

**A Thesis Submitted for the Degree of PhD at the University of Warwick**

**Permanent WRAP URL:**

<http://wrap.warwick.ac.uk/97339>

**Copyright and reuse:**

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk)

is drawn to the fact that the  
his thesis rests with its author.  
y of the thesis has been supplied  
that anyone who consults it is  
recognise that its copyright rests  
or and that no quotation from  
no information derived from it  
ished without the author's prior  
nt.

18344/77

VOL II

archent, BG

PP 49

THE  
UNIVERSITY  
OF  
WARWICK  
LIBRARY

COPY 1

Interactive Computer Programs  
for the  
Computer Aided Design  
of  
Linear Microwave Circuit  
(Programs)

Author  
Brian G. Marchent, B. Sc. (Hons.)



Date : July 1973

A thesis submitted at the  
University of Warwick for  
the degree of Doctor of  
Philosophy.

73-12459 / Doc 3  
DIS  
1973  
Ph D

\*\* ICL 4130 DE92 SYSTEM : SLAVE BACK | CORE 30K | VOL 9 DATE 29 06 73 : TIME 18.52.12

&JOBlES/R001/BM1

&ASSIGN;50;DC161CHAIN1,ES/R001/1 CPU TIME = 0000 00,054

&ASSIGN;52;DC161MICRO3,ES/R001/1 CPU TIME = 0000 00,086

&ASSIGN;54;DC161MIC3D,ES/R001/1 CPU TIME = 0000 00,118

&ASSIGN;56;DC161ML1J,ES/R001/1 CPU TIME = 0000 00,150

&ASSIGN;58;DC161ML1D,ES/R001/1 CPU TIME = 0000 00,182

&ASSIGN;60;DC161ML1A,ES/R001/1 CPU TIME = 0000 00,213

&ASSIGN;62;DC161MLC,ES/R001/1 CPU TIME = 0000 00,245

&LINES: 100001 CPU TIME = 0000 00,272

&TIME: 51 51 CPU TIME = 0000 00,300

&PRINT;501 CPU TIME = 0000 00,345  
PRINT  
ACIO

PRINT-UP OF CHAIN1 USER ES/R001/

```
CHAIN1;
"BEGIN" "COMMENT" READ MAXIMUM SIZE OF CIRCUIT;
  "INTEGER" P,NODES,NETWORKS,LOADS,VARIABLES;
  "BOOLEAN" FAULT;
  "PROCEDURE" ERROR(N); "VALUE" N; "INTEGER" N;
  "BEGIN" "PRINT" //L2SH0'CHAIN1  ERROR, SAMELINE,DIGITS(4),
    N; "I FAULT:="TRUE" "END";
  "PROCEDURE" TITLE;
  "BEGIN" "INTEGER" C,M,N,WORD1 "INTEGER" "ARRAY" A[1:63];
    "WORD" LAST; M:=0; N:=2; WORDS:=130;
    SPACE:ADVANCE(6); C:=DECODE(6); "IF" C<=62 "THEN" "GOTO" SPACE;
    CHAR:LAST:=C=27; "IF" C>61 "THEN" "GOTO" ADV;
    SHIFT:="CODE" %WORD;
      %LDWILS1
      %SMLC
      %ADDESC
      %STSWORD1 N:=N+1;
    "IF" N=4 "THEN"
      "BEGIN" M:=M+1; A[M]:=WORD1; WORDS:=N:=0;
      "IF" M=62 "THEN" "BEGIN" EROR(1); LAST:="TRUE" "END";
      "IF" LAST "THEN" GOTO END TTL;
    "ENDIF"; "IF" LAST "THEN" "BEGIN" C:=0; "GOTO" SHIFT "END";
    ADV:ADVANCE(6); C:=DECODE(6); "GOTO" CHAR;
  END TTL; ACM:=1; I:=536480; M:=1; OUTSTRING(A,M);
  "END" OF TITLE;
  SAMELINE1 DIGITS(4); FAULT:="FALSE"; "PRINT" //F11 TITLE;
  "READ" NODES,NETWORKS,LOADS,VARIABLES;
  "PRINT" //L1NUMBER OF NODES =1, NODES,
    //L1NUMBER OF NETWORKS =1, NETWORKS,
    //L1NUMBER OF PORTS = 2,
    //L1NUMBER OF LOADS =1, LOADS,
    //L1NUMBER OF VARIABLES =1, VARIABLES;
  "FOR" P:=NODES,NETWORKS,LOADS,VARIABLES "DO"
    "IF" P>LE"0 "THEN" ERROR( 2);
  "IF" "NOT" FAULT "THEN"

  "BEGIN" "COMMENT" SET ARRAYS FOR CIRCUIT DESCRIPTION;
    "REAL" FSTART,FSTEP,FSTOP,FREQ;
    "ARRAY" VNET[1:4,1:NETWORKS],VLOAD[1:3,0:LOADS],
      VAR[0:3,1:VARIABLES];
    "INTEGER" LIM,STEP,LENG,MAXLEV,PORT1,PORT2,LOOPVAR;
    "INTEGER" "ARRAY" NETCONC-NETWORKS;NETWORKS],NETTYPE[1:NETWORKS];
      LOAD,LOADTYPE[0:LOADS],VARNET,VARPAR[1:VARIABLES];
      OPTION[0:7],STOREA,STOREB[0:15-NETWORKS];
    "BOOLEAN" NEWCIT,TERM;
    "BOOLEAN" "ARRAY" NODETYPE[1:NODES];

  "PROCEDURE" SET NEW CIRCUIT;
  "BEGIN" "INTEGER" P;
    "FOR" P:=1 "STEP" 1 "UNTIL" NODES "DO" NODETYPE[P]:=TRUE;
    "FOR" P:=NETWORKS "STEP" 1 "UNTIL" NETWORKS "DO" NETCON[P]:=0;
    PORT1:=PORT2:=OPTION[0]:=LOADTYPE[0]:=0;
    "FOR" P:=0 "STEP" 1 "UNTIL" LOADS "DO" LOAD[P]:=0;
    "FOR" P:=1 STEP" 1 "UNTIL" VARIABLES "DO" VARNET[P]:=0;
    STEPI:=NETWORKS+1; STEP1:=(STEP+STEP+STEP)"DIV"NODES+1;
    LIM:=STEP*NODES;
  "END" OF SET NEW CIRCUIT;
```

```

"PROCEDURE" READ DATA1
"BEGIN" "INTEGER" K,L,M,N,P,01
    "BOOLEAN" TEST,CHECK;
    "SWITCH" S:=L1 NODES,L2 NETWORK,L3 LOAD,L4 PORT,L5 VARIABLE,
        L6 OPTIONS,L7 TITLE,L8 NEW CIRCUIT,L9 FREQ,
        L10 START,L11 RUN ENDI

"PROCEDURE" WORDS
"BEGIN" "INTEGER" C,M,WORD; "INTEGER" APRAY A[1:4]
    CHECK:=TRUE; M:=WORD:=0;
    "FOR" C:=DECODE(6) "WHILE" C<3 "DO" ADVANCE(6)
    "FOR" C:=DECODE(6) "WHILE" C>2 "DO"
    "BEGIN" ADVANCE(6); M:=M+1
        "IF" C>61 "THEN" ERROR(11) "ELSE"
        "IF" M<5 "THEN" "CODE" XLDWORD
            XLDKIL$1
            XSMLC
            XADDSC
            XSTRWORD "END"; P:=P+1
    "FOR" M:= -4523739, -4563657, -5048220, -3998540, -2483031,
        -4256471, -2974420, -7431005, -6608527, -3192718,
        154532, 3299, -3253079, -4055903, 3043 "DO"
        "IF" M=WORD "THEN" "GOTO" FOUND "ELSE" P:=P+1
    FOUND:CHECK:= P"GE"K "AND" P"LE"L1
        "IF" NOT CHECK "THEN" "BEGIN" ERROR(12); A[1]:= -2163548;
            A[2]:=0; A[3]:=WORD; A[4]:=4000;
            M:=1; OUTSTRING(A,M) "END"
    "END" OF WORDS

"PROCEDURE" CHNODE
"BEGIN" CHECK:= >0 "AND" M"LE"NODES; "IF" "NOT" CHECK "THEN"
    "BEGIN" ERROR(13); "PRINT" 'NODE ',M "END" "END"

"PROCEDURE" CHNETWORK
"BEGIN" CHECK:= >0 "AND" N"LE"NETWORKS; "IF" "NOT" CHECK "THEN"
    "BEGIN" ERROR(14); "PRINT" 'NETW ',N "END" "END"

"PROCEDURE" CHNETTYPE
"BEGIN" CHECK:=P>0 "AND" P<11; "IF" "NOT" CHECK "THEN"
    "BEGIN" ERROR(15); "PRINT" 'NETT ',P "END" "END"

"PROCEDURE" CHLOADS
"BEGIN" CHECK:= >0 "AND" N"LE"LOADS; "IF" "NOT" CHECK "THEN"
    "BEGIN" ERROR(16); "PRINT" 'LOAD ',N "END" "END"

DIGITS(4); TERM:=NEWCITI="FALSE";
DATA:=ADVANCE(6); "IF" DECODE(6)"NE"2 "THEN" "GOTO" DATA1
    K:=1; L:=11; WORD; "IF" P"LE"4 "THEN" NEWCITI="TRUE"
    "GOTO" "IF" CHECK "THEN" S[P]; "ELSE" DATA1
    L1 NODE;K:=13; L:=14; WORD;
    "IF" CHECK "THEN" "BEGIN" TEST:=P+14
READ NODE;"READ" M;
    "IF" M"NE"0 "THEN"
        "BEGIN" CHNODE; "IF" CHECK "THEN"
            "BEGIN" NODETYPE[M]:=TEST; "GOTO" READ NODE "END" "ENDIF"
            "END"; "GOTO" DATA1
    L2 NETWORK;"READ" N,M,P; CHNETWORK;
    "IF" CHECK "THEN"

```

```

"BEGIN" CHNODEI NETCON[N]:=M; M:=P; CHNODEI NETCON[-N]:=M)
"READ" P; CHNETTYPEI NETTYPE[N]:=P;
"READ" VNET[1,N],VNET[2,N],VNET[3,N],VNET[4,N]
"END" "ELSE" NETCON[N]:=NETCON[-N]:=0
"END"; "GOTO" DATA1
L3 LOAD;"READ" N,M; CHLOAD;"IF" CHECK "THEN"
"BEGIN" LOADENJ:=M; CHNODEI K:=12; L:=15; WORDI
"IF" CHECK "THEN" "BEGIN" LOADTYPE[N]:=15-P;
"READ" VLOADC1,N],VLOADC2,N],VLOADC3,N] END"
"END"; "GOTO" DATA1
L4 PORT;"READ" P,M
"IF" P=1 "THEN" PORT1:=M "ELSE" "IF" P=2 "THEN" PORT2:=M
"ELSE" "BEGIN" ERROR(17); "PRINT" 'PORT ',P "END"
CHNCDEI "GOTO" DATA1
L5 VARIABLES;"READ" M; CHECK1=M>0 "AND" M<LE"VARIABLES"
"IF" CHECK "THEN"
"BEGIN" K:=2; L:=3; WORD; TEST:=P=2
"IF" CHECK "THEN"
"BEGIN" "READ" N; "IF" N=0 "THEN"
"BEGIN" VARNETCMJ:=0; "GOTO" DATA "END"
"IF" TEST "THEN" CHNETWORK "ELSE" CHLOADS
VARNETCMJ:="IF" TEST "THEN" N "ELSE" -N
K:="IF" TEST "THEN" 4 "ELSF" 3
"READ" Q; VARPARCMJ:=Q;
"IF" Q>0 "AND" Q<LE"K "THEN"
"BEGIN" "READ" VAR[1,M],VAR[2,M],VAR[3,M];
VAR[3,M]:=VAR[3,M]-0.01*VAR[2,M] "END"
"ELSE" "BEGIN" ERROR(19); "PRINT" 'PARA ',Q "END"
"END" "END"
"ELSE" "BEGIN" ERROR(18); "PRINT" 'VARI ',M "END"
"GOTO" DATA1
L6 OPTIONS;P:=0; NEXTI;"READ" N;
"IF" N<0 OR N>52 "THEN"
"BEGIN" ERROR(20); "PRINT" 'OPTI ',N "END" "ELSE"
"BEGIN" OPTION[P]:=IF" PK7 "THEN" N "ELSE" 0;
"IF" N=0 "THEN" "GOTO" DATA1
"IF" P=7 "THEN" "BEGIN" ERROR(21); "GOTO" DATA1"END"
P:=P+1 "END"; "GOTO" NEXT1
L7 TITLE;TITLE;"GOTO" DATA1
L8 NEWCIRCUIT;NEWCITI:="TRUE"; "GOTO" DATA1
L9 FREQ;"READ" FSTART,FSTEP,FSTOP;
FSTOPI:=1.0000001*FSTOP; "GOTON" DATA1
L10 START;"IF" PORT1=0 OR PORT2=0 "THEN" ERROR(22)
"IF" OPTION[0]=0 "THEN" ERROR(23)
"IF" (FSTOP-FSTART)/FSTEP>100 "THEN" ERROR(24)
"GOTO" START1
L11 RUN END;TERM1:="TRUE"
START;"END" OF READ DATA1

"PROCEDURE" PATH SEARCH;
"BEGIN" "INTEGER" I,J,K,L,M,N,P,Q,
      STNODE,NODE,EDNODE,START, LAST,TYPE,LEVEL
      "INTEGER" "ARRAY" LTNOD,LIST[0:LIM],NODUS[1:NODES],
      NETUSCOINNETWORK6]

"PROCEDURE" SWAP(E,F) "VALUE" E,F; "INTEGER" E,F
"BEGIN" "INTEGER" K,L
      K:=STOREACJ; L:=STOREAFJ
      STOREACB:="IF" L=F "THEN" E "ELSE" L
      STOREAF:="IF" K=E THEN" F "ELSE" K

```

```

"BEGIN" CHNODEI NETCON[N]:=M; M:=P; CHNODEI NETCON[N]:=M
"READ" P; CHNETTYPEI NETTYPE[N]:=P;
"READ" VNET[1,N],VNET[2,N],VNET[3,N],VNET[4,N]
"END" "ELSE" NETCON[N]:=NETCON[N]:=0
"END"; "GOTO" DATA1
L3 LOADI "READ" N,M; CHLOADI "IF" CHECK "THEN"
"BEGIN" LOADCNJ:=M; CHNODEI K:=12; L:=15; WORDI
"IF" CHECK "THEN" "BEGIN" LOADTYPE[N]:=15-P;
"READ" VLOADE[1,N],VLOADE[2,N],VLOADE[3,N] ENDH
"END"; "GOTO" DATA1
L4 PORTI "READ" P,M
"IF" P=1 "THEN" PORT1:=M "ELSE" "IF" P=2 "THEN" PORT2:=M
"ELSE" "BEGIN" ERROR(17); "PRINT" 'PORT ',P "ENDH"
CHNCDEI "GOTO" DATA1
L5 VARIABLEI "READ" M; CHECKI:=M>0 "AND" M<LE"VARIABLES"
"IF" CHECK "THEN"
"BEGIN" K:=2; L:=3; WORDI TESTI:=P=2
"IF" CHECK "THEN"
"BEGIN" "READ" N; "IF" N<0 "THEN"
"BEGIN" VARNETCMJ:=0; "GOTO" DATA "END";
"IF" TEST "THEN" CHNETWORK "ELSE" CHLOADI
VARNETCMJ:="IF" TEST "THEN" N "ELSE" -N
K:="IF" TEST "THEN" 4 "ELSE" 3
"READ" Q; VARPARCI:=Q;
"IF" Q>0 "AND" Q<LE"K "THEN"
"BEGIN" "READ" VAR[1,M],VAR[2,M],VAR[3,M];
VAR[3,M]:=VAR[3,M]-0.01*VAR[2,M] "END"
"ELSE" "BEGIN" ERROR(19); "PRINT" 'PARA ',Q "ENDH"
"END" "END"
"ELSE" "BEGIN" ERROR(18); "PRINT" 'VARI ',M "ENDH"
"GOTO" DATA1
L6 OPTIONSI:P=0; NEXTI "READ" N
"IF" N<0 OR" N>52 "THEN"
"BEGIN" ERROR(20); "PRINT" 'OPTI ',N "ENDH" "ELSE"
"BEGIN" OPTION[P]:=;"IF" P<7 "THEN" N "ELSE" 0;
"IF" N<0 "THEN" "GOTO" DATA1
"IF" P=7 "THEN" "BEGIN" ERROR(21); "GOTO" DATA1 "ENDH"
P:=P+1 "END"; "GOTO" NEXTI
L7 TITLEI TITLEI "GOTO" DATA1
L8 NEWCIRCUITI NEWCITI:="TRUE"; "GOTO" DATA1
L9 FREQI "READ" FSTART,FSTEP,FSTOPI
FSTOPI:=1.0000001*FSTOPI "GOTO" DATA1
L10 STARTI "IF" PORT1=0 OR" PORT2=0 "THEN" ERROR(22)
"IF" OPTION[0]=0 "THEN" ERROR(23)
"IF" (FSTOP-FSTART)/FSTEP>100 "THEN" ERROR(24)
"GOTO" STARTI
L11 RUN ENDITERM:="TRUE"
STARTI "END" OF READ DATA1

"PROCEDURE" PATH SEARCHI
"BEGIN" "INTEGER" I,J,K,L,M,N,P,Q,
        STNODE,NODE,EDNODE,START,LAST,TYPE,LEVELI
        "INTEGER" "ARRAY" LTNOD,LIST[0:I|M],NODUS[1|NODES],
        NETUSC0|NETWORKS|;

"PROCEDURE" SWAP(E,F) "VALUE" E,F; "INTEGER" E,F
"BEGIN" "INTEGER" K,L;
        K:=STOREAC[E]; L:=STOREAC[F];
        STOREAC[E]:="IF" L=F "THEN" E "ELSE" L
        STOREAC[F]:="IF" K=E THEN" F "ELSE" K

```

```

L:=STOREB[C]; STOREB[E]:=STOREB[F]; STOREB[F]:=L;
"FOR" L:=E, F "DO" "BEGIN" K1=STOREA[L];
"IF" K1>0 "THEN" "BEGIN" STOREA[K]:=L;
"IF K1<=0 "THEN" STOREB[STOREB[L]]:=L "END" "END";
"END" OF SWAP;

"PROCEDURE" ENTER;
"BEGIN" N:=STEP*NODE;
"FOR" K1=LTNOD[N] "WHILE" K1<NE0 "DO"
N1="IF" N<LIM "THEN" N+1 "ELSE" 0;
LTNCN:=NODE1 LISTEN1:=N;
"END" OF ENTER;

"COMMENT" SET INITIAL VALUES;
"FOR" P:=0 "STEP" 1 "UNTIL" LIM "DO" LTNOD[P]:=0;
"FOR" P:=1 "STEP" 1 "UNTIL" NODES "DO" NODUS[P]:=0;
"FOR" P:=1 "STEP" 1 "UNTIL" NETWORKS "DO" NETUSE[P]:=0;

"COMMENT" SET UP HASH TABLE;
M:=0; NODE1:=PORT1; ENTER; NODE1:=PORT2; ENTER;
"FOR" P:=NETWORKS "STEP" 1 "UNTIL" -1 "DO" "FOR" M:=P, -P "DO"
"BEGIN" NODE1:=NETCON[M]; "IF" NODE1<NE0 "THEN" ENTER "END";

"COMMENT" PATH SEARCH;
DIGITS(4); P1:=0; LEVEL1:=0; MAXLEV1:=3; STNODE1:=PORT1;
"GOTO" FIND PATH;
BUILD1;"IF" Q>GE[P] "THEN" "GOTO" FIN1; LEVEL1:=STOREA[Q]; LAST1:=STOREB[Q];
INCRIQ1:=Q+1; "IF" Q=LAST1 "THEN" "BEGIN" Q1:=Q+2; "GOTO" BUILD1 "END";
"IF" STOREA[Q]>0 "THEN" LEVEL1:=LEVEL-2;
"IF" STOREA[Q]<=0 "THEN" "GOTO" INCRI;

"COMMENT" SHIFT RETURN POINTERS FROM PARALLEL PATHS;
N:=0; "FOR" M:=N+1 "WHILE" STOREA[M]<0 "DO" NI:=M;
"FOR" NI:=N+1 "WHILE" STOREA[N]<NE0 "AND" NI<LAST1 "DO"
"BEGIN" SWAP(N,NI); LEVEL1:=LEVEL-2; Q1:=Q+1 "END";

PATH1; STNODE1:=STOREB[Q];
"COMMENT" START FIND PATH THROUGH CIRCUIT;
FIND PATH; NODE1:=STNODE1; START1:=P1; P1:=P+1; TYPE1:=0; J1:=1;
NEXT1; M:=0; K1:=2; NI:=STEP*NODE;
"FOR" L1=LTNOD1 "WHILE" L1<NE0 "DO"
"BEGIN" "IF" L1<NODE1 "THEN"
"BEGIN" K1:=K+1; L1:=LISTEN1;
"IF" NETUSC1<0 "THEN" L1:=0 "THEN" M1:=L1
END; NI="IF" N<LIM "THEN" N+1 "ELSE" 0; "END";
"IF" J1=P1 "THEN" "BEGIN" BLANK1; "IF" K1>0 "THEN"
"BEGIN" STOREA[P1]:=1; STOREB[P1]:=NODE1;
P1:=P+1; K1:=K-1; "GOTO" BLANK1 "END" "END";
"IF" START1=0 "THEN" "BEGIN" "IF" NODE1=PORT2 THEN"
"GOTO" FST LINK "END";
"IF" M1<0 "THEN" "GOTO" BACK1;
STOREA[P1]:=0; STOREB[P1]:=M1; NETUSC1="IF" M1>0 "THEN" H "ELSE" =H1:=1;
EDNODE1:=NODE1:=NETCON[-H1]; M1:=NODUS[NODE1]; J1:=P1+1;
"IF" M1<0 "THEN" "BEGIN" NODUS[NODE1]:=1; "GOTO" NEXT1 "END";
"IF" M1>0 "THEN" "GOTO" "IF" STNODE1=EDNODE1 "THEN" LOOP "ELSE" PARPATH1;

BACK1; P1:=P+1; "IF" P=START1 "THEN" "GOTO" "IF" NODUS[STNODE1]<0
"THEN" ARM "ELSE" RESET1;
"IF" STOREA[P1]<NE0 "THEN" "GOTO" BACK1;
NODE1:=NETCON[STOREB[P1]]; "GOTO" NEXT1;

"COMMENT" FIRST LINK THROUGH BRUGUT FOUNDRY;

