*All
EIGEX	R9	EIGEX/16	EIGEX/M	EIG
EIGGO	R9	EIGGO/7	EIGGO/M	EIG
EIGLD	R9	EIGLD/16	EIGLD/M	EIG
EIGSOL	RA	EIGN/8	EIGN/M	STRP
EISPAK	RC	EISPAK/13	EISPAK/M	CEIG
ELCON	R4	F4C1/1	F4C1/M	TOPO
ELDA	R9	ELDA/14	ELDA/M	AUS
ELDATA	R9	ELDATA/12	ELDATA/M	AUS
ELEFIL	RP	ELEFIL/1	ELEFIL/M	----
ELEPLT	RP	L606/12	L606/M	PLTB
ELESTR	RP	GPF03/V70L	GPF03/M	PLTA
ELSORT	R4	F4A/13	F4A/M	TOPO
ELSUB	R4	F4C2/1	F4C2/M	TOPO
ENCODE	R2	ENCODE/1	ENCODE/M	TAB SYN
ENUMBR	RP	L6019/V70K	L6019/M	PLTB
ERABT	R2	ERABT/6	ERABT/M	*All
ERMSG1	R2	ERMSG1/8	ERMSG1/M	*All
EVCHEK	R9	EVCHEK/5	EVCHEK/M	EIG
EXCON	R4	F4C3/1	F4C3/M	TOPO

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors			
EXDEM	R5	EXDEM/15	EXDEM/M	E			
EXDKDM	RM	EXDKDM/13	EXDKDM/M	SM			
EXEQNF	R7	EXEQNF/13	EXEQNF/M	EQNF			
EXPLCT	RA	EXPLCT/11	EXPLCT/M	SYN			
EXPND1	R9	EXPND1/8	EXPND1/M	EIG			
EXPXY	RP	EXPXY/15	EXPXY/M	PXY			
EXVPRT	R7	EXVPRT/9	EXVPRT/M	VPRT			
EXXMAP	R6	EXXMAP/11	EXXMAP/M	XMAP			
F1E1	R3	F1E1/14	F1E1/M	ELD			
F1T15A	R3	F1T15A/9	F1T15A/M	TAB			
F32A	R5	F32A/15	F32A/M	E			
F34A	R5	F34A/14	F34A/M	E			
F3A	R5	F3A/14	F3A/M	E			
F3B	R5	F3B/14	F3B/M	E			
F3DX	R5	F3DX/12	F3DX/M	E			
F3GO	R5	F3GO/12	F3GO/M	E			
F3KEX	R5	F3KEX/15	F2KEX/M	EKS			
F3KGO	R5	F3KGO/12	F3KGO/M	EKS			
F3KLD	R5	F3KLD/12A	F3KLD/M	EKS			
FEUSE	RA	FEUSE/9	FEUSE/M	SYN			
FIL3	R7	FIL3/12	FIL3/M	EQNF			
FILER	RJ	FILER/12	FILER/M	EKS	MN		
FIN	R2	FIN/10	FIN/M	*All			
FINAL	RJ	FINAL/12	FINAL/M	MN			
FINSYN	RA	FINSYN/11	FINSYN/M	SYN			
FJF	R7	FJF/15	FJF/M	EQNF			
FLDEF	R3	F1E3/14	F1E3/M	ELD			
FORWRD	R7	FORWRD/8	FORWRD/M	AUS	SSOL	EIG	CEIG
FRAMEV	RP	TEK	TEK	----			
FRHEX	RJ	FRHEX/15	FRHEX/M	MN			
FRI	RC	FRI/12	FRI/M	CEIG			
FRMAVN	RP	L601D/V70J	L601D/M	PLTB			
FS3D	RJ	FS3D/15	FS3D/M	MN			
FSBTMP	RA	FSBTMP/8	FSBTMP/M	STRP			
FSUBMP	RA	FSUBMP/8	FSUBMP/M	STRP			
G3D	R5	G3D/12	G3D/M	E			
GAUSS1	RJ	GAUSS1/12	GAUSS1/M	E	EKS	M	MN
GCYLQ	R5	F32A2/1	F32A2/M	TAB	E	PLTA	
GE2D	R5	GE2D/14	GE2D/M	E			
GE3D	R5	GE3D/14	GE3D/M	E			
GEFACE	R5	GEFACE/12	GEFACE/M	E			
GELDG	R5	GELDG/12	GELDG/M	E			
GESMRY	R5	GESMRY/12	GESMRY/M	E			
GGSGO	RP	GGSGO/9	GGSGO/M	PLTA			
GGSLD	RP	GGSLD/12	GGSLD/M	PLTA			

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
GIDA	R2	GIDA/9	GIDA/M	DR PXY
GK2DP	R6	GD2DP/4	GD2DP/M	K
GL	R8	GL/4	GL/M	AUS EQNF GSF SSBT
GOALFA	RP	GOALFA/15	GOALFA/M	PXY
GOALPH	RP	GOALPH/11	GOALP	PLTB
GOCEIG	RC	GOCEIG/12	GOCEIG/M	CEIG
GODEM	R5	GODEM/8	GODEM/M	E
GOEQNF	R7	GOEQNF/9	GOEQNF/M	EQNF
GOFGKM	R6	GOFGKM/11	GOFGKM/M	FSM
GOSMA	RM	GOSMA/13	GOSMA/M	SM
GOSMB	RM	GOSMB/12	GOSMB/M	SM
GOSMC	RM	GOSMC/14	GOSMC/M	SM
GOSMX	RM	GOSMX/12	GOSMX/M	SM
GOSMX1	RM	GOSMX1/12	GOSMX1/M	SM
GOSMX2	RM	GOSMX2/12	GOSMX2/M	SM
GOSMX3	RM	GOSMX3/12	GOSMX3/M	SM
GOSS	RA	GOSS/11	GOSS/M	SYN
GOSTRP	RA	GOSTRP/9A	GOSTRP/M	STRP
GOSYN	RA	GOSYN/11	GOSYN/M	SYN
GOVEC	RP	GOVEC/15	GOVEC/M	PXY
GOXMAP	R6	GOXMAP/11	GOXMAP/M	XMAP
GPFCON	[illegible]	GPF02/12	GPF02/M	PLTA
GPLXQT	RP	GPLXQT/10	GPLXQT/M	PLTB
GQM	R6	GQM/5	GQM/M	KG
GQP	R6	GQP/5	GQP/M	KG
GSBTMP	RA	GSBTMP/8	GSBTMP/M	STRP
GSFEX	R8	GSFEX/15	GSFEX/M	GSF
GSFLD	R8	GSFLD/11	GSFLD/M	GSF
GSUBMP	RA	GSUBMP/8	GSUBMP/M	STRP
GTM	R6	GTM/5	GTM/M	KG
GTP	R6	GTP/5	GTP/M	KG
HAFMPY	RA	HAFMPY/8	HAFMPY/M	STRP
HAFTMP	RA	HAFTMP/8	HAFTMP/M	STRP
HEXNL	RJ	HEXNL/15	HEXNL/M	MN
HFB1	R6	HFB1/11	HFB1/M	FSM
HGEN	RJ	HGEN/11	HGEN/M	EKS SM
HGEND	RJ	HGEND/11	HGEND/M	EKS SM
HMBGEN	RJ	HMBGEN/11	HMBGEN/M	EKS SM
HOUSE	RA	HAS/10	HAS/M	STRP
HQT	R6	HQT/1	HQT/M	K SM
HSBTMP	RA	HSBTMP/8	HSBTMP/M	STRP
HSUBMP	RA	HSUBMP/8	HSUBMP/M	STRP
IAM	RA	IAM/8	IAM/M	STRP
ICSF	R2	ICSF/2	ICSF/M	*All
IDCODE	R2	IDCODE/8	IDCODE/M	DCU

