

RCSL: 43-GL10503
AUTHOR: JCO/JHA
EDITED: 80.08.25

01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44

; MUB09

; KEYWORD: MUS, BASIC I/O, PROCEDURES, LISTING.
; ABSTRACT: MUS BASIC I/O CONTROL PROCEDURES FOR COROUTINES.
; ASCII PAPER TAPE RCSL 43-GL10502
; REL. BINARY PAPER TAPE RCSL 43-GL10504

10002 MUB09

| | | |
|----|--------|-------------|
| | | .TITL MUB09 |
| 02 | 000001 | .TXTM 1 |
| 03 | | .XREL |
| 04 | 000012 | .RDX 10 |
| 05 | | |

```

01
02
03      ; ENTRY POINT REPEATSHARE
04      ; AC1 IRRELEVANT
05      ; AC2 ZONE
06      ; AC3 LINK
07 00000*055025 A528: STA      3   Z1,2      ; REPEATSHARE:
08 00001*021024      LDA      0   Z0,2      ;   AC0:= USERSTATUS;
09 00002*176520      SUBZL    3,3      ;   SAVE(LINK);
10 00003*055024      STA      3   ZREPEAT,2 ;   REPEAT.ZONE:=1;
11 00004*034040 A5224:LDA     3   CUR        ; REPEATSHARE2:
12 00005*051424      STA      2   SAVE,3    ;   SAVE.CUR:=ZONE;
13 00006*035005      LDA      3   ZKIND,2    ;
14 00007*024112      LDA      1   .1B9      ;   IF KIND.ZONE(9) THEN
15 00010*167405      AND      3,1 SNR      ;   BEGIN
16 00011*000403      JMP      A5226      ;     OP:=0;
17 00012*126400      SUB      1,1      ;     GOTO PREPAREMESSAGE;
18 00013*000423      JMP      A5220      ;   END;
19 00014*035020 A5226:LDA     3   ZTOP,2      ;
20 00015*025017      LDA      1   ZFIRST,2    ;
21 00016*166404      SUB      3,1 SZR        ;   IF TOP.Z<>FIRST.Z(=TOP TRANSF) THEN
22 00017*015011      DSZ      ZBLOCK,2    ;   DECR(BLOCK.ZONE);
23 00020*000401      JMP      .+1      ;
24 00021*035005      LDA      3   ZKIND,2    ;
25 00022*024116      LDA      1   .1B13      ;   OPERATION:=0;
26 00023*167405      AND      3,1 SNR      ;   IF -,KIND.ZONE(13)(=NOT POS) THEN
27 00024*000412      JMP      A5220      ;   GOTO PREPAREMESSAGE;
28 00025*024113      LDA      1   CPOSITION ;   OP:=POSITION;
29 00026*034113      LDA      3   .1B10      ;
30 00027*163405      AND      3,0 SNR      ;
31 00030*000406      JMP      A5220      ;
32 00031*035021      LDA      3   ZUSED,2    ;
33 00032*035400      LDA      3   SOPER,3    ;
34 00033*020121      LDA      0   .3        ;   IF (STATUS.SHARE AND PARITY = 1)
35 00034*163626      ANDZR   3,0 SEZ      ;   AND OPER.SHARE(14:15)=OUTPUT THEN
36 00035*024444      LDA      1   A5225      ;   OP:=POSITION+ERASURE;

```

```

01
02 00036*021006 A5220:LDA      0   ZMASK,2   ; PREPAREMESSAGE:
03 00037*045006          STA      1   ZMASK,2   ;   ACO:=SAVE(MASK,Z);
04 00040*025004          LDA      1   ZMODE,2   ;
05 00041*034147          LDA      3   .M256     ;
06 00042*137400          AND      1,3     ;
07 00043*025006          LDA      1   ZMASK,2   ;
08 00044*167000          ADD      3,1     ;
09 00045*045006          STA      1   ZMASK,2   ;
10 00046*024123          LDA      1   .6       ;   MESS0:=OPERATION + ZMODE(0:7);
11 00047*147000          ADD      2,1     ;   MESSAGEADDR:=ZONE+6(=ZMASK);
12 00050*006004          SENDMESSAGE ;   SENDMESSAGE(ZONE,MESS,BUF);
13 00051*145000          MOV      2,1     ;   AC1:=BUF;
14 00052*031424          LDA      2   SAVE,3     ;   AC2:=ZONE;
15 00053*041006          STA      0   ZMASK,2   ;   MASK.Z:=SAVE(MASK,Z);
16 00054*035021          LDA      3   ZUSED,2   ;
17 00055*045405          STA      1   SSTATE,3   ;   STATE.SHARE:=BUFADDRESS;
18 00056*004553          JSR      A5238     ;   WAITBUF(ZONE,STATUS,COUNT,SHARE);
19 00057*045023          STA      1   ZREM,2     ;   REM:=COUNT(=0 IN CONTROLMESS.);
20 00060*045405          STA      1   SSTATE,3   ;   STATE.SHARE:=0;
21 00061*024464          LDA      1   A5290     ;
22 00062*107404          AND      0,1   SZR      ;   IF (STATUS AND -,IGNORE)<>0 THEN
23 00063*000423          JMP      A5201     ;   GOTO NEUTRAL;
24 00064*055020 A5221:STA      3   ZTOP,2     ; SET STOP:
25 00065*006413 A5222:JSR    .A531     ; REP MESS:  SENDSHARE(ZONE,SHARE);
26 00066*035404 A5223:LDA      3   SNEXT,3   ; NEXTSHARE:
27 00067*055021          STA      3   ZUSED,2   ;   USED.ZONE:=SHARE;
28 00070*021020          LDA      0   ZTOP,2     ;   RESTORE(STOP);
29 00071*025405          LDA      1   SSTATE,3   ;   BUF:=STATE.SHARE;
30 00072*162415          SUB#    3,0   SNR      ;   IF SHARE=STOP THEN
31 00073*000412          JMP      A520     ;   GOTO WAIT AGAIN;
32 00074*125005          MOV      1,1   SNR      ;   IF BUF=0 THEN
33 00075*000771          JMP      A5223     ;   GOTO NEXTSHARE;
34 00076*004533          JSR      A5238     ;   WAITBUF(BUF,ZONE,MESS0,MESS1,SHARE);
35 00077*000766          JMP      A5222     ;   GOTO REP MESS;
36 00100*000321*.A531:A531 ;
37 00101*000240 A5225:1B8+1B10 ;   ERASURE+POSITION;
38
39
40