```

```

FST LINK1::=K1::=L1::=0; TYPE1::=1
  "FOR" N::=1 "STEP" 1 "UNTIL" LOADS "DO" "BEGIN" M::=LOADEN1;
    "IF" M>PORT1 "THEN" K1=N; "IF" M>PORT2 "THEN" L1=N "END";
    "IF" K1=0 "OR" L1=0 "THEN" ERROR(102); "GOTO" RESET;

"COMMENT" PARALLEL PATH FOUND IN CIRCUIT
PARPATH:TYPE1::="IF" NODETYPE[STNODE] "THEN" 8 "ELSE" 6;
  "IF" NODETYPE[EDNODE] "THEN" TYPE1::TYPE1+1;
  N1::=NODUS[EDNODE]; NODUS[EDNODE]::=1-N1;
  K1::="IF" N>LAST "THEN" 1 "ELSE" 0; L1::=0;
  STOREA[N1]::=Q1 STOREB[N1]::=P1 "GOTO" SET1;

"COMMENT" LOOP FOUND IN CIRCUIT
LOOP:TYPE1::="IF" NODETYPE[STNODE] "THEN" 5 "ELSE" 4;
  J1::=N1::=0; "FOR" M::=J1+1 STOREA[M]::=DO J1::=M;
  I1::=STOREA[J1]::=STOREB[J1]::=0; "GOTO" SET1;

"COMMENT" BRANCH ARH FOUND IN CIRCUIT
ARM:TYPE1::="IF" NODETYPE[STNODE] "THEN" 3 "ELSE" 2; N1::=Q1 P1::=J1;
  "IF" STARTED "THEN" "BEGIN" ERROR(101); "GOTO" RESET "END"; M::=P1
  I1::=K1::=L1::=0; "FOR" M::=M+1 "WHILE" M>START "DO"
  "IF" STOREA[M]::=0 "THEN" "BEGIN" "IF" STOREB[M]::=EDNODE "THEN"
  "BEGIN" P1::=M+1; STOREA[P1]::=-1; STOREB[P1]::=EDNODE;
  P1::=P1+1; "GOTO" SET "END" "END";
  "FOR" M::=1 "STEP" 1 "UNTIL" LOADS "DO"
  "IF" LOADEN1::=EDNODE "THEN" L1::=M1

"COMMENT" RESET NEW NETWORKS AND NODES STORED TO THE STORED STATES
SET1:STOREA[Q1]::=N1 STOREB[Q1]::=START1
RESET:STOREA[START1]::=LEVEL-1; STOREB[START1]::=-P1 STOREA[P1]::=TYPE1
  STOREB[P1]::=-Q1 STOREA[P+1]::=K1 STOREB[P+1]::=L1
  N1::=0; "IF" TYPE<6 "THEN" N1::=1 "ELSE" LEVEL1::LEVEL+2;
  N1::=LEVEL+N1; "IF" N>MAXLEV "THEN" MAXLEV::=N1
  N1::=START1; "FOR" N1::=N+1 "WHILE" N<P "DO"
  "BEGIN" K1::=STOREA[N1]; L1::=STOREB[N1];
  "IF" K1=0 THEN NETUSC["IF" L>0 "THEN" L::="ELSE" -L]::=-1
  "ELSE" NODUS[L]::=-N END;
  "FOR" N1::=1 "STEP" 1 "UNTIL" NODES "DO"
  "IF" NODUS[N1]>0 "THEN" NODUS[N1]::=0;
  "FOR" N1::=1 "STEP" 1 "UNTIL" NETWORKS "DO"
  "IF" NETUS[N1]>0 "THEN" NETUS[N1]::=0;
  P1::=P+2;
"COMMENT" END OF FIND PATH THROUGH CIRCUIT

  "IF" STARTED "THEN" "GOTO" BUILD;

"COMMENT" NEST PATH CORRECTLY
  K1::=0; L1::=LAST1; N1::=STOREA[Q1]; I1::=0;
NEXT1:"IF" N>K "THEN" "BEGIN" "IF" N<L "THEN" "GOTO" NEXT2 "END";
  M::=Q-1; "IF" STOREA[M]<M "THEN"
    "BEGIN" ERROR(103); "PRINT" "(LINE",N1,"GOTO" NEXT2 "END";
  K1::=STOREA[M]; I1::=STOREA[K1]; SWAP(M,Q); I1::=I1+1; Q1::=M; "GOTO" NEXT1;
NEXT2:M::=Q-1; J1::=STOREA[M];
  "IF" J>K "AND" J<L "THEN"
  "BEGIN" SWAP(M,Q); I1::=I1+1; Q1::=M; "GOTO" NEXT2 "END";
  Q1::=Q+1;

  Q1::=Q+1; "IF" STOREA[Q1]<0 "THEN" "GOTO" PATH; Q1::=Q-1;

"COMMENT" CHECK FOR PATHS CROSSING IN CIRCUIT;

```

```

M:=N:=Q; "FOR" N:=N+1 "WHILE" N<LAST "DO"
"BEGIN" K:=STOREA[N]; "IF" K>0 "THEN"
    "BEGIN" "IF" K<=M "THEN" M:=K
        "ELSE" "BEGIN" ERROR(104); "PRINT" 'LINE',N "END"
    "END" "END";
    "GOTO" INCR;

    FIN: LENGTH:=P-1;
"END" OF PATH SEARCH;

"PROCEDURE" PRINT CIRCUITS
"BEGIN" "INTEGER" K,L,M,P;
    "PROCEDURE" BL TYPE;
    "BEGIN" "SWITCH" TY:=TY1,TY2,TY3,TY4,TY5,TY6,TY7,TY8,TY9;
        "GOTO" TY[M];
        TY1;"PRINT" /MAIN LINK'; "GOTO" TY END;
        TY2;"PRINT" /BRANCH ARM TYPE Z'; "GOTO" TY END;
        TY3;"PRINT" /BRANCH ARM TYPE Y'; "GOTO" TY END;
        TY4;"PRINT" /PARALLEL LOOP TYPE Z'; "GOTO" TY END;
        TY5;"PRINT" /PARALLEL LOOP TYPE Y'; "GOTO" TY END;
        TY6;"PRINT" /PARALLEL PATH TYPE Z'; "GOTO" TY END;
        TY7;"PRINT" /PARALLEL PATH TYPE H'; "GOTO" TY END;
        TY8;"PRINT" /PARALLEL PATH TYPE G'; "GOTO" TY END;
        TY9;"PRINT" /PARALLEL PATH TYPE Y';
    TY END;"END" OF BLOCK TYPE;
    "PROCEDURE" LOD(LOAD); "VALUE" LOAD; "INTEGER" LOAD;
        "IF" LOAD<>0 "THEN"
            "PRINT" /LOAD',LOAD,' TYPE',LOADTYPE[LOAD],
            /' PARAMETERS',VLOAD[1,LOAD],VLOAD[2,LOAD],VLOAD[3,LOAD];
    P:=0; "PRINT" //L2'CIRCUIT DESCRIPTION';
    PREFIX(' '); DIGITS(3); FREEPOINT(6);
NEXT:K:=STOREA[P]; L:=STOREB[P];
    "IF" K<0 "THEN" "BEGIN" "IF" L<0 "THEN"
    "BEGIN" M:=STOREA[-L]; "PRINT" //L3'; BL TYPE "END" "END";
    "PRINT" //L',P;
    "IF" K=0 "THEN" "BEGIN" "IF" L<0 "THEN"
    "BEGIN" "PRINT" /' NETWORK',L,' NODES',NETCONC[L],/' TO',
        NETCONC[-L]; "IF" L<0 "THEN" L:=L;
        "PRINT" /' TYPE',NETTYPE[L],/' PARAMETERS',
        VNET[1,L],VNET[2,L],VNET[3,L],VNET[4,L];
    "END" "END" "ELSE" "IF" L<0 "THEN"
    "BEGIN" "IF" L<0 "THEN" "PRINT" /' END AT LINE',-L,' LEVEL',=K
        "ELSE" "PRINT" /' POINTER ON NODE',L"END";
    "ELSE" "IF" L<=0 "THEN"
    "BEGIN" M:=K; "PRINT" /' END OF ',BL TYPE;
        "IF" M>1 "THEN" "PRINT" /' ORIGIN AT LINE',-L; PI:=P+1;
        "IF" M<4 "THEN" "BEGIN" K:=STOREA[P]; L:=STOREB[P];
        "PRINT" //L'; "IF" M<2 "THEN"
        "BEGIN" "PRINT" /' PORT 1 ',1 LOD(K);
        "PRINT" //L'PORT 2 ',1 LOD(L) "END";
    "ELSE" LOD(L) "END";
    "END" "ELSE" "IF" K>=P "THEN" "BEGIN" M:=STOREA[-STOREB[L]];
        "PRINT" /' START OF ',BL TYPE;
        "PRINT" /' AT LINE',L,' RETURN AT LINE',K "END";
        "ELSE" "PRINT" /' RETURN FROM LINE',K;
    PI:=P+1; "IF" P<LENG "THEN" "GOTO" NEXT; "PRINT" //L2';
"END" OF PRINT CIRCUITS;

"PROCEDURE" SET VARIABLES(STEP,RESET);

```

```

M:=N:=Q:= "FOR" N:=N+1 "WHILE" N<LAST "DO"
"BEGIN" K:=STOREA[N]; "IF" K>0 "THEN"
    "BEGIN" "IF" K"LE" M "THEN" M:=K
        "ELSE" "BEGIN" ERROR(104); "PRINT" 'LINE',N "END"
    "END" "END";
"GOTO" INCR;

FIN: LENGTH:=P-1;
"END" OF PATH SEARCH;

"PROCEDURE" PRINT CIRCUIT
"BEGIN" "INTEGER" K,L,M,P;
"PROCEDURE" BL TYPE;
"BEGIN" "SWITCH" TY:=TY1,TY2,TY3,TY4,TY5,TY6,TY7,TY8,TY9;
    "GOTO" TYCMJ;
    TY1;"PRINT" /MAIN LINK/, "GOTO" TY END;
    TY2;"PRINT" /BRANCH ARM TYPE Z/, "GOTO" TY END;
    TY3;"PRINT" /BRANCH ARM TYPE Y/, "GOTO" TY END;
    TY4;"PRINT" /PARALLEL LOOP TYPE Z/, "GOTO" TY END;
    TY5;"PRINT" /PARALLEL LOOP TYPE Y/, "GOTO" TY END;
    TY6;"PRINT" /PARALLEL PATH TYPE Z/, "GOTO" TY END;
    TY7;"PRINT" /PARALLEL PATH TYPE Y/, "GOTO" TY END;
    TY8;"PRINT" /PARALLEL PATH TYPE G/, "GOTO" TY END;
    TY9;"PRINT" /PARALLEL PATH TYPE Y/;

TY END;"END" OF BLOCK TYPE;
"PROCEDURE" LOD(LOAD); "VALUE" LOAD; "INTEGER" LOAD;
    "IF" LOAD"NE"0 "THEN"
        "PRINT" /LOAD/,LOAD,' TYPE',LOADTYPE[LOAD],
        / PARAMETERS/,VLOAD[1,LOAD],VLOAD[2,LOAD],VLOAD[3,LOAD];
P1=0;"PRINT" //L2'CIRCUIT DESCRIPTION';
PREFIX(/ /); DIGITS(3); FREEPOINT(6);
NEXT:K:=STOREA[P]; L:=STOREB[P];
    "IF" K<0 "THEN" "BEGIN" "IF" L<0 "THEN"
        "BEGIN" M:=STOREA[-L]; "PRINT" //L3// BL TYPE "END" "END";
    "PRINT" //L//,P;
    "IF" K>0 "THEN" "BEGIN" "IF" L"NE"0 "THEN"
        "BEGIN" "PRINT" / NETWORK/,L,/ NODES/,NETCONC[L],/ TO/
            NETCONC[-L]; "IF" L<0 "THEN" L:=-L;
            "PRINT" / TYPE/,NETTYPE[L],/ PARAMETERS/,
            VNET[1,L],VNET[2,L],VNET[3,L],VNET[4,L];
        "END" "END" "ELSE" "IF" K<0 "THEN"
        "BEGIN" "IF" L<0 "THEN" "PRINT" / END AT LINE/,L,/ LEVEL/,K
            "ELSE" "PRINT" / POINTER ON NODE/,L"END";
        "ELSE" "IF" L"LE"0 "THEN"
        "BEGIN" M:=K; "PRINT" / END OF / AL TYPE;
            "IF" M>1 "THEN" "PRINT" / ORIGIN AT LINE/,L P1=P+1;
            "IF" M<4 "THEN" "BEGIN" K:=STOREA[P]; L:=STOREB[P];
            "PRINT" //L//; "IF" M<2 "THEN"
                "BEGIN" "PRINT" /PORT 1 /I LOD(K);
                "PRINT" //L/I PORT 2 /I LOD(L) "END";
            "ELSE" LOD(L) "END";
        "END" "ELSE" "IF" K"GE" P "THEN" "BEGIN" M:=STOREA[-STOREB[L]];
            "PRINT" / START OF / AL TYPE;
            "PRINT" / AT LINE/,L,/ RETURN AT LINE/,K "END";
            "ELSE" "PRINT" / RETURN FROM LINE/,K;
        "IF" P"LE" LENGTH "THEN" "GOTO" NEXT; "PRINT" //L2//;
    "END" OF PRINT CIRCUIT;

"PROCEDURE" SET VARIABLES(STEP,RESET);

```

```

"PRINT" / AT LINE1,L1 RETURN AT LINE1000
"ELSE" "PRINT" / RETURN FROM LINE1,K1
PI=P+1; "IF" P"LE"LENG "THEN" "GOTO" NEXT; "PRINT" "/L2"
"END" OF PRINT CIRCUIT;

```

"PROCEDURE" SET VARIABLES(STEP,RESET)

```

"VALUE" STEP,RESET; "BOOLEAN" STEP,RESET;
"BEGIN" "REAL" AR1
"INTEGER" M,N,P,Q; "BOOLEAN" TEST,VAREND;
VAREND:="FALSE"; N:=0;
LOOPVAR:="IF" STEP "THEN" LOOPVAR+1 "ELSE" 0;
"FOR" PI=1 "STEP" 1 "UNTIL" VARIABLES "DO"
"BEGIN" Q:=VARNET[P]; M:=VARPAR[P]; "IF" Q=0 "THEN" N:=N+1 "ELSE"
"BEGIN" "IF" Q>0 "THEN" "BEGIN" AR1:=VNET[M,Q];
"IF" STEP "THEN" "BEGIN" TEST:=VAP[3,P]<AR1
VNET[M,Q]:=AR1+VAP[2,P] "END" "ELSE"
"IF" RESET "THEN" VNET[M,Q]:=VAP[0,P]
"ELSE" "BEGIN" VARC0,P]:=AR1
VNET[M,Q]:=VARC1,P] "END" "END"
"ELSE" "BEGIN" AR1:=VLOAD[M,-Q];
"IF" STEP THEN" "BEGIN" TEST:=VAP[3,P]<AR1
VLOAD[M,-Q]:=AR1+VAP[2,P] "END" "ELSE"
"IF" RESET "THEN" VLOAD[M,-Q]:=VAP[0,P]
"ELSE" "BEGIN" VARC0,P]:=AR1
VLOAD[M,-Q]:=VAP[1,P] "END" "END"
"IF" SIGN(VAP[2,P])<0 "THEN" TEST:="NOT" TEST;
"IF" TEST "THEN" VAREND:="TRUE";
"END" "END";
TERMI:=STEP "AND" (N=VARIABLES "OR" VAREND OR" LOOPVAR<=10);

```

"END" OF SET VARIABLES;

```

"PROCEDURE" TABLE HEADING;
"BEGIN" "INTEGER" M,N,P,Q;
"BOOLEAN" TEST;
"SWITCH" HEAD1:FST1,FST2,FST3,FST4,FST5,FST6,FST7,FST8,FST9,
FST10,FST11,FST12,FST13,SEC1,SEC2,SEC3,SEC4,SEC5,
SEC6,SEC7,SEC8,SEC9,SEC10,SEC11,SEC12,SEC13;
PREFIX(/ /) DIGITS(2); "PRINT" "/L2"; PI=0;
"FOR" PI=1 "STEP" 1 "UNTIL" VARIABLES "DO"
"BEGIN" Q:=VARNET[P]; "IF" Q<>0 "THEN"
"BEGIN" "PRINT" "/L1 VARIABLE",P; M:=VARPAR[P];
"IF" Q>0 "THEN" "PRINT" / NETWORK1,Q,/ PARAMETER1,M,
' ',VNET[M,Q];
"ELSE" "PRINT" / LOAD 1,-0,/ PARAMETER1,M,
' ',VLOAD[M,-Q];
"END" "END";
DIGITS(1); PREFIX(/ /) "PRINT" "/L2S3(FREQ'S2"); PI=0;
LINE1INI:=OPTION[P]; "IF" N=0 "THEN" "GOTO" SEC LINE1;
Q1:=(N+3)"DIV"4; M1:="IF" N+1<4*Q "THEN" 1 "ELSE" 2;
PI=P+1; "GOTO" HEAD[Q];
FST11;"PRINT" / CHAR IMPED 1,M1 "GOTO" LINE11;
FST21;"PRINT" / CHAR ADMIT 1,M1 "GOTO" LINE11;
FST31|FST51;"PRINT" / INPUT IMPED 1,M1 "GOTO" LINE11;
FST41|FST61;"PRINT" / INPUT ADMIT 1,M1 "GOTO" LINE11;
FST71;"PRINT" / CHAIN 1,M1 1; "GOTO" LINE11;
FST81;"PRINT" / CHAIN 2,M1 1; "GOTO" LINE11;
FST91|FST111;"PRINT" / SPAR 1,M1 1; "GOTO" LINE11;
FST10|FST121;"PRINT" / SPAR 2,M1 1; "GOTO" LINE11;
FST131;"PRINT" / VSWR1,M1 1; "GOTO" LINE11;
SEC LINE1PI=0; "PRINT" / LS3(GHZ) 1;
LINE2INI:=OPTION[P]; "IF" N<0 "THEN" "GOTO" TABLE END;
Q1:=(N+1)"DIV"2; TEST:=(Q+Q)"NEEN NOR" Q>20;
PI=P+1; "IF" TEST "THEN" "PRINT" / 1;
"GOTO" HRADE(Q+1)"DIV"2+133;
SEC1|SEC31;"PRINT" / (OHM)'1 "GOTO" DEGI;
SEC2|SEC41;"PRINT" / (MHZ)'1 "GOTO" DEGI;

```

```
SEC5;SEC6;SEC9;SEC10;"PRINT" / (PU) ';' GOTO" DEG;
SEC7;"GOTO" "IF" NC27 "THEN" SEC5 "ELSE" SEC1;
SEC8;"GOTO" "IF" NC31 "THEN" SEC2 "ELSE" SEC5;
SEC11;SEC12;"PRINT" / (DR) ';' GOTO" LINE2;
SEC13;"PRINT" / (PU) ';' "GOTO" LINE2;
DEG;"IF" TEST "THEN" "PRINT" / "ELSE" "PRINT" / (DEG));
"GOTO" LINE2;
```

TABLE END;"END" OF TABLE HEADING\$

"PROCEDURE" FREQUENCY RESPONSE\$

```
"BEGIN" "REAL" W,AR,AI;
"ARRAY" AAC[0]15+8*MAXLEV];
"INTEGER" K,L,M,N,P,STK,STKA$;
"BOOLEAN" PRESENT$;
"BOOLEAN" "ARRAY" PRSK[1:MAXLEV];
```

```
"PROCEDURE" INVJ(I,J); "VALUE" I,J; "INTEGER" I,J;
"BEGIN" AR:=AAC[J]; AI:=AAC[J+1]; AAC[0]:=1.0/(AR*AR+AI*AI);
AAC[I]:=AR+AA[0]; AAC[I+1]:=AI*AA[0];
```

END" OF INVJ;

```
"PROCEDURE" MULTJ(I,J,K); "VALUE" I,J,K; "INTEGER" I,J,K;
"BEGIN" AR:=AAC[J]*AAC[K]-AAC[J+1]*AAC[K+1];
AI:=AAC[J+1]*AAC[K]*AAC[J]*AAC[K+1];
"IF" I"NE"0 "THEN" "BEGIN" AAC[I]:=AR; AAC[I+1]:=AI "END";
END" OF MULTJ;
```

```
"PROCEDURE" MULADJ(I,J,K,L); "VALUE" I,J,K,L; "INTEGER" I,J,K,L;
"BEGIN" MU[TJ(0,J,K)];
AAC[0]:=AR+AA[L]; AAC[I+1]:=AI+AA[L+1];
END" OF MULADJ;
```

```
"PROCEDURE" ZY LOAD(NET,ZY); "VALUE" ZY; "INTEGER" NET,ZY;
"BEGIN" "REAL" R,L,C; "INTEGER" M; M:=NET;
"IF" M>0 "THEN" "BEGIN" RI:=AA[1]; LI:=AA[2]; CI:=AA[3] "END";
"ELSE" "BEGIN" M:=M; RI:=VLOAD[1,M];
LI:=VLOAD[2,M]; CI:=VLOAD[3,M];
M:=LOADTYPE[M]; NET:=M "END";
"IF" M=1 "THEN"
"BEGIN" AR1;"IF" R>0 "THEN" 0 "ELSE" 1.0/R;
AI:=W*C+0.001-("IF" L=0 "THEN" 0 "ELSE" 1.0/(W*L));
"END" "ELSE" "IF" M=2 "THEN"
"BEGIN" AR1=R;
AI:=W*L-("IF" C=0 THEN" 0 "ELSE" 1000.0/(W*C));
"END" "ELSE" AR1:=AI:=0;
AAC[ZY]:=ARI AA[ZY+1]:=AI;
END" OF ZY LOAD;
```

```
"PROCEDURE" AADET;
"BEGIN" MULTJ( 4,16,22); MULTJ( 6,18,20);
AR:=AA[ 4]-AA[ 6]; AI:=AA[ 5]-AA[ 7];
END" OF AADET;
```

```
"PROCEDURE" A TO ZHGY;
"BEGIN" N:=K=10; AADET;
"IF" N>4 "THEN" "BEGIN" AR1:=ARI AI1:=AI "END";
AAC[ 8]:=AA[14+N]; AAC[ 9]:=AA[15+N];
AAC[10]:=ARI AA[11]:=AI;
AAC[12]:= "IF" N=20RNN=6 "THEN" 1.0 "ELSE" -1.0; AA[13]:=0;
AAC[14]:=AA[24-N]; AAC[15]:=AA[25-N];
```

```

"IF" N>9 "THEN" "BEGIN" AAC[14+N]; AAC[9]:=AAC[15+N];
AAC[10]:=ARI AAC[11]:=AI;
AAC[12]:=IF" N=20R[N=6 "THEN" 1,0 "ELSE" -1,0; AAC[13]:=0;
AAC[14]:=AAC[24+N]; AAC[15]:=AAC[25+N];

N1="IF" N<6 THEN" 18 "ELSE" 10)*N; INVJ( 4, N);
N1=6; "FOR" N=N+2 "WHILE" N<16 "DO" MULTJ( N, N, 4);
"END" OF A TO ZHGY;

"PROCEDURE ZHGY TO AI;
"BEGIN" N:=K+K+10; AADET;
AAC[6+N]:=AAC[16]; AAC[7+N]:=AAC[17];
AAC[16+N]:=AAC[22]; AAC[17+N]:=AAC[23];
"IF" N>4 "THEN" N:=N-8;
AAC[10+N]:=1,0; AAC[11+N]:=0;
AAC[12+N]:=ARI AAC[13+N]:=AI;
INVJ( 4,20); "IF" N=4 OR N=0 "THEN"
"BEGIN" AAC[4]:=AAC[4]; AAC[5]:=AAC[5] "END";
N1=6; "FOR" N=N+2 "WHILE" N<16 "DO" MULTJ( N, N, 4);
"END" OF ZHGY TO AI;

"PROCEDURE STACK;
"BEGIN" STK:=STK+1; PRSK[STK]:=PRESENT;
"IF" PRESENT "THEN"
"BEGIN" M:=N+9; FOR" N:=N+1 "WHILE" N<M "DO"
"BEGIN" STKAA:=STKAA+1; AAC[STKAA]:=AAC[N] "END";
"IF" PRESENT="FALSE"
"END" OF STACK;

"PROCEDURE RETURN;
"BEGIN" PRESENT:=PRSK[STK]; STK:=STK-1;
"IF" PRESENT "THEN"
"BEGIN" N:=24; "FOR" N:=N-1 WHILE" N>15 "DO"
"BEGIN" AAC[N]:=AAC[STKAA]; STKAA:=STKAA+1 "END";
"END";
"END" OF RETURN;

"PROCEDURE AADET;
"BEGIN" "COMMENT" SET A PARAMETERS FOR NETWORK IN AA[8] TO AAC[15];
"REAL" ATT,ANG,CS,SN,COSH,SINH;
"INTEGER" C;
"SWITCH" SS:=L1,L2,L3,L4,L5,L6,L7,L8,L9,L10;
"IF" L<0 "THEN" L:=L; Q:=NETTYPE[L];
"FOR" M:=1 "STEP" 1 "UNTIL" 4 "DO" AAC[M]:=VNETEM,L];
"IF" Q<9 "THEN"
"BEGIN" "FOR" M:=9 "STEP" 1 "UNTIL" 15 "DO" AAC[M]:=0;
AAC[8]:=AAC[14]:=1,0 "END";
"GOTO" S9[G];
L2|AAC[8]:=AAC[14]:=1,0; "GOTO" L1;
L3|M|=3; Q:=10; "GOTO" RX Q;
L4|M|=11; Q:=12;
RX Q:=AAC[3]:=0,001*AAC[3]; AAC[4]:=0,001*AAC[4];
"IF" AAC[M]>NEM0 OR" AAC[M+1]>NEM0 "THEN" INVJ( M, M);
AAC[Q]:=AAC[13+AAC[3]]; AAC[Q+1]:=AAC[2]+AAC[4]; "GOTO" L1;
L5|M|=2; Q:=10; N1:=1; "GOTO" RLC;
L6|M|=1; Q:=10; N1:=0; "GOTO" RLC;
L7|M|=2; Q:=12; N1:=0; "GOTO" RLC;
L8|M|=1; Q:=12; N1:=1; "GOTO" RLC;
RLC|ZY LOAD( M, Q); "IF" N=0 "THEN" INVJ( Q, Q); "GOTO" L1;
L9|ANG:=1,0/AAC[3]; GOTO" LINE1;
L10|ANG:=SQRT(AAC[3])/0,299793;
LINE1|ANG:=ANG*W+AAC[2]; CS:=COS(ANG); SN:=SIN(ANG); ATT:=AAC[4];
"IF" ATT=0 "THEN"
"BEGIN" AAC[8]:=0; AAC[9]:=AAC[12]:=0; AAC[13]:=SN "END" "ELSE"
"BEGIN" ATT:=0,11512923400*ATT; AAC[2];