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors		
IDCOM	R2	IDCOM/7	IDCOM/M	TAB	ELD	
IDCOM1	R2	IDCOM1/8	IDCOM1/M	AUS	VPRT	
IFONT	RP	IFONT/15	IFONT/M	PXY		
INCARD	R3	INCARD/7	INCARD/M	TAB		
INEXT	R9	INEXT/9	INEXT/M	AUS		
INF1	R2	INF1/6	INF1/M	*All		
INITAL	RJ	INITAL/12	INITAL/M	EKS	MN	
INLIB	RP	INLIB/1	INLIB/M	PLTA		
INLOC	RP	INLOC/1	INLOC/M	PLTA		
INMK	RA	INMK/10	INMK/M	STRP		
INNLSS	R5	INNLSS/15	INNLSS/M	MN		
INSTS	R8	INSTS/9	INSTS/M	GSF		
INTCA	R7	INTCA/9	INTCA/M	DR		
INVT3	RJ	INVT3/15	INVT3/M	MN		
INZ	R4	INZ/4	INZ/M	TOPO	STRP	
IO	RA	IO/9	IO/M	STRP		
ISBTMP	RA	ISBTMP/8	ISBTMP/M	STRP		
ISUBMP	RA	ISUBMP/8	ISUBMP/M	STRP		
ITHEX	RJ	ITHEX/13	ITHEX/M	EKS		
ITHEXM	RJ	ITHEXM/12	ITHEXM/M	E	M	
ITQUAD	RJ	ITQUAD/11	ITQUAD/M	EKS	SM	MN
ITTET	RJ	ITTET/12	ITTET/M	EKS		
IVT	R6	IVT/5	IVT/M	M	KG	
JF2	R8	JF2/9	JF2/M	GSF		
JNEXT	R9	JNEXT/9	JNEXT/M	AUS		
JRMK	R4	JRMK/8	JRMK/M	TOPO		
JTCSTR	RA	JTCSTR/11	JTCSTR/M	SYN		
K21	R5	K21/15	K21/M	EKS	SM	
K24	R5	K24/1	K24/M	EKS	SM	
K2D	RJ	K2D/11A	K2D/M	EKS	SM	
K3D00	RJ	K3D00/13	K3D00/M	EKS	MN	
K3D01	RJ	K3D01/13	K3D01/M	EKS		
K3DNL	RJ	K3DNL/15	K3DNL/M	MN		
KALPH	R2	KALPH/7	KALPH/M	*All		
KEXP	R2	DYNEXP/16	DYNEXP/M	*All		
KEXPE	R5	KEXPE/12	KEXPE/M	EKS		
KG34	R6	KG34/10	KG34/M	KG		
KGEXPE	R6	KGEXPE/12	KGEXPE/M	KG		
KLAM	R3	KLAM/11	KLAM/M	TAB		
KLER1	RP	L6014/V70E	L6014/M	PLTB		
KLVT	RA	KLVT/10	KLVT/M	STRP		
KMAP	R4	F4C/12	F4C/M	TOPO		
KOP	R2	KOP/7	KOP/M	DCU		
LABELH	RP	LABELH/15	LABELH/M	PXY		
LABELV	RP	LABELV/15	LABELV/M	PXY		

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
LABL	RP	L608/10	L608/M	PLTB
LADJ	R2	LADJ/6	LADJ/M	*All
LAM	R3	LAM/11	LAM/M	TAB
LCARD	R3	LCARD/14	LCARD/M	ELD
LCBD	RC	LCBD/12	LCBD/M	CEIG
LCBG	RC	LCBG/12	LCBG/M	CEIG
LDCEIG	RC	LDCEIG/12	LDCEIG/M	CEIG
LDDEM	R5	LDDEM/8	LDDEM/M	E
LDDKDM	RM	LDDKDM/13	LDDKDM/M	SM
LDEQNF	R7	LDEQNF/12	LDEQNF/M	EQNF
LDFGKM	R6	LDFGKM/12	LDFGKM/M	FSM
LDPLTB	RP	LDPLTB/16	LDPLTB/M	PLTB
LDSM	RM	LDSM/13	LDSM/M	SM
LDSS	RA	LDSS/11A	LDSS/M	SYN
LDSTRP	RA	LDSTRP/10	LDSTRP/M	STRP
LDSYN	RA	LDSYN/11	LDSYN/M	SYN
LDXMAP	R6	LDXMAP/11	LDXMAP/M	XMAP
LG	R8	LG/4	LG/M	AUS EQNF GSF SSBT
LGGL	R9	LGGL/10	LGGL/M	AUS
LINE1	RP	LINE1/15	LINE1/M	PXY
LINPLT	RP	L609/V70L	L609/M	PLTB
LIO	R2	LIO/8	LIO/M	*All
LOCATE	RA	LOCATE/8	LOCATE/M	SYN
LOCMK	RA	LOCMK/11	LOCMK/M	SYN
LSTRAN	R6	LSTRAN/5	LSTRAN/M	----
LTOC	R2	LTOC/7	LTOC/M	*All
M32	R6	M32/5	M32/M	M
M33	R6	M33/5	M33/M	M
M34	R6	M34/5	M34/M	M
M3D01	RJ	M3D01/12	M3D01/M	E M
M62	R6	M62/5	M62/M	M
M62CUR	R6	M62CUR/15	M62CUR/M	M
M63	R6	M63/5	M63/M	M
M64	R6	M64/5	M64/M	M
MAJTYP	RP	MAJTYP/13	MAJTYP/M	PLTB
MATCH	R2	MATCH/7	MATCH/M	*All
MATRIX	RA	MATRIX/11B	MATRIX/M	SYN
MFTX2	R8	MFTX2/4	MFTX2/M	GSF
MKSTR	RA	MKSTR/11	MKSTR/M	SYN
MONTOR	RP	L604/12	L604/M	PLTB
MOVEXY	RP	MOVEXY/15	MOVEXY/M	PXY
MPAUS	R9	MPAUS/16	MPAUS/M	AUS
MPCEIG	RC	MPCEIG/12	MPCEIG/M	CEIG
MPDCU	R2	MPDCU/16	MPDCU/M	DCU
MPDR	R7	MPDR/9	MPDR/M	DR

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
MPE	R5	F3/8	F3/M	E
MPEIG	R9	MPEIG/8	MPEIG/M	EIG
MPEKS	R5	F3K/7	F3K/M	EKS
MPELD	R3	MPELD/15	MPELD/M	ELD
MPEQNF	R7	MPEQNF/9	MPEQNF/M	EQNF
MPFSM	R6	MPFGKM/11	MPFGKM/M	FSM
MPGSF	R8	GSFMP/9	GSFMP/M	GSF
MPINV	R6	AF/7	AF/M	INV
MPK	R6	ASK/7	ASK/M	K
MPKG	R6	ASKG/7	ASKG/M	KG
MPKMAP	R4	PF5/12	PF5/M	PKMP
MPM	R6	ASM/8	ASM/M	M
MPMFIL	R4	PF4/11	PF4/M	PAMP
MPMN	R5	MPMN/15	MPMN/M	MN
MPPLTA	RP	MPGGS/9	MPGGS/M	PLTA
MPPLTB	RP	MPPLTB/16	MPPLTB/M	PLTB
MPPRTE	R5	PF3/11	PF3/M	PRTE
MPPS	R6	PRTSM/9	PRTSM/M	PS
MPPSF	R8	PSFMP/7	PSFMP/M	PSF
MPPSR	R9	MPPSR/16	MPPSR/M	PSR
MPPXY	RP	MPPXY/15	MPPXY/M	PXY
MPSM	RM	MPSM/12	MPSM/M	SM
MPSSBT	RA	MPSSBT/10	MPSSBT/M	SSBT
MPSSOL	R7	DS/7	DS/M	SSOL
MPSTRP	RA	MPSTRP/8	MPSTRP/M	STRP
MPSYN	RA	MPSYN/8	MPSYN/M	SYN
MPTAB	R3	MPTAB/15	MPTAB/M	TAB
MPTOPO	R4	TOPOMP/7	TOPOMP/M	TOPO
MPVPRT	R7	MPVPRT/11	MPVPRT/M	VPRT
MPXMAP	R6	MPXMAP/11	MPXMAP/M	XMAP
MTEX	RC	MTEX/14	MTEX/M	CEIG SM
MULMX	R9	MULMX/8	MULMX/M	EIG
MULTEX	R7	MULTEX/8	MULTEX/M	AUS SSOL EIG
NCALNA	RJ	NCALNA/11	NCALNA/M	EKS SM
NDEP	RJ	NDEP/15	NDEP/M	MN
NDEP2	RJ	NDEP2/15	NDEP2/M	MN
NEN	R2	NEN/6	NEN/M	*All
NEWX	R9	NEWX/8	NEWX/M	EIG
NF3D	R5	NF3D/15	NF3D/M	MN
NFBEAM	R7	NFBEAM/15	NFBEAM/M	EQNF
NFSHEL	R7	NFSHEL/11	NFSHEL/M	EQNF
NLMCLD	R5	NLMCLD/15	NLMCLD/M	MN
NODPLT	RP	L605/V70L	L605/M	PLTB
NODVAL	RP	NODVAL/16	NODVAL/M	PLTB
NORM	R9	NORM/9	NORM/M	AUS

