

## 付 録

### 付録1 関数 $g$ の上, 下界関数の単調減少特性の証明

ここでは, 6.3.3(3)項での式(6.20), (6.13)に導かれた, LRU方式とFIFO方式の性能差関数  $g$  の上界, 下界関数  $g_U, g_L$  が  $f_{LRU}$  (以下  $f$  と略す) の小さい所で単調減少であることを示す。

そのため, まず簡単のため

$$L(i) = \int_{i-1}^i l(x) dx, \quad i=1, 2, \dots \quad (A.1)$$

なる関数  $l(x)$  を導入する。  $l(x)$  は式(6.1)の  $L(i)$  の性質から,  $0 \leq u < v$  の範囲で次の性質を有する。

$$0 < l(v) < \frac{1}{v-u} \int_u^v l(x) dx < l(u) \leq 1 \quad (A.2)$$

そこで,  $g_U$  は, 式(6.2), (6.19), (6.20)などから,

$$g_U(f) = \frac{\int_0^b l(x) dx - \frac{b}{2b-1} \int_0^{2b-1} l(x) dx}{1 - \int_0^b l(x) dx} \quad (A.3)$$

となり, その導関数は次のようになる。

$$\begin{aligned} & \left(1 - \int_0^b l(x) dx\right)^2 \frac{dg_U}{db} \\ &= l(b) - \frac{1 - \int_0^b l(x) dx}{2b-1} \left\{ bl(2b-1) - \frac{\int_0^{2b-1} l(x) dx}{2b-1} \right\} \\ & \quad - \frac{b}{2b-1} l(b) \int_0^{2b-1} l(x) dx \quad (A.4) \end{aligned}$$

ここで, 式(A.2)から,

$$\begin{aligned} \text{式(A.4)右辺} &> l(b) - \frac{bl(2b-1) - l(2b-1)}{2b-1} - \frac{bl(b)}{2b-1} \\ &= \frac{b-1}{2b-1} \{ l(b) - l(2b-1) \} > 0 \quad (A.5) \end{aligned}$$

一方, 式(6.19)から  $db/df < 0$  なので, 結局,

$$\frac{dg_U}{df} = \frac{dg_U}{db} \cdot \frac{db}{df} < 0 \quad (\text{A} \cdot 6)$$

となり、 $g_U$  は単調減少関数であることが示された。

次に、 $g_L$  は、式(6.2)、(6.13)、(6.19)などから

$$g_L(f) = \frac{\frac{1}{b} \int_1^{b+1} l(x) dx - L(b+1)}{1 - \int_0^b l(x) dx} \quad (\text{A} \cdot 7)$$

となり、その導関数は次のようになる。

$$\begin{aligned} & \left(1 - \int_0^b l(x) dx\right)^2 \frac{dg_L}{db} \\ &= \left\{ \frac{1}{b} \int_1^{b+1} l(x) dx - l(b+1) \right\} \cdot \left\{ l(b) - \frac{1}{b} \left(1 - \int_0^b l(x) dx\right) \right\} \\ & \quad + \left\{ l(b) - l(b+1) \right\} \cdot \left\{ 1 - \int_0^b l(x) dx - l(b) \right\} \\ & \quad + l(b) \left\{ l(b) - L(b+1) \right\} \end{aligned} \quad (\text{A} \cdot 8)$$

ここで、式(A.2)から(A.8)の右辺の第1項第1因子、第2項第1因子および第3項は正なので、(A.8)の右辺が正であるためには、第1項第2因子と第2項第2因子が正であれば十分である。即ち、

$$1 - \int_0^b l(x) dx > l(b) > \frac{1}{b} \left(1 - \int_0^b l(x) dx\right) \quad (\text{A} \cdot 9)$$

この条件は、本実験の場合は、 $b$ がある程度大きい、 $b > 10 \sim 30$ のところ、即ち、 $f < 0.09 \sim 0.01$ と $f$ の小さいところで成立する。そして、この場合、

$$\frac{dg_L}{df} = \frac{dg_L}{db} \cdot \frac{db}{df} < 0 \quad (\text{A} \cdot 10)$$

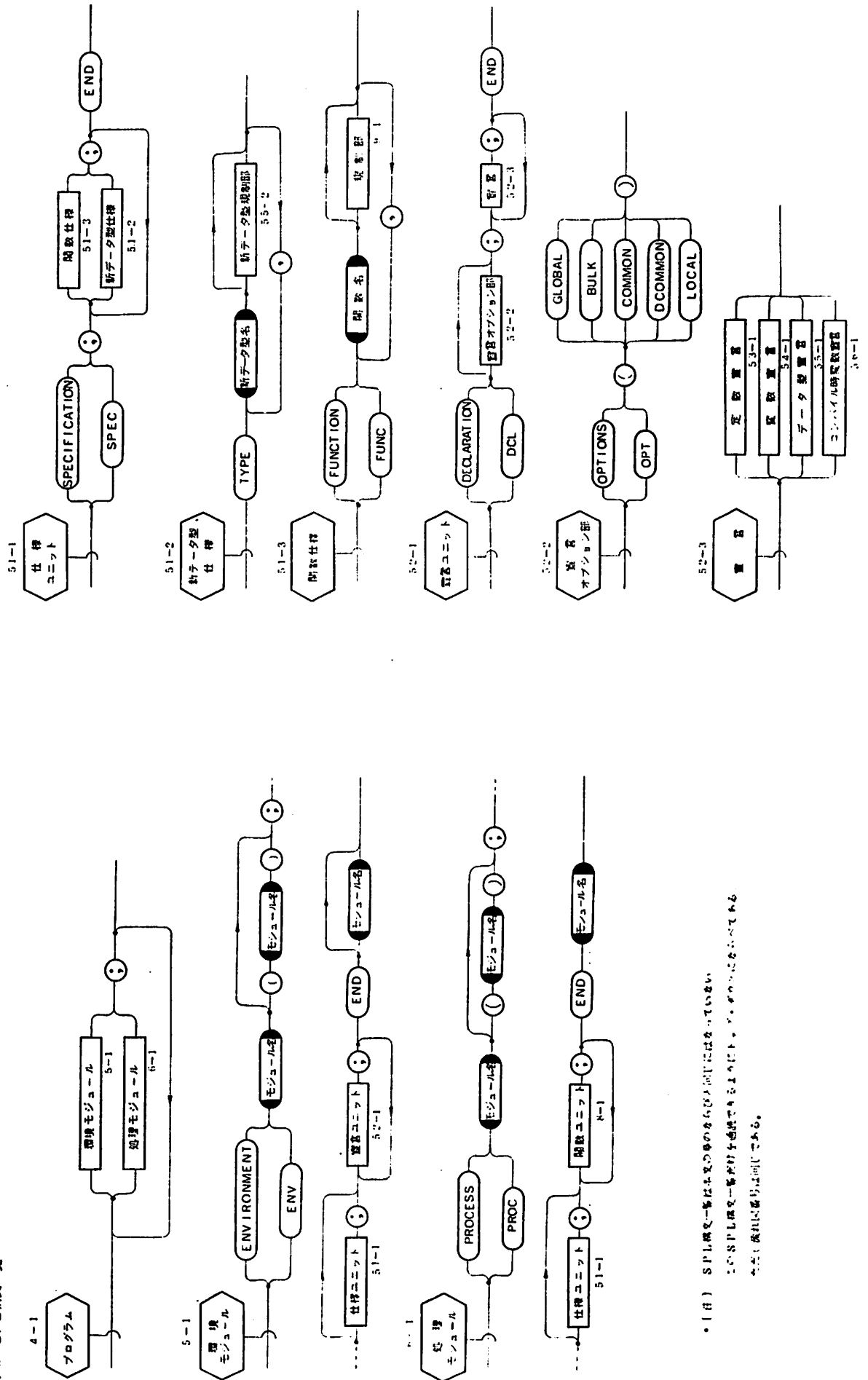
となるので、結局、 $g_L$  は $f$ の小さい所で単調減少関数となる。

付録 2. SPL構文一覧  
 文献 [H4] より転載する。

付録 1

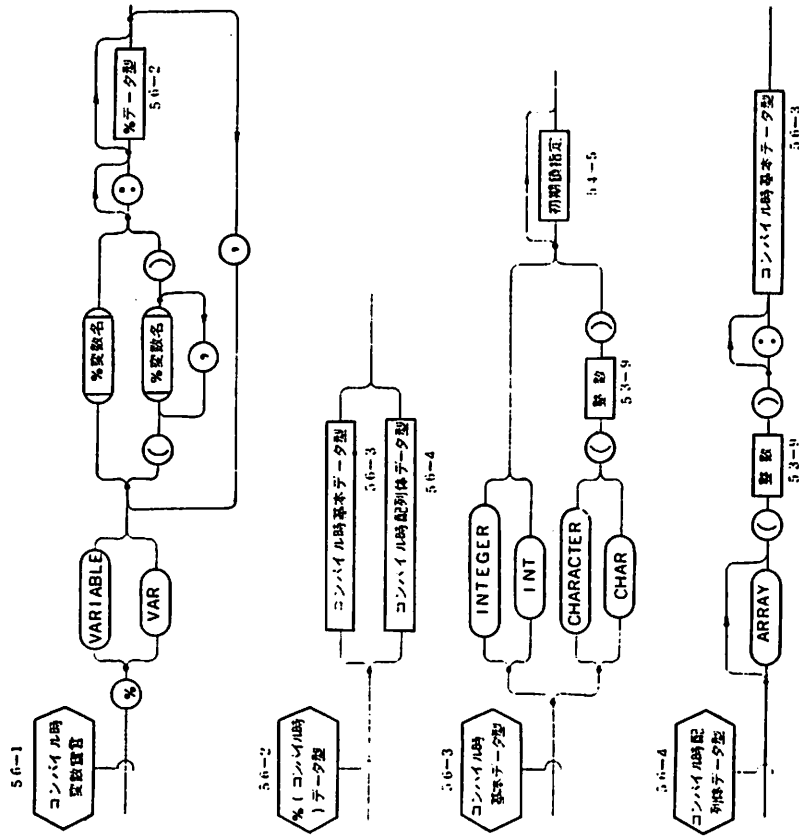
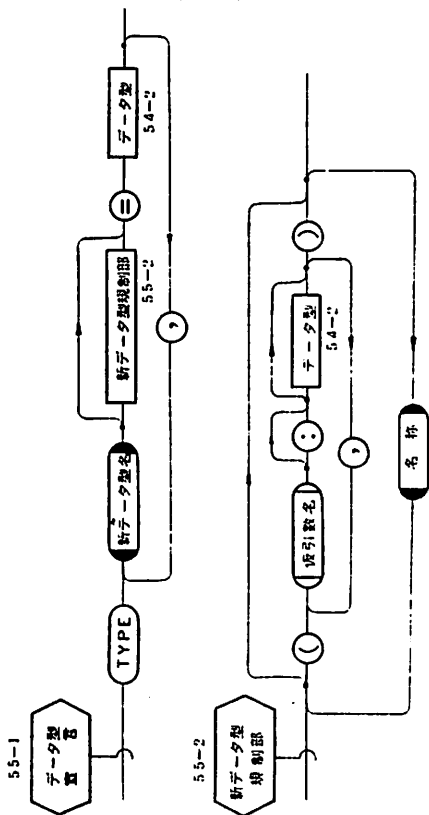
付録 1. SPL構文一覧\*