```

```

    "IF" Q=9 "THEN" ATT=ATT+FREQ/AAC[3];
    SINH=0.5*EXP( ATT); ATT=0.5*EXP(-ATT);
    COSH=SINH+ATT; SINH=SINH-ATT;
    AAC[8]=COSH+CS; AAC[9]=SINH+SN;
    AAC[12]=SINH+CS; AAC[13]=COSH+SN;
    "END"; AAC[14]=AAC[8]; AAC[15]=AAC[9]; Q=10;
    ZOIMI=11; "FOR" M=4+1 "WHILE" M<14 "DO"
        "BEGIN" AAC[Q]=AAC[M]*AAC[1]; Q=Q+1 "END";
    "IF" Q=12 "THEN" "BEGIN" AAC[1]=1,0/AAC[1] "GOTO" ZO "END";
    L1;"END" OF SET A PARAMETERS FOR NETWORK;

"COMMENT" START CHAIN MATRIX ANALYSIS
    W1=6,2631853072+FREQ;
    P1=STK1=0; STKA1=23; PRESENT1="FALSE";

ELEMENT1(P1=P+1) K1=STOREA[P]; L1=STOREB[P];
    "IF" K>0 "THEN" "BEGIN" "IF" L=LERO "THEN"
        "GOTO" "IF" K<2 "THEN" COMPLETE "ELSE"
        "IF" K<4 "THEN" ARM END "ELSE"
        "IF" K<6 "THEN" LOOP "ELSE" PPBRANCH
        "ELSE" "GOTO" "IF" K>P "THEN" PPEND "ELSE" STORE "END";
    "IF" L=L0 "THEN" "GOTO" ELEMENT1;
    AA.NET1;
AAMULT1;"IF" PRESENT "THEN" "BEGIN" M=12;
    "FOR" M=M+4 "WHILE" M<24 "DO"
        "BEGIN" N=N+2; MULTJ( 4, M, 0); MULTJ( 6, M, 10);
        MULADJ( M, N, 12, 4); MULADJ( N, N, 14, 6) "END";
    "END" "ELSE" "BEGIN" N=13;
    "FOR" M=8 STEP 1 "UNTIL" 15 "DO"
        "BEGIN" N=N+1; AAC[N]=AAC[M]; "END";
    PRESENT1="TRUE" "END"; "GOTO" ELEMENT1;

STOREIN1=15; STACK1 P1=L1 "GOTO" ELEMENT1;

PPBRANCHIA TO ZHGVI N1=7; STACK1 P1=L1 "GOTO" ELEMENT1;

PPEND1K1=STOREA[L]; A TO ZHGVI RETURN;
    M1=STOREA[L+1]; "IF" M1=NEND "THEN" RETURN; N1=15;
    "FOR" M=8 STEP 1 "UNTIL" 15 "DO"
        "BEGIN" N=N+1; AAC[N]=AAC[M]; AAC[N]=AAC[M]; "END";
    ZHGVI TO A1 "IF" M=0 "THEN" RETURN "ELSE" PRESENT1="FALSE";
    "GOTO" AA.MULT1;

LOOP1M1K1="IF" M<9 "THEN" 6 "ELSE" 9; A TO ZHGVI;
    AAC[2]=AAC[8]+AAC[10]+AAC[12]+AAC[14];
    AAC[3]=AAC[9]+AAC[11]+AAC[13]+AAC[15];
    K1=M-2; "GOTO" ARM SET1;

ARM ENDIM1=STOREB[P+1]; ZY LOAD(M, 2);
    "IF" M<2 "THEN" "BEGIN" M=16; N1=18 "END"
    "ELSE" "BEGIN" N1=16; M=18 "END";
    MULADJ( 4, N, 2, M); MULADJ( 6, N, 4, 2, M, 4);
    N1="IF" K<3 "THEN" 6 "ELSE" 4;
    INVJ( N, N); MULTJ( 2, 4, 6);

ARM SET1 M1="IF" K<3 "THEN" 10 "ELSE" 12;
    "FOR" N=9 STEP 1 "UNTIL" 15 "DO" AAC[N]=0;
    AAC[8]=AAC[14]; AAC[10]=AAC[23]; AAC[M+1]=AAC[3];
    RETURN; P1=L1 "GOTO" AA.MULT1;

COMPLETE1(P1=P+1)
    M1=STOREA[P]; ZY LOAD( M ,8); "IF" M<2 "THEN" INVJ( 8, 8);
    M1=10; "IF" M<2 THEN" INVJ(10,10);

```

```
ARM SETI      M1="IF" K<3 "THEN" 10 "ELSE" 12
  "FOR" N:= 9 "STEP" 1 "UNTIL" 15 "DO" AA[N]:=0;
  AA[8]:=AA[14]; AA[1]:=AA[2]; AA[M+1]:=AA[3];
  RETURN; PI:=L; "GOTO" AAMULT;
```

```
COMPLETE:PI=P+1;
  M1=STOREA[P]; ZY LOAD(    M , 8); "IF" M<2 "THEN" INVJ( 8, 8);
  M1=STOREB[P]; ZY LOAD(    M,10); "IF" M<2 THEN" INVJ(10,10);
  AA[12]:=AA[8]; AA[13]:=AA[9]; AA[14]:=AA[10]; AA[15]:=AA[11];

"COMMENT" CALCULATE INPUT IMPEDANCES IN AA[24] TO AA[31];
  MULADJ(24,16,10,18); MULADJ(28,20,10,22);
  MULADJ(26,22, 9,18); MULADJ(30,20, 8,16);

"COMMENT" CALCULATE SCATTERING PARAMETERS IN AA[32] TO AA[39];
  MULADJ(32,28,12,24); MULADJ(38,30,14,26);
  AA[34]:=AA[36]; I=2; D=SORT(AA[8]*AA[10]); AA[35]:=AA[37]; I=0;
  AADET; AA[4]:=AR; AA[5]:=AI; MULTJ(34, 4,34);
  MULADJ( 6, 8,28,24); INVJ( 6, 6);
  NI=30; "FOR" NI=N+2 "WHILE" NI<40 "DO" MULTJ( N, N, 6);

"BEGIN" "COMMENT" PRINT RESULTS;
  "INTEGER" CI;
  "BOOLEAN" TEST,ODD;
  "SWITCH" WR1,WR2,WR3,WR4,WR5,WR6,WR7,WR8,WR9,
  WR10,WR11,WR12,WR13;
  PREFIX(' ', ); "PRINT" "/L",FREEPOINT(5),FREQ; PI=0;
  WRITEINI=OPTION[P]; "IF" N=0 "THEN" "GOTO" WRITE END;
  M=(N+1)"DIV"2; TEST; R(M+M)"NE"NI; ODD; "FALSE";
  PI=P+1; Q1=2; "GOTO" WR1(M+1)"DIV"2;

WR11Q1="IF" NI< 3 "THEN" 8 "ELSE" 10; "GOTO" OUTPUT;
WR21INVJ( 2,"IF" NI< 7 "THEN" 8 "ELSE" 10); ODD1="TRUE"; "GOTO" OUTPUT;
WR31INVJ( 2,"IF" NI<11 "THEN" 20 "ELSE" 30);
  MULTJ( 2, 2,"IF" NI<11 "THEN" 24 "ELSE" 24); "GOTO" OUTPUT;
WR41INVJ( 2,"IF" NI<15 "THEN" 24 "ELSE" 24); ODI="TRUE";
  MULTJ( 2, 2,"IF" NI<15 "THEN" 28 "ELSE" 30); GOTO" OUTPUT;
WR51MULTJ( 2,"IF" NI<19 "THEN" 8 "ELSE"
  "IF" NI<19 "THEN" 28 "ELSE" 30; INVJ( 4, 2));
  MULTJ( 2, 4,"IF" NI<19 "THEN" 24 "ELSE" 26); "GOTO" OUTPUT;
WR61MULTJ( 2,"IF" NI<23 "THEN" 8 "ELSE" 10,
  "IF" NI<23 "THEN" 28 "ELSE" 30);
  INVJ( 4,"IF" NI<23 "THEN" 24 "ELSE" 26);
  MULTJ( 2, 4, 2); "GOTO" OUTPUT;
WR71WR81M=M-10; ODD1=M-15; "GOTO" OUTPUT;
WR91WR101Q1=M-M- 21 "GOTO" OUTPUT;
WR111WR121Q1=M-M-10; AR1=AAEQ3; AI1=AA[Q+1];
  "PRINT" PREFIX(' ', ),FREEPOINT(5),
  4.342944819*LN(AR+AR+AI*AI),';' "GOTO" WRITE;
WR131Q1="IF" NI<51 "THEN" 32 "ELSE" 38; AR1=AA[Q]; AI1=AA[Q+1];
  AR1=SORT(AR+AR+AI*AI); "IF" "NOT"TEST "THEN" AR1=AR1
  "PRINT" PREFIX(' ', ),FREEPOINT(5),
  (1+AR)/(1-AR),';' "GOTO" WRITE;
OUTPUT1AR1=AA[Q]; AI1=AA[Q+1];
  "IF" ODD "THEN" AR1=1000;0*AR; AI1=1000;0*AI "END";
  "IF" TEST "THEN" "PRINT" FREEPOINT(4),AR,PREFIX(' ', ),AI
  "ELSB" "PRINT" FREEPOINT(4),SQR(AR+AR+AI*AI),
  PREFIX(' ', ),ALIGNED(3,2),
  "IF" AR<0 "THEN" 90*SIGN(AI) "ELSE"
  ((("IF" AR<0 "THEN" 180*SIGN(AI) "ELSE" 0)
  +97.29577951*ARCTAN(AI/AR))"

"GOTO" WRITE;
WRITE END;"END" OF PRINT RESULTS;
"END" OF FREQUENCY RESPONSE;
```

```
"BEGIN" "COMMENT" MAIN PROGRAM
  SET NEW CIRCUIT
DATA INI READ DATA1
  "IF" TERM "THEN" "GOTO" STOP1
  "IF" FAULT "THEN" "GOTO" DATA INI
  "IF" "NOT" NEWCIT "THEN" "GOTO" TABLE1
  PATH SEARCH
  PRINT CIRCUIT
  "IF" FAULT "THEN" "GOTO" DATA INI
TABLE1 SET VARIABLES("FALSE","FALSE")
  FREQ1=FSTART1
  TABLE HEADINGS1
RESPI FREQUENCY RESPONSE1
  FREQ1=FREQ+FSTEP1
  "IF" FREQ<FSTOP "THEN" "GOTO" RESPI
  SET VARIABLES("TRUE","FALSE")
  "IF" "NOT" TERM "THEN" "GOTO" TABLE1
  SET VARIABLES("FALSE","TRUE")
  "GOTO" DATA INI
STOP1"END" OF MAIN PROGRAM

"END" OF READ DATA1
"END" OF READ MAXIMUM SIZE OF CIRCUIT
```

CPU TIME = 0000 14.018

PRINT1521

PRINT-UP OF MICRO3 USER ES/R001/

MICRO3  
"BEGIN" "COMMENT" PROGRAM MICRO3 BY B G MARCHENTI  
    "INTEGER" SMICROJ  
    "INTEGER" INCHAN, OUCHAN, LP, EXT, IWSI  
    "INTEGER" "ARRAY" IST[113000]  
    "INTEGER" "ARRAY" STRING[015]  
    "REAL" AC, AC1, BC, BC1, ONE, ONEI, PI, PI2I  
    "REAL" VLIGHT, DBNEP, PADDEG, CPXDBI  
  
"PROCEDURE" DUMMY;  
"BEGIN" "COMMENT" DUMMY TO ENTER SUBPROGRAMS AC1=0; NEND"  
  
"PROCEDURE" PCALL(PRG, ENTRY); "VALUE" PRG, ENTRY; "INTEGER" PRG, ENTRY;  
"BEGIN" "COMMENT" ENTER SUBPROGRAMS  
    "INTEGER" FAR, SON, BL, IP, IS, KI  
    "CODE" XLDSSIST  
        XLDSENTRY  
        XBBS22  
        XGETASIST  
        XINDEXSPRG  
        XBLIMSO  
        XBBSSONI INSIM0I  
        "IF" SON="LE0" "THEN"  
        "BEGIN" "CODE" XLDRISSON  
            XJILS108  
            XINCSSIWSI "GOTO" FIN "END";  
    PAR1=SMICROJ  
    BLINKSIKI=41  
NEXTLI KI=K+2I "IF" K"GE"BL "THEN" "GOTO" SETI  
    "IF" K=8 "THEN" "GOTO" NEXTLI  
    ISI=SON+KI (PI=FAR+KI)  
    "CODE" XLDRIIS  
        XJILS219  
        XRL1ISIP  
        XBBIISIS  
        XLDRISSIP  
        XJILS221I  
    "CODE" XLDRISSON  
        XINDEXSK  
        XRTASIS  
        XLDRISSPAR  
        XINDEXSK  
        XRTASIP  
        XBLIIISIS  
        XBBSAC  
        XBLIIISIP  
        XBBIISIS  
        XRLSAC  
        XBBIISIP; "GOTO" NEXTLI  
SETI "IF" PAR1=NEPSMICRO "THEN" "GOTO" FIN;  
    DUMMY;  
    PAR1=SONI SON1=SMICROJ "GOTO" BLINKS;  
FINI "END" PRGCAL;  
  
"PROCEDURE" PFIND(PRG); "VALUE" PRG1 "INTEGER" PRG1  
"BEGIN" "COMMENT" FIND START OF PROGRAM IN NICE TABLEI  
    "INTEGER" START, BL, II STARTI=BLI=0I  
    "CODE" XGETASISTRING

```

%INDEXLS1
XBLIMSO
%4BS208
XLDILS20
XLDRS187
XJILS212
XJFSMISSD
XADDRILS40
XJILS218
XADDRILS4
XSTRSTART
SEH JEDI    I1=IWS#START1
    IF START#0 "THEN" "GOTO" FIN1
    GOF1 XLDILS2
    INCSSI
    ISL XTSLDRIISI
    JNSBFIN
    ADDSSI
    ADDSSBL
    JB$BNEXT
ASBFIN
"CODE" XGETASIST
%INDEXSPRG
XSTRSI
XBLSTART
X4B1ISII
PFIND1
"PROCEDURE" PROLOG(ST); "STRING" ST
"BEGIN" "INTEGER" I1
"CODE" XGETASSTRING
%INDEXLS1
XTRSI
XGETSSST
XLDIMSI
XLDRIMSO
XZSINSTR
X4B1ISI
SSINSTR PFIND(10)
"IF" IWS"NEW" "THEN" "BEGIN" IWS#0; "GOTO" FIN "END"
"CODE" XI,DRSI
XLDIMSO
XSTIMSI
XLDIMSI
XSTIMSI
STRING[1]:=71691521 STRING[2]:=01
"CODE" XGETSSST
XLDIMSI
XLDRSI
XSTIMSI
XJILS188
INCSSIWSI
STRING[5]:=4000
FIN1 END# PROLOG1
"PROCEDURE" ZBROJ(A,JA,ZERO); "VALUE" A,JA; "REAL" A,JA; "INTEGER" ZERO
"BEGIN" "COMMENT" SET ZERO TO 0 IF COMPLEX NUMBER A IS ZERO
    ZERO1="IF" A#0 "AND" JA#0 "THEN" 0 "ELSE" 1
    END# ZERO1
"PROCEDURE" AMAXJ(A,JA,MAX); "VALUE" A,JA; "REAL" A,JA,MAX
    "COMMENT" MAXIMUM ABSOLUTE VALUE OF A OR JA IN MAX

```

```
"PROCEDURE" ZEROJ(A,JA,ZERO); "VALUE" A,JA; "REAL" A,JA; "INTEGER" ZERO;
"BEGIN" "COMMENT" SET ZERO TO 0 IF COMPLEX NUMBER A IS ZERO
    ZEROI="IF" A=0 "AND" JA=0 "THEN" 0 "ELSE" 1
"END" ZEROJ
```

```
"PROCEDURE" AMAXJ(A,JA,MAX); "VALUE" A,JA; "REAL" A,JA,MAX;
"BEGIN" "COMMENT" MAXIMUM ABSOLUTE VALUE OF A OR JA IN MAX
    "REAL" F,JF; F=ABS(A); JF=ABS(JA);
    MAXI="IF" JF < F "THEN" F "ELSE" JF
"END" AMAXJ
```

```
"PROCEDURE" ASSJ(A,JA,B,JB); "VALUE" A,JA; "REAL" A,JA,B,JB;
"BEGIN" "COMMENT" COPY COMPLEX NUMBER A INTO B
    B=A; JBI=JA;
"END" ASSJ
```

```
"PROCEDURE" ADDJ(A,JA,B,JB,C,JC);
    "VALUE" A,JA,B,JB; "REAL" A,JA,B,JB,C,JC;
"BEGIN" "COMMENT" TO COMPLEX A ADD B AND PLACE IN C
    CI=A+B; JC=JA+JB;
"END" ADDJ
```

```
"PROCEDURE" SUBJ(A,JA,B,JB,C,JC);
    "VALUE" A,JA,B,JB; "REAL" A,JA,B,JB,C,JC;
"BEGIN" "COMMENT" FROM COMPLEX A SUBTRACT B AND PLACE IN C
    CI=A-B; JC=JA-JB;
"END" SUBJ
```

```
"PROCEDURE" INVJ(A,JA,B,JB); "VALUE" A,JA; "REAL" A,JA,B,JB;
"BEGIN" "COMMENT" INVERT COMPLEX NUMBER A AND PLACE IN B
    "REAL" FI; FI=A*JA-JA; BI=A/FI; JRI=-JA/FI
"END" INVJ
```

```
"PROCEDURE" MULTJ(A,JA,B,JB,C,JC);
    "VALUE" A,JA,B,JB; "REAL" A,JA,B,JB,C,JC;
"BEGIN" "COMMENT" MULTIPLY COMPLEX A AND B AND PLACE IN C
    CI=A*B-JA*JB; JC=A*JB+JA*B;
"END" MULTJ
```

```
"PROCEDURE" DIVJ(A,JA,B,JB,C,JC);
    "VALUE" A,JA,B,JB; "REAL" A,JA,B,JB,C,JC;
"BEGIN" "COMMENT" DIVIDE COMPLEX A BY B AND PLACE IN C
    "REAL" F,JF;
    INVJ(B,JB,F,JF); MULTJ(A,JA,F,JF,C,JC);
"END" DIVJ
```

```
"PROCEDURE" MODJ(A,JA,MOD); "VALUE" A,JA; "REAL" A,JA,MOD;
"BEGIN" "COMMENT" FIND THE MODULUS OF COMPLEX A AND PLACE IN MOD
    MODI=SQRT(A*A+JA*JA);
"END" MODJ
```

```
"PROCEDURE" ARGJ(A,JA,ARG); "VALUE" A,JA; "REAL" A,JA,ARG;
"BEGIN" "COMMENT" FIND ARGUMENT OF COMPLEX A AND PLACE IN ARG
    "INTEGER" SI,JI; "REAL" ANG;
    SI=SIGN(A); JI=SIGN(JA);
    ANGI="IF" SI=0 "THEN" JSI=0.5*PI "ELSE" ARCTAN(JA/A);
    "IF" SI<0 "THEN" ANGI="IF" JSI<0 "THEN" ANG=PI "ELSE" ANG=PI;
    ARGI=RADDEG*ANG;
"END" ARGJ
```

```
"PROCEDURE" DBJ(A,JA,DB); "VALUE" A,JA; "REAL" A,JA,DB;
"BEGIN" "COMMENT" FIND MODULUS OF COMPLEX A IN DB
    DBI=CPXDR=LN(A*A+JA*JA);
"END" DBJ
```

```

"PROCEDURE" LENGTH(H,LEN) "VALUE" HI "INTEGER" H,LEN
"BEGIN" "INTEGER" K1
  "CODE" %GETASIST
  %INDEXSH
  %LDIMSO
  %LDKILS2
  %SMLC
  %ANDILS2047
  %STSK1  LEN:=K1
"END" LENGTH

"PROCEDURE" DPCINT(H,POINT) "VALUE" HI "INTEGER" H,POINT
"BEGIN" "INTEGER" K1
  "CODE" %GETASIST
  %INDEXSH
  %LDIMSO
  %LDKILS63
  %STSK1  POINT:=K1
"END" DPOINT

"PROCEDURE" INREAL(FPOINT,H)
  "VALUE" FPOINT,HI "REAL" FPOINT "INTEGER" HI
  "CODE" %FLSFPOINT
  %GETASIST
  %INDEXSH
  %WFIMSO

"PROCEDURE" REALOF(H,FP) "VALUE" HI "INTEGER" HI "REAL" FP
"BEGIN" "REAL" FPOINT
  "CODE" %GETASIST
  %INDEXSH
  %FLIMSO
  %WFSPPOINT FPI:=FPOINT
"END" REALOF

PRGLOD('ACIO 1); PRGLOD('SPR 1)
INCHAN1=501 OUTCHAN1=LP1#4
RESTART;"FOR" IWS1=2 "STEP" 1 "UNTIL" 9 "DO" IST[IWS1]=0
IST[1]:=251 IST[4]:=31 IST[5]:=41 !IST[15]:=4000
ONEI:=1.01 CNEII:=01 PI1:=3.1415926535901 PI21:=PI+PI
VLIGHT1:=2997930001 RADDEG1:=57.2957795130821
DBNEP1:=0.11512925464971 CPXDB1:=4.342944819031
PRGLOD('MICRO3 1); SMICRO1:=IST[10]
"PRINT" PUNCH(LP), "F"

DATA1
"BEGIN" "COMMENT" SEGMENT READ DATA FROM INPUT STREAM
  "INTEGER" WORD,RING,BEAD,J,K,M,N
  "REAL" FL1
  "SWITCH" MAINWD1=SNET ,SJUNCT,SPORT ,SCIRT ,SZO ,
    SFREQ ,SOUT ,SSPEC ,SVARY ,SANALY ,
    SOPTIM ,SLABEL ,SDATA ,SError ,SLP ,
    SRERUN ,SEND ,SNFLT ,SDELET ,SLOAD ,
    STRUCT
"COMMENT" SYNTAX PROCEDURES PRCHAR, NEXTCH, DATERR,
SYNTAX, FNCDRD AND ZTYPE1
"INTEGER" CHAR,COLUMN,TYPE,INTREP1
"BOOLEAN" FAULT
"REAL" FLREP1