```

```

01
02 ; PROCEDURE WAITTRANSFER(ZONE);
03 ;
04 ; WAITS FOR A MESSAGE BY CALLING FREESHARE(ZONE,STATUS,SHARE).
05 ; ZTOP:=SFIRST, ZFIRST(=TOP TRANSFERRED):=SFIRST+COUNT,
06 ; IF MODE=OUTPUT THEN ZREM:=ZSHAREL-IF Z.KIND(10) THEN 2 ELSE 0.
07 ; IF (STATUS AND REP) THEN REPEATSHARE2;
08 ; CALL: RETURN:
09 ; AC0 DESTROYED
10 ; AC1 DESTROYED
11 ; AC2 ZONE ZONE
12 ; AC3 LINK DESTROYED
13
14 000024 ZREPE=Z0 ; REF.REPEAT.ZONE:=REF.Z0.ZONE;
15 00102*055025 A52: STA 3 Z1,2 ; WAITTRANSFER:
16 00103*020122 A521: LDA 0 .5 ;
17 00104*041024 STA 0 ZREPEAT,2 ; REPCOUNT:=5;
18 00105*006210 A520: FREESHARE ; WAIT AGAIN: FREESHARE(Z,STATUS,SHARE);
19 00106*025406 A5201:LDA 1 SFIRST,3 ; NEUTRAL, CONTROL ERROR:
20 00107*045020 STA 1 ZTOP,2 ; TOP.ZONE:=FIRST.SHARE;
21 00110*035023 LDA 3 ZREM,2 ;
22 00111*167000 ADD 3,1 ;
23 00112*045017 STA 1 ZFIRST,2 ; FIRST.ZONE:=TOP.ZONE+COUNT(=ZREM);
24 00113*025004 LDA 1 ZMODE,2 ;
25 00114*125222 MOVZR 1,1 SZC ;
26 00115*125223 MOVZR 1,1 SNC ; IF (MODE.Z AND 1B14 AND 1B15)<>0 THEN
27 00116*000403 JMP A5206 ; !OUTPUT I TRANSPUTMESSAGE!
28 00117*035022 LDA 3 ZSHAREL,2 ; ZREM:=SHAREL.ZONE
29 00120*055023 STA 3 ZREM,2 ;
30 00121*024115 A5206:LDA 1 .1B12 ; REPEAT=1B12;
31 00122*035005 LDA 3 ZKIND,2 ; IF KIND.ZONE(12)=0 THEN
32 00123*167405 AND 3,1 SNR ; REP:=1B15
33 00124*176521 SUBZL 3,3 SKP ; ELSE
34 00125*034421 LDA 3 A5291 ; REP:=REPBITS;
35 00126*117405 AND 0,3 SNR ; IF (STATUS AND REP)<>0 THEN
36 00127*000403 JMP A5207 ; BEGIN
37 00130*015024 DSZ ZREPEAT,2 ; IF DECR(REPCOUNT)>0 THEN
38 00131*000653 JMP A5224 ; GOTO REPEATSHARE2; ! AC0 = STATUS
39 ; END;
40
41