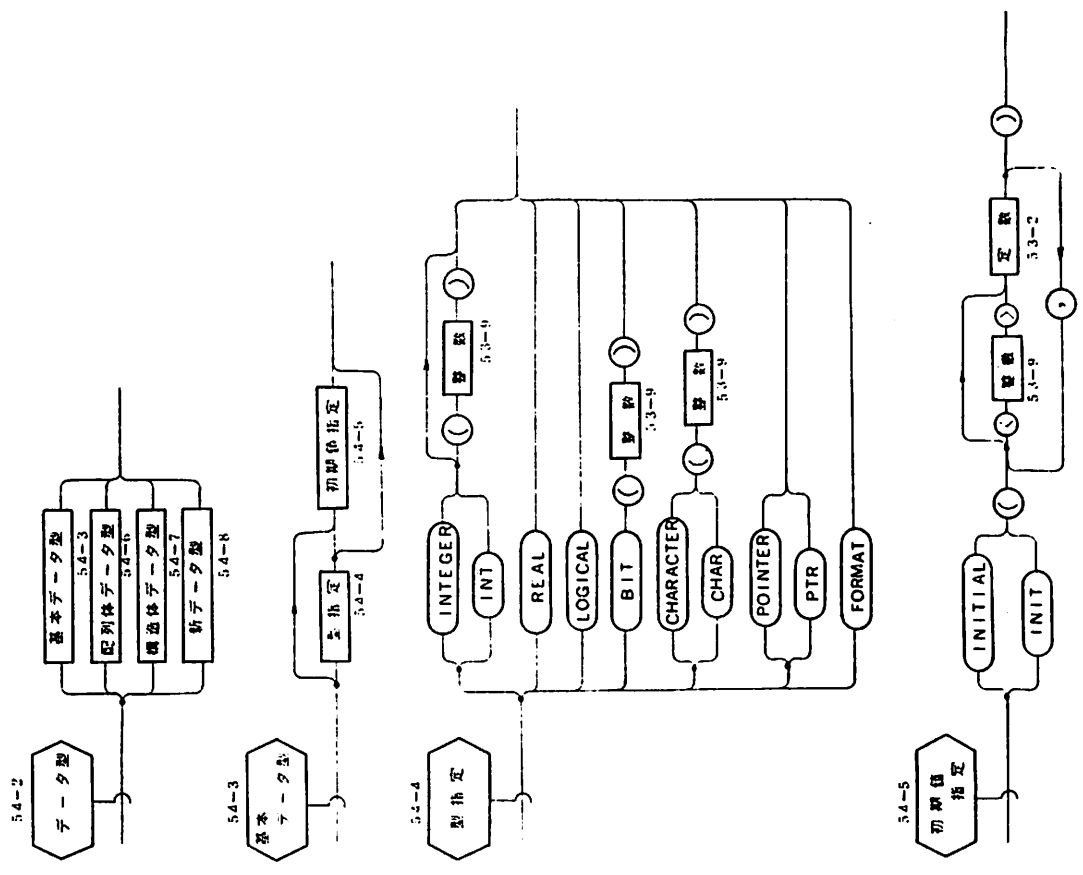
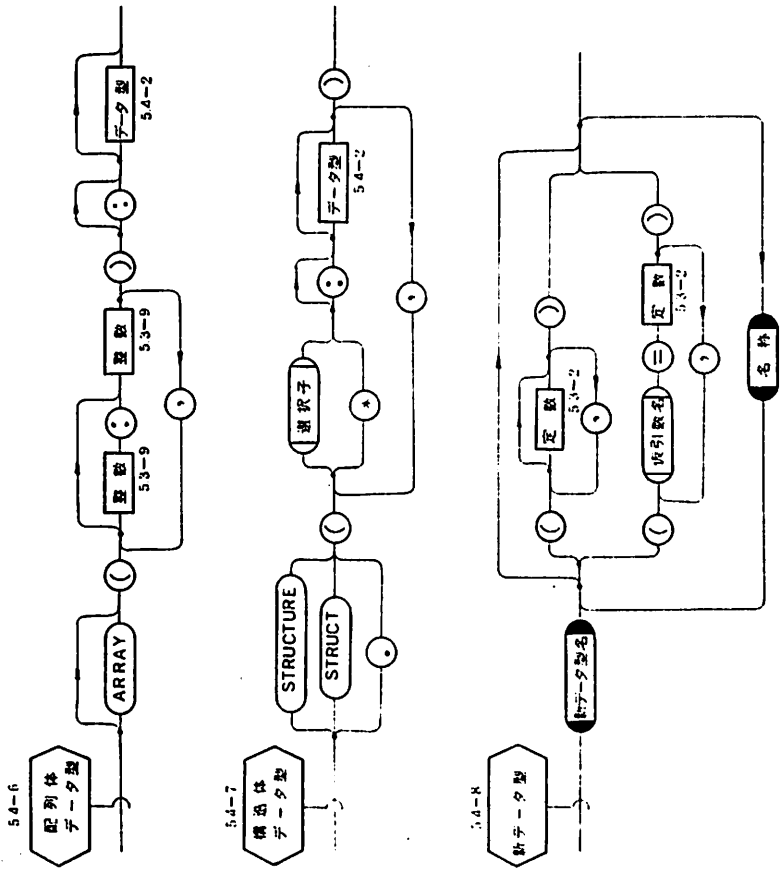
付録 1



\* (注) SPL構文一覧は本文の巻の巻の巻にはなっていない。  
 このSPL構文一覧が対象言語であることにより、その中にはなっていない。  
 本表は表紙に添付してある。



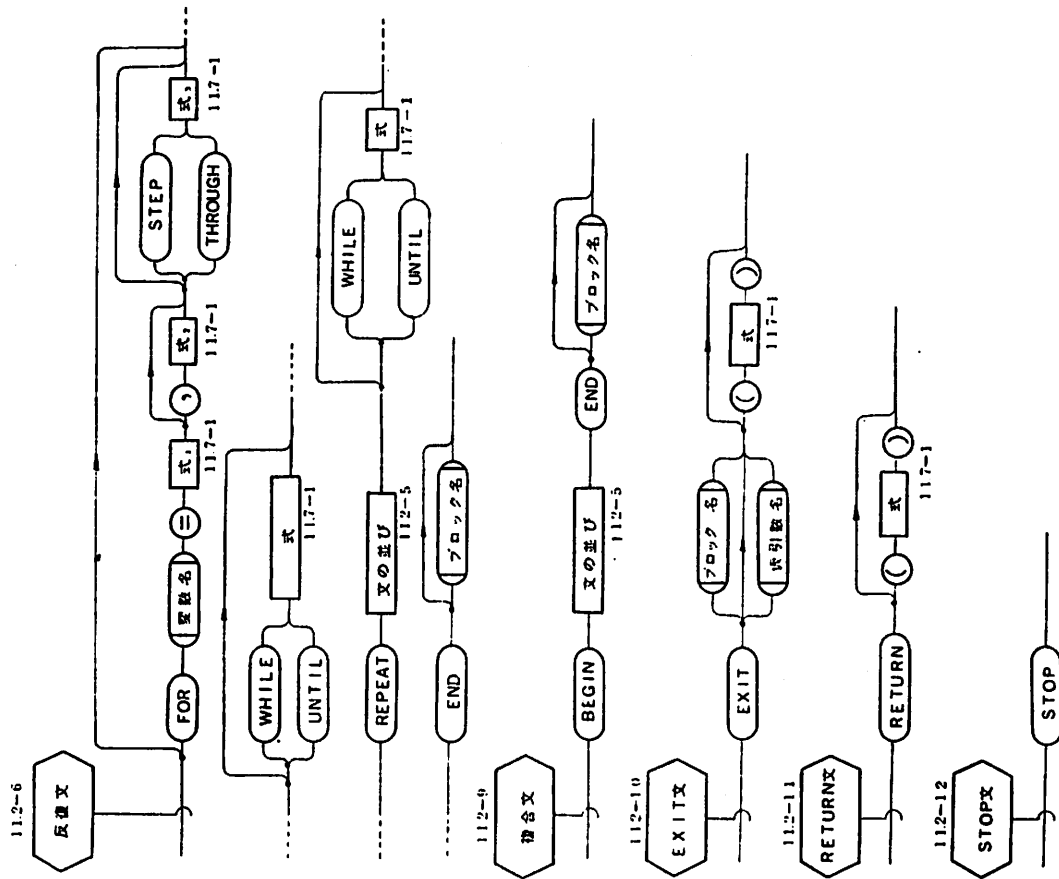
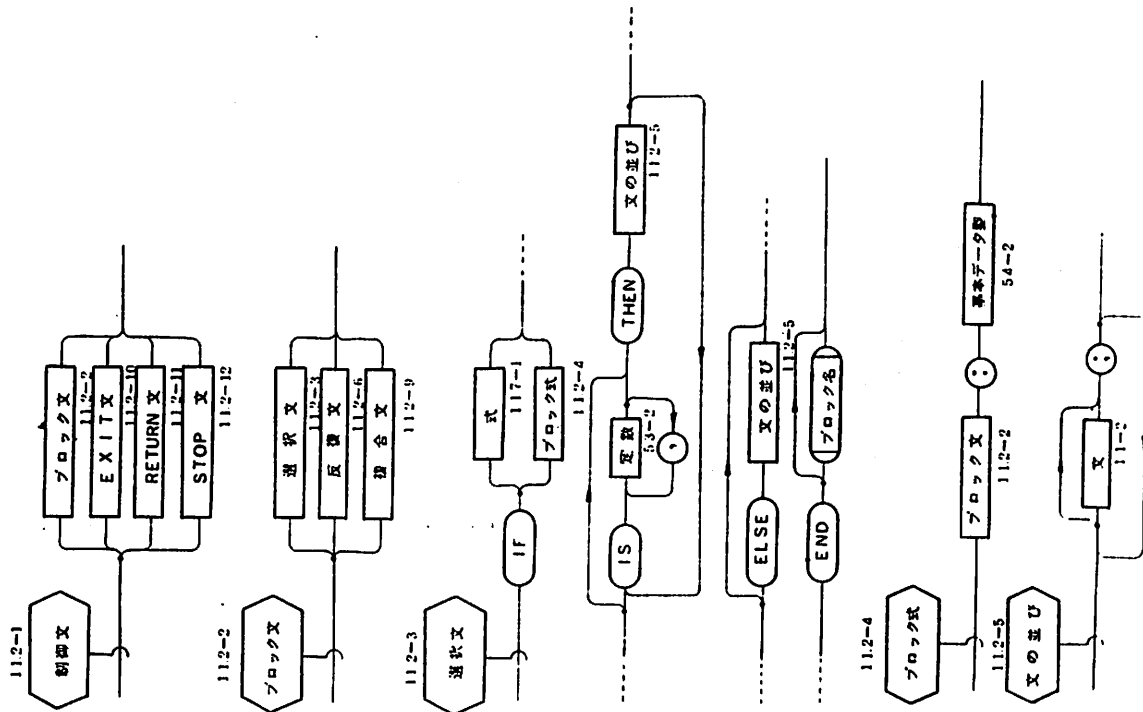