```

"COMMENT" SYNTAX PROCEDURES PRCHAR, NEXTCH, DATERR,  
SYNTAX, FNDWRD AND ZTYPE;  
"INTEGER" CHAR,COLUMN,TYPE,INTREP  
"BOOLEAN" FAULT;  
"REAL" FLREP;

"PROCEDURE" PRCHAR(CHAR); "VALUE" CHAR; "INTEGER" CHAR;  
"BEGIN" "INTEGER" M;  
"CODE" XGETASSTRING  
  %INDEXI\$0  
  XLDSCHAR  
  XANDN1\$63  
  %JNZSOUT  
  XLDSCHAR  
  XPUTIM\$0  
  %LDIL\$62  
  XPUTIM\$0  
  %JFSIN  
  \$SOUTSLD:LS62  
  XPUTIM\$0  
  XLDSCHAR  
  XPUTIM\$0  
  \$TINSLDIL\$32  
  XPUTIM\$0  
  XPUTIM\$0;  
M:=0; OUTSTRING(STRING,M);  
"END" PRCHAR;

"PROCEDURE" NEXTCH;  
"BEGIN" "IF" INCHAN"NE"LP "THEN"  
  "BEGIN" PUNCH(LP); PRCHAR(CHAR); PUNCH(OUCHAN) "END";  
  "IF" CHAR=2 "THEN"  
    "BEGIN" "IF" OUCHAN"NE"LP "THEN" "PRINT" '+';  
      COLUMN:=COLUMN+1;  
      ADVANCE(OUCHAN); CHAR:=DECODE(INCHAN);  
    "END" NEXTCH;

"PROCEDURE" DATERR(N); "VALUE" N; "INTEGER" N;  
"BEGIN" "INTEGER" I,J,L;  
  JI:=OUCHAN;  
PRI: PUNCH(J); "IF" NCO "THEN"  
  "PRINT" //L\*\* CIRCUIT ERROR!, -N "ELSE"  
  "BEGIN" "PRINT" //L\*\* SYNTAX ERROR!, N, I-1;  
    L:=1; OUTSTRING(STRING,L) "END";  
  "PRINT" //L\*\*;  
  "IF" J"NE"LP "THEN" "BEGIN" J:=LP; "GOTO" PR "END";  
  PUNCH(OUCHAN); FAULT:="TRUE";  
  "IF" NCO "THEN" "GOTO" FIN;  
  "IF" INCHAN"NE"LP "THEN"  
    "FOR" I:=1"STEP"1"UNTIL"COLUMN"DO"PRINT" PUNCH(LP), I-1;  
    "IF" INCHAN"NE"OUCHAN "THEN" "BEGIN" CHAR:=2;"GOTO" RUBOUT"END";  
    "IF" TYPE=1 "THEN" "GOTO" RUBOUT;  
NEXTI: I:=CHAR; NEXTCH;  
  "IF" I=5 "AND" CHAR=9 "THEN" "GOTO" RUBOUT;  
  "IF" I"NE"27 "THEN" "GOTO" NEXTI;  
RUBOUT:"GOTO" DATA1;  
FIN:"END" DATERR;

"PROCEDURE" SYNTAX(TYPES,ERR);  
  "VALUE" TYPES,ERR; "INTEGER" TYPES,ERR;  
"BEGIN" "REAL" FNO,FNOA; "INTEGER" NO,J,K,STS,STN,DIGI;  
  "BOOLEAN" SNEG;

"PROCEDURE" STKCHAI;  
"BEGIN" "IF" STN>0 "THEN" GOTO" STK1

```

    "IF" STS=4 "THEN" "GOTO" FIN;
STS=4; STS>STS+1; "IF" STS<3 "THEN" "GOTO" STK1
STRING[STS]=0; STRING[STS+1]=4000
STK1 STN=STN-1;
"CODE" XGETASSTRING
XINDEXSSTS
XLDSCCHAR
XANDILS63
XLDKSSTN
XSMLC
XANDS1MS01
FIN: NEXTCH; "END" STKCHAI

"PROCEDURE" NUMBER1
"BEGIN" K1=NO1=0; FNOA1=0
NEXT1 "IF" CHAR>15 "AND" CHAR<26 "THEN"
"BEGIN" K1=K1+1; J1=CHAR+16; STKCHAI
"IF" K1>7 "THEN" "BEGIN" NO1=10+NO+J1; FNOA1=NO "END"
"ELSE" FNOA1=10,0+FNOA1+J1
"GOTO" NEXT "END"; DIGI=DIGI+K1
"IF" SNEG "THEN" "BEGIN" NO1+=NO1; FNOA1+=FNOA1 "END"
"END" NUMBER1

"PROCEDURE" INTEG1
"BEGIN" SNEG1=CHAR+13
"IF" CHAR=11 "OR" CHAR=13 "THEN" STKCHAI
NUMBER1
"END" INTEG1

ENTER1 STS1=STN1=K1=DIGI=0; SNEG1="FALSE"
STRING[1]=STRING[2]=0; STRING[3]=4000
BLA1 "IF" CHAR<3 "THEN" "BEGIN" NEXTCH; "GOTO" BLA "END"
"GOTO" TALPHAI
NALPHAI;"IF" CHAR>15 "AND" CHAR<26 "THEN" "GOTO" ALPHA1
TALPHAI;"IF" CHAR>32 "AND" CHAR<59 "THEN" "GOTO" ALPHA1
"GOTO" MISAI
ALPHA1 J1=CHAR1 STKCHAI K1=K1+1 "GOTO" NALPHAI
MISAI "IF" CHAR=5 "THEN" "GOTO" MISS1
"IF" K1NE0 "THEN" "BEGIN" TYPE1=0; "GOTO" FIN "END"
INTEG1
"IF" CHAR=14 "THEN" "GOTO" DPOINT1
"IF" CHAR=37 "THEN"
"BEGIN" STKCHAI
    FNO1="IF" K1NE0 "THEN" FNOA "ELSE"
    "IF" SNEG "THEN" ONE=0 "ELSE" ONE=1
    "GOTO" EXPONT "END"
"IF" CHAR=5 "THEN" "GOTO" MISS1
IP1 STS1NE0 "THEN"
"BEGIN" "CODE" XLDSTYPES
XANDILS2
    XSTSJ1 "IF" J1=0 "THEN" "GOTO" INTR1
    TYPE1=2; "IF" K1>4 "THEN" DATERR(5)
    INTREP1=NO1 "GOTO" NUM "END"
"IF" CHAR=27 "THEN"
"BEGIN" TYPE1=1; STKCHAI "GOTO" FIN "END"
MISS1 J1=CHAR1 NEXTCH
"IF" J1NE5 "THEN" "GOTO" ENTER1
"IF" CHARNE5 "THEN" "GOTO" ENTER1
NEXTCH; "GOTO" DATAIN1

```

MISSI J1=CHAR1 NEXTCH  
"IF" J1"NE"5 "THEN" "GOTO" ENTER1  
"IF" CHAR"NE"5 "THEN" "GOTO" ENTER1  
NEXTCH; "GOTO" DATAINI

DPOINT:STKCHAI  
INTR1 FNO:=FNOA1 NUMBER1  
FNO:=FNO+FNOA10\*(+K)  
"IF" CHAR"NE"37 "THEN" "GOTO" REAL1 STKCHAI  
EXPONT:INTEGER1 FNO:=FNC\*10\*N01  
REAL1 "IF" CHAR=5 "THEN" "GOTO" MISS1 TYPE:=41 FLREP1=FNO1  
NUM1 "IF" DIG=0 "THEN" DATERR(6)  
FINI "CODE" %LOSTYPES  
%ANDTYPE  
%STSJ1  
"IF" J=0 "THEN" DATERR(ERR) "ELSE"  
"IF" TYPE=1 "AND" TYPES<16 "THEN" "GOTO" DATAINI  
"END" SYNTAX1  
  
"INTEGER" "PROCEDURE" FNDWRD(N,ST,ERR);  
"VALUE" N,ERR; "INTEGER" N,ERR; "STRING" ST;  
"BEGIN" "INTEGER" J1 J1=0;  
"CODE" %GETASSTRING  
%INDEX1LS1  
%BL1HS0  
%WBS208  
%GETSSST  
%LDSN  
%JILS212  
%JFSMISS  
%ADDILS1  
%STSJ  
S1MISS1 FNDWRD1:=J1  
"IF" J1>0 "THEN" "GOTO" FINI  
"IF" ERR"NE"0 "THEN" DATERR(ERR);  
FINI "END" FNDWRD1  
  
"INTEGER" "PROCEDURE" ZTYPE(PARS); "INTEGER" PARS;  
"BEGIN" "INTEGER" PAR1  
PAR1:=ZTYPE1:=FNDWRD(10,'R X L SRX <<  
>>SRCL G B C PGB PGCL ',0);  
"IF" PAR1>5 "THEN" PAR1:=PAR1-5;  
PARS1:="IF" PAR1<4 "THEN" 1"ELSE" "(F" PAR1<5"THEN" 2"ELSE" 3)  
"END" ZTYPE1  
  
"COMMENT" DATA STRUCTURE PROCEDURES PRIOR, KEY, LOCNAME, DUMP, DEFINE,  
INSERT AND ELIMIN;  
"PROCEDURE" PRIOR(H,PRI); "VALUE" H; "INTEGER" H,PRI;  
"BEGIN" "INTEGER" K1  
"CODE" %GETASIST  
%INDEXSH  
%LD1HS0  
%LOKILS3  
%SHLG  
%ADDILS63  
%STS1 PRI1:=K1  
"END" PRIOR1  
  
"PROCEDURE" KEY(H)LENGTH(DL)POINTERS((DP)PRIOR1(PR))  
"VALUE" H,DL,DP,PRI; "INTEGER" H,DL,DP,PRI;  
"CODE" %GETASIST  
%INDEXSH  
%LD1HS2048  
%ADDSDL  
%ADDSDP

```

    XSTIMSO
    XLDSPR
    XPUTIMSO
    XLDSDP
    XPUTIMSOI

"INTEGER" "PROCEDURE" LOCNAME(RING,NAME)
    "VALUE" RING,NAME; "INTEGER" RING,NAME;
"BEGIN" "INTEGER" J,K,L; J:=IST[RING];
    "FOR" K:=J+1 "WHILE" K<NE"RING "AND" JNE"0 "DO"
        "BEGIN" DPOINT(J,L); "IF" IST[K+L]>NAME "THEN" "GOTO" FOUND;
            J:=IST[K]; "END"; J:=0;
FOUND; LOCNAME:=J;
"END" LOCNAME;

"PROCEDURE" DUMP(H); "VALUE" H; "INTEGER" H;
"BEGIN" "INTEGER" K,M,N,P;
    LENGTH(H,K); KEY(H,K,0,7); M:=1;
    "FOR" N:=IST[M+1] "WHILE" N<NE"0 "DO"
        "BEGIN" LENGTH(N,P); "IF" K<LE"P "THEN" "GOTO" OUT;
            "ELSE" M:=N "END";
        OUT; IST[H+1]:=N; IST[M+1]:=H;
"END" DUMP;

"INTEGER" "PROCEDURE" DEFINE(DL)POINTERS((DP>PRIORI(PR));
    "VALUE" DL,DP,PR; "INTEGER" DL,DP,PR;
"BEGIN" "INTEGER" A,I,J,K;
    A:=DL+DR; J:=1;
SERCH; I:=IST[J+1];
    "IF" I=0 "THEN" "GOTO" NEW; LENGTH(I,K); K:=K-A;
    "IF" K<2 "THEN" "BEGIN" "IF" KNE"0 "THEN"
        "BEGIN" J:=I; "GOTO" SERCH "END"; I
        IST[EJ+1]:=IST[I+1]; "IF" K=0 "THEN" "GOTO" FOUND;
        J:=I+A+1; K:=K-1; KEY(J,K,0,7);
        DUMP(J); "GOTO" FOUND;
NEW; I:=IST[1]+1; IST[1]:=J:=A;
    "IF" J>RANGE(IST,1) "THEN"
        "BEGIN" "PRINT" //L100 [ST OFLO]; "GOTO" RUNEND "END";
FOUND; KEY(I,DL,DP,PR); K:=I-A; DEFINE:=I;
    "FOR" I:=I+1 "WHILE" I<LE"K "DO" IST[I]:=0;
"END" DEFINE;

"PROCEDURE" INSERT(RING,BEAD,NAME);
    "VALUE" RING,BEAD,NAME; "INTEGER" RING,BEAD,NAME;
"BEGIN" "INTEGER" I,J,K,L;
    DPOINT(BEAD,L); IST[BEAD+L+1]:=NAME; J:=IST[RING]; I:=RING;
    "FOR" K:=J+1 "WHILE" K<NE"RING "AND" JNE"0 "DO"
        "BEGIN" DPOINT(J,L); "IF" IST[K+L]>NAME "THEN" "GOTO" OUT;
            I:=K; J:=IST[K]; "END";
OUT; IST[BEAD+1]:=J; IST[1]:=BEAD;
"END" INSERT;

"PROCEDURE" ELIMIN(HEAD); "VALUE" HEAD; "INTEGER" HEAD;
"BEGIN" "INTEGER" LH,STACK,LNEXT,NP,L,J,K,PR;
    "IF" HEAD=0 "THEN" "GOTO" FIN;
CHECK; LH:=; "IF" IST[HEAD]<0 "THEN" HEAD "ELSE" IST[HEAD];
    "IF" LH<0 "THEN" "GOTO" FIN;
    "IF" IST[LH]>=0 "THEN" "GOTO" FIN; STACK:=0;
    PRICRL(H,J); "IF" J<7 "THEN" "GOTO" FIN;
NEW; DPOINT(LH,NP); NP:=LH+NP; PRIOR(LH,PR); L:=LH;

```

```

    "IF" HEAD=0 "THEN" "GOTO" FINI
CHECK1 LH:= "IF" IST[HEAD]<0 "THEN" HEAD "ELSE" IST[HEAD]
    "IF" LH>0 "THEN" "GOTO" FINI
    "IF" IST[LH]>0 "THEN" "GOTO" FINI STACK1=0
PRICR(LH,J); "IF" J=7 "THEN" "GOTO" FINI
NEW: DPOINT(LH,NP); NP1=LH+NPI PRIOR(LH,PR); L=LH

```

```

    "IF" PR=7 "THEN" "GOTO" UNSTAK
    "IF" PR=1 "THEN" "GOTO" POINTS1
    "IF" PR="NE"3 "THEN" "GOTO" RING1 K1=LH+1
JSTEP1 J1=IST[K+1]; "IF" J=0 "THEN" "GOTO" RING1
    "IF" IST[J+2]=IST[STACK+2] "THEN" "GOTO" JOUT1
    K1=J1 "GOTO" JSTEP1
JOUT1 IST[K+1]:=IST[J+1]; DUMP(J)
    "IF" IST[LH+2]>0 "THEN" "GOTO" UNSTAK
RING1 K1=LH RGSTEPI J1=IST[K+1]
    "IF" J=0 "THEN" "GOTO" POINTS1
    "IF" J="NE" LH "THEN" "BEGIN" K1=J1 "GOTO" RGSTEP "END"
    IST[K+1]:=IST[LH+1]; IST[LH+1]:=0
POINTS1 L1=L+1
    "IF" L>NP "THEN" "BEGIN" DUMP(LH); "GOTO" UNSTAK "END"
    LNEXT1=IST[L]; "IF" LNEXT1=0 "THEN" "GOTO" POINTS1
    "IF" PR="NE"0 "THEN" IST[L]:=0
    PRICR(LNEXT1,J); "IF" J=7 "THEN" "GOTO" POINTS1
    J1=STACK1 STACK1=DEFINE(1,1,1); IST[STACK+1]:=J1
    IST[STACK+2]:=LH; LH=LNEXT1 "GOTO" NEW1
UNSTAK1 "IF" STACK1=0 "THEN" "GOTO" CHECK1
    LH=IST[STACK+2]; J1=STACK1 STACK1=IST[STACK+1]
    DUMP(J); "GOTO" NEW1
FINI "END" ELIMINI

```

```

"COMMENT" CIRCUIT STRUCTURE PROCEDURES NEHCIT AND JUNCT1
"INTEGER" "PROCEDURE" NEHCIT(CIT); "VALUE" CIT1 "INTEGER" CIT1
"BEGIN" INTEGER" K1
    K1=LQCNAM(4,CIT); "IF" K=0 "THEN"
    "BEGIN" K1=DEFINE(4,4,0); INSERT(4,K,CIT) "END"
    NEHCIT1=K1; IST[K+2]:=K+1; IST[K+3]:=K+2; IST[K+4]:=K+3
"END" NEHCIT1

"PROCEDURE" JUNCT(HNET,JUN,SUF);
    "VALUE" HNET,JUN,SUF; "INTEGER" HNET,JUN,SUF
"BEGIN" "INTEGER" H,K,L1
    H1=IST[HNET+SUF];
    "IF" H="NE"0 "THEN"
    "BEGIN" K1=H+1;
    "FOR" L1=IST[K+1] "WHILE" IST[L+2]>0 "DO" K1=L1
    IST[K+1]:=IST[L+1]; DUMP(L)
    "IF" IST[4+2]=0 "THEN" BLIMIN(H); "END"
    K1=IST[3]+3; H1=LQCNAM(K,JUN);
    "IF" H=0 "THEN" "BEGIN" H1=DEFINE(3,2,3); INSERT(K,H,JUN) "END"
    K1=DEFINE(0,2,1); IST[K+2]:=HNET;
    IST[K+1]:=IST[H+2]; IST[H+2]:=K1; IST[HNET+SUF]:=H1
"END" JUNCT1

```

```

"COMMENT" CIRCUIT CHECKING PROCEDURES CHCIT AND CHOPTS1
"PROCEDURE" CHCIT(BEAD); "VALUE" BEAD1 "INTEGER" BEAD1
"BEGIN" "INTEGER" I,J,K,M,N,RING,P,INT1 RING1=31
NEXTB1 RING1=IST[RING+1];
    "IF" RING1=3 "THEN" "BEGIN" DATERR(-1); GOTO" C1TERR "END"
    "IF" RING1>BEAD "THEN" "GOTO" NEXTB1
    "PRINT" "/(* CIRCUIT", IST[BEAD+5]);
    J1=RING1-BEAD+2; INT1=0
    "FOR" J1=IST[J+1] "WHILE" J1<RING "DO"
    "BEGIN" IST[J+5]:=0; INT1=INT1+1; "IF" IST[J+4]=0 "THEN"
        "BEGIN" DATERR(-2); "GOTO" JUNERR "END" "END"
    IST[BEAD+6]:=INT1
JUNERR1 J1=RING1-BEAD+1

```

```

"FOR" J:=ISTCJ+1] "WHILE" J<NE"RING "DO"
"BEGIN" I:=0 DPOINT(J,M) P:=1
"FOR" K:=3 "STEP" 1 "UNTIL" M <DO
"BEGIN" N:=ISTCJ+K] "IF" N>0 "THEN"
"BEGIN" DATERR(-3); "GOTO" NETERR "END"
"IF" ISTEN+4]>0 "THEN" I:=I+P; P:=P+P
"END"
"IF" ISTCJ+2]>NE"0 "THEN" I:=I+4096; ISTCJ+M+3]:=I
"END"
NETERR: J:=RING:=BEAD+3; K:=J:=0; P:=1
"FOR" J:=ISTCJ+1] "WHILE" J<NE"RING "DO"
"BEGIN" K:=K+1; N:=ISTCJ+2]; ISTEN+5]:=K
"IF" ISTEN+4]>0 "THEN" I:=I+P; P:=P+P "END"
"IF" K>0 "CR" K>8 "THEN" DATERR(-4)
ISTCBEAD+6]:=K; ISTCREAD+7]:=I
INT:=INT-K; K:=0; J:=RING:=BEAD+2
"FOR" J:=ISTCJ+1] "WHILE" J<NE"RING "DO"
"BEGIN" M:=J+5; N:=ISTEN];
"IF" N>0 "THEN" ISTEN]:=K:=K+1 "ELSE" ISTEN]:=N+INT "END"
NTPORT: J:=RING:=BEAD+1
"FOR" J:=ISTCJ+1] "WHILE" J<NE"RING "DO"
"BEGIN" DPOINT(J,K); K:=K+2; I:=~ISTCJ+K+4]
"IF" I>0 "THEN"
"BEGIN" CHCIT(I); N:=ISTC1+6];
"IF" N>NE"K "THEN" DATERR(-6) "END" "END"
CITERRI;"END" CHCIT;
"PROCEDURE" CHOPTS(OPTIM); "VALUE" OPTIM; "BOOLEAN" OPTIM;
"BEGIN" "INTEGER" I,J,K,M,N
"IF" ISTC5]>4 "THEN" DATERR(-21)
"IF" OPTIM "THEN" "GOTO" OP1
"IF" ISTC6]>0 "THEN" DATERR(-22)
N:=6; "GOTO" OUT1
OP1: "IF" ISTC9]>0 "THEN" DATERR(-29)
N:=8; "FOR" N:=ISTEN+1] "WHILE" N<NE"0 "DO"
"BEGIN" M:=ISTEN+2]; I:=5; "IF" M>2 "THEN" ZOP1
I:=LOCNAME(4,ISTEN+5)]+2
"IF" I=2 "THEN" "BEGIN" DATERR(-26); GOTO" OPI "END"
ZOP1: I:=I+LOCNAME(1,ISTC[N+3])];
"IF" I>0 "THEN" "BEGIN" DATERR(-27); "GOTO" OPI "END"
"IF" M>1 "THEN"
"BEGIN" M:=ISTC1+2]; "IF" M>NE"0 "THEN" I:=M "END"
M:=ISTC[N+4]; DPOINT(I,K); K:=K+3+J+J; LENGTH(I,I);
"IF" M>LE"0" OR"K>I "THEN" DATERR(-28) "ELSE" ISTC[N+6]:=J;
"END"; OPI; N:=7
OUT1: "IF" ISTEN+1]>0 "THEN" DATERR(-23); M:=ISTC[ISTC3+6]
"FOR" N:=ISTC[N+1] "WHILE" N<NE"0 "DO"
"FOR" I:=N+3,N+4 "DO"
"BEGIN" I:=ISTC1]; "IF" I>LE"0" OR" I>M "THEN" DATERR(-24) "END"
CHFINI;"END" CHOPTS;
PUNCH(OUTCHAN); READER(INCHAN);
PREFIX(' ') DIGITS(4);
CHAR:=2; FAULT:=FALSE; "PRINT" "'L'** DATA'L'";
"COMMENT" STATEMENTS FOR SYNTAX ANALYSIS AND FORMATION OF DATA
STRUCTURE;
DATAINISYNTAX(9, 111
WORD: FNDWRD(21,'NET JUNCT PORTS CIRCUIT ZO <<
>>FREQ OUTPUT SPEC VARY ANALYSE OPTIMISE <<
```