```

```

01
02 00132*025006 A5207:LDA      1   ZMASK,2   ;
03 00133*107400      AND      0,1           ;   USERSTATUS:=STATUS AND MASK.ZONE;
04 00134*122400      SUB      1,0           ;   STATUS:=STATUS-USERSTATUS;
05 00135*034410      LDA      3   A5290      ;   IF (STATUS AND -,IGNORE)<>0 THEN
06 00136*163404      AND      3,0 SZR        ;   GOTO BREAK5;
07 00137*000411      JMP      A524          ;
08
09
10      A527:                               ; FINIS:
11 00140*125005      MOV      1,1 SNR        ;   IF USERSTATUS=0 THEN
12 00141*003025      JMP@     Z1,2          ;   RETURN;
13 00142*035025      LDA      3   Z1,2          ;   RESTORE(LINK);
14 00143*045024      STA      1   Z0,2          ;   Z0.ZONE:=USERSTATUS;
15 00144*003007      JMP@     ZGIVEUP,2 ;   GOTO GIVEUP.ZONE;
16      000002 .RDX 2
17 00145*161777 A5290:1110001111111111 ;   -,IGNORE BITS 1B3+1B4+1B5;
18 00146*020141 A5291:0010000001100001 ;   REPEAT BITS 1B2+1B9+1B10+1B15;
19 00147*001005 A5292:0000001000000101 ;   REJECTED BITS 1B6+1B13+1B15;
20      000012 .RDX 10
21 00150*105000 A524: MOV      0,1           ; BREAK5:
22 00151*035006      LDA      3   ZMASK,2   ;
23 00152*175222      MOVZR   3,3 SZC        ;   IF ZMASK.ZONE(1B15)=1 THEN
24 00153*000765      JMP      A527          ;   GOTO FINIS;
25 00154*020122      LDA      0   .5           ;
26 00155*030040      LDA      2   CUR          ;
27 00156*006012      BREAKPROCESS ;   BREAKPROCESS(5,STATUS,CUR);
28
29

```

```

01 ;PROCEDURE FREESHARE(ZONE,STATUS,SHARE);
02 ;
03 ; WAITS FOR THE MESSAGEBUFFER ADDRESSED BY SADDR.SHARE.
04 ; AND RETURN WITH STATUS OF THE OPERATION.
05 ; REM.Z:=MESS1.BUF(=COUNT).
06 ; IF (STATUS<>REJECTED AND BUF>0) THEN
07 ; BEGIN
08 ;   ZFILE:=MESS2.BUF(=FILE).
09 ;   IF KIND=POS THEN ZBLOCK:=MESS3.BUF ELSE INCR(ZBLOCK);
10 ;   IF KIND=-,BLOCKED THEN SCOUNT:=SCOUNT-COUNT AND SADDR:=SADDR+CO
11 ; END;
12 ; STATE.SHARE(=BUFADDRESS):=0;
13 ;
14 ;AC0          CALL:          RETURN:
15 ;AC1          STATUS
16 ;AC2          ZONE          DESTROYED
17 ;AC3          LINK         ZONE
18 ;                          SHARE=USED.ZONE
19 00157*055014 A523: STA      3   ZSIZE,2 ; FREESHARE:
20 00160*004451 JSR      A5238 ; WAITBUF(ZONE,MESS0.BUF,MESS1.BUF);
21 00161*045023 STA      1   ZREM,2 ; REM.ZONE:=COUNT(=MESS1.BUF);
22 00162*024765 LDA      1   A5292 ; AC1:=REJECTED BITS;
23 00163*035405 LDA      3   SSTATE,3 ; AC3:=BUF;
24 00164*175004 MOV      3,3 SZR ; IF (STATUS AND REJECT)=0 AND BUF>0
25 00165*107404 AND      0,1 SZR ; BEGIN
26 00166*000437 JMP      A5233 ; !BUF<0 THEN STATUS=1B13 BY WAITBU
27 00167*025410 LDA      1   MESS2,3 ;
28 00170*045010 STA      1   ZFILE,2 ; FILE.ZONE:=MESS2.BUF(=FILE);
29 00171*025005 LDA      1   ZKIND,2 ;
30 00172*020112 LDA      0   .1B9 ;
31 00173*123404 AND      1,0 SZR ; IF KIND.ZONE(9)=0 THEN
32 00174*000413 JMP      A5235 ; BEGIN
33 00175*020116 LDA      0   .1B13 ; POSITION=1B13;
34 00176*123405 AND      1,0 SNR ; IF KIND.ZONE(POS)=1 THEN
35 00177*000404 JMP      A5234 ; BEGIN
36 00200*025411 LDA      1   MESS3,3 ; BLOCK.Z:=MESS3.BUF(=BLOCK);
37 00201*045011 STA      1   ZBLOCK,2 ; END
38 00202*000405 JMP      A5235 ; ELSE
39 00203*025023 A5234:LDA 1   ZREM,2 ; IF COUNT(=ZREM)<>0 THEN
40 00204*125004 MOV      1,1 SZR ; BEGIN
41 00205*011011 ISZ      ZBLOCK,2 ; INCR(BLOCK.Z);
42 00206*000401 JMP      .+1 ; END;
43 ; END;

```

```

01
02 00207*035021 A5235:LDA      3  ZUSED,2  ;      AC3:=SHARE;
03 00210*025005      LDA      1  ZKIND,2  ;
04 00211*125220      MOVZR   1,1  ;      BLOCKED=1814;
05 00212*125222      MOVZR   1,1  SZC    ;      IF Z.KIND(BLOCKED)=0 THEN
06 00213*000410      JMP      A5232  ;      BEGIN
07 00214*025023      LDA      1  ZREM,2  ;      !UPDATE SHARE, NOT BLOCKED!
08 00215*021401      LDA      0  SCOUNT,3 ;
09 00216*122400      SUB      1,0  ;      COUNT.SHARE:=
10 00217*041401      STA      0  SCOUNT,3 ;      COUNT.SHARE-COUNT;
11 00220*021402      LDA      0  SADDR,3  ;      ADDRESS.SHARE:=
12 00221*123000      ADD      1,0  ;      ADDRESS.SHARE+COUNT;
13 00222*041402      STA      0  SADDR,3  ;      END;
14 00223*035405 A5232:LDA      3  SSTATE,3 ;      AC3:=BUF;
15 00224*021406      LDA      0  MESS0,3 ;      AC0:=STATUS(=MESS0.BUF);
16 00225*126400 A5233:SUB      1,1  ;      END;
17 00226*035021      LDA      3  ZUSED,2  ;
18 00227*045405      STA      1  SSTATE,3 ;      STATE.SHARE:=0(=FREE);
19 00230*003014      JMP#    ZSIZE,2  ; RETURN;
20