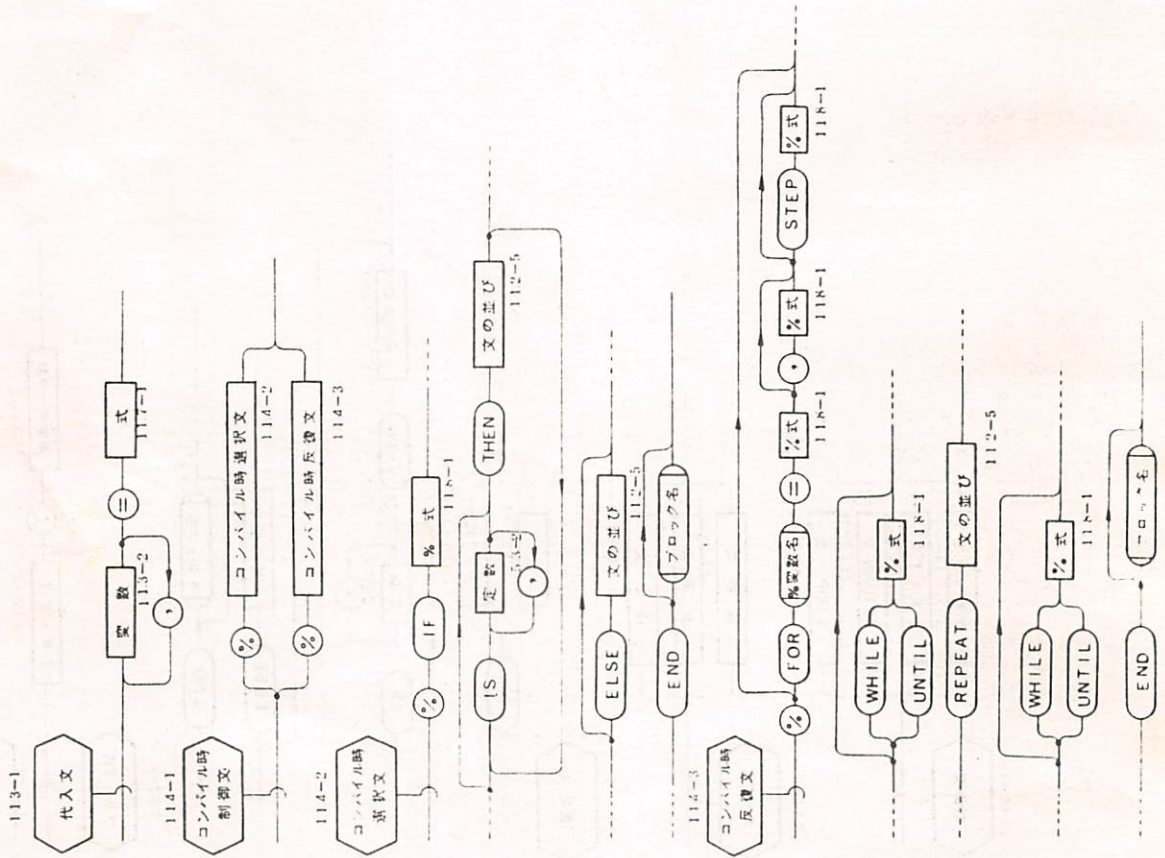
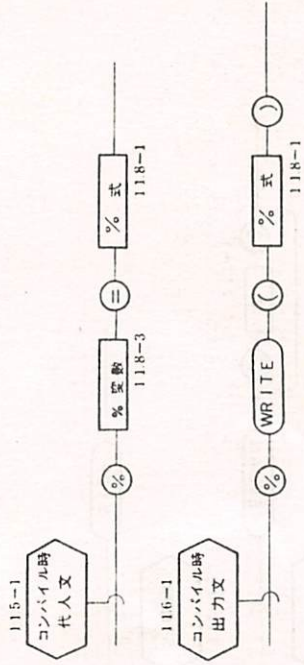


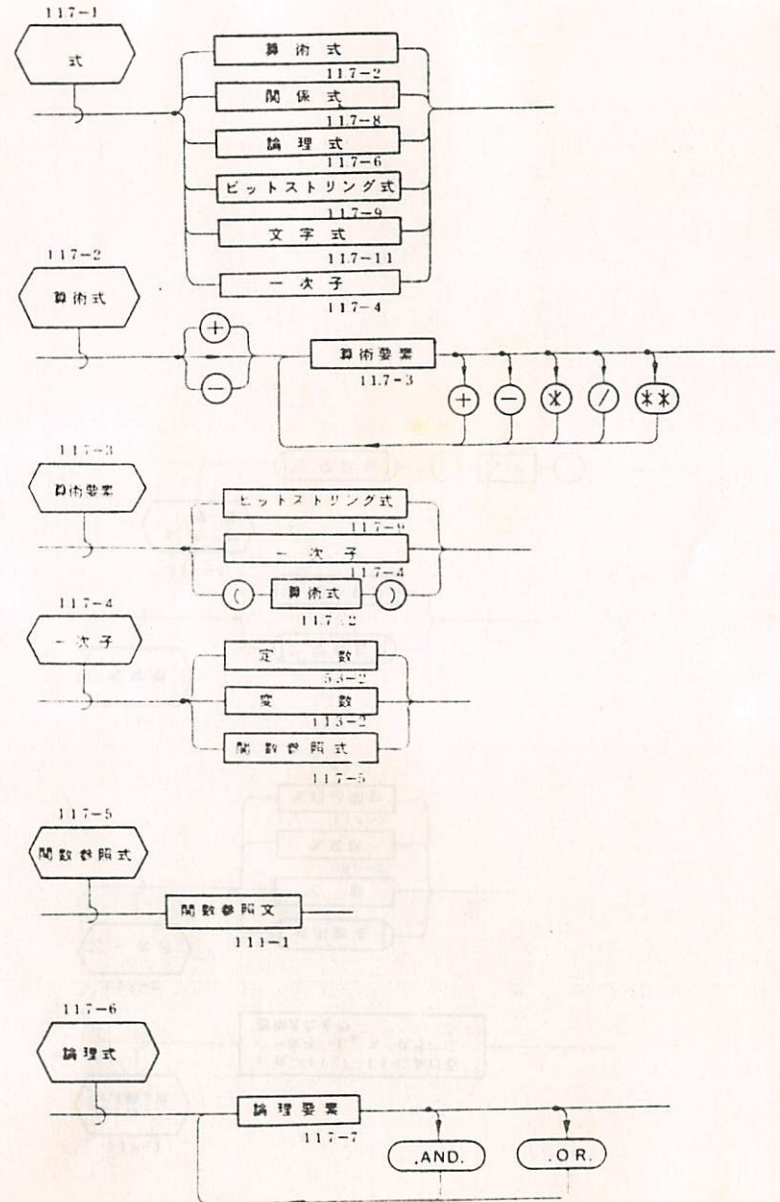
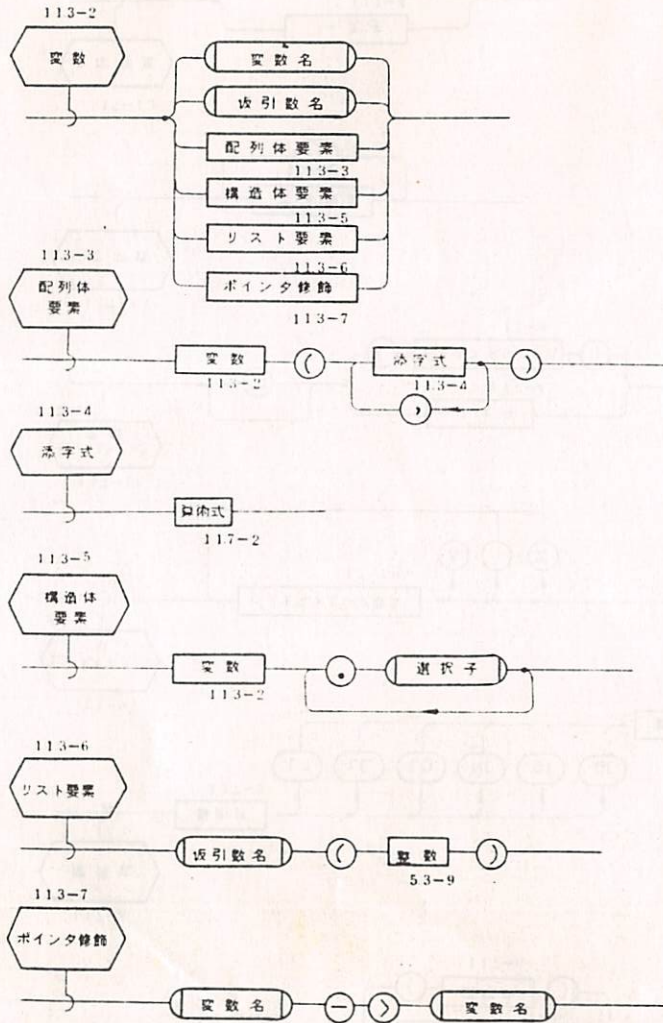