"COMMENT" STATEMENTS FOR SYNTAX ANALYSIS AND FORMATION OF DATA  
STRUCTURE;  
DATAINISYNTAX( 9, 1);  
WORD:=FNDWRD(21,'NET JUNCT PORTS CIRCUIT ZO <<  
>>FREQ OUTPUT SPEC VARY ANALYSE OPTIMISE<<

>>LABEL DATA ERROR RESULT NEWRUN END <<  
>>NCFAULT DELETE LOAD STRUCTUR1, 2);  
"IF" WORD<4 "THEN"  
"BEGIN" "IF" IST[3]:=0 "THEN" IST[3]:=NEWCIT(0) "END"  
"GOTO" MAINWORD];  
  
"COMMENT" NET STATEMENT;  
SNET1 SYNTAX( 2,11); II=INTREP1 RING1=IST[3]+21 BEAD1=LOCNAM(RING,1)  
NETNAM:SYNTAX( 9,12); WORD1=FNDWRD(3,'PAR CONN JUNCT 1, 0);  
"IF" WORD1=0 "THEN" "GOTO" NOTWRD1  
"IF" BEAD1=0 "THEN" DATERR(14);  
"IF" WORD1=2 "THEN" "GOTO" NTCONN1;  
"IF" WORD1=3 "THEN" "GOTO" NTJUNT1;  
  
KI=IST[BEAD1+2]; "IF" KI=0 "THEN" KI=BEAD1;  
SYNTAX( 2,16); DPOINT(K,J); J1=J+2\*INTREP1+INTREP1 LENGTH(K,N);  
"IF" INTREP1 LE 0 "OR" J1 GE N "THEN" DATERR(17);  
SYNTAX( 4,15); INREAL(FLREP,K+J);  
"GOTO" NETNAM1;  
NTCONN1:SYNTAX( 2,18); DPOINT(BEAD,M); J1=INTREP1+21;  
"IF" INTREP1 LE 0 "OR" J1 M "THEN" DATERR(19);  
SYNTAX( 2,20); JUNCT(BEAD,INTREP1,J);  
"GOTO" NETNAM1;  
NTJUNT1:DPOINT(BEAD,N);  
"FOR" J1=3 STEP 1 "UNTIL" M "DO"  
"BEGIN" SYNTAX( 2,21); JUNCT(BEAD,INTREP1,J) "END";  
"GOTO" NETNAM1;  
  
NOTWRD1  
"BEGIN" "COMMENT" READ NETWORK TYPE AND PARAMETERS;  
"INTEGER" RTYPE,RPAR,EETYPE,ECONN,EPAR,JI  
"SWITCH" NET1=ZONLY,CIRCU1,SER1,SHNT,TEE,PHI,NETSET,  
LINE,CPLINE1;  
ECONN1=21 EPAR1=11 RTYPE1=SH1=N1=0;  
EETYPE1=FNDWRD( 8,'CIRCUIT SER1 SHNT TEE PHI <<  
>>TRANSF LINE CPLINE 1,2-1;  
"GOTO" NET1(EETYPE1+2);  
  
ZONLY1: ETYPE1=01 J1=ZTYPE1(EPAR1); "IF" J1=0 "THEN" "GOTO" EXTPRO1;  
ECONN1=11 "GOTO" SRSH1;  
CIRCU1:SYNTAX( 2,23); ETYPE1=NEWCIT(INTREP1);  
SYNTAX( 2,24); ECONN1=INTREP1;  
EPAR1=01 "GOTO" NETSET1;  
SER11: SHNT1:SYNTAX( 2,22);  
J1=ZTYPE1(EPAR1); "IF" J1=0 "THEN" DATERR(22);  
SRSH1 ETYPE1=64+J1+EETYPE1 "GOTO" NETSET1;  
TEE1 PHI1;  
LINE1 EPAR1=4 "GOTO" LGWL1;  
CPLINE1:ECONN1=41 EPAR1=5;  
LGWL1: SYNTAX( 8,25); RTYPE1=J1=FNDWRD( 2,'HL LG 1,26);  
RPAR1=EPAR1 "GOTO" SRSH1;  
EXTPRO1: IF IWSR0 "THEN" DATERR(27); M1=IST[10]; N1=IST[11];  
SYNTAX( 8,28); IST[12]:=STRING1; IST[13]:=STRING2; IST[14]:=0;  
PCALL(10,0);  
RTYPE1=IST[10]; RPAR1=IST[11];  
EETYPE1=IST[12]; ECONN1=IST[13]; EPAR1=IST[14];  
"IF" EPAR1<0 THEN" DATERR(-EPAR1);  
  
NETSET1: IF IWSR0 (BEAD1) BEAD1=K1=DEFINE(EPAR1-EPAR1+3,ECONN1+2,2);  
INSERT(RING,READ,1); INT(BEAD1+ECONN1+4)=EETYPE1;

```

    "IF" RTYPE=0 "THEN" "GOTOB RPARS"
    IST[BHEAD+2]=K1=DEFINE(RPAR+RPAR+3,0,1)
    IST[K+1]=M; IST[K+2]=N; IST[K+3]=RTYPE;
    RPARS: DPOINT(K,N); N=N+2; LENGTH(K,M);
    "FOR" H=N+2 "WHILE" M>H "DO"
    "BEGIN" SYNTAX( 4,15) INREAL(FLREP,K+N) "END"
    "END" READ NETWORK TYPE AND PARAMETERS
    "GOTO" NATNAH

    "COMMENT" JUNCT STATEMENT
    SJUNCT:SYNTAX( 3,31) WORD1=FNDWRD( 2,'SERIES PARALLEL',32)
    RING1=IST[3]+31 WORD2=WORD+WORD-31
    NEXTJ: SYNTAX( 3,33) BEAD1=LOCNAM(RING,INTREP)
    "IF" BEAD1=0 "THEN" DATERR(34)
    IST[BHEAD+4]=WORD1 "GOTO" NEXTJ

    "COMMENT" PORTS STATEMENT
    SPORT; RING1=IST[3]+41 ELIMIN(RING)
    PTNUM1: SYNTAX( 3,41) BEAD1=DEFINE(0,2,2)
    IST[BHEAD+13]=IST(RING1) IST(RING1)=BEAD1
    JUNCT(BEAD,INTREP,2) RING1=BEAD+1 "GOTO" PTNUM1

    "COMMENT" CIRCUIT STATEMENT
    SCIRI: SYNTAX( 2, 3) SYNTAX(17, 4) IST[3]=NEWCIT(INTREP)
    "GOTO" DATA1

    "COMMENT" ZO STATEMENT
    SZ01: SYNTAX( 2,45) BEAD1=LOCNAM(5, INTREP) SYNTAX( 8,46)
    WORD1=ZTYPE(J) "IF" WORD1=0 "THEN" DATERR(47)
    J1=J+J+31 ELIMIN(BEAD1)
    BEAD1=DEFINE(J,1,2) INSERT(5,BEAD,INTREP) IST[BHEAD+3]=WORD1
    K1=31 "FOR" K1=K+2 "WHILE" K1<=J "DO"
    "BEGIN" SYNTAX( 4,48) INREAL(FLREP,BEAD+K) "END"
    SYNTAX( 1,49)

    "COMMENT" FREQ STATEMENT
    SFREQ1 ELIMIN(6) IST[6]=0 K1=51
    NXTRQ1:SYNTAX(13,51)
    WORD1="IF" TYPE=8 "THEN" FNDWRD(2,'STEPLIN STEPLOG ',52) "ELSE" =31
    "IF" TYPE=4 "THEN" FL1=FLREP "ELSE"
    "BEGIN" SYNTAX( 4,53) FL1=FLREP;
    SYNTAX( 4,54) SYNTAX( 2,55) "END"
    BEAD1=DEFINE(WORD+5,1,1) INREAL(FL,BEAD+2)
    "IF" TYPE=2 "THEN"
    "BEGIN" "IF" WORD1=1 "THEN"
    "BEGIN" FL1=(FLREP-FL)/INTREP1 FLREP1=FLREP+0,1*FL "END"
    "ELSE" "BEGIN" FL1=(FLREP/FL)+(ONE/INTREP)
    FLREP1=(ONE+0,1*(FL-ONE))/FLREP "END"
    INREAL(FLREP,BEAD+4) INREAL(FL,BEAD+6)
    "END"
    IST[K+1]=BEAD1 K1=BEAD1
    "GOTO" NXTRQ1

    "COMMENT" OUTPUT AND SPEC STATEMENT
    SOUT1 SSPEC1 ELIMIN( WORD ) IST[WORD]=0 MINWORD=7; RING1=WORD+1
    NXTOPT1:SYNTAX( 9,61)
    NEXTOP1:INTREP1=1 K1=0
    WORD1=FNDWRD( 9,/ZY SPAR ZO ZIN YO <<
    >>YIN VSHR ZYMX SPMX 1,62)
    "IF" WORD1<0 "THEN" SYNTAX( 2,63)
    "ELSE" "IF" MINWORD>0 "THEN" DATERR(62) J1=INTREP1

```

```

"IF" RTYPE=0 "THEN" "GOTOB" RPARS1
IST[BHEAD+2]I=K1=DEFINE(RPAR+RPAR+3,0,1)
IST[K+1]I=M1 IST[K+2]I=N1 IST[K+3]I=RTYPE
RPARS1 DPOINT(K,N) I N=N+2 LENGTH(K,M)
"FOR" I I=N+2 "WHILE" N<H "DO"
"BEGIN" SYNTAX( 4,15) INREAL(FLREP,K+N) "END"
"END" READ NETWORK TYPE AND PARAMETERS
"GOTO" NRTRAN1

"COMMENT" JUNCT STATEMENT
SJUNCTISYNTAX( 8,31) WORD1=FNDWRD( 2,'SERIES PARALLEL',32)
RING1=IST[3]+31 WORD1=WORD+WORD-31
NEXTJ1 SYNTAX( 3,33) BEAD1=LOCNAM(RING,INTREP)
"IF" BEAD=0 "THEN" DATERR(34)
IST[BHEAD+4]I=WORD1 "GOTO" NEXTJ1

"COMMENT" PORTS STATEMENT
SPORT1 RING1=IST[3]+41 ELIMIN(RING)
PTNUM1 SYNTAX( 3,41) BEAD1=DEFINE(0,2,2)
IST[BHEAD+1]I=IST[RING1] ISTER[NG1]=BEAD1
JUNCT(BEAD,INTREP,2) RING1=BEAD+1 "GOTO" PTNUM1

"COMMENT" CIRCUIT STATEMENT
SCIRTI SYNTAX( 2, 3) SYNTAX(17, 4) IST[3]=NEWCIT(INTREP)
"GOTO" DATA1H

"COMMENT" ZO STATEMENT
SZOI SYNTAX( 2,45) BEAD1=LOCNAM(5,INTREP) SYNTAX( 8,46)
WORD1=ZTYPE(J1) "IF" WORD=0 "THEN" DATERR(47)
J1=J+J+31 ELIMIN(BEAD)
BEAD1=DEFINE(J1,2) INSERT(5,BEAD,INTREP) IST[BHEAD+3]I=WORD1
K1=31 "FOR" K1=K+2 "WHILE" K1<=J "DO"
"BEGIN" SYNTAX( 4,48) INREAL(FLREP,BEAD+K) "END"
SYNTAX( 1,49)

"COMMENT" FREQ STATEMENT
SFREQ1 ELIMIN(6) IST[6]=01 K1=51
NXTFRQ1 SYNTAX(13,51)
WORD1="IF" TYPE=8 "THEN" FNDWRD(2,'STEPLIN STEPLOG ',52) "ELSE" -3
"IF" TYPE=4 "THEN" FL1=FLREP "ELSE"
"BEGIN" SYNTAX( 4,53) FL1=FLREP
SYNTAX( 4,54) SYNTAX( 2,55) "END"
BEAD1=DEFINE(WORD+5,1,1) INREAL(FL,BEAD+2)
"IF" TYPE=2 "THEN"
"BEGIN" "IF" WORD=1 "THEN"
"BEGIN" FL1=(FLREP+FL)/INTREP FLREP1=FLREP+0,1*FL "END"
"ELSE" "BEGIN" FL1=(FLREP/FL)+(ONE/INTREP)
FLREP1=(ONE+0,1*(FL-ONE))+FLREP "END"
INREAL(FLREP,BEAD+4) INREAL(FL,BEAD+6)
"END"
IST[K+1]I=BEAD1 K1=BEAD1
"GOTO" N4TFRQ1

"COMMENT" OUTPUT AND SPEC STATEMENT
SOUT1 SSPEC1 ELIMIN( WORD ) IST[WORD]=01 M1=WORD-71 RING1=WORD+1
NXTOPT1 SYNTAX( 9,61)
NEXTOP1 INTREP1=1 K1=01
WORD1=FNDWRD( 9,/ZY SPAR ZO ZIN YO <<
>>VIN VSWR ZVMX SPMX 1,62)
"IF" WORD<8 "THEN" SYNTAX( 2,63)

"ELSE" "IF" M1=NE"0 "THEN" DATERR(62) J1=INTREP1

```

NEXTOP:INTREP:=1; K:=3;  
WORD1=FNDWRD( 9,'ZY SPAR ZO ZIN YO <<  
>>YIN VSWR ZYMX SPMX 1,62);  
"IF" WORD<>"THEN" SYNTAX( 2,63)

"ELSE" "IF" M="NE"0 "THEN" DATERR(62); JI=INTREP;  
"IF" WORD<>"THEN" SYNTAX( 2,63))  
"IF" WORD"NE"7 "THEN"  
"BEGIN" SYNTAX( 8,64);  
K="IF" M=0 "THEN" FNDWRD( 3,/CMPX HODARG DBARG 1,65)  
"ELSE" FNDWRD( 5,/REAL IMAG HOD ARG DB 1,65)\*4;  
"IF" (K=3 "OR" K=9) "AND" WORD"NE"2  
"AND" WORD"NE"9 "THEN" DATERR(66))  
"END" "ELSE" "IF" M="NE"0 "THEN" K=4; N=4;  
"IF" M=0 "THEN" "GOTO" BOUTI SYNTAX( 4,67))  
ADDSPC:FL1=FLREP; SYNTAX( 4,68); N=10;  
BOUTI BEAD1=DEFINE(N,1,1); INSERT(RING+1,BEAD,WORD);  
IST[BEAD+3]:=JI IST[BEAD+4]:=INTREP; IST[BEAD+5]:=K;  
"IF" M=0 "THEN" "GOTO" NXTCHI  
INREAL(FL,BEAD+6); INREAL(FLREP,BEAD+8); INREAL(ONE,BEAD+10);  
WEIGHT:SYLTAX(13,61); "IF" TYPE=4 "THEN" "GOTO" ADDSPC;  
"IF" FNDWRD( 1,/WEIGHT ', 0)=0 "THEN" "GOTO" NEXTOP;  
SYNTAX( 4,69); INREAL(FLREP,BEAD+10); "GOTO" WEIGHT;

"COMMENT" LABEL STATEMENT;  
SLABEL[I]:=14;  
STEP1: "IF" I=24 "THEN" "GOTO" NXTCHI;  
I:=I+1; IST[I]:=0; IST[I+1]:=4000; N=3;  
NLETTI: "CODE" XGETASIST  
XINDEXSI  
XLDSCHAR  
XANDILS63  
XLDSH  
XSMC  
XADDSSIM\$0;  
NXTCHI: NEXTCHI; N=N-1;  
"IF" CHAR=27 "THEN" "GOTO" DATAIN;  
"GOTO" "IF" N<0 "THEN" STEP1 "ELSE" NLETTI;

"COMMENT" VARY STATEMENT;  
SVARY1: ELIMIN( 9 ); IST[9]:=0; RING:=8; M:=0;  
NXTVAR:SYNTAX( 9,101);  
NXTV1: WORD1=FNDWRD( 3,/NET ZO CIRCUIT 1,102); SYNTAX( 2,103);  
"IF" WORD=3 "THEN" "BEGIN" M:=INTREP; "GOTO" NXTVAR "END";  
I:=INTREP; SYNTAX( 8,107); FNDWRD( 1,/PAR 1,107);  
SYNTAX( 2,104); BEAD1=DEFINE(9,1,1);  
IST[RING+1]:=BEAD1; RING:=BEAD1;  
IST[BEAD+2]:=WORD1; IST[BEAD+3]:=I; IST[BEAD+4]:=INTREP;  
IST[BEAD+5]:=M; FL1=-n761 INREAL(FL,BEAD+7);  
FL1=-FL1; INREAL(FL,BEAD+9);  
SYNTAX(13,101); "IF" TYPE=0 "THEN" "GOTO" NXTV1;  
FL1=FLREP; SYNTAX( 4,105); "IF" FLREP<FL "THEN" DATERR(106);  
INREAL(FL,BEAD+7); INREAL(FLREP,BEAD+9); "GOTO" NXTVAR;

"COMMENT" ANALYSE STATEMENT;  
SANALY1:SYNTAX(17, 4);  
"IF" FAULT "THEN" DATERR(110);  
CHCIT(IST[3]); CHOPTS("FALSE");  
"IF" FAULT "THEN" "GOTO" DATAIN;  
IST[10]:=0; "PRINT" //L1\*\* ANALYSIS/L1\*\*;  
"GOTO" COMPI;

"COMMENT" OPTIMISE STATEMENT;  
SOPT1:ISYNTAX( 9,111); PFIND(10);  
"IF" IN9=0 "THEN" DATERR(114); IST[14]:=0;

```

SYNTAX( 2,113) I IST[12]:=INTREP I SYNTAX(17,4)
CHCIT(IST[3]) I CHOPTS("TRUE")
"IF" FAULT "THEN" "GOTO" DATAINI
PCALL(10,0) I I=IST[14] I IF" ICO "THEN" DATERR(-1)
"IF" FAULT "THEN" DATERR(110)
"PRINT" //L/* OPTIMISE/L*/
"GOTO" COMP1

"COMMENT" DATA STATEMENTS
SDATAI SYNTAX( 2,21) I SYNTAX(17, 4)
INCHAN=INTREP I READER(INCHAN); "GOTO" ACCEPT1

"COMMENT" ERROR STATEMENTS
SERRORSYNTAX( 2,22) I SYNTAX(17, 4)
OUCHAN=INTREP I PUNCH(OUCHAN); "GOTO" ACCEPT1

"COMMENT" RESULT STATEMENTS
SLPI SYNTAX( 2,23) I SYNTAX(17, 4)
LP1=INTREP I "GOTO" ACCEPT1

ACCEPT1:"PRINT" //L/* ASSIGNED/L// CHAR#2 I "GOTO" DATAINI

"COMMENT" RERUN STATEMENTS
SRERUNSYNTAX(17, 4) I "PRINT" PUNCH(LP), //L2 LAST ADDR #1,IST[1]
"PRINT" //L/* RESTART/L// "GOTO" RESTART1

"COMMENT" END STATEMENTS
SEND1 SYNTAX(17, 4)
RUNEND1:"PRINT" PUNCH(LP), //L2 LAST ADDR #1,IST[1],//L2//
"PRINT" //L/* RUNEND/L// STOP1

"COMMENT" NOFAULT STATEMENTS
SNFLTI SYNTAX(17, 4) I FAULT#="FALSE"; "GOTO" DATAINI

"COMMENT" DELETE STATEMENTS
SDELETISYNTAX( 9,71)
WORD1=FNDWORD( 4, /NET JUNCT CIRCUIT ZO 1,72)+1
"IF" WORD4 "THEN"
"BEGIN" "IF" IST[3]=0 THEN" DATERR(73) I WORD1=IST[3]+WORD "END"
SYNTAX( 2,74) I BEADI=LOCNAME(WORD,INTREP)
"IF" BEADE0 "THEN" DATERR(75) "ELSE" ELIMIN(BEAD)
"IF" IST[3]=READ "THEN" IST[3]:=0
"GOTO" SDELET1

"COMMENT" LOAD STATEMENTS
SLOADI SYNTAX( 9,121) I PROGLOD(' ',10,'')
"IF" IWS#NE0 "THEN" DATERR(122); GOTO SLOAD1

"COMMENT" STRUCTURE STATEMENTS
STRUCTISYNTAX(17, 4); PUNCH(LP); CHAR#2
"PRINT" //L2' DATA STRUCTURE'
"FOR" I:=1 ASTBP# 1 "UNTIL" 14 "DO"
"PRINT" //L1'ISTE',I,' ) #1,IST[19]
"PRINT" //L1' OUTSTRING(IST,I) I I=26
DBEADI "IF" ICISTE1 "THEN"
"BEGIN" DPCINT(I,J); J:=I+J; LENGTH(I,K); K:=K; PRIOR(I,M)
"PRINTA //L1'ADDR #1,I,DIGITS(2),
' PRIOR #1,M, DPOINT #1;
"FOR" I:=I+1 "WHILE" I<LENJ "DO" "PRINT" IST[1]
I:=I+1 "PRINT" / DATA #1