```



```

01 ; PROCEDURE WAITBUF(ZONE,STATUS,COUNT,SHARE);
02 ; CALL: RETURN;
03 ; AC0 STATUS=MESS0.BUF
04 ; AC1 COUNT=MESS1.BUF
05 ; AC2 ZONE ZONE
06 ; AC3 LINK SHARE=USED.ZONE
07
08 00231*055023 A5238:STA 3 ZREM,2 ; WAITBUF:
09 00232*021005 LDA 0 ZKIND,2 ; AC0:=KIND.ZONE;
10 00233*035021 LDA 3 ZUSED,2 ; AC3:=SHARE;
11 00234*145000 MOV 2,1 ; AC1:=ZONE;
12 00235*031405 LDA 2 SSTATE,3 ; AC2:=BUF;
13 00236*150513 NEGL# 2,2 SNC ; IF BUF>0 THEN
14 00237*000425 JMP A5239 ; BEGIN
15 00240*101133 MOVZL# 0,0 SNC ; IF KIND.ZONE(0)=1 THEN
16 00241*000406 JMP A5236 ; BEGIN
17 00242*034040 LDA 3 CUR ;
18 00243*021433 LDA 0 PC,3 ;
19 00244*135000 MOV 1,3 ;
20 00245*041417 STA 0 ZFIRST,3 ; ZFIRST.ZONE:=PC.CUR;
21 00246*006337 CWANSWER ; CWANSWER(ZONE,BUF);
22 00247*034040 A5236:LDA 3 CUR ; END;
23 00250*045424 STA 1 SAVE,3 ; SAVE.CUR:=ZONE;
24 00251*006005 WAITANSWER ; WAITANSWER(STATUS,COUNT,BUF);
25 00252*035424 LDA 3 SAVE,3 ;
26 00253*035421 LDA 3 ZUSED,3 ;
27 00254*035400 LDA 3 SOPER,3 ;
28 00255*020117 LDA 0 .1B14 ;
29 00256*116405 SUB 0,3 SNR ; IF SHARE.OPERA = 2 THEN
30 00257*101121 MOVZL 0,0 SKP ; STATUS:= 1B13; ! OTTO BIT !
31 00260*021006 LDA 0 MESS0,2 ;
32 00261*030040 LDA 2 CUR ;
33 00262*031024 LDA 2 SAVE,2 ; AC2:=ZONE:=SAVE.CUR;
34 00263*000407 JMP A5237 ; END
35 00264*141004 A5239:MOV 2,0 SZR ; ELSE BEGIN
36 00265*020116 LDA 0 .1B13 ; MESS0:=IF BUF=0 THEN 0 ELSE 1B13;
37 00266*131000 MOV 1,2 ; AC2:=ZONE;
38 00267*126400 SUB 1,1 ; MESS1:=0;
39 00270*034040 LDA 3 CUR ;
40 00271*051424 STA 2 SAVE,3 ; SAVE.CUR:=ZONE;
41 00272*035021 A5237:LDA 3 ZUSED,2 ; END; AC3:=SHARE;
42 00273*003023 JMP# ZREM,2 ; RETURN;
43