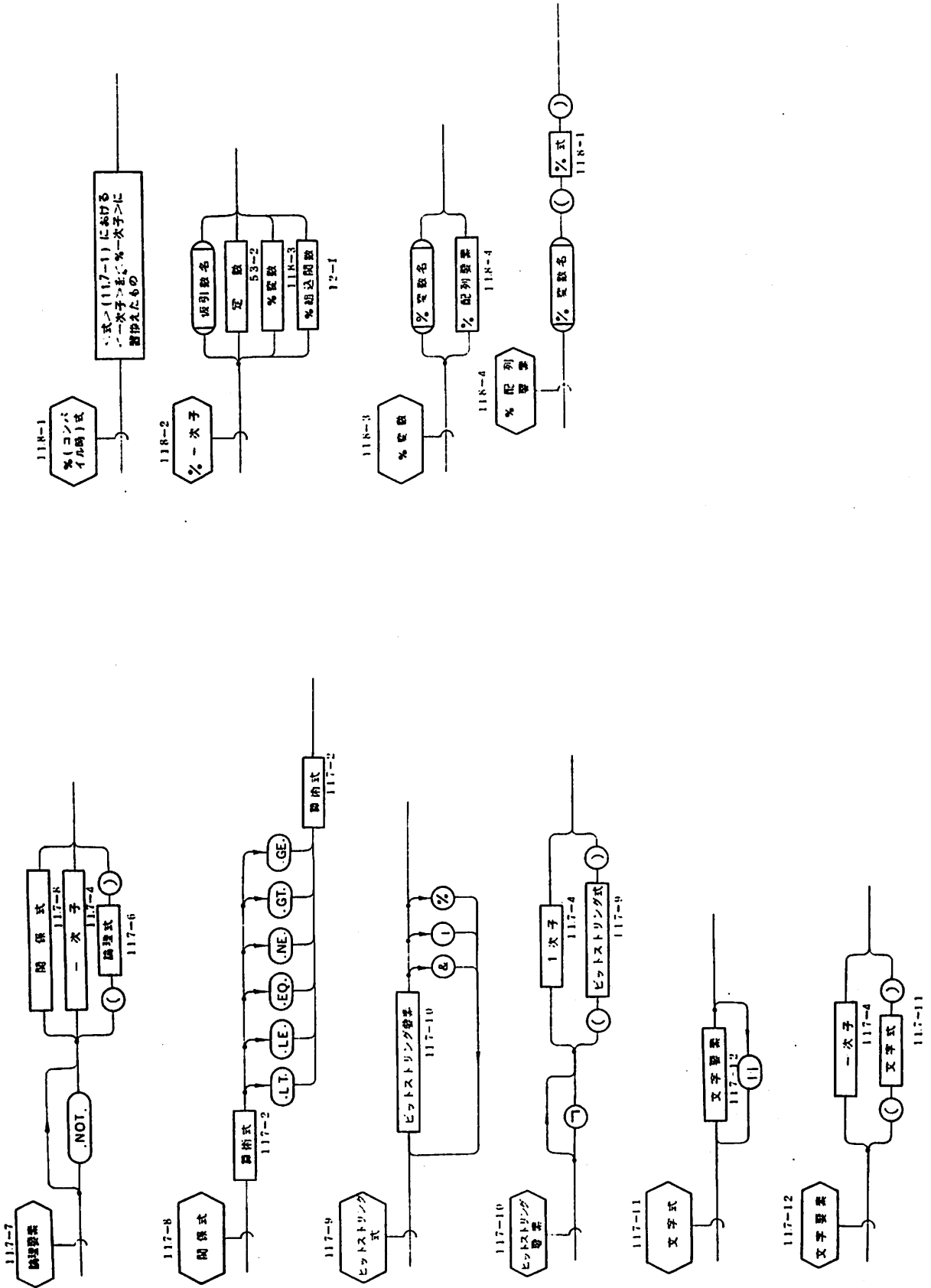


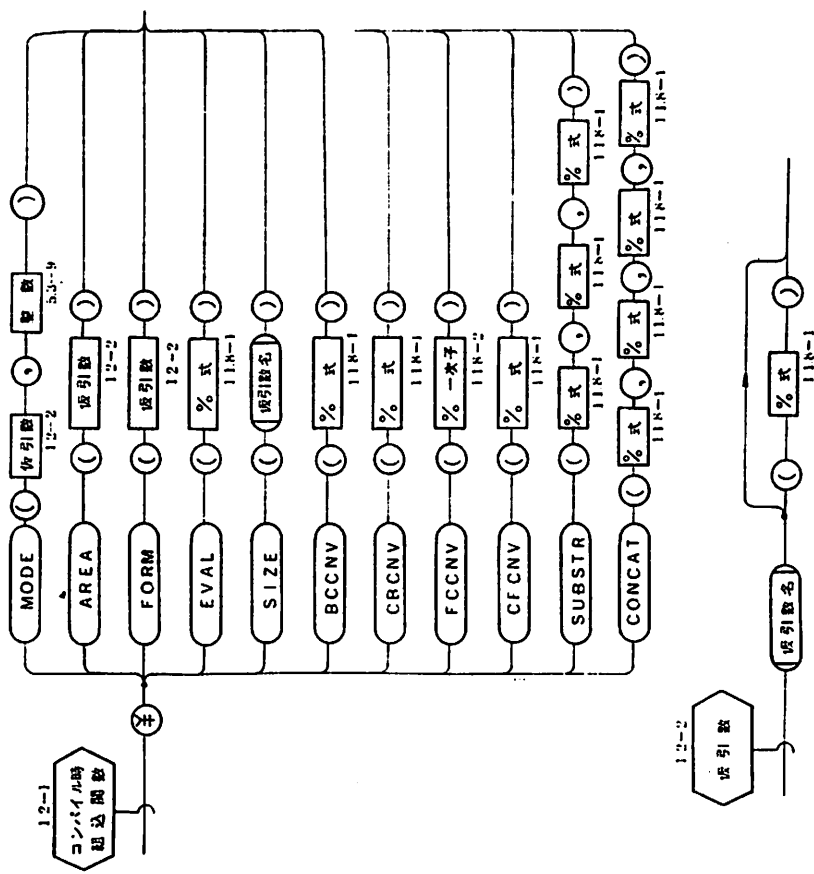


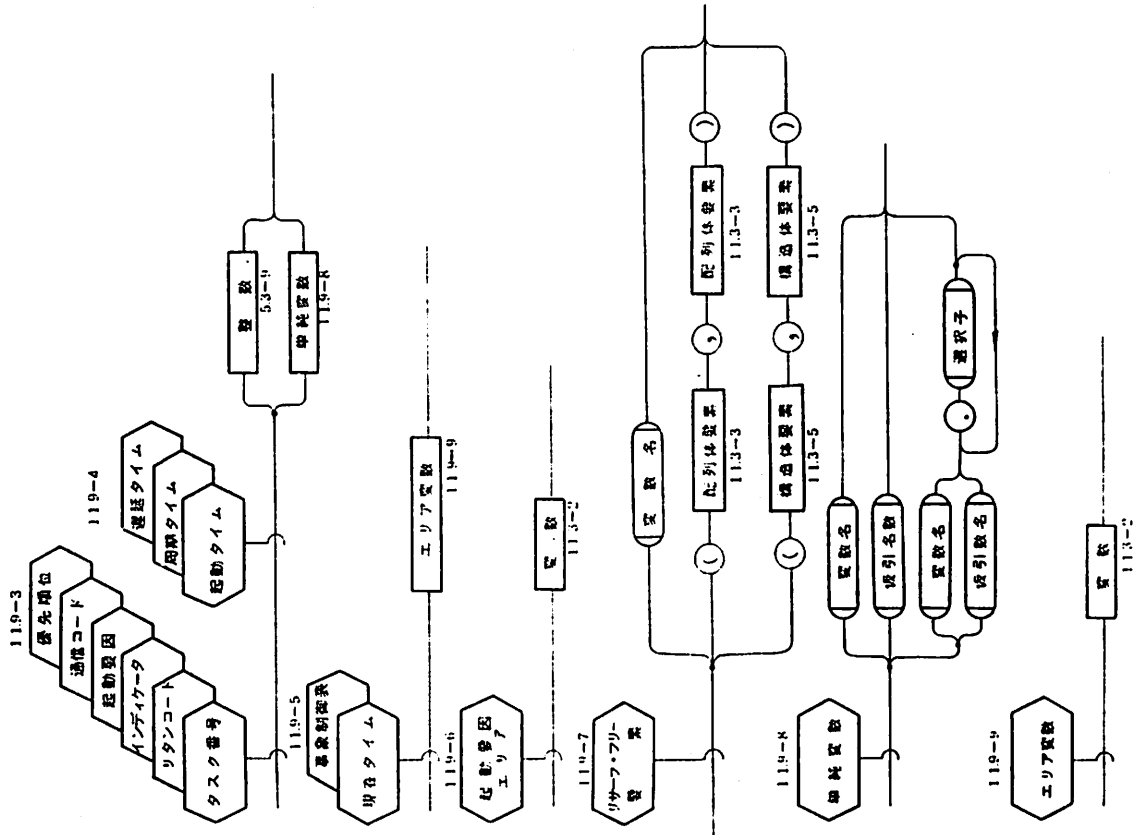
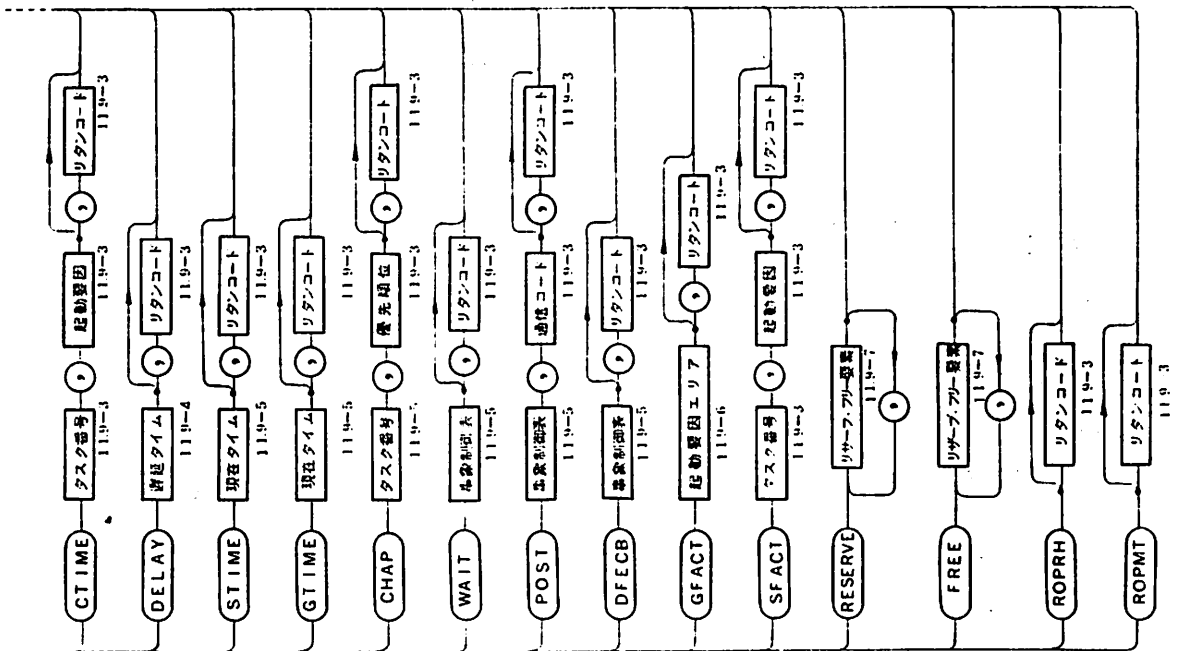