```

```
"PRINT" /*L'ADDR =',I,DIGITS(2),  
    ' PRIOR =',M,' DPOINT ='  
"FOR" II=I+1 "WHILE" I"LE"J "DO" "PRINT" IST[II]  
II=I+1; "PRINT" / DATA E'
```

```
"FOR" II=I+1 "WHILE" I"LE"K "DO" "PRINT" IST[II]  
"GOTO" CBREAD  
"PRINT" /*(211) PUNCH(OUCHAN); "GOTO" DATAIN)  
  
COMPI EXT1=IST[IST[3]+6]  
"END" SEGMENT READ DATA FROM INPUT STREAM  
  
"BEGIN" "COMMENT" SEGMENT CIRCUIT ANALYSIS  
"INTEGER" OPTION,FQBEAD,FQ,FQ1  
"REAL" FREQ,K,PSUM  
"ARRAY" MIXZY,SPARC[1:EXT1:EXT*EXT],ZOC[1:EXT*EXT]  
"INTEGER" "ARRAY" ROW[1:8]  
"ARRAY" BUFC[1:8,1:32]  
  
"PROCEDURE" CMPXEC(AA,ELIM,ROWS,COLS,BACKSB,BB)  
"VALUE" BLIM,ROWS,COLS,BACKSB; "ARRAY" AA,BB  
"INTEGER" ELIM,ROWS,COLS; "BOOLEAN" BACKSB  
"BEGIN" "COMMENT" TRIANGULARISATION OF A COMPLEX MATRIX IN AAC , J  
FOR ROWS 1 TO ELIM USING GAUSS ELIMINATION, WITH ROW  
INTERCHANGES, FOR A SPARSE MATRIX; BACK SUBSTITUTION  
IS CARRIED OUT, IF REQUIRED, FOR ROWS 1 TO ELIM;  
THE RESULTS ARE PLACED IN ARRAY BB;  
"INTEGER" I,JR,K,KR,M,N  
"REAL" PIVR,PIVI  
  
"COMMENT" TRIANGULARISATION; KRIE=1  
"FORM" K:=1 "STEP" 1 "UNTIL" ELIM "DO"  
"BEGIN" KRIEKR+2; ACI=0;  
"COMMENT" FIND LARGEST ELEMENT FOR PIVOT ON COLUMN K;  
"FOR" II=K "STEP" 1 "UNTIL" ELIM "DO"  
"BEGIN" AMAXJ(AACI,KR),AAC[I,KR+1],ACI);  
"IF" ACK<ACI "THEN" "BEGIN" ACI=ACI; NI=I "END" "END";  
"IF" ACK>=20 "THEN"  
"BEGIN" "PRINT" PUNCH(OUCHAN),/* SINGULAR'L';  
"GOTO" SINGLE "END";  
"COMMENT" INTERCHANGE CODEWORDS FOR ROWS K AND NI  
"IF" N=NI "THEN"  
"CODE" XGETASA  
XINDEXK  
XSTRSH  
XJILS219  
XRLS16  
XWBSPIVR  
XGETASA  
XINDEXN  
XSTRSN  
XJILS219  
XLDRSH  
XJILS221  
XRLSPIVR  
XWBCS16  
XLDRSN  
XJILS221  
"COMMENT" INVERT PIVOT;  
INVJ(AACK,KR3,AACK,KR+1J,PIVR,PIVI);  
ASSJ(PIVR,PIVI,AACK,KR1,AACK,KR+1J);  
"COMMENT" ELIMINATE COLUMN K;  
II=KI "FOR" II=I+1 "WHILE" I"LE"ROWS "DO"  
"BEGIN" ZEROJ(AACI,KR),AACI,KR+1J,N); "IF" N=NEMO "THEN"
```

```

"BEGIN" MULTJ(AA[1],KR],AA[1,KR+1],PIVR,PIVI,AC,ACI))
    JRI=KR] "FOR" JR:=JR+2 "WHILE" JR<COLS "DO"
    "BEGIN" ZEROJ(AA[K,JR],AA[K,JR+1],N); "IF" N==N "THEN"
    "BEGIN" MULTJ(AA[K,JR],AA[K,JR+1],AC,ACI,BC,BCI))
        SUBJ(AA[1],JR],AA[1,JR+1],BC,BCI,AA[1,JR],AA[1,JR+1])
    "END" "END"
    "END" "END"
"END" TRIANGULARISATION

"COMMENT" BACK SUBSTITUTION IF REQUIRED
"IF" BACKSB "THEN"
"BEGIN" K1=ELIM+1; KRI=K+ELIM; N=NKR-2
"FOR" K:=K-1 "WHILE" K>0 "DO"
"BEGIN" KRI=KR-2; JR:=N]
    "FOR" JRI=JR+2 "WHILE" JR<COLS "DO"
    "BEGIN" MULTJ(AA[K,JR],AA[K,KR+1],AA[K,JR+1],PIVR,PIVI))
        ASSJ(PIVR,PIVI,AA[K,JR],AA[K,JR+1])
    I:=K] "FOR" I:=I-1 "WHILE" I>0 "DO"
    "BEGIN" MULTJ(AA[I,JR],AA[I,KR+1],PIVR,PIVI,AC,ACI))
        SUBJ(AA[I,JR],AA[I,JR+1],AC,ACI,AA[1,JR],AA[1,JR+1])
    "END" "END"
"END" "END" BACK SUBSTITUTION

"COMMENT" TRANSFER RESULTS INTO ARRAY BB ;
SINGLE;
KRI=ELIM+ELIM; I:=0; N="IF" BACKSB "THEN" 0 ELSE ELIM;
"FOR" M:=M+1 "WHILE" M<=ROWS "DO"
"BEGIN" I:=I+1; JRI:=0; N:=KR]
    "FOR" N:=N+1 "WHILE" N<=COLS "DO"
    "BEGIN" JRI:=JR+1; BBC[I,JR]:=AA[M,N]; "END"
"END";
"END" CMPXEQ;

"PROCEDURE" ANALYS(HCIT,JUNCTS,PORTS,AAMIX);
    "VALUE" HCIT,JUNCTS,PORTS;"INTEGER" HCIT,JUNCTS,PORTS;"ARRAY" AAMIX;
"BEGIN" "COMMENT" THIS PROCEDURE WILL CARRY OUT A CIRCUIT ANALYSIS ON
    A CIRCUIT BEAD AT ADDRESS HCIT . THE NUMBER OF INTERNAL
    JUNCTIONS IS GIVEN BY JUNCTS AND PORTS GIVES THE NUMBER
    OF EXTERNAL PORTS, THE RESULTS OF THE CIRCUIT ANALYSIS
    IS STORED IN THE ARRAY AAMIX
    "INTEGER" RING,NET,CONN,I,J,K,M,N,TYPCON;
    "INTEGER" STYPE,ZTYP,NETYP,EPARS,RPARS;
    "BOOLEAN" ZYEX;
    "REAL" ZC,ZCI,ATT,ANG;
    "ARRAY" MATRIXC1(JUNCTS,1|JUNCTS+JUNCTS);
    "SWITCH" NETTYPE1= ZONLY ,SERI ,SHNT ,TEE ,PHI ,
        TRANSF,LINE ,CPLINE,SPARS ,ZPAR ,
        YPAR ;

"PROCEDURE" EXCHANGE;
"BEGIN" "COMMENT" EXCHANGE VARIABLES TO CHANGE FROM THE PRESENT
    BUF TYPCON TO THE REQUIRED TYPCON
    "INTEGER" IP,IP2,J,JP,JR,II
    IP:=0; IP2:=11
    EXNEXT:IP:=IP+1; "IF" IP>CONN "THEN" "GOTO" FINI
    "CODE" XLDSTYPCON
        XANDSIP2
        XSTSJ
        XLDSSSTYPE

```

```
EXNEXT:IP1=IP+1; "IF" IP>CONN "THEN" "GOTO" FINI  
"CODE" XLDSTYPCON  
XANDSIP2  
XST3J  
XLDSTYTYPE
```

```
XANDSIP2  
XSUBSSJ1  
IP2I=IP2+IP2I "IF" J=0 "THEN" "GOTO" EXNEXT;  
JP1=IP+IP-1; AMAXJ(BUF[IP,JP],BUF[IP,JP+1],AC);  
"IF" AC<=-20 "THEN"  
"BEGIN" "PRINT" PUNCH(OUCHAN),/* EX SINGULAR/L */  
"GOTO" FIN "END";  
INVJ(BUF[IP,JP],BUF[IP,JP+1],AC,ACI); BC1:=ACI BC11:=ACII  
ASSJ(AC,ACI,BUF[IP,JP],BUF[IP,JP+1]);  
JR1:=1; "FOR" J1=1 "STEP" 1 "UNTIL" CONN "DO"  
"IF" JR>NE JP "THEN"  
"BEGIN" MULTJ(BUF[IP,JR],BUF[IP,JR+1],BC,  
BCI,BUF[IP,JR],BUF[IP,JR+1]); JR1=JR+2 "END";  
"FOR" I:=1 "STEP" 1 "UNTIL" CONN "DO" "IF" I>NE IP "THEN"  
"BEGIN" JR1:=1; "FOR" J1=1 "STEP" 1 "UNTIL" CONN "DO"  
"BEGIN" "IF" JA>JP "THEN" "GOTO" NXTJ;  
MULTJ(BUF[I,JP],BUF[I,JP+1],BUF[IP,JR],  
BUF[IP,JR+1],BC,BCI);  
ADDJ(BUF[I,JR],BUF[I,JR+1],BC,  
BCI,BUF[I,JR],BUF[I,JR+1]); NXTJ1; JR1=JR+2 "END";  
MULTJ(BUF[I,JP1],BUF[I,JP+1],AC,ACI,BUF[I,JP],BUF[I,JP+1]);  
"END" J GOTO EXNEXT;  
FINI "END" EXCHGR;
```

```
"PROCEDURE" IMPE01  
"BEGIN" "COMMENT" FORM Z OR Y IN AC,ACI FOR TYPE=ZTYP AND  
PARAMETERS AT ADDRESS=EPARS;  
"INTEGER" JUMP1  
"BOOLEAN" BB;  
ACI:=0; ACII:=0; BB:=ZTYP<6;  
JUMP1;"IF" BB "THEN" ZTYP "ELSE" ZTYP=5;  
BB:=ZYEX "EQUIV" BB;  
"IF" JUMP=1 "THEN" REALOF(EPARS,AC) "ELSE"  
"IF" JUMP=2 THEN" REALOF(EPARS,ACI) "ELSE"  
"IF" JUMP=3 "THEN"  
"BEGIN" REALOF(EPARS,ACI); ACI:=W*ACI "END" "ELSE"  
"IF" JUMP=4 "THEN"  
"BEGIN" REALOF(EPARS,AC); REALOF(EPARS+2,ACI) "END" "ELSE"  
"BEGIN" REALOF(EPARS+2,AC); REALOF(EPARS+4,ACI);  
"IF" ACI>NE0 "THEN" ACI:=ONE/(W*ACI);  
ACI:=W*ACI-ACII; REALOF(EPARS,AC) "END";  
"IF" BB "THEN" INVJ(AC,ACI,AC,ACI);  
"END" IMPE01
```

```
"PROCEDURE" STOZY(LINE); "VALUE" LINE; "BOOLEAN" LINE;  
"BEGIN" "INTEGER" I,J,JR,KR,M,N,P,O,J2,I2; "REAL" YC,YCI;  
"IF" LINE "THEN"  
"BEGIN" ACI:=EXP(-ATT); ACI=ACI*COS(ANG);  
BCI:=ACI*SIN(ANG); BC1=-ACI ACI1=-BCI "END";  
J2:=1; JR:=1; KRI:=CONN=CONN-1; INVJ(ZC,ZC1,YC,YCI);  
"FOR" J1=1 "STEP" 1 "UNTIL" CONN "DO"  
"BEGIN" JR1:=JR+2; KRI:=KR+2;  
"CODE" XLDSTYPCON  
XANDSJ2  
XSTSP1; J2:=J2+J2; I2:=1;  
"IF" P=0 "THEN" "BEGIN" Q1:=JR1 PI=KR "END"  
"ELSE" "BEGIN" PI:=JR1 Q1=KR "END";  
"FOR" I1=1 "STEP" 1 "UNTIL" CONN "DO"  
"BEGIN" "CODE" XLDSTYPCON  
XANDS12
```

```

XSTSII [21+12+12]
"IF" M=0 "THEN" "BEGIN" NI=JRI NI<KR "END"
"ELSE" "BEGIN" NI=JRI NI>KR "END"
"IF" LINE "THEN"
"BEGIN" "IF" I="NE" J "THEN" "GOTO" LAI
ASSJ(ONE,ONE,I,BUF[I,P],BUF[I,P+1])
ASSJ(YC,YCI,BUF[I,Q],BUF[I,Q+1])
"GOTO" IEND "END"
ASSJ(BUF[I,JR],BUF[I,JR+1],AC,AC)
RCI=AC BCII=AC
LAI ASSJ(BC,RCI,BUF[I,M],BUF[I,M+1])
ASSJ(AC,ACI,BUF[I,N],BUF[I,N+1])
"IF" I=J "THEN"
"BEGIN" BUF[I,JR]=BUF[I,JR]+ONE
BUF[I,KR]=BUF[I,KR]+ONE "END"
MULTJ(BUF[I,Q],BUF[I,Q+1],YC,YCI,BUF[I,Q],BUF[I,Q+1])
IEND "END" STEP I
"END" STEP J
CMPXEQ(BUF,CONN,CONN,4*CONN,"TRUE",BUF)
"END" STOZY

"COMMENT" CLEAR MATRIXI
KI=JUNCTS+JUNCTS "FOR" I=1 "STEP" 1 "UNTIL" JUNCTS "DO"
"FOR" J=1 "STEP" 1 "UNTIL" K "DO" MATRIX[I,J]=0

"COMMENT" FORM MIXED MATRIX FOR CIRCUIT IN MATRIXI
RINGI=NETI=HCIT+1
"FOR" NETI=IST[NET+1] "WHILE" NET=NEWRING "DO"
"BEGIN" DPOINT(NET,CONN) CONNI=CONN=2

"COMMENT" CLEAR BUFFER MATRIXI
KI=CONN+CONN "FOR" I=1 "STEP" 1 "UNTIL" CONN "DO"
"FOR" J=1 "STEP" 1 "UNTIL" K "DO" BUF[I,J]=0
!

"COMMENT" FORM EQUIVALENT CIRCUIT FOR NETWORKI
EPARSI=NET+CONN+6 TYPCONI=IST[EPARS-1]
NETYPI=IST[EPARS-2] IF" NETYP<0 "THEN" "GOTO" CIRCUIT
ZTYP=NETYP DIV 64 NETYP=NETYP-64 ZTYP
"IF" TYPCON>GEN "THEN" "GOTO" NETOKI NI=0
AGI TYPCONI=TYPCON+4096 NI=N+1 "IF" TYPCON<0 "THEN" "GOTO" AGI
"COMMENT" FORM EQUIVALENT NETWORKI
RPARSI=IST[NET+2]+4 KI=RPARS=3
"IF" IST[K]>0 "THEN"
"BEGIN" "COMMENT" FORM EQUIVALENT LINE OR CPLINEI KI<RPARS
WLLGI REALOF(RPARS,AC) INREAL(AC,K) REALOF(RPARS+2,RC)
REALOF(RPARS+6,BC) ACI=DBNEP*BC*BCI
REALOF(RPARS+4,BC) ACI=FBBC/BCI
"IF" ZTYP=1 "THEN" ACI=AC/SORT(BC) "ELSE" ACI=AGI/VLIGHTI
INREAL(AC,K+4) INREAL(AC,K+6)
"IF" NETYP=7 "THEN" "BEGIN" "IF" KI<RPARS "THEN"
"BEGIN" KI=K+8I RPARSI=RPARS+8I "GOTO" WLLG "END" "END"
"END" "ELSE"
"BEGIN" "COMMENT" CALL SUBPROGRAM TO FORM EQUIVALENT NETWORKI
IST[13]=EPARSI IST[14]=RPARSI
PCALL(K,N)
"END"

"COMMENT" FORM MIXED MATRIX FOR NETWORK IN BUFFERI
NETOKI
"GOTO" NETYPECNETYP+13I

```

"COMMENT" FORM MIXED MATRIX FOR NETWORK IN BUFFER  
NETCK:  
"GOTO" NETTYPE(NETYP+1)

CFAULT:"PRINT" PUNCH(OUTCHAN), /\* CONN FAULT NETWORK', DIGITS(6),  
ISTCERARS-3], /\*L\*/ "GOTO" DATA1  
CIRCUITANALYS(=NETYP,ISTCS-NETYP],CONN,BUF)  
STYPEI=ISTC7-NETYP] "IF" STYPE"NE" TYPCON "THEN" EXCHGE  
"GOTO" ADDBUF  
ZONLYI ZYEXI=TYRCCN"NE"0] IMPEDI  
ASSJ(AC,ACI,BUF[1,1],BUF[1,2]) "GOTO" ADDBUF  
SERII IF" TYPCON=0 "THEN" "GOTO" CFAULT; KI=3-TYPCONI  
ZYEXI=K=0] IMPEDI BC1=ACI BCII=ACI "GOTO" S03I  
SHNTI "IF" TYPCON=3 "THEN" "GOTO" CFAULT; KI=TYPCONI  
ZYEXI=K"NE"0] IMPEDI BC1=ACI BCII=ACI  
S03I "IF" K"NE"0 "THEN" "GOTO" S12I  
ASSJ(AC,ACI,BUF[1,1],BUF[1,2]) ASSJ(AC,ACI,BUF[2,3],BUF[2,4])  
ASSJ(BC,BCI,BUF[1,3],BUF[1,4]) ASSJ(BC,BCI,BUF[2,1],BUF[2,2])  
"GOTO" ADDBUF  
S12I JI=K+K-1] ASSJ(AC,ACI,BUF[K,J],BUF[K,J+1]) I I=3-TYPCONI  
BUF[1,TYRCCN+TYPCON-1] IF ONEI BUF[1,1]=ONEI  
"GOTO" ADDBUF  
TEEI STYPEI=0] "GOTO" TEEXI  
PHII STYPEI=3I TEEXI JI=ZTYP"0] V"64] ZTYP1=ZTYP=64+JI  
IMPEDI BC1=ACI BCII=ACI ZTYP1=JI IMPEDI  
ADDJ(AC,ACI,BC,BCI,AC,ACI)  
ASSJ(AC,ACI,BUF[1,1],BUF[1,2]) ASSJ(AC,ACI,BUF[2,3],BUF[2,4])  
"IF" NETYP=4 "THEN" "BEGIN" BC1=-BCI BCII=-BCI "END"  
ASSJ(BC,BCI,BUF[1,3],BUF[1,4]) ASSJ(BC,BCI,BUF[2,1],BUF[2,2])  
"IF" STYPE"NE" TYPCON "THEN" EXCHGE "GOTO" ADDBUF  
TRANSF: "IF" TYPCON=0 "THEN" "GOTO" CFAULT  
"IF" TYPCON=3 "THEN" "GOTO" CFAULT  
REALOF(EPARS,AC); "IF" TYPCON=2 "THEN" ACI=ACI  
BUF[2,1]=ACI BUF[1,3]=ACI "GOTO" ADDBUF  
LINEI  
CPLINEI KI=0I CP2I REALOF(EPARS,ZC)I REALOF(EPARS+2,ZCI)  
REALOF(EPARS+4,ATT)I REALOF(EPARS+6,ANG)I ANGI=H=ANGI  
"IF" ZTYP=1 "THEN" ATT=ATT\*SQRT(FREQ)I CONNI=2I  
STOZY("TRUE"); "IF" NETYP"NE"7 "THEN" "GOTO" ADDBUF  
"IF" K"NE"0 "THEN" "GOTO" CPLI  
"FOR" I I=1.2 "DO"  
"FOR" J I=1 "STBPN" 1 "UNTIL" 4 "DO" BUF[I+2,J]=BUF[1,J]  
EPARSI=EPARS+8I KI=1I "GOTO" CP2I  
"FOR" I I=1.2 "DO"  
"FOR" J I=1 "STBPN" 1 "UNTIL" 4 "DO"  
"BEGIN" H I=1 N I=J+4I  
ACI=BUFEI,J)+BUFI,I,J) ACII=BUFEI,J)-BUFI,I,J)  
BUFEI,J)=BUFEI,N)I=0,5\*ACI BUFEI,J)=BUFI,I,N)I=0,5\*ACII  
"END"; CONNI=4I  
"CODE" XLDSTYPCON  
XANDILS3  
XGTSSYPEI STYPEI=4\*STYPE+STYPEI  
"IF" STYPE"NE" TYPCON "THEN" EXCHGE "GOTO" ADDBUF  
SPARSI REALOF(EPARS,ZC)I REALOF(EPARS+2,ZCI)  
EPARSI=EPARS+4I "GOTO" MOVEI  
ZPARI STYPEI=0I "GOTO" MOVEI  
YPARI STYPEI=-1I MOVEI KI=CONN+CONN  
"FOR" I I=1 "STBPN" 1 "UNTIL" CONN "DO"  
"FOR" J I=1 "STBPN" 1 "UNTIL" K "DO"  
"BEGIN" REALOF(EPARS,BUF[I,J])I EPARSI=EPARS+2 "END"  
"IF" NETYP=8 "THEN" STOZY("FALSE") "ELSE" EXCHGE  
"GOTO" ADDBUF

"COMMENT" ADD BUFFER TO FULL MIXED MATRIX]

```

ADDHUF:JI=NET+21 "FOR" I:=1 "STEP" 1 "UNTIL" CONN "DO"
  "BEGIN" J:=J+1; ROWC[J]:=IST[ISTEJ]+5; "END"
  "FOR" I:=1 "STEP" 1 "UNTIL" CONN "DO"
  "FOR" J:=1 "STEP" 1 "UNTIL" CONN "DO"
  "BEGIN" M:=ROWC[J]; N:=ROW[J];
    K:=J+1; NI=N+1;
    ADDJ(MATRIX[M,N],MATRIX[N,1],BUF[I,K],
          BUF[I,K+1],MATRIX[M,N],MATRIX[N,1]);
  "END"

"END" SET UP MIXED MATRIX IN BUFFER FOR SINGLE NETWORK

"COMMENT" REDUCE MATRIX TO EXTERNAL PORTS FOR CIRCUIT IN AAMIXI
  CMPXEQ(MATRIX,JUNCTS-EXT,JUNCTS,JUNCTS+JUNCTS,"FALSE",AAMIX)

"IF" HCIT"NE"IST[3] "THEN" "GOTO" FIN
"COMMENT" FORM CHARACTERISTIC IMPEDANCES OR ADMITTANCES
  JI=1 NET=IST[5]
  FOR" I:=1 "STEP" 1 "UNTIL" EXT "DO"
  "BEGIN" "IF" NET"NE"4 "THEN"
    "BEGIN" EPARS=NET+5; ZTYP=IST[NET+3]; ZYEX="FALSE"
      IHPED=NET=IST[NET+1] "END"
      ASSJ(AC,ACI,ZO[J],ZO[J+1]); JI=J+2
    "END"

"COMMENT" SET UP EQUATIONS AND SOLVE TO GIVE SCATTERING PARAMETERS
  TYPCON=IST[HCIT+7]; MI=KJ=1; NI=EXT+EXT+1
"FOR" JI=1 "STEP" 1 "UNTIL" EXT "DO"
"BEGIN" "CODE" XLDSTYPCON
  XANESK
  XTESTTYPE
  "FOR" I:=1 "STEP" 1 UNTIL EXT DO"
  "BEGIN" ASSJ(MIXZY[I,M],MIXZY[I,M+1],AC,ACI)
    ASSJ(ZO[M],ZO[M+1],ZC,ZCI)
  "IF" STYPE=0 "THEN"
    "BEGIN" "IF" I=J "THEN"
      "BEGIN" ACI=AC+ZCI; ACII=BC[I]=AC)+ZCII; BCII=AC-ZC END"
      "ELSE" ASSJ(AC,ACI,BC,BCI)
    "END" "ELSE"
    "BEGIN" BCII=ACI+ZCII; BCII=AC-ZCII; ZCII=ACI+ZCI; ZCI=AC-ZCI
      ACII=BCI+ZCII; BCII=BC+ZCI; BCII=BCI+ZCII
      "IF" I=J "THEN" "BEGIN" ACI=AC-ONE; BCII=BC-ONE "END"
    "END"
    ASSJ(AC,ACI,BUF[I,M],BUF[I,M+1])
    ASSJ(BC,BCI,BUF[I,N],BUF[I,N+1])
  "END" STEP I
  K=K+1; MI=M+2; NI=N+2
"END" STEP JI

CMPXEQ(BUF,EXT,EXT,4*BXT,"TRUE",SPAR)

"IF" IST[IST[5]+1]"NE"0 "THEN"
"BEGIN" MI=1; "FOR" I:=1 "STEP" 1 "UNTIL" EXT "DO"
  "BEGIN" BUF[1,I]=SQRT(ZO[M]); MI=M+2 "END"
  MI=1; "FOR" J:=1 "STEP" 1 "UNTIL" EXT "DO"
  "BEGIN"
    "FOR" I:=1 "STEP" 1 "UNTIL" EXT "DO"
    "IF" I=N"J "THEN"
    "BEGIN" ZCII=BUF[1,I]/BUF[1,J];
      SPARE[1,M]=ZCII*SPARE[1,M]
    "END"
  "END"