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
NORMRI	RC	NORMRI/12	NORMRI/M	CEIG
NPRS	R7	NPRS/14	NPRS/M	EQNF
NSECTS	R2	NSECTS/6	NSECTS/M	*All
NTOC	R2	NTOC/9	NTOC/M	*All
NUM1	R9	NUM1/16	NUM1/M	AUS
NWORDS	R2	NWORDS/6	NWORDS/M	*All
OP1	R9	OP1/8	OP1/M	AUS SSOL CEIG SM
OPN1	R5	OPN1/15	OPN1/M	MN
OUTF1	R3	OUTF1/7	OUTF1/M	TAB
OUTSOL	RA	OUTSOL/9	OUTSOL/M	STRP
OUTZ	R4	OUTZ/4	OUTZ/M	TOPO STRP
P3CALC	R9	P3CALC/16	P3CALC/M	PSR
P44	R8	P44/8	P44/M	PSF
PACK46	RP	L608A/V70H	L608A/M	PLTB PXY
PCALC	R9	PCALC/16	PCALC/M	PSR
PEJECT	R3	PEJECT/10	PEJECT/M	TAB
PF3A	R5	PF3A/11	PF3A/M	PRTE
PFB1	R6	PFB1/12	PFB1/M	FSM
PLAMS	R8	PLAMS/11	PLAMS/M	PSF
PLREAD	RP	PLREAD/9	PLREAD/M	----
PLT3D	RP	PLT3D/12	PLT3D/M	PLTB
PLTCLS	RP	PLTCLS/M	PLTCLS/M	PLTB
PLTEXT	RP	L6018/V70L	L6018/M	PLTB
PLTOPN	RP	PLTOPN/M	PLTOPN/M	PLTB
PLTR	RP	L602/12	L602/M	PLTB
POWER	R9	POWER/16	POWER/M	AUS
PRECON	R4	F4D/12	F4D/M	TOPO
PREOPT	RP	PREOPT/10	PREOPT/M	PLTB
PREPEL	R5	PREPEL/11	PREPEL/M	E
PREXT	R7	PREXT/9	PREXT/M	DR
PRIN3D	R8	PRIN3D/12	PRIN3D/M	PSF
PRINT1	R2	PRINT1/9	PRINT1/M	DCU
PRINV	R6	PRINV/9	PRINV/M	PS
PRLINE	R2	PRLINE/9	PRLINE/M	*All
PRNLS1	R5	PRNLS1/15	PRNLS1/M	MN
PRNLSS	R5	PRNLSS/15	PRNLSS/M	MN
PROCOM	RP	GPF10/9	GPF10/M	PLTA
PRP2	R9	PRP2/9	PRP2/M	AUS
PRT34	R8	PRT34/11	PRT34/M	PSF
PRTHST	R9	PRTHST/6	PRTHST/M	EIG
PRTVEC	R9	PRTVEC/5	PRTVEC/M	EIG
PS21	R8	PS21/10	PS21/M	PSF
PS22	R8	PS22/8	PS22/M	PSF
PS23	R8	PS23/8	PS23/M	PSF
PS24	R8	PS24/11	PS24/M	PSF

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
PS3D	R8	PS3D/12	PS23/M	PSF
PSFEX	R8	PSFEX/12	PSFEX/M	PSF
PSFLD	R8	PSFLD/12	PSFLD/M	PSF
PSM	R6	PSM/9	PSM/M	PS
PSMDP	R6	PSMDP/9	PSMDP/M	PS
PSN	R8	PSN/11	PSN/M	PSF
PSPACE	RP	PSPACE/12A	PSPACE/M	PLTB
PTAB1	R2	PTAB1/9	PTAB1/M	DCU
PUP	R7	PUP/8	PUP/M	TAB AUS EQNF GSF PLTB MN E
PWR	R9	PWR/16	PWR/M	AUS
QGENP	RP	GPF09/V70I	GPF09/M	PLTA
QH	R6	QH/1	QH/M	K SM
QTEQ	RA	QTEQ/11	QTEQ/M	SYN
RANF	R9	RANF/7	RANF/M	EIG CEIG
RATIOS	RP	L6013/V70E	L6013/M	PLTB
RBINT	R7	RBINT/9	RBINT/M	DR
RBVEC	R9	RBVEC/7	RBVEC/M	AUS
RDIND	R2	RDIND/7	RDIND/M	*All
RDMAT	R9	RDMAT/9	RDMAT/M	AUS
RDTAB	R3	F1T/15	F1T/M	TAB
REAC	R7	REAC/9	REAC/M	SSOL
READ	R2	READ/7	READ/M	*All
READD	RA	READD/9	READD/M	STRP
READER	R2	READER/9	READER/M	*All
REC	RA	REC/11	REC/M	SYN
RECMAD	RA	DMAT/10	DMAT/M	STRP
RECMAT	R9	RECMAT/5	RECMAT/M	EIG
RED	R6	RED/8	RED/M	INV
REDDP	R6	REDDP/8	REDDP/M	INV
REDUCE	R9	REDUCE/16	REDUCE/M	EIG
REPOSZ	RA	REPOSZ/9	REPOSZ/M	STRP
RGEN	RJ	RGEN/9	RGEN/M	EKS SM
RIFIN	RC	RIFIN/12	RIFIN/M	CEIG
RIGI	R9	RIGI/9	RIGI/M	AUS
RINV	R9	RINV/9	RINV/M	AUS
RIO	R2	RIO/9	RIO/M	*All
RLOOP	R9	RLOOP/13	RLOOP/M	AUS
RMAT	R9	RMAT/9	RMAT/M	AUS
RMBT	R3	RMBT/11	RMBT/M	TAB
RMP	R8	RMP/10	RMP/M	FSM GSF
ROOTS	RC	ROOTS/13	ROOTS/M	CEIG
RPOINT	R5	F32A3/1	F32A3/M	E
RPRO	R9	RPRO/9	RPRO/M	AUS
RRINZ	R4	RRINZ/4	RRINZ/M	TOPO STRP
RRMK	R9	RRMK/7	RRMK/M	EIG

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
RROUTZ	RA	RROUTZ/9	RROUTZ/M	STRP
RSEL1	RC	RSEL1/13	RSEL1/M	CEIG
RSET	R2	RSET/16	RSET/M	*All
RTRA	R9	RTRA/9	RTRA/M	AUS
RTV	RP	RTV/9	RTV/M	PLTB
RWINDZ	R4	RWINDZ/6	RWINDZ/M	TOPO STRP
S11	R9	S11/8	S11/M	AUS
S1V	R9	S1V/8	S1V/M	AUS
S21	R9	S21/8	S21/M	AUS
S22	R9	S22/8	S22/M	AUS
S2V	R9	S2V/8	S2V/M	AUS
SASCON	R3	F1T15B/10	F1T15B/M	TAB
SBA	RA	SBA/11	SBA/M	SSBT
SBA1	RA	SBA1/10	SBA1/M	SSBT
SBA2	RA	SBA2/10	SBA2/M	SSBT
SBB	RA	SBB/10A	SBB/M	SSBT
SBB1	RA	SBB1/10	SSB1/M	SSBT
SCLPLT	RP	L6012/V70I	L6012/M	PLTB
SCNTOC	R9	SCNTOC/9	SCNTOC/M	AUS
SCOMP	R8	SCOMP/13	SCOMP/M	GSF
SDPLAY	RP	SD02/16	SD02/M	PLTB
SDPXQT	RP	SDPXQT/11	SDPXQT/M	PLTB
SE21	R8	SE21/15	SE21/M	PSF
SECT2	R3	F1T091/12	F1T091/M	TAB
SEGMPY	R9	SEGMPY/16	SEGMPY/M	AUS
SEQGEN	R3	F1T17/11	F1T17/M	TAB
SETFNT	RP	SETFNT/15	SETFNT/M	PXY
SFETCH	RP	SD03/11B	SD03/M	PLTB
SHADE	RP	L607/V70L	L607/M	PLTB
SHRINK	R9	SHRINK/8	SHRINK/M	AUS
SI	R3	SI/6	SI/M	TAB
SKEWP	R7	SKEWP/9	SKEWP/M	EQNF GSF
SLABL	RP	SD05/16	SD05/M	PLTB
SMLD	R6	SMLD/15	SMLD/M	K M KG
SMSB	R8	SMSB/11	SMSB/M	PSF PLTB
SMSPDP	RC	SMSPDP/12	SMSPDP/M	CEIG SM
SMUL	RC	SMUL/12	SMUL/M	CEIG SM
SMULT	R7	SMULT/8	SMULT/M	AUS SSOL EIG
SNEW	RJ	SNEW/15	SNEW/M	MN
SNEW2	RJ	SNEW2/15	SNEW2/M	MN
SPECIO	RP	SPECIO/1	SPECIO/M	PLTA
SPFGKM	R6	SPFGKM/12	SPFGKM/M	FSM
SPMOVE	RM	SPMOVE/12	SPMOVE/M	SM
SPMX	RM	SPMX/12	SPMX/M	SM
SPROD	R9	SPROD/16	SPROD/M	AUS