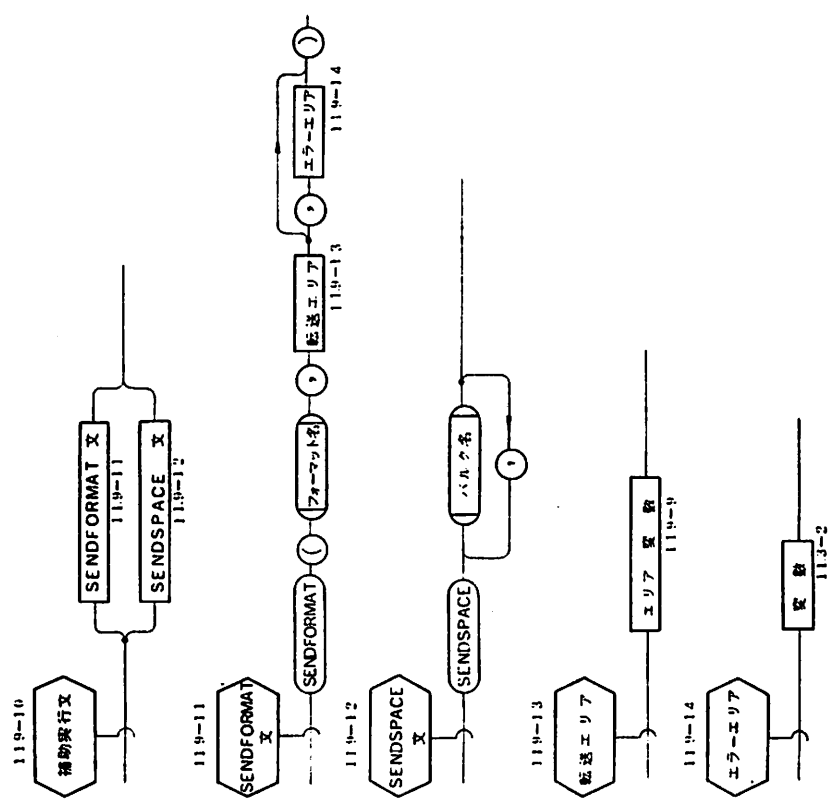
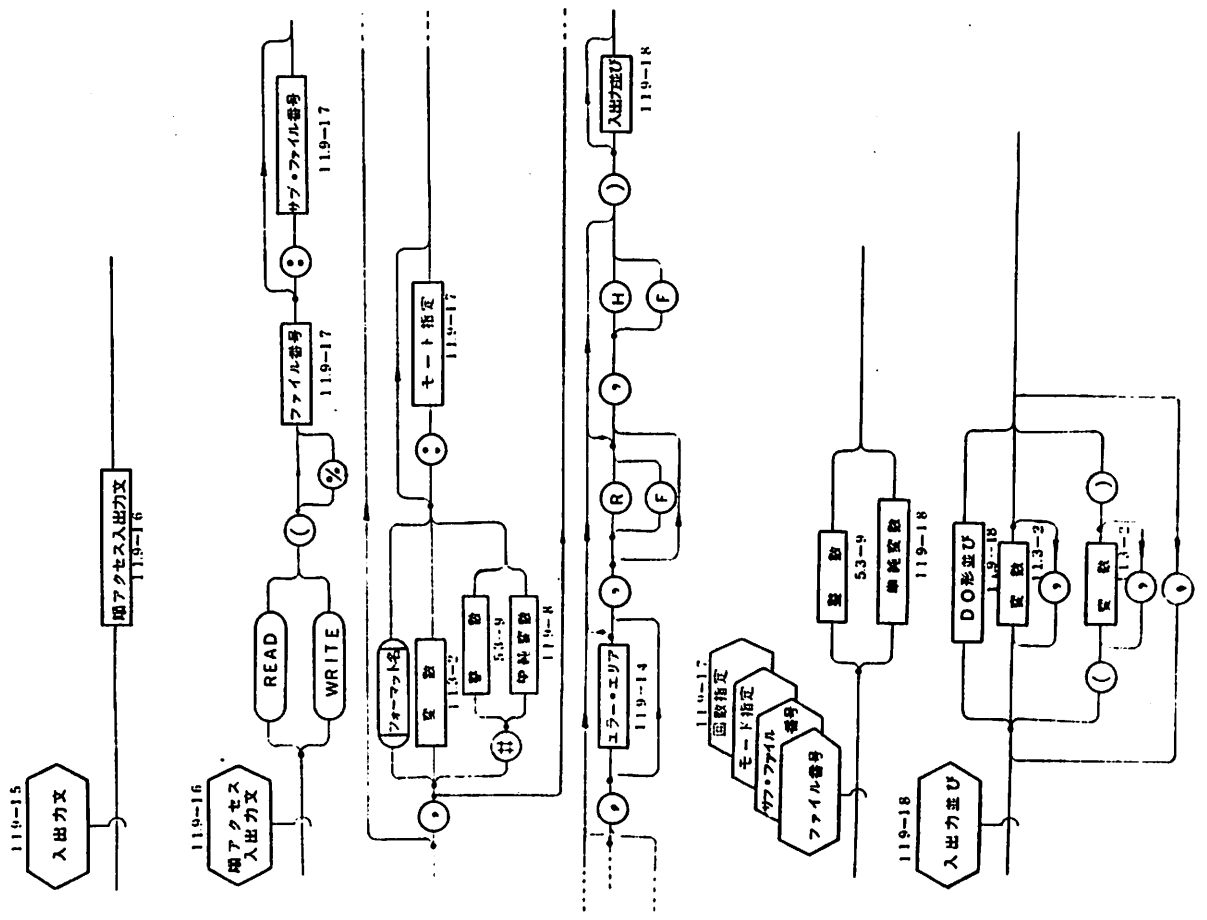


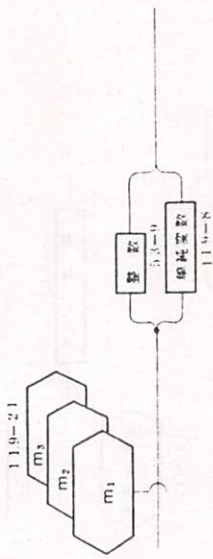
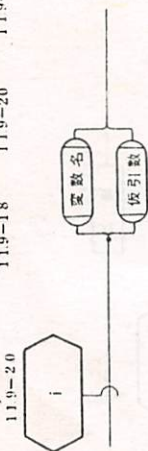
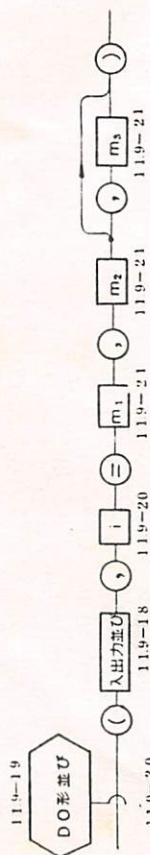
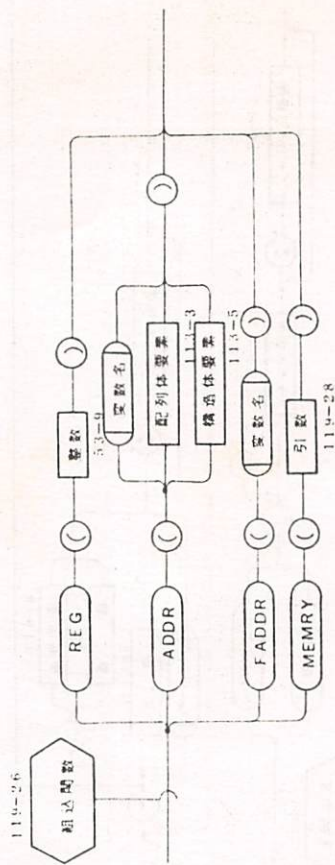
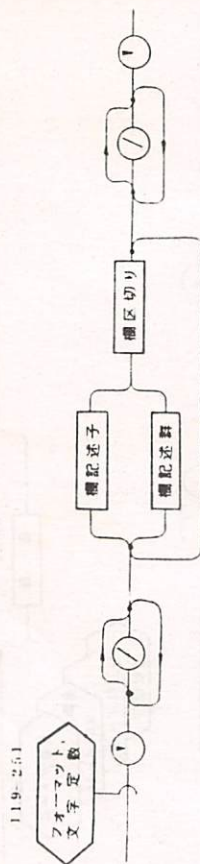
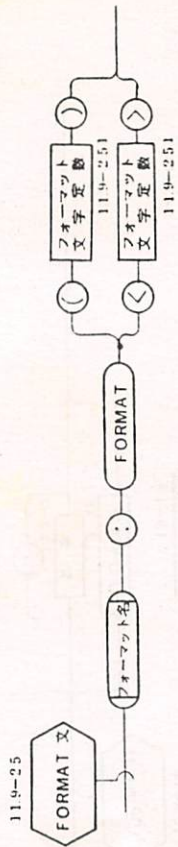