```

```

01 ; PROCEDURE TRANSFER(ZONE,LENGTH,OPERATION);
02 ; STARTS THE TRANSFER DESCRIBED BY OP AND LENGTH TO USED SHARE.
03 ; UPDATES USED SHARE TO POINT AT NEXT SHARE IN THE ZONE.
04 ; CALL: RETURN:
05 ; AC0 OPERATION DESTROYED
06 ; AC1 LENGTH DESTROYED
07 ; AC2 ZONE ZONE
08 ; AC3 LINK DESTROYED
09
10 00274*021004 A530: LDA 0 ZMODE,2 ; START TRANSFER:
11 00275*055024 A53: STA 3 Z0,2 ; OP:=MODE.ZONE;
12 00276*035021 LDA 3 ZUSED,2 ; TRANSFER: SHARE:=USED.ZONE;
13 00277*045401 STA 1 SCOUNT,3 ; MESS1.BUF=COUNT.SHARE:=LENGTH;
14 00300*025406 LDA 1 SFIRST,3 ;
15 00301*045402 STA 1 SADDR,3 ; MESS2.BUF=ADDR.SHARE:=FIRST.SHARE;
16 00302*041400 STA 0 SOPER,3 ; MESS0.BUF=OPERA.SHARE:=OP;
17 00303*025011 LDA 1 ZBLOCK,2 ;
18 00304*045403 STA 1 SSPECIAL,3 ; MESS3.BUF=SPECIAL.SHARE:=BLOCK.ZONE;
19 00305*004414 JSR A531 ; SENDSHARE(ZONE,SHARE);
20 00306*035404 LDA 3 SNEXT,3 ;
21 00307*055021 STA 3 ZUSED,2 ; USED.ZONE:=NEXT.SHARE;
22 00310*003024 JMP# Z0,2 ; RETURN;
23
24 ; PROCEDURE STARTSHARE(ZONE,OPERATION,SHARE);
25 ; SUPPLY OPERATION WITH CHANNELNUMBER (BIT 0:7 OF MODE)
26 ; AND SEND MESSAGE BY SENDSHARE
27 ; CALL: RETURN:
28 ; AC0 OPERATION DESTROYED
29 ; AC1 DESTROYED
30 ; AC2 ZONE ZONE
31 ; AC3 LINK USED.ZONE
32 ;
33 00311*055023 A5310:STA 3 ZREM,2 ; STARTSHARE:
34 00312*035004 LDA 3 ZMODE,2 ;
35 00313*024147 LDA 1 .M256 ;
36 00314*137400 AND 1,3 ;
37 00315*163000 ADD 3,0 ; OPERATION:=OPERATION+MODE(0:7) SHIFT
38 00316*035021 LDA 3 ZUSED,2 ;
39 00317*041400 STA 0 SOPER,3 ; OPERATION,SHARE:= OPERATION;
40 00320*101001 MOV 0,0 SKP ; SENDSHARE(ZONE,SHARE); RETURN;
41 ; PROCEDURE SENDSHARE(ZONE,SHARE);
42 ; SENDS A MESSAGE BUILT UP IN USED SHARE OF THE ZONE. THE ADDRESS
43 ; OF THE MESSAGE BUFFER IS STORED IN STATE,SHARE.
44 ; CALL: RETURN:
45 ; AC0 DESTROYED
46 ; AC1 DESTROYED
47 ; AC2 ZONE ZONE
48 ; AC3 LINK USED.ZONE
49
50 00321*055023 A531: STA 3 ZREM,2 ; SENDSHARE:
51 00322*025021 LDA 1 ZUSED,2 ; ADDRESS(=SHARE):=ZONE.USED;
52 00323*141000 MOV 2,0 ; SAVE(ZONE);
53 00324*006004 A532: SENDMESSAGE ; SENDMESSAGE(SHARE,ZONE,BUF);
54 00325*135000 MOV 1,3 ;
55 00326*051405 STA 2 SSTATE,3 ; STATE.SHARE:=BUF;
56 00327*111000 MOV 0,2 ;
57 00330*003023 JMP# ZREM,2 ; RETURN;
58

```

```

01 ; PROCEDURE INBLOCK(ZONE);
02 ; CALL: RETURN:
03 ; AC0 DESTROYED
04 ; AC1 DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07
08 00331*055025 A54: STA 3 Z1,2 ; INBLOCK:
09 00332*035021 A541: LDA 3 ZUSED,2 ; SHARE:= USED.ZONE;
10 00333*021405 LDA 0 SSTATE,3 ;
11 00334*101004 MOV 0,0 SZR ; IF STATE.SHARE<>0 !BUSY! THEN
12 00335*002404 JMP# .A521 ; GOTO WAITTRANSFER;
13 00336*025022 LDA 1 ZSHAREL,2 ; LENGTH:=SHARELENGTH.ZONE;
14 00337*004735 JSR A530 ; TRANSFER(ZONE,LENGTH);
15 00340*000772 JMP A541 ; GOTO INBLOCK;
16 00341*000103*.A521: A521 ; RETURN;
17
18 ; PROCEDURE OUTBLOCK(ZONE);
19 ; CALL: RETURN:
20 ; AC0 DESTROYED
21 ; AC1 DESTROYED
22 ; AC2 ZONE ZONE
23 ; AC3 LINK DESTROYED
24
25 00342*055025 A55: STA 3 Z1,2 ; OUTBLOCK:
26 00343*025022 LDA 1 ZSHAREL,2 ; LENGTH:=SHARELENGTH.ZONE;
27 00344*021023 LDA 0 ZREM,2 ; REM:=ZREM.ZONE;
28 00345*035005 LDA 3 ZKIND,2 ;
29 00346*175300 MOVS 3,3 ;
30 00347*177120 ADDZL 3,3 ;
31 00350*177133 ADDZL# 3,3 SNC ; IF KIND.ZONE(11)=1 THEN
32 00351*000404 JMP A551 ; BEGIN
33 00352*106404 SUB 0,1 SZR ; IF REM<>LENGTH THEN
34 00353*025022 LDA 1 ZSHAREL,2 ; LENGTH:=SHARELENGTH.ZONE;
35 00354*102400 SUB 0,0 ; REM:=0;
36 ; END;
37 00355*106404 A551: SUB 0,1 SZR ; LENGTH:=LENGTH-REM;
38 00356*004716 JSR A530 ; IF LENGTH<>0 THEN TRANSFER(Z,LENGTH);
39 00357*002762 JMP# .A521 ; WAITTRANSFER(ZONE);
40 ; RETURN;
41
42
43