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors			
SPRT	R8	SPRT/12	SPRT/M	PSF			
SPTRN3	R6	SPTRN3/12	SPTRN3/M	K	M	KG	FSM
SPTRN6	R6	SPTRN6/12	SPTRN6/M	K	M	KG	
SQUARE	R9	SQUARE/5	SQUARE/M	EIG			
SRTOS	R8	SRTOS/11	SRTOS/M	PSF			
SS1V	R9	SS1V/8	SS1V/M	AUS			
SS2V	R9	SS2V/8	SS2V/M	AUS			
SSHL	RP	SSHL/11	SSHL/M	PLTB			
SSMK	R9	SSMK/5	SSMK/M	AUS			
SSPREP	R9	SSPREP/8	SSPREP/M	AUS			
SSTM	RJ	SSTM/12A	SSTM/M	EKS	SM	MN	
SSUM	R9	SSUM/16	SSUM/M	AUS			
STATIO	R2	STATIO/7	STATIO/M	*All			
STCHOL	RA	STCHOL/8	STCHOL/M	STRP			
STEXPE	R8	STEXPE/12	STEXPE/M	GSF			
STORE	RA	STORE/8	STORE/M	STRP			
STORS	RJ	STORS/11	STORS/M	K	SM		
STORS3	R6	STORS3/12	STORS3/M	K			
STRDIA	RA	STRDIA/8A	STRDIA/M	STRP			
STRK2D	R8	STRK2D/11C	STRK2D/M	GSF			
STRLST	RP	STRLST/10	STRLST/M	PLTB			
STRNE	RJ	STRNE/12	STRNE/M	EKS	MN		
STRPRT	RA	STRPRT/10	STRPRT/M	STRP			
STRS21	R8	STRS21/15	STRS21/M	GSF			
STRS3D	R8	STRS3D/13	STRS3D/M	GSF			
STRSYM	RA	STRSYM/8	STRSYM/M	STRP			
STRTAB	R3	STRTAB/10	STRTAB/M	TAB			
SVEC2	R9	SVEC2/9	SVEC2/M	AUS			
SVEC3	R9	SVEC3/10	SVEC3/M	AUS			
SVV	R9	SVV/16	SVV/M	AUS			
SYMINV	RJ	SYMINV/14	SYMINV/M	EKS	CEIG	SM	MN
SYMVRT	RJ	SYMVRT/11	SYMVRT/M	EKS	SM		
SYSM	RC	SYSM/12	SYSM/M	CEIG			
T3D01	R7	T3D01/12	T3D01/M	EQNF			
T3D02	R7	T3D02/12A	T3D02/M	EQNF			
TCB	RJ	TCB/9	TCB/M	EKS	SM		
TCLOCK	R2	TCLOCK/7	TCLOCK/M	*All			
TCOL	RM	TCOL/13	TCOL/M	SM			
TCOL1	RM	TCOL1/13	TCOL1/M	SM			
TDMBRN	R7	TDMBRN/14	TDMBRN/M	EQNF			
TERMIN	RP	TERMIN/15	TERMIN/M	PXY			
TFB1	R6	TFB1/12	TFB1/M	FSM			
TGEN	RJ	TGEN/11	TGEN/M	EKS	SM		
THAFMP	RA	THAFMP/8	THAFMP/M	STRP			
THSBMP	RA	THSBMP/8	THSBMP/M	STRP			

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
TIC1	RP	TIC1/15	TIC1/M	PXY
TIC2	RP	TIC2/15	TIC2/M	PXY
TIC3	RP	TIC3/15	TIC3/M	PXY
TINT	R7	TINT/9	TINT/M	DR
TIO	R2	TIO/6	TIO/M	DCU
TITL	R9	TITL/9	TITL/M	AUS
TK3D	R5	TK3D/15	TK3D/M	MN
TKU	RJ	TKU/11	TKU/M	K SM
TLAB1	RP	TLAB1/15	TLAB1/M	PXY
TOCO	R2	TOCO/9	TOCO/M	ELD AUS EQNF SSOL GSF MN
TOPOEX	R4	TOPOEX/12	TOPOEX/M	TOPO
TOPOLD	R4	TOPOLD/9	TOPOLD/M	TOPO
TR1	R7	TR1/9	TR1/M	DR
TR1A	R7	TR1A/9	TR1A/M	DR
TRAML1	RA	TRAML1/11	TRAML1/M	SYN
TRAML2	RA	TRAML2/11	TRAML2/M	SYN
TRAN3	R6	TRAN3/5	TRAN3/M	SM
TRAN6	R6	TRAN6/5	TRAN6/M	SM
TRGEN	RA	TRGEN/8	TRGEN/M	SYN
TRI	RA	TRI/11	TRI/M	SYN
TRIL	RJ	TRIL/11	TRIL/M	K SM
TRIL3	R6	TRIL3/12	TRIL3/M	K
TRIMUL	RJ	TRIMUL/11	TRIMUL/M	K SM
TRINV6	R3	TRINV6/11B	TRINV6/M	TAB
TRIOUT	RJ	TRIOUT/11	TRIOUT/M	K SM
TRIPRO	R7	TRIPRO/9	TRIPRO/M	EQNF GSF
TRISQ3	R6	TRISQ3/11	TRISQ3/M	FSM
TRMC	RJ	TRMC/15	TRMC/M	MN
TRMCO	R5	TRMCO/15	TRMCO/M	MN
TSUBMP	RA	TSUBMP/8	TSUBMP/M	STRP
TT10X3	RJ	TT10X3/9	TT10X3/M	EKS SM
TT6X3	RJ	TT6X3/9	TT6X3/M	EKS SM
TTE	R3	TTE/14	TTE/M	ELD
TTE1	R3	TTE1/14	TTE1/M	ELD
TTGEN	RJ	TTGEN/9	TTGEN/M	EKS SM
TX2N	R5	TX2N/7	TX2N/M	EKS SM
TXPR	R2	TXPR/9	TXPR/M	TAB PSF PLTB
U3D	R8	U3D/12	UED/M	GSF
UBEND	R8	UBEND/11	UBEND/M	GSF
UEVAL	R7	UEVAL/8	UEVAL/M	AUS SSOL EIG CEIG
ULOC3	R5	ULOC3/15	ULOC3/M	MN
UMBRN	R8	UMBRN/11C	UMBRN/M	GSF
UNION	R9	UNION/9	UNION/M	AUS
VIEWST	RP	GPF06/V70J	GPF06/M	PLTA
VLD	R9	VLD/9	VLD/M	AUS

Table 7-4 (Continued)

Name	File	Symbolic	Relocatable	Processors
VMISES	RJ	VMISES/15	VMISES/M	MN
VR2	R8	VR2/15	VR2/M	GSF PSF
WARPT	RJ	WARPT/11	WARPT/M	EKS SM
WNU7	R4	F4B/11	F4B/M	TOPO
WR	R2	WR/7	WR/M	*All
WRTIND	R2	WRTIND/7	WRTIND/M	*All
WRTJKC	R3	F1T15C/1	F1T15C/M	TAB
XATN2	R9	XATN2/16	XATN2/M	AUS
XBLOCK	R3	F1T051/8	F1T051/M	TAB
XKALER	RP	L603	L603/M	PLTB
XLIO	R9	XLIO/9	XLIO/M	AUS
XNTI	R9	XNTI/16	XNTI/M	AUS
XPARA	RM	XPARA/13	XPARA/M	SM
XPRNT	R9	XPRNT/16	XPRNT/M	EIG
XSI	RP	XSI/9	XSI/M	PLTA
XTRANS	RC	XTRANS/12	XTRANS/M	CEIG
XTY	R9	XTY/9	XTY/M	AUS
XTYD	RC	XTYD/12	XTYD/M	CEIG
XTYG	RC	XTYG/12	XTYG/M	CEIG SM
XUEVL	RC	XUEVL/12	XUEVL/M	CEIG
XXMN	R5	XXMN/15	XXMN/M	MN
XY1	RP	XY1/15	XY1/M	PXY
XY2	RP	XY2/15	XY2/M	PXY
XY3	RP	XY3/15	XY3/M	PXY
XY3S	RP	XY3S/15	XY3S/M	PXY
XY4	RP	XY4/15	XY4/M	PXY
XY5	RP	XY5/15	XY5/M	PXY
XY6	RP	XY6/15	XY6/M	PXY
XY7	RP	XY7/15	XY7/M	PXY
XYEXT	RP	XYEXT/15	XYEXT/M	PXY
XYLD	RP	XYLD/15	XYLD/M	PXY
XYTEXT	RP	XYTEXT/15	XYTEXT/M	PXY
YPARA	RM	YPARA/12	YPARA/M	SM
ZRMP	R8	ZRMP/12	ZRMP/M	EQNF GSF

8. PROCESSOR COMMAND SUMMARY

This section contains a summary of processor commands and resets. The following resets are common to all processors:

CORE n
ABORT

The following commands are common to all processors:

ERABt
FIN
FORMat
ONLIne
IOUT

Resets

none

Commands

TOC
DISAble
ENABle
PRINt
COPY
XCOPy
LIBLib
NTAPe
ABORt
NCPL
SCALe
CLEAn
TREAd
RETRieve
REPOsition
XLOAd
REWInd
TITLe
CHANge
DUPLicate
STORe
STATus
TWRIte
EXIT
STOP

Resets
ncne

Commands

TEXT	
MATC	MATErial CONStants
NSW	DISTribited weight
ALTRef	ALTErnate REFErence frames
JLOC	JOINt LOCAtions
JREF	JOINt REFErence frames
MREF	BEAM ORIEntation
BRL	BEAM RIGId links
BA	E21 SECTion properties
BB	BEAM S6X6
BC	E23 SECTion properties
BD	E24 SECTion properties
SA	SHELl SECTion properties
SB	PANEl SECTion properties
CON	CONStraint definitions
JSEQ	JOINt ELIMination sequence
RMASs	RIGId MASSes
QJJT	
TITLe	
RK	
RM	
STRA	
STARt	
STOP	
MOD	
NREF	
UPDAte	