```

```

M:=1; "FOR" I:=1 STEP 1 UNTIL EXT "DO"
"BEGIN"
"FOR" J:=1 STEP 1 UNTIL EXT "DO"
"IF" INE(J) "THEN"
"BEGIN" ZC:=BUF[1,I]/BUF[1,J]
SPAR[I,M]:=ZC*SPAR[I,M]

```

  

```

SPAR[I,M+1]:=ZC*SPAR[I,M+1] "END"; M:=M+2 "END"
"END"; FINI "END" CIRCUIT ANALYSIS PROCEDURE ANALYSI

"PROCEDURE" OUTPUT(OPTION); "VALUE" OPTION; "INTEGER" OPTION
"BEGIN" "INTEGER" I,J,K,L,MOD1
"SWITCH" WR1=WR1,WR2,WR3,WR4,WR5,WR6,WR7,WR8,WR9;

"PROCEDURE" PRMODES
"BEGIN" "COMMENT" OUTPUT RESULT IN CORRECT MODE ON LP1
"IF" MOD=0 "THEN" "PRINT" FREEPOINT(5),/,I,AC,/      ' "ELSE"
"IF" MOD=1 "THEN" "PRINT" AC,PREFIX(' J'),AC1 "ELSE"
"BEGIN" "IF" MOD=2 "THEN" MODJ(AC,AC1,BC) "ELSE" DBJ(AC,AC1,BC)
ARQJ(AC,AC1,BC))"PRINT"BC,GAMELINE,ALIGNED(4,2),BC1"END"
"END"; WRITE1

"PROCEDURE" PRARAY(ST,MAT); "STRING" ST; "ARRAY" MAT
"BEGIN" SCALED(6); DIGITS(2); "PRINT" //L2//,ST
"IF" MOD=1 "THEN" "PRINT" /CMPX\ "ELSE"
"IF" MOD=2 "THEN" "PRINT" /MODARG\ "ELSE" "PRINT" /DBARG\
"PRINT" / AT FREQ =',FREEPONT(5),FREQEN=9,' GHZ'
K:=0; "FOR" J:=1 STEP 1 UNTIL EXT "DO"
"BEGIN" "IF" K=0 "THEN" "BEGIN" K:=3; "PRINT" //LS3//"END"
"ELSE" K:=K+1 "IF" MOD=1 "THEN" "PRINT" //S6//
"PRINT" //S11//PORT =',J END"
"FOR" I:=1 STEP 1 UNTIL EXT "DO"
"BEGIN" "PRINT" //L1//PORT =',I, K:=4;
"FOR" J:=1 STEP 1 UNTIL EXT "DO"
"BEGIN" "IF" K=0 "THEN"
"BEGIN" K:=3; "PRINT" //LS11// "END" "ELSE" K:=K+1
L:=J-J-1; ASSJ(MAT[I],L),MAT[I,L+1],AC,AC1))
PRMODE "END" "END"
"PRINT" //L2//;
"END"; PRARAY

I:=IST[OPTION+3]; J:=IST[OPTION+4]; J:=J+1;
K:=IST[OPTION+2]; MOD:=IST[OPTION+5]; "GOTO" WR[K];
WR1: ASSJ(MIXZY[I,J],MIXZY[I,J+1],AC,AC1)); "GOTO" WRITE1
WR2: ASSJ(SPAR[I,J],SPAR[I,J+1],AC,AC1)); "GOTO" WRITE1
WR3: ASSJ(ZO[J],ZO[J+1],AC,AC1)); "GOTO" WRITE1
WR5: INVJ(ZO[J],ZO[J+1],AC,AC1)); "GOTO" WRITE1
WR4: MULTJ(ZO[J],ZO[J+1],SPAR[I,J],SPARE[I,J+1],AC,AC1));
AC1=AC*ZO[J]; AC11=AC1*ZO[J+1];
SUBJ(ONE,ONE,I,SPAR[I,J],SPARE[I,J+1],BC,BC1));
"IF" K=4 "THEN" DIVJ(AC,AC1,BC,BC1,AC,AC1)
"ELSE" DIVJ(BC,BC1,AC,AC1,AC,AC1)) "GOTO" WRITE1
WR7: MODJ(SPAR[I,J],SPAR[I,J+1],AC1);
AC1=(ONE*AC)/(ONE-AC)) "GOTO" WRITE1
WR8: PRARAY('ZYM ',MIXZY)) "GOTO" WRFINI
WR9: PRARAY('SPMX ',SPAR)) "GOTO" WRFINI

WRITE1: "IF" MOD<4 "THEN" PRMODE "ELSE"
"IF" MOD>6 "THEN" AC1=AC1 "ELSE"
"IF" MOD>7 "THEN" MODJ(AC,AC1,AC) "ELSE"
"IF" MOD>8 "THEN" ARQJ(AC,AC1,AC) "ELSE"
"IF" MOD>9 "THEN" DBJ(AC,AC1,AC) ;
WRFINI "END" PROCEDURE OUTPUT1

```

```

FUNCTION(LP)
  "IF" ISTC10]"NEQ" "THEN" "GOTO" OPTIM
  "PRINT" "[L2]", IWS:=15; OUTSTRING(IST,IWS)
FOR=ISTC10+2]-8] "IF" FOR0 "THEN"
"BEGIN" "COMMENT" PRINT TABLE HEADING$;
  "INTEGER" OPTION,I,J,K
  "SWITCH" LINE1=FST1,FST2,FST3,FST4,FST5,FST6,FST7]
  "SWITCH" UNITS=UNIT1,UNIT2,UNIT3,UNIT4,UNIT5,UNIT6,UNIT7]
  "SWITCH" LINE2=UNITZY,"IF" J=3 "THEN" UNIT7 "ELSE" UNITS[2+J]
    UNITS[J],UNITS[J],UNITS[J+4],UNITS[J+4],UNIT3]
  DIGITS(2); SAMELINE; "PRINT" "[L2S3'FREQ/SC]", OPTION=6]
SLINE1|OPTION=ISTCOPTION+1]; "IF" OPTION=0 "THEN" "GOTO" SECL
  I:=ISTCOPTION+3]; JI=ISTCOPTION+4]; KI=ISTCOPTION+2]
  "GOTOM" "IF" KKA "THEN" LINE1[KJ] "ELSE" SLINE1]
FST1| "PRINT" / MIXED ",I,J" "GOTOM" SLINE1]
FST2| "PRINT" / SPAR ",I,J" "GOTOM" SLINE1]
FST3| "PRINT" / CHAR IMPED ",I" "GOTOM" SLINE1]
FST4| "PRINT" / INPUT IMPED ",I" "GOTOM" SLINE1]
FST5| "PRINT" / CHAR ADMIT ",I" "GOTOM" SLINE1]
FST6| "PRINT" / INPUT ADMIT ",I" "GOTOM" SLINE1]
FST7| "PRINT" / VSWR ",I," "V" "GOTOM" SLINE1]
SECL | "PRINT" "[L3]"(GHZ) "OPTION=6]
SLINE2|OPTION=ISTCOPTION+1]; "IF" OPTION=0 "THEN" "GOTO" FINHD1
  II=ISTCOPTION+2]; JI=ISTCOPTION+5]
  "GOTOM" "IF" I<9 "THEN" LINE2[I] "ELSE" SLINE2]
UNITZY|KI=0] FOR II=3,4 "DO"
  KI=K+ISTCISTCISTC3+4]+ISTCOPTION+1]+5]
  "GOTOM" UNITSEK+2+J]
UNIT1| "PRINT" / (OHM) "GOTOM" SLINE2]
UNIT2| "PRINT" / (OHM) (DEG) "GOTOM" SLINE2]
UNIT3| "PRINT" / (PU) "GOTOM" SLINE2]
UNIT4| "PRINT" / (PU) (DEG) "GOTOM" SLINE2]
UNIT5| "PRINT" / (MHO) "GOTOM" SLINE2]
UNIT6| "PRINT" / (MHO) (DEG) "GOTOM" SLINE2]
UNIT7| "PRINT" / (DB) (DEG) "GOTOM" SLINE2]
FINHD1 "PRINT" "[L1]
"END" PRINT TABLE HEADING$;

FORREAD=5]
NXTFRQ|FORREAD=1]ISTCFORREAD+1]; "IF" FORREAD0 "THEN" "GOTO" DATA1
  REALOF(FORREAD+2,FREQ)
ANAL1 W=FPI2*FREG1 ANALYS(ISTC3],ISTCISTC3+8],EXT,MIXZY)

"COMMENT" PRINT RESULTS]
  OPTION=6; PREFIX(' ');
  FREEPOINT(4);
  "IF" FOR0 "THEN" "PRINT" FREEPOINT(5),FREQ=9]
"FOR" OPTION=ISTCOPTION+1] "WHILE" OPTION=NEQ0 "DO" OUTPUT(OPTION)
  "PRINT" "[L1]
  LENGTH(FORREAD,OPTION); "IF" OPTION>3 "THEN"
  "BEGIN" REALOF(FORREAD+6,AC); REALOF(FORREAD+4,AC)]
    FREQ="IF" OPTION<8 "THEN" FREQ+AC "ELSE" FREQ*AC
    "IF" FREQ<AC1 "EQUIV" AC0> "THEN" "GOTO" ANAL1
  "END";
  "GOTOM" NXTFRQ1

"COMMENT" ANALYSIS FOR OPTIMISATION]
OPTIM1 ISTC123]=ISTC12]-11 "IF" ISTC12]<0 "THEN" GOTO" PRVARY1
  OPTION=7; ESUM1=0; SCALED(6);
"FOR" OPTION=ISTCOPTION+1] "WHILE" OPTION=NEQ0 "DO"
  "BEGIN" REALOF(OPTION+6,FREQ); W=FPI2*FREQ;

```

OPTION=7; ESUM=0I SCALED(6)  
"FOR" OPTION=IST[OPTION+1] "WHILE" OPTION"NE"0 "DO"  
"BEGIN" REALOF(OPTION+6,FREQ); HI=PI2\*FREQ;

ANALYS(IST[3],IST[IST[3]+8],EXT,MIXZY);  
OUTPUT(OPTION); REALOF(OPTION+8,AC); AC1=AC\*AC1;  
"IF" AC1"NE"0 "THEN" AC1=AC/AC1; INREAL(AC,OPTION+12);  
REALOF(OPTION+10,AC); ESUM1=ESUM+AC1\*AC\*AC1;  
"END" CALC OPTIONS FOR SPEC;  
"PRINT" /\* ESUM =',ESUM,'/L'\*/;  
INREAL(ESUM,13);  
"COMMENT" CALL OPTIMISATION PROGRAM;  
PCALL(10,1); "GOTO" OPT1;  
"COMMENT" PRINT VALUES OF VARIABLES;  
PRVARY;"PRINT" /\*L VARIABLE VALUES/L'\*/;  
OPTION=8; "FOR" OPTION=IST[OPTION+1] "WHILE" OPTION"NE"0 "DO"  
"BEGIN" FQBEAD1=IST[OPTION+6]; "IF" IST[OPTION+2]=1 "THEN"  
"BEGIN" FQ1=IST[FQBEAD+2]; "IF" FQ"NE"0 "THEN" FQBEAD1=FQ "END";  
FQ1=IST[OPTION+4]; DPOINT(FQBEAD,FQ1);  
FQBEAD1=FQBEAD+FQ1\*2+FQ\*FQ1; REALOF(FQBEAD,AC);  
"PRINT" 'C1',IST[OPTION+5];  
"IF" IST[OPTION+2]=1 "THEN" "PRINT" /\* NET \*/"ELSE" "PRINT" /\* Z0 \*/;  
"PRINT" IST[OPTION+3],/\* PAR \*/,FQ,/\* = \*/,AC,'/L'\*/;  
"END" PRINT VARIABLES;  
"GOTO" DATA1;  
"END" SEGMENT CIRCUIT ANALYSIS;  
"END" PROGRAM MICRO3 BY B G MARCHENT;

&PRINT|54|

CPU TIME = 0000 36,505

PRINT-UP OF MIC3D      USER ES/R001/

```
DC 3
IL MIC3D
FC PC
IS 3D
DC 3
FC ""T,
IS FSTEPS.
FC J
IS ,DISPL[112000]
FL PRGLOD()
IS MIC3D ,+
DC +
FL RESTARI
IS FSTEPST=0
FL DATAI
FL STRUCT
IS ,SDISPL
FL DATAINI
FC =2
CS 2
FC DCR
IS DISPLAY ,
DC +
FL STRUCTI
FL "END"
FL "PRINT"
FE
IB
SDISPLISYNTAX(17, 4); "IF" FSTEPS=0 "THEN" "GOTO" DATAINI
"PRINT" //(* DISPLAY PLOT/L*) "GOTO" PLOT
*
FL "ARRAY"
FC JJ
IS ,AA[0]EXT+EXT+4*EXT+EXT]
FL ADDBUFI
FL CMPXEQ(BUF,
FL "END"
IB
"IF" IST[10]=0 "THEN"
"BEGIN" "COMMENT" TRANSFER RESULTS TO WORKFILE CHANNEL 25
  DCHECK(25); AA[0]:=FREQ; M:=0; K:=EXT+EXT; N:=K+EXT;
  "FOR" J:=1 "STEP" 1 "UNTIL" K "DO"
    "BEGIN" M:=M+1; AA[M]:=Z0[J]; "END"
  "FOR" I:=1 "STEP" 1 UNTIL" EXT "DO"
    "FOR" J:=1 "STEP" 1 "UNTIL" K "DO"
      "BEGIN" M:=M+1; AA[M]:=SPARE[I,J];
        AA[M]:=N; J:=MIXZYC[I,J]; "END"
      M:=RANGE(AA,1); DWRITE(25,AA,0,M); FSTEPST=FSTEPS+1
    "END" TRANSFER RESULTS TO WORKFILE CHANNEL 25
*
FL "END" PRINT TABLE
FC IB
IS DOPEN(25,0,0); FSTEPST=0
FC OM
IS "BEGIN" DCHECK(25); "GOTO" DATA "END"
DE
```

```

FL "END" PRINT TABLE
FC 131
IS DOPEN(25,0,1); FSTEP$1=0
FC 0"
IS "BEGIN" DCHECK(25); "GOTO" DATA "END"
DE

FL "END" SEGMENT
IB

PLOT1
"BEGIN" "COMMENT" SEGMENT PLOT RESULTS ON CRT DISPLAY
  "INTEGER" "ARRAY" BUF[1:1000],LAB[1:18]
  "INTEGER" GRAPH,PLOT,GRID,OPTION,MODE,
    IND1,INDJ,FN,XSTEP,XNO,YSTEP,YNO,I,J,K,L
  "REAL" XLOW,YLOW,XHIGH,YHIGH,XINC,YINC,XSTART,YSTART,MARK1
  "ARRAY" PLY,PLY[1:5,1:FSTEP$1]

  "INTEGER" "PROCEDURE" SEEPEN(ITEML,ITEMU)
    "VALUE" ITEMU,ITEMU1 "INTEGER" ITEMU,ITEMU1
  "BEGIN" "INTEGER" J,FRAMES,SEEN1
  AG11 SEEN1=ACTION(5,FRAMES)
    "IF" SEEN1=ITEML "THEN" "GOTO" AG11
    "IF" SEEN1>ITEMU "THEN" "GOTO" AG11
    SEEN1=SEEN1
  AG21 FRAMES1=2 J1=ACTION(6,FRAMES)
    "IF" J1=0 "THEN" "GOTO" AG21
    "FOR" J1=ITEML "STEP" 1 "UNTIL" ITEMU "DO" DELETE(J)
  "END" SEEPEN

  "PROCEDURE" LABBL1
  "BEGIN" "INTEGER" I,JI
    NEWBUF(BUF); POINT(100,1000+30*GRAPH,0) I SETP(2,2,0,0)
    "PRINT" GRAPH,() I I=1 OUTSTRING(LAB,I)
    I=2000+GRAPH; JI=-1 "IF" GRAPH<0 "THEN" JI=0
    "IF" IND1=0 "THEN" DRAW(I,J) "ELSE" REPLACE()
  "END" LABEL1

  "INTEGER" "PROCEDURE" MENU(SUF,WORDS,ST)
    "VALUE" SUF,WORDS; "INTEGER" SUF,WORDS1 "STRING" ST
  "BEGIN" "INTEGER" LABADD,J,K1
    XGETASLAB
    XINDEX1LS6
    XSTRSLABADD1
    "FOR" K1=1 "STEP" 1 "UNTIL" WORDS "DO"
    "BEGIN" NEWBUF(BUF); POINT(880,810-70*K,0) I SETP(2,2,0,0)
    "CODE" XGETSSST
      XADRSK
      XADRSK
      XSUBRILS2
      XBLIMSO
      X'BIIISLABADD1
      JI=61 OUTSTRING(LAB,J) I DRAH(K,0)
    "END"
    MENU1=K1=SEEPEN(1,WORDS)
    "IF" SUF=0 "THEN" "GOTO" FINI
    "CODE" XGETASLAB
      XINDEXSSUF
      XSTRSLABADD
      XGETSSST
      XADDPSK
      XADRSK
      XSUBRILS2
      XBLIMSO
      X'TIIISLABADD1 LABEL1
  FINI "END" MENU1

```

```
FC 1#  
IS DOPEN(25,0,0); FSTEPS=0;  
FC 0"  
IS "BEGIN" DCHECK(25); "GOTO" DATA "END"  
DE
```

```
PL "END" SEGMENT  
IB  
  
PLOT;  
"BEGIN" "COMMENT" SEGMENT PLOT RESULTS ON CRT DISPLAY;  
"INTEGER" "ARRAY" BUF[1:1000],LAB[1:8];  
"INTEGER" GRAPH,PLOT,GRID,OPTION,MODE,  
INDI,INDJ,FQ,XSTEP,XNO,YSTEP,YNO,I,J,K,LI  
"REAL" XLOW,YLOW,XHIGH,YHIGH,XINC,YINC,XSTART,YSTART,MARKI  
"ARRAY" PLX,PLY[1:5,1:FSTEPS];  
  
"INTEGER" "PROCEDURE" SEEPEN(ITEML,ITEMU);  
"VALUE" ITEML,ITEMU; "INTEGER" ITEML,ITEMU;  
"BEGIN" "INTEGER" J,FRAMES,SEEN;  
AG1: SEEN:=ACTION(5,FRAMES);  
"IF" SEEN<ITEML "THEN" "GOTO" AG1;  
"IF" SEEN>ITEMU "THEN" "GOTO" AG1;  
SEEPEN:=SEEN;  
AG2: FRAMES:=2; J:=ACTION(6,FRAMES);  
"IF" J>N#0 "THEN" "GOTO" AG2;  
"FOR" J:=ITEML "STEP" 1 "UNTIL" ITEMU DO#0 DELETE(J);  
"END" SEEPEN;  
  
"PROCEDURE" LABEL;  
"BEGIN" "INTEGER" I,JI  
NEWBUF(BUF); POINT(100,1000-30*GRAPH,0); SETP(2,2,0,0);  
"PRINT" GRAPH,(); I:=I+1; OUTSTRING(LAB,I);  
I:=2000*GRAPH; J:=I-1; "IF" GRAPH>1 "THEN" J:=0;  
"IF" IND1=0 "THEN" DRAW(I,J) "ELSE" REPLACE(I);  
"END" LABEL;  
  
"INTEGER" "PROCEDURE" MENU(SUF,WORDS,ST);  
"VALUE" SUF,WORDS; "INTEGER" SUF,WORDS; "STRING" ST;  
"BEGIN" "INTEGER" LABADD,J,K;  
XGETASLAB  
XINDEX$6  
XSTRSLABADD;  
"FOR" K:=1 "STEP" 1 "UNTIL" WORDS DO#0  
"BEGIN" NEWBUF(BUF); POINT(880,810-70*K,0); SETP(2,2,0,0);  
"CODE" XGETSSST  
XADDRSK  
XADDRSK  
XSUBRIL$2  
XBLIM$0  
XWB$1ISLABADD;  
J:=6; OUTSTRING(LAB,J); DRAW(K,0);  
"END";  
MENU:=K:=SEEPEN(1,WORDS);  
"IF" SUF=0 "THEN" "GOTO" FIN;  
"CODE" XGETASLAB  
XINDEX$3UF  
XSTRSLABADD  
XGETSSST  
XADDRSK  
XADDRSK  
XSUBRIL$2  
XLDIM$0  
XST$1ISLABADD; LABEL;  
FIN "END" MENU;
```

```

"INTEGER" "PROCEDURE" SEPORT(SUF); "VALUE" SUF; "INTEGER" SUF;
"BEGIN" "INTEGER" I,KI
  KI=IST[IST+3]+6;
  "FOR" I:=1 "STEP" 1 "UNTIL" K "DO"
    "BEGIN" NEWBUF(BUF); POINT(880,810-70*K,0);
      S2TF(2,2,0,0); "PRINT" I; DRAW(1,0) "END";
    I+=SEPORT; I+=SEOPEN(1,K);
    LAB[SUF]:=I+(I+14)*("IF" SUF=2 "THEN" 64 "ELSE" 4096); LABEL;
  "END" SEPORT;

"PROCEDURE" SCALE(LOW,HIGH,SLSTEP,SLSTAR,RANGE,STEP,NUMBER);
  "VALUE" LOW,HIGH,RANGE;
  "REAL" LOW,HIGH,SLSTEP,SLSTAR; "INTEGER" RANGE,STEP,NUMBER;
"BEGIN" "REAL" AR,EX; "INTEGER" JI
  AR:=HIGH-LOW;
  "IF" AR<=0 "THEN" AR:=-AR; EX:=ONE;
  AR:=100,0*AR/RANGE; EX:=0,1*EX; "GOTO" ST1 "END";
ST1;"IF" AR>10,0 "THEN" "BEGIN" AR:=0,0*AR; EX:=10,0*EX; "GOTO" ST2 "END";
ST2;"IF" AR>10,0 "THEN" "BEGIN" AR:=0,1*AR; EX:=10,0*EX; "GOTO" ST2 "END";
  "FOR" J:=2,5,10 "DO" "IF" J>AR "THEN" "GOTO" SCLJ;
  SCL:=SLSTEP*AR+EX*JI; SLSTAR:=EX*AR+JI;
  NUMBER:=JI*(HIGH-EX)/AR+1; STEP:=RANGE DIV JI;
  "END" OF SCALE;

LAB[5]:=LAE[8]:=393216;
SAMELINE; DIGITS(2); FREEPOINT(4);
GRAPH:=0; RESET(DISPL,21); PUNCH(9);

"COMMENT" SET UP MENU FOR SELECTING RESULTS TO PLOT;
NEWPLT;PLOT:MENU(0,"IF" GRAPH=0 "THEN" 2 "ELSE" 4,
  'DATA' PLOT NEWPLOT ADDDUMP '');
  "IF" PLOT=1 "THEN" "GOTO" DATA;
  "IF" PLOT=4 "THEN"
  "BEGIN" "PRINT" PUNCH(OUCHAN),"/L"+$1 ADVANCE(INCHAN);
  "IF" DECODE(INCHAN)=14 "THEN"
  "BEGIN" "PRINT" PUNCH(OUCHAN),"/L"+$1 DISPLAY DUMP/L$1;
  "COMMENT" ENTER FPLOT;
  JI:=6644692; JI:=4243456; KI:=14;
  "CODE" XLDRLS;
    XJIL$188;
    XJAIL$0;
    "PRINT" PUNCH(OUCHAN),"/L"+$1 END DUMP/L$1;
  "END"; "GOTO" NEWPLT "END";
  "FOR" I:=1 "STEP" 1 "UNTIL" 4 "DO" LAB[I]:=0;
  "IF" GRAPH=5 "THEN" PLOT:=2;
  "IF" PLOT=2 "THEN"
  "BEGIN" RESET(CISPL,21); GRAPH:=0; NEWBUF(BUF);
    POINT(50,1000,0); IJ:=151 OUTSTRING(IST,I);
    DRAW(0,0) "END";
  GRAPH:=GRAPH+1;
  "IF" PLOT=2 "THEN" GRID:MENU(0,3,'RECT POLAR SMITH ');
  "IF" GRID=3 "THEN" "BEGIN" OPTION:=2; LAB[1]:=3209102 "END";
  "ELSE" OPTION:=MENU(1,8=GRID,
    'ZY SPAR ZO ZIN YO YIN VSWR ');
  INDI:=IND; I:=SEPORT(2);
  "IF" OPTION<3 "THEN" INDI:=SEPORT(3);
  "IF" GRID=1 "AND" OPTION<7 "THEN"
    MODE:=MENU(4,"IF" OPTION=2 "THEN" 5 "ELSE" 4,
      'REAL IMAG MOD ARG DB ');
  I:=EXT+EXT; JI:=I+EXT; KI:=I+J+JI DUPEN(25,0,0);