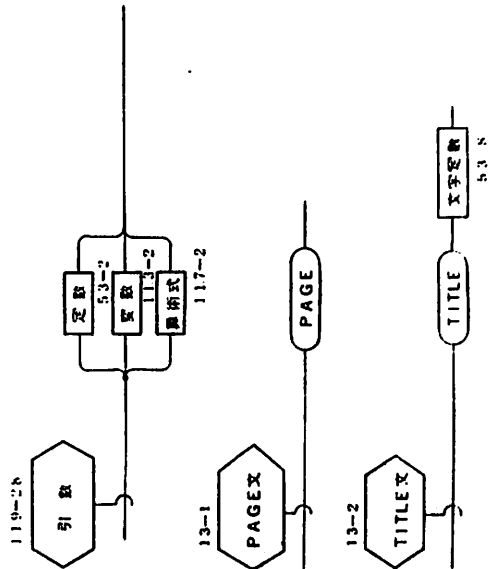
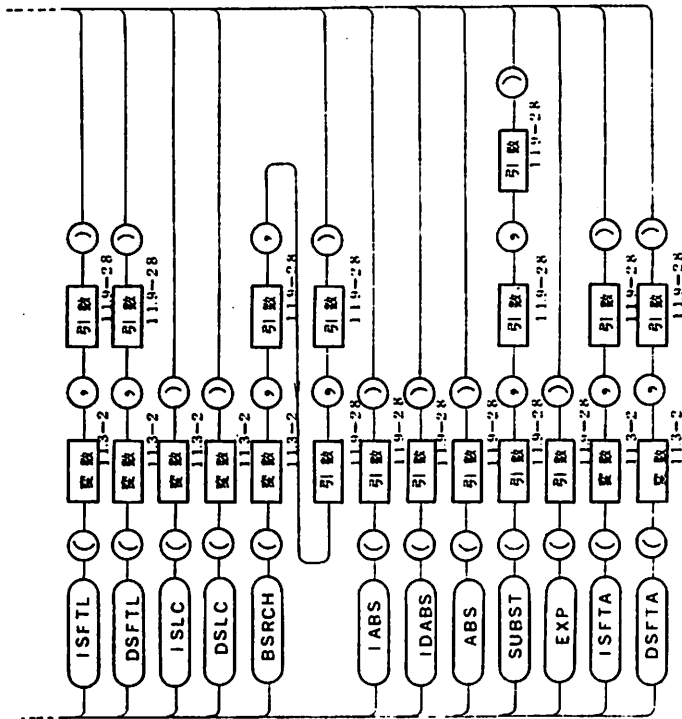
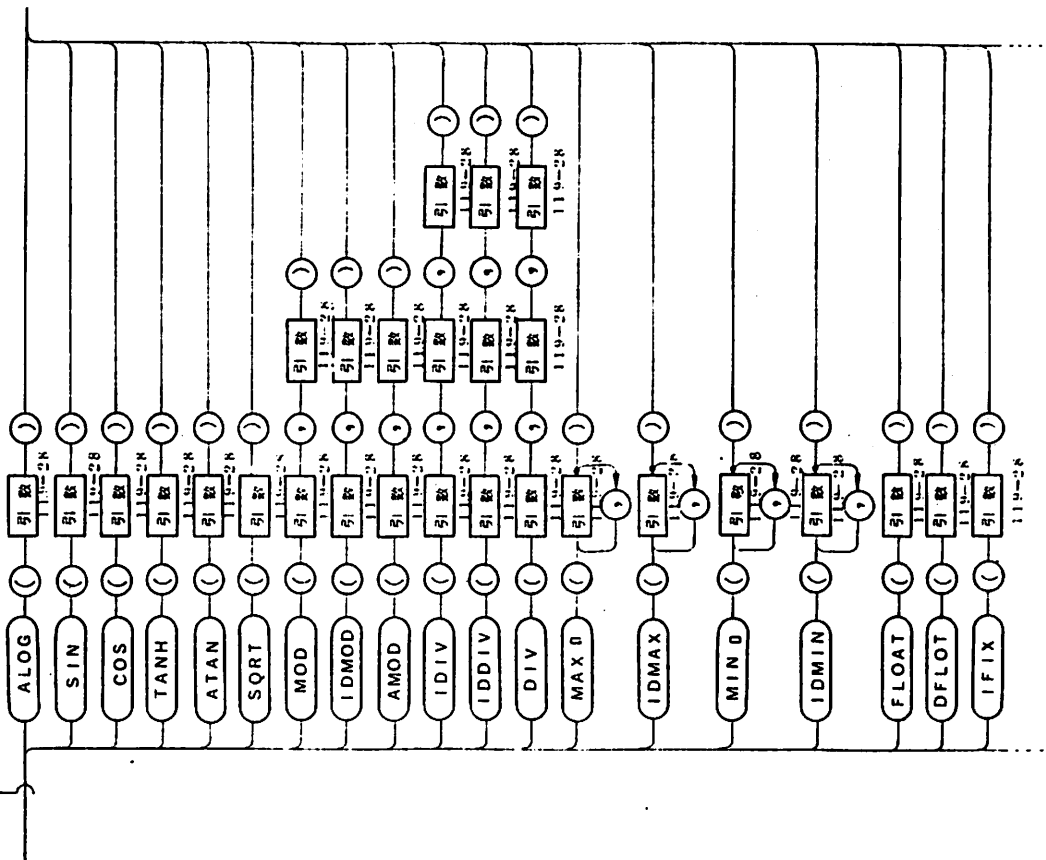






119-27

基本外部関数



付録 3. SPLプログラムの例

本文 4. 3 節で言及したページングアルゴリズム解析プログラムのソースリストを下に示す。なお、次の点に注意のこと。

- (1) 第1カラムが「C」の行は コメント行である。
- (2) データ型のデフォルトは整数型である。
- (3) 手続き属性 (OPT) のデフォルトはOPEN (インライン展開指定) である。

```

C----- 00020000
ENVIRONMENT E_A; 00030000
DCL; 00040000
  VAR PSIZE, /* PAGE SIZE (W) */ 00050000
    NPAGE; /* NUMBER OF PAGES = MAXADR/PSIZE */ 00060000
  VAR PSEQ:PAGE_SEQUENCE, /* SEQUENCE OF REFERENCED PAGE NUMBER */00070000
    NDATA; /* NUMBER OF DATA */ 00080000
  CONST MAXADR=16384, /* MAXIMUM ADDRESS (W)=128*1024 */ 00090000
    SSIZE=64, /* M/D SECTOR SIZE (W) */ 00100000
    MAXPG=256; /* MAXIMUM NUMBER OF PAGES = MAXADR/SSIZE*/00110000
  CONST ENDMK=-1; /* END MARK OF SEQUENCE */ 00120000
END; 00130000
END E_A; 00140000
00150000
C----- 00160000
ENVIRONMENT E_B (E_A); 00170000
SPEC; 00180000
  FUNCTION GETADR(ADRX:INT(2)) OPT(SUB); /* ASS */ 00190000
END; 00200000
DCL; 00210000
  CONST MAXNO=1000; /* MAXIMUM NUMBER OF DATA */ 00220000
  TYPE PAGE_SEQUENCE=ARRAY(MAXNO); /*TYPE OF REFERENCED PAGE NUMBER*/00230000
END; 00240000
END E_B; 00250000
00260000
C----- 00270000
ENVIRONMENT E_C (E_A); 00280000
DCL; 00290000
  VAR LTABLE(MAXPG); /* REF. FREQ. OF LRU_STACK */ 00300000
  VAR STABLE(MAXPG):(FREQ,PGNO); /* REF. FREQ. OF EACH PAGE */ 00310000
/* - IN ORDER OF FREQUENCY - */ 00320000
END; 00330000
END E_C; 00340000
00350000
C----- 00360000
ENVIRONMENT E_C1 (E_C); 00370000
DCL; 00380000
  VAR LSTACK(MAXPG), /* MAPPING TABLE & LRU STACK */ 00390000
    TOP, 00400000
    BOTTOM; 00410000
END; 00420000
END E_C1; 00430000

```

```

C-----
ENVIRONMENT E_C2(E_C);
  DCL;
  VAR TTABLE(MAXPG); /* REF. FREQ. OF EACH PAGE */
                          /* - IN ORDER OF PAGE NUMBER */
  END;
END E_C2;
00450000
00460000
00470000
00480000
00490000
00500000
00510000
00520000
00530000
C-----
ENVIRONMENT E_C3(E_C);
  DCL;
  CONST DUMMY=0;
  END;
END;
00540000
00550000
00560000
00570000
00580000
00590000
00600000
C-----
ENVIRONMENT E_D (E_A);
  DCL;
  CONST BLOOP=4; /* LOOP BY VARYING BUFFER SIZE */
  VAR BSIZE(BLOOP) INITIAL(1024,2048,4096,8192,16384);
                          /* BUFFER SIZE(W) */
  VAR BPAGE; /* NUMBER OF PAGES IN BUFFER */
  CONST ALGNO=5; /* NUMBER OF PAGING ALGORITHM */
  VAR FAULT(ALGNO); /* NUMBER OF FAULTS BY EACH ALGORITHM */
  CONST FINUFO=1, /* 1 = FINUFO (LRU/CYCLE) */
        FIFO =2, /* 2 = FIFO */
        FIVE =3, /* 3 = LRU/5BITS (FIVE) */
        OPX =4, /* 4 = OPTIMUM */
        LRU =5; /* 5 = LRU/STACK */
  END;
END E_D;
00610000
00620000
00630000
00640000
00650000
00660000
00670000
00680000
00690000
00700000
00710000
00720000
00730000
00740000
00750000
00760000
00770000
C-----
ENVIRONMENT E_D1(E_D);
  DCL;
  VAR T(MAXPG):STRUCT(PTR,REF), /* MAPPING TABLE OF FINUFO */
      TOP,
      BOTTOM,
      COUNT; /* NUMBER OF PAGES IN CORE */
  END;
END E_D1;
00780000
00790000
00800000
00810000
00820000
00830000
00840000
00850000
00860000
00870000
C-----
ENVIRONMENT E_D2(E_D);
  DCL;
  VAR T(MAXPG), /* MAPPING TABLE OF FIFO */
      TOP,
      BOTTOM,
      COUNT; /* NUMBER OF PAGES IN CORE */
  END;
END E_D2;
00880000
00890000
00900000
00910000
00920000
00930000
00940000
00950000
00960000
00970000
C-----
ENVIRONMENT E_D3 (E_D);
  DCL;
  VAR T(MAXPG):STRUCT(PTR,REF), /* MAPPING TABLE OF FIVE */
      TOP,
      BOTTOM,
      COUNT; /* NUMBER OF PAGES IN CORE */
  END;
END E_D3;
00980000
00990000
01000000
01010000
01020000
01030000
01040000
01050000
01060000
01070000
C-----
ENVIRONMENT E_D4(E_D);
  DCL;
  VAR T(MAXPG), /* MAPPING TABLE OF OPTIMUM */
      TOP,
      BOTTOM,
      COUNT; /* NUMBER OF PAGES IN CORE */
  END;
END E_D4;
01080000
01090000
01100000
01110000
01120000
01130000
01140000
01150000
01160000
01170000
C-----
ENVIRONMENT E_D5 (E_D);
  DCL;
  VAR LRATIO(ALGNO):REAL, /* RATIO ON LRU */
      ORATIO(ALGNO):REAL, /* RATIO ON OPTIMUM */
      FAULTR(ALGNO):REAL; /* FAULTRATE */
  END;
END E_D5;
01180000
01190000
01200000
01210000
01220000
01230000
01240000