TAB/RMASS
REPEat
CM
ZERO

TAB/SA
ISOTropic
MEMBrane
PLATe
UNCOupled
COUPled
LAMInate
W
INVM
INBT
NMAT

TAB/CON
ZERO
RELEase
SYMMetry plane
ANTIsymmetry plane
FIXEd plane
NONZero
LZERo
RZERo

TAB/JSEQ
REPEat

Resets	Default
LIB	1
LREC	896
KS3D	3

Commands

E21
E22
E23
E24
E25
E31
E32
E33
E41
E42
E43
E44
F41
F61
F81
S41
S61
S81
STOP
EXPE
GROUP
NMATerial
NSECt
NNSW
NOFF
NREF
NDES
NPROp
MOD JOINT
MOD GROUP
MOD NSECT
MOD NMAT
MOD NNSW
MOD NREF
MOD NOFF
INC NSECT
INC NMAT
INC NNSW
INC NREF
INC NOFF

Resets	Default
BLIB	1
LRKMap	696
LRAMap	1792
LR7	896
MAXSub	1400
ILMAx	0
LAPR	0
TIME	0
SA	0
PRTKmap	0
PRTAmap	0
PRT7	0
HLIB	1
ILIB	1

Commands

none

Resets	Default
G	1.
PRTT	1
PRTE	1
LIM	50
RCH	.0001
LZERo	.001
MWARp	.05

Commands

STOP

T

IERR

Resets	Default
BLIB	1
ELIB	1
GAZEro	1.E-20
CIZEro	1.E-20
TIME	0
ZK2D	.0001
GIPT	2
NS4	1
NS6	3
NS8	3

Commands

STOP
K3D
K2D
FLUSh

Resets	Default
SEG1	2
SEG2	7

Commands

STOP

ele type e.g. E43

K

Resets	Default
LREC	2240
BLIB	1
ELIB	1
HLIB	1
OUTLib	1
SPDP	1
TIME	0
SA	0
MCURve	1

Commands

none

M

Resets	Default
LREC	2240
BLIB	1
ELIB	1
HLIB	1
OUTLIB	1
G	1.0
IBEAm	0
INERt	0
TIME	0
SA	0
MCURve	1

<u>Commands</u>

none

Resets	Default
LREC	2240
BLIB	1
ELIB	1
HLIB	1
OUTLib	1
AZERo	1.E-10
IZERo	1.E-10
IKG2	0
IKG3	0
TIME	0
SA	0
MCURve	1

Commands

none

Resets	Default
K	K
DZERo	1.E-5
CON	1
KLIB	1
KILIb	1
NJMAx	50
LRA	3584
ILIB	klib

Commands
none

Resets   Default
none

Commands
STOP
LIB
J
data set name

Resets	Default
TL1	21
TL2	22
IGKM	3
LREC	2240
TIME	0
SA	0
BLIB	1
ELIB	1
HLIB	1
OUTLib	1
SPDP	1

Commands

G

Resets	Default
CLIB	1
ELIB	1
SLIB	1
ULIB	1
FLIB	1
SET	1
CON	1
ZDE	1.E-20
LAYErs	5
NJSS	10
GIPT	2
NS4	1
NS6	3
NS8	3
ZYIEld	.001

Commands

STOP
TRMC
INSS
NF
TK
PRINt

Resets	Default
SET	1
L1	1
L2	1
ELIB	1
INLIb	1
FEFLib	1
ISLIb	1
ISBL	896

Commands
STOP
T3D

Resets	Default
K	K
KLIB	1
KILIb	1
QLIB	1
TIME	0
EP	1
L1	1
L2	all records
CON	1
REAC	1
SET	1
NMAX	0

Commands

none

<u>Resets</u>

none

<u>Commands</u>

LIB
COMPonents
HEADing
VECTors
JOINts
LINEs
ZERO (same as FILTer)
FILTer
I
J
STOP
PRINTt
TPRInt

Resets	Default
ELIB	1
QLIB	1
L1	1
L2	0
SET	1
CON	MASK
LREC	5600
IEA	1
EMBEd	0
ACCUm	0
KGF	0

Commands

SOURce
STOP

Resets	Default
BLIB	1
QLIB	1
L1	1
L2	1
SET	1
IEA	1
LINEs	56
DISPlay	1
NODEs	1
CROSs	1
Z3DA	.00001
Z3DR	.00001

Commands
STOP
CFILter
DIV
SFILter

PSR

Resets	Default
INLIb	1
OUTLib	1

Commands
"etype"

Resets
none

Commands
STOP
DTEX
TR1
BACK

Controls	Default
DT	calculated (1/8 period)
NTERms	10
INLIb	1
OUTLib	1
N2	MASK

DR/
TR1

Controls	Default
PLIB	0
QXLIb	0
QX1Lib	0
QX2Lib	1
ALIB	1
QR2Lib	1
QRLIb	0
QR1Lib	0
INLIb	1
CASE	1
LB	896
N2	MASK
T1	
T2	

DR/
BACK

Controls	Default
M1	1
M2	
DT	1.0
TSTArt	0.0
LRZ	896
PRINt	1
N3REpeat	1
N4REpeat	1
FMAX	-1.E+20
FMIN	1.E+20
BIG	1.E+20
STAT	0
SOURce	1
DEST	1

Commands

Z
EXT
ZC
T
Y

Resets
none

Commands
INLIb
OUTLib
DEFIne
RIGId
UNIOn
SUM
PRODuct
XTYDiag
STYSym
XTY
SQRT
RECIp
SQUAre
NORM
RPROd
RTRAn
RINV
LTOG
GTOL
STOP
NODQ
ELDAta
ALPHa
SYSVec
TABLe
U
NI
NJ
ZERO
SSID
SSPRep
SSM
SSK

Commands
COS
ACOS
SIN
ASIN
TAN
ATAN
ATN2
COSH
SINH
TANH
EXP
ALOG
AL10
ABS
IFIX
FLOAt
POWEr
SRSS
NUM1
XNT1
XNT2
XNT3
XNT4
CBR
CBD
ACBR
ACBD

Commands (TABLE)
TRAN
BLOCk
CASE
STOP

Commands (ELDATA)
GSHIft
ESHIft
I
DDATa
OPERation
GROUp
G
E
SUM
XSUM
MULTiply
DIVIde
TEMP
DISL
PRES
CASE

Commands (SYSVEC)
JSHIft
I
DDATa
OPERation
J
JOINt
SUM
XSUM
MULTiply
DIVIde
SOURce
ILIM
JLIM
SSKIp
SBASe
DBASe
L1
L2

Resets	Default
SYSL	1
LR	896
CON	1
TOLR	0.1E-5
TOLM	1.0E-30
TOLK	1.0E-30

Commands

STOP
K
M
FUNC

Resets	Default
FRQ1	-10.E10
FRQ2	10.E10
TOL	1.E-15
SOURce	1
DEST	1
INT	0
NOUT	26

Commands

none

Resets	Default
JMG	1

Commands

STOP

MODEs

Resets	Default
CM	.0
SOURce	1
DEST	1
LTEMp	21
SET	1
CON	1
N	0
HIST	0
NDYN	6
CONV	1.E-5
NREQ	0
V1	.0
V2	.0
NUIN	0
ZERO	1.E-20
CMEThod	1
CBAL	0
ZMOD	1.E-30
ZRI	1.E-4
ZVEC	1.E-30
KSEL	2
MSEL	1
RRPR	0

Commands

none

Resets	Default
AZER	1.E-10
IZER	1.E-10
G	1.0
IKG2	0
IKG3	0
IBEA	0
INER	0
LREC	2240
TIME	0
SA	0
BLIB	1
ELIB	1
HLIB	1
OUTL	21
NUPAra	1
N4PAra	1
NPARas	0
NUMS	0
NUT	1
NUEIg	1
N3EIg	1
N4EIg	1
NUDK	22
NUXD	23
NUVX	24
PZERo	1.E-20
NUDP	0
NSRR	0
NSEE	0
KTARget	0
KDPX	1
DPZEro	1.E-20
NUDM	25
NEGL	0
ZDV	1.E-20

Commands
OPER
AOPE

Resets	Default
GGSL	16

Commands

SPEC
LCONtrol
VIEWs
ROTAte
SYM
AXES
STITle
S2TItle
MARGin
STOP
ANTIsym
LINEs
TEXT
JLABel
LROTate
CONNect
LOCLabel
ALL
PRIN

Resets	Default	(TEK version only)
BAUD	1200	
NDEV	4016	
CHRS	2	

Commands

```
STOP
PLOT
INLIb
PSLI
SET
CASEs
VECTors
CON
DNORm
ECHO
OPTIons
DISP
LAMI
DISP= UNDE
      STAT
      VIBR
      BUCK
      TABLe
```

9. REFERENCES

1. Whetstone, W.D., "SPAR Structural Analysis System Reference Manual - System Level 13A. Volume 1 - Program Execution" NASA CR-158970-1, December 1978.

2. Oden, J.T. and M.L. Pearson, "SPAR Improved Structure/Fluid Dynamic Analysis Capability," LMSC-HREC TR D867285, August 1983.