```

```

01 ; PROCEDURE WAIT ALL(ZONE);
02 ; WAITS FOR ALL PENDING MESSAGES OF THE ZONE, USING WAIT
03 ; TRANSFER FOR OUTPUT MESSAGES AND WAIT ANSWER FOR THE
04 ; REMAINING MESSAGES.
05 ; CALL: RETURN:
06 ; AC0 UNCHANGED
07 ; AC1 UNCHANGED
08 ; AC2 ZONE ZONE
09 ; AC3 LINK USED,ZONE
10
11 00360*055026 A700: STA 3 Z2,2 ; WAIT ALL:
12 00361*041030 STA 0 Z4,2 ;
13 00362*045027 STA 1 Z3,2 ;
14 00363*021004 LDA 0 ZMODE,2 ;
15 00364*024121 LDA 1 .3 ;
16 00365*123626 ANDZR 1,0 SEZ ; IF MODE.ZONE(14:15)=3 THEN
17 00366*006206 OUTBLOCK ; OUTBLOCK(ZONE);
18 00367*035021 A7009:LDA 3 ZUSED,2 ; SET STOP:
19 00370*055025 STA 3 Z1,2 ; STOP:=USED,ZONE;
20 A7000: ; TEST OPERATION:
21 00371*035404 LDA 3 SNEXT,3 ; SHARE:=NEXT.SHARE;
22 00372*055021 STA 3 ZUSED,2 ; USED.ZONE:=SHARE;
23 00373*021405 LDA 0 SSTATE,3 ;
24 00374*101005 MOV 0,0 SNR ; IF STATE.SHARE=0 THEN
25 00375*000412 JMP A7004 ; GOTO NEXT;
26 00376*021004 LDA 0 ZMODE,2 ;
27 00377*024121 LDA 1 .3 ; OP:=MODE.ZONE;
28 00400*123627 ANDZR 1,0 SBN ; IF COMMAND.OP<>OUTPUT THEN
29 00401*000403 JMP A7002 ; GOTO ONLY WAIT;
30 00402*006202 WAITTRANSFER ; WAIT TRANSFER(ZONE);
31 00403*000764 JMP A7009 ; GOTO SET STOP;
32 A7002: ; ONLY WAIT:
33 00404*004625 JSR A5238 ; WAITBUF(ZONE,STATUS,COUNT,SHARE);
34 00405*102400 SUB 0,0 ;
35 00406*041405 STA 0 SSTATE,3 ; STATE.SHARE:=0;
36 00407*025025 A7004:LDA 1 Z1,2 ; NEXT:
37 00410*166414 INE 3,1 ; IF SHARE<>STOP THEN
38 00411*000760 JMP A7000 ; GOTO TEST OPERATION;
39 00412*006202 A7005:WAITTRANSFER ; WAITTRANSFER(ZONE);
40 00413*025027 LDA 1 Z3,2 ;
41 00414*021030 LDA 0 Z4,2 ;
42 00415*003026 JMP# Z2,2 ; RETURN;
43
44
45
46

```

10013 MUB09

```
01 ; PROCEDURE SETPOSITION(ZONE,FILE,BLOCK);
02 ; CALL: RETURN:
03 ; AC0 BLOCK DESTROYED
04 ; AC1 FILE DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07
08 00416*055031 A70: STA 3 Z5,2 ; SETPOSITION:
09 00417*004741 JSR A700 ; WAIT ALL(ZONE);
10 00420*041011 STA 0 ZBLOCK,2 ; BLOCK,ZONE:= BLOCK;
11 00421*045010 STA 1 ZFILE,2 ; FILE,ZONE:= FILE;
12 00422*035021 LDA 3 ZUSED,2 ; SHARE:= USED.ZONE;
13 00423*041403 STA 0 SSPEC,3 ; MESS3.BUF=SPEC.SHARE:=BLOCK;
14 00424*020113 LDA 0 CPOSITION ; AC0=OPERATION:=POSITION;
15 00425*045402 STA 1 SADDR,3 ; MESS2.BUF=ADDRESS.SHARE:=FILE;
16 00426*004663 JSR A5310 ; STARTSHARE(ZONE,OPERATION,SHARE);
17 00427*006202 WAITTRANSFER ; WAITTRANSFER(ZONE);
18 00430*003031 JMP# Z5,2 ; RETURN;
19
20
21
```

```

01 ; PROCEDURE CLOSE(ZONE,RELEASE);
02 ; CALL: RETURN:
03 ; AC0 DESTROYED
04 ; AC1 RELEASE, 0=NO, 1=YES DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07
08 00431*055031 A71: STA 3 Z5,2 ; CLOSE: Z5:=LINK;
09 00432*004726 JSR A700 ; WAIT ALL(ZONE);
10 00433*035004 LDA 3 ZMODE,2 ; MODE:=MODE.ZONE;
11 00434*102400 SUB 0,0 ; OP:=SENSE;
12 00435*175222 MOVZR 3,3 SZC ; IF MODE(14:15)=3 THEN
13 00436*175223 MOVZR 3,3 SNC ; BEGIN !OUTPUT I TRANSPUTMESS!
14 00437*000402 JMP A714 ; OP:=TERMINATION;
15 00440*020114 LDA 0 CTERMIN ; END;
16 A714: ;
17 00441*125004 MOV 1,1 SZR ; IF RELEASE=YES(=1) THEN
18 00442*024423 LDA 1 A713 ; OP:=OP+
19 00443*123000 ADD 1,0 ; CONTROL BITS;
20 00444*035021 LDA 3 ZUSED,2 ; ! AC0 = OPERATION !
21 00445*126400 SUB 1,1 ;
22 00446*045401 STA 1 SCOUNT,3 ; MESS1.BUF=COUNT.SHARE:=0;
23 00447*045402 STA 1 SADDR,3 ; MESS2.BUF=ADDRESS.SHARE:=0;
24 00450*004641 JSR A5310 ; STARTSHARE(ZONE,OPERATION,SHARE);
25 00451*126400 SUB 1,1 ;
26 00452*045004 STA 1 ZMODE,2 ; MODE.ZONE:=0;
27 00453*004705 JSR A700 ; WAIT ALL(ZONE);
28 00454*035005 LDA 3 ZKIND,2 ; IF ZONE.KIND = DISC THEN
29 00455*024114 LDA 1 .1B11 ; BEGIN
30 00456*137405 AND 1,3 SNR ; AREA-PROC(REMOVE-AREAPROCESS,
31 00457*003031 JMP 0 Z5,2 ; ZONE);
32 00460*024107 LDA 1 .1B6 ; CATPROC(REMOVE-CATALOG-PROCESS,
33 00461*004431 JSR A732 ; ZONE);
34 00462*024533 LDA 1 A739 ; END;
35 00463*035031 LDA 3 Z5,2 ; RETURN;
36 00464*000475 JMP A738 ; END;
37 ;
38 00465*000114 A713: 1B9+1B12+1B13 ; CONTROL BITS: DISCON+CONV+RESER;