```



```

C-----
FUNCTION PUSH (PGNO) TO LRU STACK;
  VAR COUNT; /* STACK DISTANCE */
  VAR PTR; /* SEARCH POINTER */
  PREPTR;
  IF LSTACK(PGNO) .EQ. 0
    THEN LTABLE(NPAGE)=LTABLE(NPAGE)+1;
    IF TOP .EQ. 0 /* FIRST REFERENCE */
      THEN TOP=PGNO;
      BOTTOM=PGNO;
      LSTACK(PGNO)=PGNO;
    ELSE LSTACK(PGNO)=TOP;
      LSTACK(BOTTOM)=PGNO;
      TOP=PGNO;
    END;
  ELSE COUNT=1; /* BEING IN STACK */
    PTR=TOP;
    PREPTR=BOTTOM;
    UNTIL PTR .EQ. PGNO
      REPEAT PREPTR=PTR; /* STACK DISTANCE */
      PTR=LSTACK(PTR);
      COUNT=COUNT+1;
    END;
    LTABLE(COUNT)=LTABLE(COUNT)+1;
    IF PTR .EQ. TOP
      THEN ;
      ELSE IF PTR .EQ. BOTTOM /* UPDATE LRU_STACK */
        THEN TOP=PTR;
        BOTTOM=PREPTR;
        ELSE LSTACK(PREPTR)=LSTACK(PTR);
          LSTACK(PTR)=TOP;
          TOP=PTR;
          LSTACK(BOTTOM)=TOP;
        END;
      END;
    END;
END PUSH;
C-----
FUNCTION GET FAULT_LRU (FLRU) OF BUFFER SIZE(BPAGE);
  VAR SUM; /* SUMMATION */
  VAR I; /* LOOP VAR */
  SUM=NDATA;
  FOR I=1,BPAGE
    REPEAT SUM=SUM-LTABLE(I);
  END;
  FLRU=SUM-MINO(BPAGE,LTABLE(NPAGE)); /* REDUCE FIRST LOADING */
END GET;
END C1;
C-----
PROCESS C2 (E_C2);
C-----
FUNCTION EXAMINE STATIC REFERENCE FREQUENCY;
  VAR NO; /* LOOP VAR */
  VAR PGNO; /* CURRENT PAGE NUMBER */
  INITIALIZE EXAMINE_STATIC_DISTRIBUTION;
  FOR NO=1,NDATA
    REPEAT GET (NO)TH PAGE_NUMBER(PGNO);
    COUNT (PGNO) TO TTABLE;
  END;
  SORT TTABLE TO STABLE;
END EXAMINE;
C-----
FUNCTION INITIALIZE EXAMINE_STATIC_DISTRIBUTION;
  VAR NO; /* LOOP VAR */
  FOR NO=1,NPAGE
    REPEAT TTABLE(NO)=0;
  END;
END INITIALIZE;
C-----
FUNCTION COUNT (PGNO) TO TTABLE;
  TTABLE(PGNO)=TTABLE(PGNO)+1;
END COUNT;
C-----
FUNCTION SORT TTABLE TO STABLE;
  VAR MAXPNO; /* PAGE NUMBER OF MAX FREQ. */
  VAR INDEX1,INDEX2; /* LOOP VAR */
  FOR INDEX1=1,NPAGE
    REPEAT MAXPNO=1;
    FOR INDEX2=2,NPAGE
      REPEAT IF TTABLE(INDEX2) .GT. TTABLE(MAXPNO)
        THEN MAXPNO=INDEX2;
      END;
    END;
  END;
  STABLE(INDEX1).FREQ=TTABLE(MAXPNO);
  STABLE(INDEX1).PGNO=MAXPNO;
  TTABLE(MAXPNO)=-1;
END;
END SORT;
END C2;
02140000
02150000
02160000
02170000
02180000
02190000
02200000
02210000
02220000
02230000
02240000
02250000
02260000
02270000
02280000
02290000
02300000
02310000
02320000
02330000
02340000
02350000
02360000
02370000
02380000
02390000
02400000
02410000
02420000
02430000
02440000
02450000
02460000
02470000
02480000
02490000
02500000
02510000
02520000
02530000
02540000
02550000
02560000
02570000
02580000
02590000
02600000
02610000
02620000
02630000
02640000
02650000
02660000
02670000
02680000
02690000
02700000
02710000
02720000
02730000
02740000
02750000
02760000
02770000
02780000
02790000
02800000
02810000
02820000
02830000
02840000
02850000
02860000
02870000
02880000
02890000
02900000
02910000
02920000
02930000
02940000
02950000
02960000
02970000
02980000
02990000
03000000
03010000
03020000
03030000

```

```

-----03050000
PROCESS C3 (E_C3):                                03060000
C-----03070000
FUNCTION COMPARE LRU WITH STATIC:                 03080000
VAR LSUM,SSUM,DSUM;                               /* SUM=LRU.STATIC.DIFERENCE */ 03090000
VAR SSUMX;                                        03100000
VAR (LRATE,SRATE,DRATE):REAL;                   /* FAULT RATE = SUM/NDATA */ 03110000
VAR NO;                                           /* LOOP VAR */                 03120000
PRINT_TITLE(' NO          LTABLE LSUM  LRATE      - 03130000
              -          STABLE  SSUM   SRATE    S.PGNO- 03140000
              -          DSUM    DRATE');        03150000

LSUM=NDATA;                                       03160000
SSUMX=NDATA;                                       03170000
FOR NO=1,NPAGE                                    03180000
  REPEAT LSUM=LSUM-LTABLE(NO);                   03190000
    IF NO .NE. 1                                  03200000
      THEN SSUMX=SSUMX-STABLE(NO-1).FREQ;        03210000
    END;                                           03220000
    IF ( NO .LE. 64 ) .OR. ( MOD(NO,32) .EQ. 0 ) 03230000
      THEN SSUM=SSUMX-(LTABLE(1)*FLOAT(SSUMX)/NDATA)+0.5; 03240000
      DSUM=SSUM-LSUM;                             03250000
      LRATE=FLOAT(LSUM)/NDATA;                   03260000
      SRATE=FLOAT(SSUM)/NDATA;                 03270000
      DRATE=SRATE-LRATE;                       03280000
      WRITE(6,F100) NO,LTABLE(NO),LSUM,LRATE,   03290000
              STABLE(NO).FREQ,SSUM,SRATE,      03300000
              STABLE(NO).PGNO,DSUM,DRATE;     03310000
      F100:FORMAT('1H0,I4,2(I20,I8,F10.3),I10,I20,- 03320000
                  -F10.3');                    03330000
    END;                                           03340000
  END;                                           03350000
END COMPARE;                                       03360000
END C3;                                           03370000
-----03380000
-----03390000
PROCESS D (E_D):                                  03400000
C-----03410000
FUNCTION APPLY PAGING ALGORITHM TO PAGE_SEQUENCE; 03420000
VAR I;                                           /* LOOP VAR */                 03430000
PRINT_TITLE('APPLICATION OF PAGING ALGORITHMS'); 03440000
FOR I=1,BLOOP                                    03450000
  REPEAT BPAGE=BSIZE(I)/PSIZE;                 03460000
    APPLY FINUFO;                               /* PROC. D1 */                 03470000
    APPLY FIFO;                                 /* PROC. D2 */                 03480000
    APPLY FIVE;                                /* PROC. D3 */                 03490000
    APPLY OPTIMUM;                             /* PROC. D4 */                 03500000
    COMPARE RESULTS;                           /* PROC. D5 */                 03510000
  END;                                           03520000
END APPLY;                                       03530000
END D;                                           03540000
-----03550000
-----03560000
PROCESS D1 (E_D1):                                03570000
C-----03580000
FUNCTION APPLY FINUFO:                            03590000
VAR NO;                                           /* LOOP VAR */                 03600000
VAR PGNO;                                         /* CURRENT PAGE NUMBER */     03610000
INITIALIZE APPLY_FINUFO;                       03620000
FOR NO=1,NDATA                                    03630000
  REPEAT GET (NO)TH PAGE_NUMBER (PGNO);        03640000
    DEMAND (PGNO) WITH FINUFO;                 03650000
  END;                                           03660000
END APPLY;                                       03670000
-----03680000
-----03690000
FUNCTION INITIALIZE APPLY_FINUFO:                03700000
VAR NO;                                           /* LOOP VAR */                 03710000
FAULT(FINUFO)=0;                                03720000
FOR NO=1,NPAGE                                    03730000
  REPEAT T(NO).PTR=0;                           03740000
    T(NO).REF=0;                                03750000
  END;                                           03760000
COUNT=0;                                       03770000
END INITIALIZE;

```

```

C-----
FUNCTION DEMAND (PGNO) WITH FINUFO;
VAR P,
    PREP;
IF T(PGNO).PTR .NE. 0          /* PGNO IN CORE ? */
THEN T(PGNO).REF=1;          /* Y */
ELSE IF COUNT .LT. BPAGE     /* N */ /* UNUSED BUFFER ? */
THEN IF COUNT .EQ. 0        /* Y */ /* FIRST ? */
THEN TOP=PGNO;
    BOTTOM=PGNO;
    T(PGNO).PTR=PGNO;
ELSE T(PGNO).PTR=TOP;      /* N */
    T(BOTTOM).PTR=PGNO;
    BOTTOM=PGNO;
END;
COUNT=COUNT+1;
ELSE P=TOP;                  /* N */ /* SWAPPING */
PREP=BOTTOM;
UNTIL T(P).REF .EQ. 0
REPEAT T(P).REF=0;
    PREP=P;
    P=T(P).PTR;
END;
T(PGNO).PTR=T(P).PTR;
T(P).PTR=0;
T(PREP).PTR=PGNO;
TOP=T(PGNO).PTR;
BOTTOM=PGNO;
FAULT(FINUFO)=FAULT(FINUFO)+1;
END;
END DEMAND;
END D1;
-----
C
PROCESS D2 (E_D2);
C-----
FUNCTION APPLY FIFO;
VAR NO;                      /* LOOP VAR */
VAR PGNO;                    /* CURRENT PAGE NUMBER */
INITIALIZE APPLY_FIFO;
FOR NO=1, NDATA
REPEAT GET (NO)TH PAGE_NUMBER (PGNO);
    DEMAND (PGNO) WITH FIFO;
END;
END APPLY;
C-----
FUNCTION INITIALIZE APPLY_FIFO;
VAR NO;                      /* LOOP VAR */
FAULT(FIFO)=0;
FOR NO=1, NPAGE
REPEAT T(NO)=0;
END;
COUNT=0;
END INITIALIZE;
C-----
FUNCTION DEMAND (PGNO) WITH FIFO;
IF T(PGNO) .NE. 0          /* PGNO IN CORE ? */
THEN;                      /* Y */
ELSE IF COUNT .LT. BPAGE   /* N */ /* UNUSED BUFFER ? */
THEN IF COUNT .EQ. 0      /* Y */ /* FIRST ? */
THEN TOP=PGNO;
    BOTTOM=PGNO;
    T(PGNO)=PGNO;
ELSE T(PGNO)=TOP;        /* N */
    T(BOTTOM)=PGNO;
    BOTTOM=PGNO;
END;
COUNT=COUNT+1;
ELSE T(PGNO)=T(TOP);      /* N */ /* SWAPPING */
T(BOTTOM)=PGNO;
T(TOP)=0;
TOP=T(PGNO);
BOTTOM=PGNO;
FAULT(FIFO)=FAULT(FIFO)+1;
END;
END DEMAND;
END D2;