3. Pearson, M.L., "SPAR Improved Structure/Fluid Dynamic Analysis Capability, Phase II - Final Report," LMSC-HREC TR D951490, June 1984.

Appendix A

SPAR REFERENCE MANUAL UPDATES

Appendix A

Included as an attachment to this appendix are update pages to the SPAR Structural Analysis System Reference Manual (NASA CR 158970-1) dated December 1978. These updates describe changes and additions to the manual which are applicable to SPAR System Level 16.

Attachment to Appendix A

Update pages to the SPAR Structural Analysis System Reference Manual (NASA CR 158970-1)

Table 1-2: SPAR Processor Functions (continued)

Name and Section Reference		Function
AUS	5.1	The Arithmetic Utility System, comprised of an array of subprocessors in the following categories: - Data set constructors, providing a general means of furnishing input data for use by SPAR. Applied load data of all types (mechanical, thermal, pressure, dislocational, transient dynamic) is usually defined via these subprocessors. - Matrix arithmetic operations, e.g. sums, products, unions. - Special functions, including subprocessors used in performing substructure analysis.
EQNF	6.2	Computes fixed-joint forces associated with thermal, dislocational, and pressure loading. Computes element generalized initial strain arrays.
SSOL	6.3	Computes joint motions and reactions due to static loading.
GSF	7.1	Produces data sets containing element stresses and internal loads. GSF is used to compute both static and dynamic stresses.
PSF	7.2	Produces tabular stress reports from data sets generated by GSF.
PSR	7.3	Produces multi-block, table-format data sets containing principal stresses for both 2-d and 3-d element types.
EIG	8	Solves high-order eigenproblems involving system matrices in SPAR's sparse matrix format. Used to solve both vibrational and buckling eigenproblems.
CEIG	13	Computes complex modes and frequencies of damped, spinning structures. System matrices are in SPAR's standard sparse matrix format, permitting analysis of systems of very high order.
DR	9	Computes linear transient modal response.

which will cause the processor not to make an error abort if it encounters a serious error (e.g., if required input data sets do not exist), and

RESET CORE = n$ (available on UNIVAC, only)

which will result in issuance of an executive request to change the available data space to n words.

On CDC systems, the user controls core size through RFL cards.

The statement, DATA SPACE = n, appearing at the beginning of execution of each program, indicates n = available data space.

Most SPAR programs generate little or no printed output. In some programs, the kind and quantity of output are controlled by a command (not a reset parameter) in the following form:

ONLINE = n$

where n = 0 for minimum printout, 1 for normal printout, and 2 for maximum printout. If desired, the ONLINE statement may be used more than once within the same program execution.

Table 5-1 Summary of AUS Subprocessors

Miscellaneous	General Arithmetic		Data Set Constructors	Substructure
INLIB	SUM	COSH	TABLE	SSPREP
OUTLIB	PRODUCT	SINH	SYSVEC	SSM
DEFINE	UNION	TANH	ELDATA	SSK
ZERO	XTY	EXP	ALPHA	SSID
FIND	XTYSYM	ALOG		
	NORM	AL10		
	RIGID	ABS		
	RECIP	IFIX		
	SQUARE	FLOAT		
	SQRT	POWER		
	RPROD	SRSS		
	RTRAN	NUM1		
	RINV	XNT1		
	COS	XNT2		
	ACOS	XNT3		
	SIN	XNT4		
	ASIN	CBR		
	TAN	CBD		
	ATAN	ACBR		
	ATN2	ACBD		

5.1.2 General Arithmetic Operations

Table 5.1.2-1 summarizes commands in this category. All are in the following form, except where specifically noted otherwise.

lib Z= c Oper(c_1 X_1, c_2 X_2, ---)

where Oper is one of the operation names, such as SUM, PRODUCT, etc., and the X_i's are short-form names identifying source data. If a short-form name X has not appeared in a DEFINE X= ... statement, it is assumed that X is a data set named X MASK MASK MASK that is contained in the current primary data source library identified by the last INLIB statement. The c's are floating-point constants which may be omitted (default is 1.0).

The data set produced as a result of the command will be stored in the library designated by lib, or in the current destination library designated by the last OUTLIB command if lib is omitted. The name of the output data set will depend on the form of Z, as summarized below:

Form of Z	Output Data Set Name
N1	N1 AUS 1 1
N1 N2	N1 N2 1 1
N1 N2 n3	N1 N2 n3 1
N1 N2 n3 n4	N1 N2 n3 n4

Table 5.1.2-1 Summary of General Arithmetic Operations

Command Forms	Meaning
Z= SUM(X,Y)	Z = X + Y (system matrices)
Z= PRODUCT(X,Y)	Z = X Y (system matrices)
Z= UNION(X_1,X_2,---)	Z = [X_1 | X_2 | X_3 ---]
Z= XTY(X,Y)	Z = X^tY
Z= XTYSYM(X,Y)	Z = X^tY, symmetric
Z= XTYDIAG(X,Y)	Z = X^tY, diagonal
Z= NORM(X,j,k,v)	System vector renormalization
Z= RIGID(j)	Rigid body motion vectors
Z= RECIP(X)	Each element z = 1./x
Z= SQRT(X)	Each element z = sign(x) $\sqrt{\|x\|}$
Z= SQUARE(X)	Each element z = x^2
Z= RPROD(X,Y)	Z = X Y (rectangular matrices)
Z= RTRAN(X)	Z = X^t (rectangular matrices)
Z= RINV(X)	Z = X^{-1} (square matrices)
Z= LTOG(X)	Converts system vector components from local joint reference frames to global
Z= GTOL(X)	Complement of LTOG
Z= COS(X)	Each element z = cos(x)
Z= ACOS(X)	Each element z = arccos(x)
Z= SIN(X)	Each element z = sin(x)
Z= ASIN(X)	Each element z = arcsin(x)
Z= TAN(X)	Each element z = tan(x)

Table 5.1.2-1 (Continued)

Command Forms	Meaning
Z= ATAN(X)	Each element z = arctan(x)
Z= ATN2(X,Y)	Each element z = arctan(x/y)
Z= COSH(X)	Each element z = cosh(x)
Z= SINH(X)	Each element z = sinh(x)
Z= TANH(X)	Each element z = tanh(x)
Z= EXP(X)	Each element z = e^x
Z= ALOG(X)	Each element z = ln(x)
Z= AL10(X)	Each element z = $\log_{10}(x)$
Z= ABS(X)	Each element z = abs(x)
Z= IFIX(X)	Each element z = ifix(x)
Z= FLOAT(X)	Each element z = float(x)
Z= POWER(X,p)	Each element z = x^p
Z= SRSS(X,Y)	Each element z = $\sqrt{x^2 + y^2}$
Z= NUM1(X,Y)	Numerical integration
Z= XNT1(XY,A)	Linear interpolation
Z= XNT2(XY,A)	Log-log interpolation
Z= XNT3(XY,A)	Linear-log interpolation
Z= XNT4(XY,A)	Log-linear interpolation
Z= CBR(X,Y)	Matrix multiplication
Z= CBD(X,Y)	Matrix multiplication
Z= ACBR(X,Y)	Matrix multiplication
Z= ACBD(X,Y)	Matrix multiplication

Examples.

K+KG= SUM(K, 4.7 KG)	System stiffness matrix including effects of prestress.
M1= SUM(RMASS, DEM)	Diagonal system matrix composed of rigid mass data plus the lumped-mass equivalent of all distributed element mass.
M2= SUM(CEM, RMASS)	SPAR-format consistent mass matrix, plus rigid-mass data.
K24= SUM(K, -24000. M)	Shifted stiffness matrix to be used in EIG to compute eigenvalues near 24,000.

Core Requirement. One block of X plus one block of Y.

Note: For operations involving type A or type D data sets, the core requirement stated above does not apply. If insufficient core is available to hold entire blocks, the blocks are loaded in segments using the available core.

5.1.2.2 PRODUCT. The general form of the command is as follows:

$$Z= PRODUCT(c_x X, c_y Y)$$

This statement means that Z is c_x times c_y times X post-multiplied by Y . In standard applications, X is of type S, DS, or D, and Y is of type V . Z will have the same number of blocks (vectors) as Y .

Example. Construct inertia force vectors due to rigid-body acceleration. The command R= RIGID(j) would result in production of a 6-block data set containing system rigid-body motions in SYSVEC format. Where M is the system mass matrix,

$$MR= PRODUCT(M, R)$$

would produce a 6-block data set, in SYSVEC format, containing inertia force vectors due to unit rigid-body accelerations.

In addition to the above, PROD can also be used to perform element-by-element multiplication of data sets, provided that X and Y have the same block length (NI*NJ), and both contain only real data. Where:

x_i = the ith element in the first block of X,

y_{ij} = the ith element in the jth block of Y, and

z_{ij} = the ith element in the jth block of Z,

$z_{ij} = x_i y_{ij}$.

Core Requirement. One block each of X, Y, and Z.

Note: For operations involving type A or type D data sets, the core requirement stated above does not apply. If insufficient core is available to hold entire blocks, the blocks are loaded in segments using the available core.