```

```

01 ; PROCEDURE OPEN(ZONE,OPERATION);
02 ; CALL: RETURN:
03 ; AC0 OPERATION DESTROYED
04 ; AC1 DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07
08 00466*055026 A72: STA 3 Z2,2 ; OPEN:
09 00467*041004 STA 0 ZMODE,2 ; AC0=MODE.ZONE:=OPERATION;
10 00470*020127 LDA 0 .12 ; OPERATION:= CONV+RESER;
11 00471*035005 LDA 3 ZKIND,2 ; IF ZONE.ZKIND = DISC THEN
12 00472*024114 LDA 1 .1B11 ; BEGIN
13 00473*137405 AND 1,3 SNR ; CATPROC(CREATE-CATALOG-PROCESS,
14 00474*000405 JMP A720 ; ZONE);
15 00475*024570 LDA 1 A7390 ; AREA-PROC(CREATE-AREAPROCESS,
16 00476*004462 JSR A7380 ; ZONE:OPERATION,
17 00477*024106 LDA 1 .1B5 ; RESERVEMODE);
18 00500*004412 JSR A732 ; END;
19 A720: ; BUF.MESS1=SHARE.COUNT:=
20 00501*035021 LDA 3 ZUSED,2 ; RESERVEMODE;
21 00502*045401 STA 1 SCOUNT,3 ;
22 00503*025012 LDA 1 ZCONV,2 ; MESS2.BUF=ADDR.SHARE:=
23 00504*045402 STA 1 SADDR,3 ; CONV.ZONE;
24 00505*025011 LDA 1 ZBLOCK,2 ; MES3.BUF=SSPEC.SHARE:=
25 00506*045403 STA 1 SSPEC,3 ; ZBLOCK.ZONE;
26 00507*004602 JSR A5310 ; STARTSHARE(ZONE,OPERATION,SHARE);
27 00510*006202 WAITTRANSFER ; WAITTRANSFER(ZONE);
28 00511*003026 JMP# Z2,2 ; RETURN;

```

```

01 ; PROCEDURE AREA-PROC(ZONE,ACTION,RESERVEMODE);
02 ; THE PROCEDURE COMMUNICATES WITH THE CAT PROCESS TO CREATE
03 ; OR REMOVE AREAPROCESSES, WHEN THE BACKING STORAGE IS A
04 ; DISC; OPERATION=RESERVATION AT OUTPUT ELSE 0;
05 ; CALL: RETURN;
06 ; AC0 OPERATION
07 ; AC1 ACTION FOR CAT RESERVEMODE
08 ; AC2 ZONE ZONE
09 ; AC3 LINK DESTROYED
10
11 A732: ; AREAPROC
12 00512*055027 STA 3 Z3,2 ; ZONE.Z3:= LINK;
13 00513*035021 A733: LDA 3 ZUSED,2 ; MESS:= ZUSED.ZONE;
14 00514*045400 STA 1 SOPER,3 ; MESS0.MESS:= ACTION;
15 00515*051401 STA 2 SCOUNT,3 ; MESS1.MESS:= ZONE;
16 00516*020434 LDA 0 .A734 ;
17 00517*041023 STA 0 ZREM,2 ; ! ZREM=RETURN FOR SENDSHARE !
18 00520*141000 MOV 2,0 ; IF CONV.ZONE <> 0 THEN
19 00521*031012 LDA 2 ZCONV,2 ; REC=AC2:= CONV.ZONE
20 00522*151225 MOVZR 2,2 SNR ; ELSE REC=AC2:= ADDR("CAT");
21 00523*030431 A7340:LDA 2 A731 ; AC0:= SAVE(ZONE);
22 00524*165000 MOV 3,1 ;
23 00525*006426 JSR# .A532 ; SENDSHARE(ZONE,SHARE);
24 00526*006202 A734: WAITTRANSFER ; WAITTRANSFER(ZONE);
25 00527*035012 LDA 3 ZCONV,2 ; IF ZCONV.ZONE <> 0 THEN
26 00530*175225 MOVZR 3,3 SNR ; BEGIN
27 00531*000412 JMP A7341 ;
28 00532*021402 LDA 0 +2,3 ; ZNAME(5).ZONE:=
29 00533*024143 LDA 1 .255 ; ZZCONV(5).ZONE;
30 00534*123405 AND 1,0 SNR ;
31 00535*000406 JMP A7341 ;
32 00536*035002 LDA 3 ZNAME+2,2 ; END;
33 00537*167400 AND 3,1 ;
34 00540*136400 SUB 1,3 ;
35 00541*163000 ADD 3,0 ;
36 00542*041002 STA 0 ZNAME+2,2 ;
37 00543*035004 A7341:LDA 3 ZMODE,2 ;
38 00544*020116 LDA 0 CRESERV ; OPERATION:=RESERVATION;
39 00545*024117 LDA 1 .1B14 ;
40 00546*167414 AND# 3,1 SZR ; IF Z.MODE(14)= 1 THEN
41 00547*125400 INC 1,1 ; RESERVEMODE:=4
42 00550*125400 INC 1,1 ; ELSE RESERVEMODE:=3;
43 00551*003027 JMP # Z3,2 ; RETURN;
44 00552*000526*.A734:A734 ;
45 00553*000324*.A532:A532 ;
46 00554*000555*A731: .+1 ; NAME OF CAT-PROCESS;
47 00555*041501 .TXT .CAT<0><0>. ;
48 052000
49 000000