```

```

"BEGIN" "COMMENT" TRANSFER RESULTS FROM DISC CHANNEL 25
"INTEGER" ADDR,SUFJ,L,N
"REAL" FREQ
"ARRAY" AAC0[K],Z0[E1:I],SPAR,MIXZY[1:J]
"SWITCH" PL1=PL1,PL2,PL3,PL4,PL5,PL6,PL7

SUFJ1=INDJ+INDJ-11 ADDR1=(INDI-1)*I+SUFJ1
FQ1=0 K1=K+1 DREAD(25,AA,0,K)
NEXT1 FQ1=FQ1+1 "IF" FQ>FSTEPS "THEN" "GOTO" FIN1
DCHECK(25)1 FREQ1=AAC0[J] L1=0
"FOR" NI=1 "STEP" 1 "UNTIL" I "DO"
    "BEGIN" L1=L1+1 Z0[N1]=AACL1 "END"
"FOR" NI=1 "STEP" 1 "UNTIL" J "DO"
    "BEGIN" L1=L1+1 SPAREN1=AACL1; MIXZY[N1]=AACL1 "END"
    "IF" FQ<FSTEPS "THEN" DREAD(25,AA,0,K)
    "GOTO" PL1[OPTION1]

PL11 ASSJ(MIXZY[ADDR],MIXZY[ADDR+1],AC,AC1); "GOTO" RESULT1
PL21 ASSJ(SPARE[ADDR],SPARE[ADDR+1],AC,AC1); "GOTO" RESULT1
PL31 ASSJ(Z0[SUFJ1],Z0[SUFJ1+1],AC,AC1); "GOTO" RESULT1
PL51 INVJ(Z0[SUFJ1],Z0[SUFJ1+1],AC,AC1); "GOTO" RESULT1
PL41
PL61 MULTJ(Z0[SUFJ1],Z0[SUFJ1+1],SPARE[ADDR],SPARE[ADDR+1],AC,AC1)
    AC1=AC+Z0[SUFJ1]; AC1=AC-Z0[SUFJ1+1]
    SUBJ(ONE,ONE1,SPAP[ADDR],SPAP[ADDR+1],BC,BC1)
    "IF" OPTION=4 "THEN" DIVJ(AC,AC1,BC,BC1,AC,AC1)
        "ELSE" DIVJ(BC,BC1,AC,AC1,AC,AC1) "GOTO" RESULT1
PL71 MODJ(SPARE[ADDR],SPARE[ADDR+1],AC1)
    AC1=(ONE+AC1)/(ONE-AC1); "GOTO" FNEXT1
RESULT1 "IF" GRID>1 "THEN" "GOTO" POLAR
    "IF" MODE=1 "THEN" AC1=AC "ELSE"
    "IF" MODE=3 "THEN" MODJ(AC,AC1,AC1) "ELSE"
    "IF" MODE=4 "THEN" ARGJ(AC,AC1,AC1) "ELSE"
    "IF" MODE=5 "THEN" DBJ(AC,AC1,AC1)
FNEXT1 AC1=FREQ*-91
POLAR1 PLXEGRAPH,F0J1=AC1 PLYEGRAPH,F0J1=AC1
    "GOTO" NEXT1
FIN1 "END" CALCULATE OUTPUT OPTIONS

"COMMENT" DETERMINE LIMITS FOR GRID1
    XLOW1=YLOW1=-70; XHIGH1=YHIGH1=-XLOW1
    "FOR" I1=1 "STEP" 1 "UNTIL" GRAPH "DO"
    "FOR" J1=1 "STEP" 1 "UNTIL" FSTEPS "DO"
    "IF" GRID=1 "THEN"
    "BEGIN" "COMMENT" RECTANGULAR GRID
        "IF" PLX[I1,J1]<XLOW "THEN" XLOW1=PLX[I1,J1]
            "IF" PLX[I1,J1]>XHIGH "THEN" XHIGH1=PLX[I1,J1]
        "IF" PLY[I1,J1]<YLOW "THEN" YLOW1=PLY[I1,J1]
            "IF" PLY[I1,J1]>YHIGH "THEN" YHIGH1=PLY[I1,J1]
    "END" "ELSE"
    "BEGIN" "COMMENT" POLAR GRID
        AC1=PLX[I1,J1]-PLX[I1,J1]*PLY[I1,J1]*PLY[I1,J1]
        "IF" AC>XHIGH "THEN" XHIGH1=AC1
    "END" DETERMINE LIMITS1
    "IF" GRID=NE1 "THEN" XHIGH1=SQRT(XHIGH1)
    "COMMENT" DRAW GRID1      NEWBUF(BUF))
    "IF" GRID=1 "THEN"
    "BEGIN" "COMMENT" DRAW RECT GRID1
        SCALE(XLOW,XHIGH,XINC,XSTART,700,XSTEP,XNO)
        SCALE(YLOW,YHIGH,YINC,YSTART,700,YSTEP,YNO)

```

```

POINT(150, 95,0); SETP(1,2,0,0);
JI=YNO+YSTEP+5; MARKI=XSTART;
"FOR" FQI=0 STEP" 1 "UNTIL" XNO "DO"
"BEGIN" "IF" J<0 "THEN" VECTOR(0,J,1);
    VECTOR(-48,-17,0); "PRINT" MARKI VECTOR(-48,17,0);
    JI=-JI; MARKI=MARKI+XINC;
    "IF" J<0 "THEN" VECTOR(0,-J,1);
    VECTOR(XSTEP,0,0) "END";
POINT(145,100,0); II=XNO+XSTEP+5; MARKI=YSTART;
"FOR" FQI=0 "STEP" 1 "UNTIL" YNO "DO"
"BEGIN" "IF" I<0 "THEN" VECTOR(I,0,1);
    VECTOR(-97,0,0); "PRINT" MARKI VECTOR(1,0,0);
    II=-II; MARKI=MARKI+YINC;
    "IF" I<0 "THEN" VECTOR(-I,0,1);
    VECTOR(0,YSTEP,0) "END";
POINT(420, 20,0); "PRINT" "FREQ (GHZ)";
"END" "ELSE" "IF" GRID=2 "THEN"
"BEGIN" "COMMENT" DRAW POLAR GRID;
    SCALE(-10,XHIGH,XINC,XSTART,400,XSTEP,XNO);
    YINC1=XINC1 XSTART1=YSTART1=0; YSTEP1=XSTEP1;
    POINT(450,450,0); SETP(1,2,0,0); II=0; MARKI=0;
    "FOR" FQI=1 "STEP" 1 "UNTIL" XNO "DO"
    "BEGIN" VECTOR(XSTEP,0,0); II=I+XSTEP1; APC(I,0,I,0,0);
        MARKI=MARKI+XINC1; VECTOR(-48,-22,0);
        "PRINT" MARKI VECTOR(-48,22,0) "END";
    MARKI=II; ACI=ACI; KI=KI;
    "FOR" FQI=0 "STEP" 1 "UNTIL" 5 "DO"
    "BEGIN" II=MARK*COS(ACI); JI=MARK*SIN(ACI); ACI=ACI+0.523598776;
        KI=KI; "IF" K<0 "THEN" "BEGIN" II=-II; JI=-JI "END";
        POINT(450-I,450-J,0); VECOR(I+I,J+J,1) "END";
"END" "ELSE"
"BEGIN" "COMMENT" DRAW SMITH GRID;
    XSTART1=YSTART1=0; XINC1=YINC1=1.0; XSTEP1=YSTEP1=400;
    POINT( 50,450,0); SETP(1,2,0,0); FREEPOINT(2);
    VECTOR(800,0,1); ARC(400,0,400,0,0);
    "FOR" MARKI=0,2,0,5,1,0,2,0,5,0,10,0 "DO"
    "BEGIN" POINT(850,450,0); II=400,0/(MARK+1,0); ARC(I,0,-I,0,0);
        VECTOR(-32,-22,0); "PRINT" MARKI VECTOR(-32,22,0);
        ARC(-I,0,I,0,0); KI=400,0/MARKI; ACI=MARK+MARK+1,0;
        I=800,0/ACI; JI=800,0*MARK/ACI; ARC(0,K,-I,K-J,0);
        VECTOR(-32,-22,0); "PRINT" -MARKI;
        VECTOR(-64,J+J+44,0); "PRINT" MARKI VECTOR(-32,-22,0);
        ARC(-I,J-K,0,-K,0);
    "END";
    FREEPOINT(4);
"END" DRAW GRID;
"IF" PLOT=3 "THEN" REPLACE(1000) "ELSE" DRAW(1000,0);

"COMMENT" PLOT GRAPHS;
II=0; "FOR" II=I+1 "WHILE" I<GRAPH "DO" DELETE(100+I);
"FOR" II=1 "STEP" 1 "UNTIL" GRAPH "DO"
"BEGIN" NEWBUF(BUF);
    "IF" GRID<2 "THEN" POINT(150,100,0) "ELSE" POINT(450,450,0);
    SETP(2,2,0,0); XNOI=YNOI=0;
"FOR" JI=1 "STEP" 1 "UNTIL" FSTEPS "DO"
"BEGIN" KI=XSTEP*(PLXE[I,J]-XSTART)/XINC1;
    LI=YSTEP*(PLYE[I,J]-YSTART)/YINC1;
    VECTOR(K-XNO,L-YNO,J-1); XNOI=KI; YNOI=LI;
    "IF" JI=1 "CR" J=FSTEPS "THEN"
    "BEGIN" VECTOR(-16,0,0); "PRINT" DIGITS(1),I; VECTOR(-16,0,0) "END";

```

```
L=STEP*(PLT+1,JJ=START//TING,  
VECTOR(K=XNO,L=YNO,J=1)) XNO:=K; YNO:=L  
"IF" J=1 "CR" J=FSTEPS "THEN"  
"BEGIN"VECTOR(-16,0,0);"PRINT" DIGITS(1),I;VECTOR(-16,0,0)"END"  
  
"END" STEP JJ  
DRAW(100*I,0);  
"END" PLOT GRAPHS;  
"GOTO" NEWPLT;  
  
"END" PLOT RESULTS ON CRT DISPLAY;  
'  
SH
```

dPRINT:56:

CPU TIME = 0000 41,220

PRINT-UP OF ML1U      USER ES/RD01/

MCINS & &  
MCSKIP NT,[WITHC ]WITH]  
MCSKIP MDT,[ ]

MCDEF ZEROJ WITH ( , , ) AS [{"CODE": "XCLS&A1E",

XFLS22  
XFSIG  
%COMPILE\$0  
%JNZSNZ  
XFLS24  
XFSIG  
\$\$NZ\$ST\$&A3E}]]

MCDEF AMAXJ WITH ( , , ) AS [{"CODE": "XCLS&A1E",

XFLS22  
XFMOD  
XWFS26  
XFLS24  
XFMOD  
XFCPS26  
%JNNSL1  
XFLS26  
SSL1\$WFS&A3E}]]

MCDEF ASSJ WITH ( , , ) AS [{"CODE": "XCLS&A1E",

XWCS&A3E}]]

MCDEF ADDJ WITH ( , , , , ) AS [{"CODE": "XCLS&A1E",

XCAS&A3E  
XWCSE&A5E  
XJOFSRESET  
SSRESET}]]

MCDEF SUBJ WITH ( , , , , ) AS [{"CODE": "XCLS&A1E",

XCSS&A3E  
XWCSE&A5E  
XJOFSRESET  
SSRESET}]]

MCDEF INVJ WITH ( , , , ) AS [{"CODE": "XCL\$NONE",

XCDS&A1E  
XWCSE&A3E  
XJOFSRESET  
SSRESET}]]

MCDEF MULTJ WITH ( , , , , ) AS [{"CODE": "XCLS&A1E",

XCMSE&A3E  
XWCSE&A5E  
XJOFSRESET  
SSRESET}]]

MCDEF DIVJ WITH ( , , , , ) AS [{"CODE": "XCLS&A1E",

XCDS&A3E  
XWCSE&A5E  
XJOFSRESET  
SSRESET}]]

MCDEF MODJ WITH ( , , ) AS [{"BEGIN": "CODE": "XCLS&A1E",

RESET]]

MCDEF MODJ WITH ( , , ) AS [{"BEGIN" "CODE" "XCLSFA1E

"XFLS22  
%FMS22  
%WFS26  
%FLS24  
%FMS24  
%FAS26  
%WF\$EA3E] [A3E:=SCRT(A3E) "END"]]

MCDEF DBJ WITH ( , , ) AS [{"BEGIN" "CODE" "XCLSFA1E

"XFLS22  
%FMS22  
%WFS26  
%FLS24  
%FMS24  
%FAS26  
%WF\$EA3E] [A3E:=CFXDB=LN(A3E) "END"]]

MCSET S10=-50

&PRINT:58;

CPU TIME = 0000 41,735

PRINT-UP OF ML17      USER ES/R001/

MCINS & &  
MCSKIP HT,[WITHC JWITH]  
MCSKIP MDT,[ ]

MCDEF \$ WITH LIT WITH ( ) AS [EMCGO L1 UNLESS EA1E BC N  
[L1E\$EA1E]]

MCDEF INREAL WITH ( , ) AS [EMCGO L1  
[L1E"BEGIN" IWSI=EA1E "CODE" XFL\$EA1E  
XGETASIST  
XINDEXSIWS  
XWFIMSO "ENDH" ]]

MCDEF REALOF WITH ( , ) AS [EMCGO L1  
[L1E"BEGIN" IWSI=EA1E "CODE" XGETASIST  
XINDEXSIWS  
XFLIMSO  
XWF\$EA2E "END" ]]

MCDEF DPOINT WITH ( , ) AS [{"CODE" XGETASIST  
XINDEXSLIT(EA1E)  
XL0IMSO  
XANDIL\$63  
XSTS\$EA2E}]

MCDEF LENGTH WITH ( , ) AS [{"CODE" XGETASIST  
XINDEXSLIT(EA1E)  
XL0IMSO  
XL0KIL\$2  
XSMLC  
XANDIL\$2047  
XSTS\$EA2E}]

MCDEF PRIOR WITH ( , ) AS [{"CODE" XGETASIST  
XINDEXSLIT(EA1E)  
XL0IMSO  
XL0KIL\$3  
XSMLC  
XANDIL\$63  
XSTS\$EA2E}]

MCDEF KEY WITH ( , , , ) AS [{"CODE" XGETASIST  
XINDEXSLIT(EA1E)  
XL0ILS2048  
XADDSSLIT(EA2E)  
XADDSSLIT(EA3E)  
XSTSIMSO  
XL0SSLIT(EA4E)  
XPUTIMSO  
XL0SSLIT(EA3E)  
XPUTIMSO}]

MCSET S10=-50

&PRINT:60;

CPU TIME = 0000 42,153

PRINT-UP OF ML1A      USER FS/R001/

MCINS & L  
MCSKIP M,<WITH< >WITH>  
MCSKIP MT,[WITH< ]WITH]  
  
MCDEF % N1 OPT TN1 OR [N1 OR ,N1 OR ]N1 OR NL OR I OR " ALL AS [[MCGO L6  
L5]MCGO L5 IF ED1E = \$  
[[%]]EA1\$ED1E MC GO L4  
L5]MCGO L1 IF ED2E = [  
[%]]EA1\$EA2E ED2E MC GO L4  
L1]MCSET T1 = 2  
[%]]GETAS\$EA2E  
L2]MCSET T1 = T1 + 1  
[%INDEX]]MC GO L3 UNLESS EA1E BC N  
I[L3]EA1E  
MC GO L2 IF EDT1E = ,  
[%]]EA1E;M\$0 EDT1+1\$EL4E]]

MCSET S10=-50

&PRINT#621

CPU TIME = 0000 42,375

PRINT-UP OF MWC            USER ES/R001/

COMPONENTS  
"BEGIN" "COMMENT" BASIC PROGRAM FOR MICROWAVE COMPONENTS  
  "ARRAY" IST[11]  
  STOP  
"BEGIN" "INTEGER" ENTRY,RTYPE,RPAR,EENTRY,EPAR  
  "COMMENT" ADD OTHER VARIABLES AS REQUIRED  
"PROCEDURE" DUMMY; ENTRY:\$0  
  "COMMENT" THIS DUMMY PROCEDURE MUST OCCUR AS BLOCK 3  
"SWITCH" NETYP1=N1,N2,N3  
"SWITCH" ECTY11=E1,E2,E3  
"SWITCH" ECTY21=F1,F2,F3  
"PROCEDURE" FNDWRD(N,ST)  
  "VALUE" N1 "INTEGER" N1 STRING# ST  
"BEGIN" "COMMENT" FIND WORD IN STRING ST TO MATCH CONTENTS  
  OF RTYPE AND RPAR WITH EPAR GIVING THE ERROR  
  NUMBER IF NO MATCH IS FOUND;  
  "CODE" XBL\$RTYPE  
    %WRS208  
    %GET\$ST  
    XLDSN  
    XJIL\$212  
    %JAIL\$2  
    %NEG\$SEPAR  
    %ADDIL\$1  
    %STS\$RTYPE1  
    "IF" EPAR<0 THEN" "GOTO" EXIT1  
"END" FNDWRD1  
"PROCEDURE" INREAL(FPOINT,H)  
  "VALUE" FPOINT,H1 "REAL" FPOINT1 "INTEGER" H1  
  "CODE" XFL\$FPOINT  
    %GETAIS1  
    %INDEXSH  
    %WFIMSO1  
"PROCEDURE" REALOF(H,FP); "VALUE" H; "INTEGER" H; "REAL" FP1  
"BEGIN" "REAL" FPOINT1  
  "CODE" %GETAIS1  
    %INDEXSH  
    %FLIMSO1  
    %WF\$FPOINT1; FP1=FPOINT1  
"END" REALOF1  
"COMMENT" COPY PARENT VARIABLES TO LOCAL VARIABLES  
"CODE" XAL\$22  
  %STAS1  
  %STSENTRY1  
  RTYPE1=IST[12]; RPAR1=IST[13]; EPARI=IST[14];  
  "IF" ENTRY\$NE"0 "THEN" "GOTO" EQ1  
"COMMENT" BASIC ENTRY1  
  EPARI=1002; FNDWRD( 3,/LIST OF WORDS )  
  "COMMENT" COMMON AREA FOR BASIC ENTRY1

```
"GOTO" NETTYPE[RTYPE];  
"COMMENT" SET RPAR, ETYPE, ECONN, EPAR FOR NETWORK;  
N1I "GOTO" EXIT;  
N2I "GOTO" EXIT;  
N3I "GOTO" EXIT;  
  
"COMMENT" FIRST ENTRY;  
EQ1I RTYPEI=IST[RPAR-1]; "IF" ENTRY"NE"1 "THEN" "GOTO" EQ2I  
ETYPEI=EPAR-1; IST[ETYPE]I=IST[ETYPE]-4096;  
"COMMENT" COMMON AREA FOR FIRST ENTRY;  
"GOTO" ECTY1[RTYPE];  
  
"COMMENT" SET PARAMETERS FOR FIRST ENTRY;  
E1I "GOTO" EQ2I;  
E2I "GOTO" EQ2I;  
E3I "GOTO" EQ2I;  
  
"COMMENT" SECOND ENTRY STARTING WITH COMMON AREA;  
EQ2I "GOTO" ECTY2[RTYPE];  
  
F1I "GOTO" RETURNI;  
F2I "GOTO" RETURNI;  
F3I "GOTO" RETURNI;  
  
"COMMENT" TRANSFER VARIABLES TO PARENT PROGRAM;  
EXITI IST[10]I=RTYPEI IST[11]I=RPARI  
IST[12]I=ETYPEI IST[13]I=ECONN; IST[14]I=EPARI;  
  
RETURNI "END";
```

```
&ENDI  
CPU TIME = 0000 43,636 REAL TIME 00 01 10  
A
```

CPU TIME = 0000 43,634