5.1.2.6 RIGID. The general form of the command is as follows:

$$Z = \text{RIGID}(j)$$

Z will be in SYSVEC form, containing six vectors (blocks) that define rigid-body motion of the system. The first three blocks correspond to unit translations in global directions 1, 2, and 3. The second three blocks correspond to unit rotations about axes parallel to the global frame, passing through joint j. If the integer j is omitted, a default value of 1 is assumed.

Core Requirement. 18 times the number of joints in the structure.

5.1.2.7 RECIP, SQRT, SQUARE. These commands apply to single or multiblock data sets comprised entirely of real data. The output, Z, will be in the same form (block length, number of words, etc.) as the input, X. In the following definitions, z_i and x_i are the ith elements of Z and $c_x X$, respectively.

Z= RECIP($c_x X$) indicates $z_i = 1.0/x_i$.

Z= SQRT($c_x X$) indicates $z_i = (\text{sign of } x_i)\sqrt{|x_i|}$

Z= SQUARE($c_x X$) indicates $z_i = x_i^2$.

The zero-test parameter established by the last ZERO= e statement (see Section 5.1.1) is used to avoid error stops in RECIP and SQRT. In these operations, $z_i = x_i$ if the magnitude of x_i is less than e.

Core Requirement. No minimum requirement. Uses available core to load blocks in segments if insufficient core space is present to load an entire block.

5.1.2.10 COS, ACOS, SIN, ASIN, TAN, ATAN, COSH, SINH, TANH, EXP, ALOG, AL10, ABS

The general form of this class of commands is as follows:

lib Z= c OPER(cx X)

X may be a single or multiblock data set and must contain only real data. The output, Z, will be in the same form (block length, number of words, etc.) as the input, X. In the following definitions, z and x are corresponding elements of Z and X, respectively.

Z= c COS(cx X) indicates z = c*cos(cx*x)

Z= c ACOS(cx X) indicates z = c*arccos(cx*x)

Z= c SIN(cx X) indicates z = c*sin(cx*x)

Z= c ASIN(cx X) indicates z = c*arcsin(cx*x)

Z= c TAN(cx X) indicates z = c*tan(cx*x)

Z= c ATAN(cx X) indicates z = c*arctan(cx*x)

Z= c COSH(cx X) indicates z = c*cosh(cx*x)

Z= c SINH(cx X) indicates z = c*sinh(cx*x)

Z= c TANH(cx X) indicates z = c*tanh(cx*x)

Z= c EXP(cx X) indicates $z = c*e^{(cx*x)}$

Z= c ALOG(cx X) indicates z = c*ln(cx*x)

Z= c AL10(cx X) indicates $z = c*\log_{10}(cx*x)$

Z= c ABS(cx X) indicates z = c*abs(cx*x)

Core Requirement. No minimum requirement. Uses available core to load blocks in segments if insufficient core space is present to load an entire block.

5.1.2.11 IFIX, FLOAT, POWER. These commands apply to single or multiblock data sets. IFIX and POWER operate on real data. FLOAT operates on integer data. The output, Z, will be in the same form (block length, number of words, etc.) as the input, X. In the POWER command, p is a floating point constant which must be present. The floating point constants c and cx default to 1.0 if omitted. In the following definitions, z and x are corresponding elements of Z and X, respectively.

```
Z=   IFIX( cx X)    indicates  z = ifix(cx*x)
Z= c FLOAT(   X)    indicates  z = c*float(x)
Z= c POWER(cx X,p)  indicates  z = c*(cx*x)^p
```

<u>Core Requirement.</u> POWER requires one block of X. IFIX and FLOAT have no minimum requirement. They use available core to load blocks in segments if insufficient core space is present to load an entire block.

5.1.2.12 ATN2, SRSS. These commands apply to single or multiblock data sets comprised of real data only. It is required that X and Y have the same block length and number of blocks. The output, Z, will have the same form (block length, number of words, etc.) as X and Y. In the following definitions, z, x, and y are corresponding elements of Z, X, and Y, respectively.

Z= c ATN2(cx X, cy Y) indicates $z = c*\arctan(cx*x/cy*y)$

Z= c SRSS(cx X, cy Y) indicates $z = c*\sqrt{(cx*x)^2 + (cy*y)^2}$

The zero-test parameter established by the last ZERO= e statement is used to identify the situation where both cx*x and cy*y in ATN2 are zero, in which case z is set to zero. The current value of the parameter e is also used to identify zero values of $(cx*x)^2$ and $(cy*y)^2$ in SRSS.

Core Requirement. Two times the block length of X and Y.

5.1.2.13 NUM1. The general form of the command is as follows:

lib Z= c NUM1(cx X, cy Y)

X is a single-block data set of real data containing n abscissa values. Y is a multiblock data set of real data consisting of m blocks containing n ordinate values each. The data set, Z, consists of one block containing m values derived by straight-line integration, each value being the integral of the curve represented by the corresponding block or ordinate values.

The following error codes are produced by NUM1.

Code	Error
1	n less than 2
2	Y does not contain n numbers per block

Core Requirement. Minimum of m + 2n words.

5.1.2.13 XNT1, XNT2, XNT3, XNT4. The form of these commands is as follows:

lib Z= XNT1(XY,A)

XY is a single-block data set containing n pairs of real numbers, (x_i, y_i), defining a piecewise linear function of X. A contains m real numbers representing abscissa values for which y values are to be determined. The output, Z, contains m ordinate values corresponding to the abscissa values in A.

XNT2 is similar to XNT1 except that straight-line interpolation is performed assuming logarithmic (base 10) x and y. XNT3 assumes linear x and logarithmic y. XNT4 assumes logarithmic x and linear y.

The following error codes are produced by this subprocessor:

Code	Error
1	NJ less than 2 in XY
2	NI not equal to 2 for XY
3	Empty A

Core Requirement. m plus two times n words.

5.1.2.13 CBR, CBD, ACBR, ACBD. The form of these commands is as follows:

lib Z= CBR(X,Y)

X is a multiblock data set representing a rectangular matrix. Each block of X contains a column of the matrix. Y may be single or multiblock. CBR performs the matrix product of X and Y. If Y is multiblock, the block length must equal the number of blocks in X. If Y is single-block, TOC item NI must equal the number of blocks in X.

The output Z is a multiblock data set containing n blocks, where n is equal to the number of blocks of Y (multiblock), or the TOC item NJ for a single-block Y. The block length of Z is equal to the block length of X.

CBD is used for the special case where Y is a single-block data set representing a diagonal matrix. In this case Z has the same block size and number of blocks as X.

ACBR and ACBD perform the same functions as CBR and CBD except that each number in the data set X is replaced by its absolute value before the multiplication takes place.

Core Requirement. The number of words contained in Y plus two times the block length of X.

7.3 PSR - PRINCIPAL STRESS GENERATOR

PSR reads multiblock, table-format stress data sets and computes and stores principal stresses in similar data sets. Input data sets have names of the form: ES Eij n3 n4. Output data sets have names of the form: PSTR Eij n3 n4.

PSR is applicable to two-dimensional element types E31-E33 and E41-E43, and three-dimensional solid element types S41-S81.

The order of stress quantities in the data sets produced by PSR for 2-d element types are as follows: 1) ANG, 2) MAX PS, 3), MIN PS, 4) MAX SHR, and 5) SEFF, effective stress.

The order of stress quantities for 3-d solid element types are as follows: 1) NS1, 2) NS2, 3) NS3, 4) SS1, 5) SS2, 6) SS3, 7) ONS, and 8) OSS, octahedral shear stress.

RESET Controls

Name	Default Value	Meaning
INLIB	1	Source library for ES Eij n3 n4 data sets.
OUTLIB	1	Destination library for PSTR Eij n3 n4 data sets generated by PSR.

Execution Control

The PSR processor is executed as follows:

```
@XQT PSR
 "etype"

   or

 "etype"   n3   n4
```

Examples:

```
@XQT PSR
 E43                 (reads     ES     E43    mask    mask)
                     (creates   PSTR   E43    n3      n4  )
        or

 E33    1    2       (reads     ES     E33    1        2  )
                     (creates   PSTR   E33    1        2  )
```

Core Requirements

Block length of input data set x 8/3 (2-d)

Block length of input data set x 7/3 (3-d)

10.2 PLTB- PRODUCTION OF GRAPHICAL DISPLAYS

Function. As shown on Figure 10-1, PLTB (or PLTB/TEK when using Tektronix scopes on the U-1110), is used to produce graphical displays. To cause images corresponding to plot specifications spec1 through spec2 to be displayed, the following command is given:

PLOT spec1, spec2

The form of display resulting from a PLOT command will depend on the current values of an array of execution control parameters which the user selects through the control statements summarized below. Display formats include undeformed plots, static deformations, vibrational modes, buckling modes, or stress displays either from stress data sets produced by GSF, or from table-format data sets created by the user.

When plotting from data sets produced by GSF, complete element stress data sets must be created during the GSF execution; that is, the user must not restrict GSF output to a limited number of element groups, if it is to be read by PLTB.