```

```

03780000
03790000
03800000
03810000
03820000
03830000
03840000
03850000
03860000
03870000
03880000
03890000
03900000
03910000
03920000
03930000
03940000
03950000
03960000
03970000
03980000
03990000
04000000
04010000
04020000
04030000
04040000
04050000
04060000
04070000
04080000
04090000
04100000
04110000
04120000
04130000
04140000
04150000
04160000
04170000
04180000
04190000
04200000
04210000
04220000
04230000
04240000
04250000
04260000
04270000
04280000
04290000
04300000
04310000
04320000
04330000
04340000
04350000
04360000
04370000
04380000
04390000
04400000
04410000
04420000
04430000
04440000
04450000
04460000
04470000
04480000
04490000
04500000
04510000
04520000
04530000
04540000
04550000
04560000

```

```

-----04580000
PROCESS D3(E_D3);                                04590000
C-----04600000
FUNCTION APPLY FIVE;                             04610000
  VAR NO;                                         /* LOOP VAR */    04620000
  VAR PGNO;                                       /* CURRENT PAGE NUMBER */ 04630000
  INITIALIZE APPLY_FIVE;                         04640000
  FOR NO=1,NDATA                                  04650000
    REPEAT GET (NO)TH PAGE_NUMBER (PGNO);        04660000
      DEMAND (PGNO) WITH FIVE;                   04670000
  END;                                             04680000
END APPLY;                                       04690000
C-----04700000
FUNCTION INITIALIZE APPLY_FIVE;                  04710000
  VAR NO;                                         /* LOOP VAR */    04720000
  FAULT(FIVE)=0;                                  04730000
  FOR NO=1,NPAGE                                  04740000
    REPEAT T(NO).PTR=0;                          04750000
      T(NO).REF=0;                               04760000
  END;                                             04770000
  COUNT=0;                                        04780000
END INITIALIZE;                                  04790000
C-----04800000
FUNCTION DEMAND (PGNO) WITH FIVE;               04810000
  VAR P,                                          04820000
  PREP;                                          04830000
  VAR M,                                         /* PAGE NUMBER OF MINIMUM */ 04840000
  PREM;                                         /* PREPOINTER OF M */      04850000
  VAR I;                                         /* LOOP VAR */            04860000
  IF T(PGNO).PTR .NE. 0                         /* PGNO IN CORE ? */      04870000
    THEN T(PGNO).REF = T(PGNO).REF ! "10000"B;  04880000
    ELSE IF COUNT .LT. BPAGE                     /* N */ /* UNUSED BUFFER ? */ 04890000
      THEN IF COUNT .EQ. 0                       /* Y */ /* FIRST ? */      04900000
        THEN TOP=PGNO;                           /* Y */                    04910000
          BOTTOM=PGNO;                             04920000
          T(PGNO).PTR=PGNO;                        04930000
        ELSE T(PGNO).PTR=TOP;                      /* N */                    04940000
          T(BOTTOM).PTR=PGNO;                     04950000
          BOTTOM=PGNO;                             04960000
        END;                                       04970000
      COUNT=COUNT+1;                             04980000
    ELSE P=TOP;                                   /* N */ /* SWAPPING */    04990000
      PREP=BOTTOM;                                05000000
      M=TOP;                                       05010000
      PREM=BOTTOM;                                05020000
      FOR I=1,BPAGE                                05030000
        REPEAT IF T(M).REF .GT. T(P).REF         05040000
          THEN M=P;                                05050000
          PREM=PREP;                               05060000
        END;                                       05070000
        T(P).REF=T(P).REF/2;                      05080000
        PREP=P;                                    05090000
        P=T(P).PTR;                                05100000
      END;                                       05110000
      T(PGNO).PTR=T(M).PTR;                       05120000
      T(M).PTR=0;                                  05130000
      T(M).REF=0;                                  05140000
      T(PREM).PTR=PGNO;                           05150000
      IF M .EQ. TOP                                05160000
        THEN TOP=PGNO;                             05170000
        ELSE IF M .EQ. BOTTOM                       05180000
          THEN BOTTOM=PGNO;                        05190000
        END;                                       05200000
      END;                                       05210000
      FAULT(FIVE)=FAULT(FIVE)+1;                 05220000
    END;                                       05230000
  T(PGNO).REF=32;                                 /* 32 = "10000"B */    05240000
END;                                             05250000
END DEMAND;                                       05260000
END D3;                                           05270000

```



```

C-----05290000
PROCESS D4 (E_D4);                                05300000
C-----05310000
FUNCTION APPLY OPTIMUM;                            05320000
  VAR NO;                                           /* LOOP VAR */    05330000
  VAR PGNO;                                         /* CURRENT PAGE NUMBER */ 05340000
  INITIALIZE APPLY_OPTIMUM;                        05350000
  FOR NO=1,NDATA                                    05360000
    REPEAT GET (NO)TH PAGE_NUMBER (PGNO);         05370000
      DEMAND (PGNO) WITH OPTIMUM AT (NO)TH;       05380000
  END;                                              05390000
END APPLY;                                         05400000
C-----05410000
FUNCTION INITIALIZE APPLY_OPTIMUM;                 05420000
  VAR NO;                                           /* LOOP VAR */    05430000
  FAULT(OPX)=0;                                     05440000
  FOR NO=1,NPAGE                                    05450000
    REPEAT T(NO)=0;                                 05460000
  END;                                              05470000
  COUNT=0;                                         05480000
END INITIALIZE;                                    05490000
C-----05500000
FUNCTION DEMAND (PGNO) WITH OPTIMUM AT (NO)TH;    05510000
  VAR P;                                           05520000
  VAR PREP;                                        05530000
  VAR MAX;                                         /* MAX INTERVAL TO NEXT REF. */ 05540000
  VAR M;                                           /* PAGE NUMBER OF MAXIMUM */    05550000
  VAR PREM;                                        /* PREPOINTER OF M */          05560000
  VAR I;                                           05570000
  VAR TIME;                                        05580000
  IF T(PGNO) .NE. 0                                /* PGNO IN CORE ? */          05590000
    THEN;                                          /* Y */                        05600000
    ELSE IF COUNT .LT. BPAGE                       /* N *//* UNUSED BUFFER ? */ 05610000
      THEN IF COUNT .EQ. 0                         /* Y *//* FIRST ? */         05620000
        THEN TOP=PGNO;                             /* Y */                        05630000
          BOTTOM=PGNO;                              05640000
          T(PGNO)=PGNO;                             05650000
        ELSE T(PGNO)=TOP;                           /* N */                        05660000
          T(BOTTOM)=PGNO;                           05670000
          BOTTOM=PGNO;                              05680000
        END;                                        05690000
      COUNT=COUNT+1;                              05700000
    ELSE P=TOP;                                     /* N *//* SWAPPING */       05710000
      PREP=BOTTOM;                                 05720000
      MAX=0;                                        05730000
      FOR I=1,BPAGE                                05740000
        REPEAT GET INTERVAL(TIME) OF (P) AT (NO)TH; 05750000
          IF MAX .LT. TIME                         05760000
            THEN MAX=TIME;                         05770000
              M=P;                                  05780000
              PREM=PREP;                           05790000
          END;                                      05800000
          PREP=P;                                   05810000
          P=T(P);                                   05820000
        END;                                        05830000
      T(PGNO)=T(M);                                05840000
      T(M)=0;                                       05850000
      T(PREM)=PGNO;                                 05860000
      IF M .EQ. TOP                                05870000
        THEN TOP=PGNO;                             05880000
        ELSE IF M .EQ. BOTTOM                       05890000
          THEN BOTTOM=PGNO;                         05900000
        END;                                        05910000
      END;                                         05920000
      FAULT(OPX)=FAULT(OPX)+1;                     05930000
    END;                                          05940000
  END;                                          05950000
END DEMAND;                                       05960000

```

```

C-----
FUNCTION GET INTERVAL(TIME) OF (P) AT (NO)TH;
VAR I; /* LOOP VAR */
VAR PGNO; /* PAGE NUMBER */
TIME=1;
FOR I=NO+1,NDATA
  REPEAT GET (I)TH PAGE_NUMBER (PGNO);
    IF P .EQ. PGNO
      THEN EXIT LOOP;
    END;
    TIME=TIME+1;
  END LOOP;
END GET;
END D4;
-----
C
PROCESS D5 (E_D5);
-----
C
FUNCTION COMPARE RESULTS;
CALCULATE RATIOS TO LRU_AND_OPTIMUM;
PRINT RESULTS;
END COMPARE;
-----
C
FUNCTION CALCULATE RATIOS TO LRU_AND_OPTIMUM;
VAR I; /* LOOP VAR */
GET FAULT_LRU (FAULT(LRU)) OF BUFFER SIZE(BPAGE);
FOR I=1,ALGNO
  REPEAT LRATIO(I)=FLOAT(FAULT(I))/FAULT(LRU);
    ORATIO(I)=FLOAT(FAULT(I))/FAULT(OPX);
    FAULTR(I)=FLOAT(FAULT(I))/NDATA;
  END;
END CALCULATE;
-----
C
FUNCTION PRINT RESULTS;
VAR I; /* LOOP VAR */
VAR ANAME(5) : CHARACTER(10) INITIAL('FINUFO','FIFO','FIVE',
  'OPTIMUM','LRU');

WRITE(6,F100) BPAGE;
F100:FORMAT('1H0,'*** BUFFER SIZE =',I3,' PAGES ***');
WRITE(6,F200);
F200:FORMAT('1H0,22X,'FAULT RATE',3X,'LRU RATIO',3X,'OPT -
  -RATIO');
FOR I=1,ALGNO
  REPEAT WRITE(6,F300) ANAME(I),FAULTR(I),LRATIO(I),ORATIO(I);
    F300:FORMAT('1H0,4X,5A2,6X,3F12.3');
  END;
END PRINT;
END D5;
-----
C
PROCESS X; /* COMMON FUNCTION GROUP OVER SYSTEM */
-----
C
FUNCTION PRINT_TITLE (TITLEX:CHARACTER(132));
WRITE(6,F100) TITLEX;
F100:FORMAT('1H1,60A2');
END PRINT_TITLE;
END X;

```

```

05970000
05980000
05990000
06000000
06010000
06020000
06030000
06040000
06050000
06060000
06070000
06080000
06090000
06100000
06110000
06120000
06130000
06140000
06150000
06160000
06170000
06180000
06190000
06200000
06210000
06220000
06230000
06240000
06250000
06260000
06270000
06280000
06290000
06300000
06310000
06320000
06330000
06340000
06350000
06360000
06370000
06380000
06390000
06400000
06410000
06420000
06430000
06440000
06450000
06460000
06470000
06480000
06490000
06500000
06510000
06520000
06530000

```