The description of data set requirements (name and contents) for table-format data sets to be plotted by PLTB is given later in this section.

The PLOT statement and all control statements described below may appear any number of times during a single PLTB execution.

Control Statement	Meaning
DISPLAY=UNDEformed, STATic deformation, VIBRational mode, or BUCKling mode.	Display mode selection. Default is DISPlay=UNDEformed.
DISPLAY=TABLe N1 "etype" nset	Direct display of table-format data from data set identified by the name: N1 "etype" nset ngroup
DISPLAY=SX,TXY, . . . or LAMINATE=SX,node,layer,TXY,... (This form permits stress displays for elements with laminate section properties).	Selected stress or internal load data is displayed. See examples in Section 10.3. A complete list of available stress quantity display symbols is given in Table 10.2-1. The following form is also permitted (underlined quantities may be omitted) DISPLAY=SX/<u>div</u>,<u>node</u>,<u>loc</u>,TXY,... SX is divided by <u>div</u>. <u>Div must be greater than or equal to 1.</u> <u>node</u> indicates the element node (1, 2, etc.) at which the stress is to be evaluated. For 3 and 4 node elements, node 0 is the center of the element. (Note than node must be present for laminate displays.) For 3 and 4 node elements, <u>loc</u> values of 0, 1, and -1 indicate mid, outer, and inner surfaces, corresponding to points C, A, and B (in order) on Figure 7.3-1 (PSF) For laminate section types, "layer" indicates the layer for which stresses are to be displayed. (Must be present).

Control Statement (Cont.)	Meaning (Cont.)
	Note: "node" and "layer" may be omitted for internal load (stress resultant) displays for laminate sections. "layer" is meaningless for internal load displays. Principal stress quantities are not available for laminate sections.
DNORM=dnorm	When plotting deformed structures, (i.e., if DISPLAY=STAT, or VIBR, or BUCK), joint displacements are normalized to dnorm. This command must be given, since there is no default value.
INLIB=inlib SET=nset CON=ncon CASES=case1, case2 or VECTORS=vect1, vect2 *(The control statements CASES and VECTORS are synonymous.)*	The following source data, if needed as a result of the prevailing DISPLAY statement, will reside in inlib. STAT DISP nset ncon VIBR MODE nset ncon BUCK MODE nset ncon STRS EIJ nset i, for i=case1, case1+1, ... case2 Default values are INLIB=1, SET=1, CON=1, CASES=1.
OPTIONS=n1, n2, ...	List of options. See Table 10.2-2

Table 10.2-1 Summary of Available Stress Display Symbols

Symbol	Meaning	Applicable Element Type E21	E31 E41	E32 E42	E33 E43	E44	LAM
SMAX	Maximum P/A + My/I beam stress	X					
SMIN	Minimum " " " "	X					
P/A	Axial beam stress	X					
S1	Dir. 1 beam shear stress	X					
S2	Dir. 2 " " "	X					
TS	Beam twisting stress	X					
SX	Normal stress, element x-dir.		X	X	X		x
SY	" " " y-dir.		X	X	X		x
TXY	In-plane shear stress		X	X	X	X	x
PS1	Maximum principal stress		X	X	X		
PS2	Minimum " "		X	X	X		
TMAX	Maximum shear stress		X	X	X		
ANG	Angle between x-axis and PS1 vector		X	X	X		
NX	Normal stress resultant, x-dir.		X		X		x
NY	" " " y-dir.		X		X		x
NXY	In-plane shear stress resultant		X		X	X	x
PN1	Maximum principal stress resultant		X		X		x
PN2	Minimum " " "		X		X		x
NMAX	Maximum shear stress resultant		X		X		x
NANG	Angle between x-axis and PN1 vector		X		X		x
MX	Bending stress resultant, x-dir.			X	X		x
MY	" " " y-dir.			X	X		x
MXY	Twisting stress resultant			X	X		x
QX	Transverse shear resultant, x-dir.			X	X		x
QY	" " " y-dir.			X	X		x

The P/A display is also available for E23 and E24 elements.

Table 10.2-2 Meaning of OPTION Numerical Codes

Description of Option	Option Numeric Code	
Specification Control		
Plot error free specifications only	1	
Plot all specifications ignoring error status	2	
Plot all specifications appearing on a single PLOT command to the same scale	3	
Frame Labeling		
Omit deformation identification label	4	
Omit specification titles	5	
Omit "SPEC" identification	6	
Omit "SCALE"	7	
Omit all frame labeling	8	
Collapse margin and omit all labels	9	
Geometric Construction		
Dotted deformed structure	24	
Curved lines, deformed structure	25	
Superimpose deformed/undeformed structures	26	
Dotted undeformed structure	27	
Plot Content	Large Char.	Small Char.
Joint numbers displayed	10	11
Joint elimination order displayed	12	13
Joint labels displayed	14	15
Element index numbers displayed	16	17
Element group-index numbers displayed	18	19
Element section property group displayed	20	21
Element stress display size	28	29
3-Node elements are shaded	22 (no size control)	
4-Node elements are shaded	23 (no size control)	
Automatic Hardcopy & Frame Advance		
Tektronix version only	30	

Notes:
- Select no more than one from options 10,11,12,13.
- Select no more than one from options 14,15.
- Select no more than one from options 16,17,18,19,20,21,28,29.
- Options 16-21 may only be used in conjunction with DISPLAY=UNDEformed.

Table-Format Data Sets

A separate data set must be constructed for each group of each element type which is to be plotted. The data set may contain either one value per element, assumed to be at the center, values at each of the nodes, or values at each of the nodes plus the center of the element (NNODES+1 values). Stress displays may be created for 2-node, 3-node, or 4-node elements. Depending on the number of values contained in the data set per element, stresses will be displayed at the center of the element, at the corners, or both, accordingly.

The data sets may be created in AUS and must have names of the following form:

"name1" "etype" nset ng

where,

"name1" is any name supplied by the user, which may describe the quantity contained in the data set,

"etype" is a valid 2-d element type (E21,E23,E24,E31, E32,E33,E41,E42,E43,E44),

nset is supplied by the user and may correspond to a load set designation, and

ng is the element group number to which the data set corresponds.

Example: ES E43 1 1

Descriptive information for frame labelling purposes may be placed in a data set named:

TABL TITL nset mask

where nset refers to the nset value in the stress data set names. If such a data set is present, the contents (up to 60 characters) will be displayed at the top of the plot frame.

The data set plotting is invoked in PLTB or PLTB/TEK with the DISPLAY command as follows:

DISPLAY=TABLe NAME1 "etype" nset

where, NAME1 "etype" nset, refers to the first three names of the data set desired to be plotted. The fourth name, ng, is not required on the DISPLAY command since a plot specification may contain elements from different groups, and the data set corresponding to the group designation of the elements being plotted is read automatically.

Rigid Links

Rigid links, if any (see BRL discussion in TAB) are ignored by PLTB in generating plots.

Reset Controls

RESET NDEV=4010 (for 4010 models)

RESET NDEV=4014 (for 4014 models <u>without</u> enhanced graphics)

Note: Defaults to 4014 models <u>with</u> enhanced graphics.

RESET CHRS=n (defaults to 2)

Note: This reset applies to 4014 models with enhanced graphics only.

CHARACTER SIZE 1	Optional "large" character size
CHARACTER SIZE 2	Default "large" character size (Options 10, 12, etc.)
CHARACTER SIZE 3	Optional "large" character size
CHARACTER SIZE 4	"Small" character size (options 11, 13, 15, etc.) (May also be selected for "large" character size, CHRS=4)

Core requirements

Where J is the number of joints in the structure, the data space required by PLTB is a follows:

For plotting undeformed structures:	2000 + J
For plotting deformed structures:	2000 + 13J
For plotting stresses:	2000 + J + the length of one block of input stress data, plus the length of the shell section property table (SA)

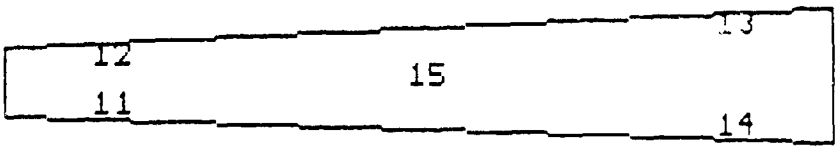
```
@XQT AUS
 OUTLIB=2
 ALPHA:TABL TITL 3 1
 1'SX AT NODES AND CENTER
 TABL(NI=5,NJ=1):SX5 E43 3 1
 I=1,2,3,4,5:J=1:11.0,12.0,13.0,14.0,15.0
```

```
@XQT PLTA
 SPEC 2
 S2TITL' E43 ELEMENTS
 VIEW 1
 E43

@XQT PLTB
 OPTION=29
 INLIB=2
 DISPLAY=TABL SX5 E43 3
 PLOT 2
```

SX AT NODES AND CENTER

ID= 1/1/1



PRECEDING PAGE BLANK NOT FILMED

10-3-1 to 10-3-12

SPEC
2.1

E43 ELEMENTS

0 1
SCALE

C-7