```



```

01 ; PROCEDURE CATPROC(ACTION,ZONE);
02 ; CALL: RETURN:
03 ; AC0 SPECIAL DESTROYED
04 ; AC1 ACTION DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07 ;
08 A7380: ; ENTRY POINT FROM OPEN:
09 00560*102400 SUB 0,0 ; SPECIAL:= 0;
10 00561*055027 A738: STA 3 Z3,2 ; CATPROC:
11 00562*045031 STA 1 Z5,2 ;
12 00563*035021 LDA 3 ZUSED,2 ; Z3,ZONE:= LINK;
13 00564*045400 STA 1 SOPER,3 ; MESS:= ZUSED.ZONE;
14 00565*041403 STA 0 SSPEC,3 ; MESS3.MESS:= SPECIAL;
15 00566*021012 LDA 0 ZCONV,2 ; MESS0.MESS:= ACTION;
16 00567*101225 MOVZR 0,0 SNR ; IF ZCONV.ZONE = 0 THEN
17 00570*003027 JMP# Z3,2 ; RETURN;
18 00571*041401 STA 0 SCOUNT,3 ; MESS1.MESS:= CONV.ZONE DIV 2;
19 00572*020422 LDA 0 .A738 ; ! ZREM=RETURN FOR SENDSHARE !
20 00573*041023 STA 0 ZREM,2 ; AC0:= SAVE(ZONE);
21 00574*141000 MOV 2,0 ; REC=AC2:= ADDR("CAT");
22 00575*000726 JMP A7340 ; SENDSHARE(ZONE,SHARE);
23 00576*006202 A7381:WAITTRANSFER ; WAITTRANSFER(ZONE);
24 00577*025031 LDA 1 Z5,2 ;
25 00600*020465 LDA 0 A7390 ;
26 00601*106404 SUB 0,1 SZR ; IF ACTION = FREECAT THEN
27 00602*003027 JMP# Z3,2 ; RETURN;
28 00603*035012 LDA 3 ZCONV,2 ;
29 00604*175220 MOVZR 3,3 ;
30 00605*025402 LDA 1 +2,3 ; ZCONV(5).ZONE:=
31 00606*020147 LDA 0 .M256 ; ZFILE.ZONE;
32 00607*107400 AND 0,1 ;
33 00610*021010 LDA 0 ZFILE,2 ;
34 00611*123000 ADD 1,0 ;
35 00612*041402 STA 0 +2,3 ;
36 00613*003027 JMP# Z3,2 ; END;
37 00614*000576*.A738:A7381
38 00615*001400 A739: 3#7 ; ACTION(REMOVE-CATALOG-PROCESS)

```

```

01 ; PROCEDURE CREATEENTRY(ZONE,TYPE,SIZE);
02 ; CALL: RETURN:
03 ; AC0 TYPE DESTROYED
04 ; AC1 SIZE DESTROYED
05 ; AC2 ZONE ZONE
06 ; AC3 LINK DESTROYED
07
08 00616*055027 A74: STA 3 Z3,2 ; Z3.ZONE:=LINK(CHECKAREAPROCESS);
09 00617*034102 LDA 3 .1B1 ; ACTION:=CREATE ENTRY;
10 00620*000417 JMP A80 ; GOTO COMMON;
11
12 ; PROCEDURE LOOKUPENTRY(ZONE,STORAGE ADDRESS);
13 ; CALL: RETURN:
14 ; AC0 DESTROYED
15 ; AC1 STORAGE ADDRES DESTROYED
16 ; AC2 ZONE ZONE
17 ; AC3 LINK DESTROYED
18
19 00621*055027 A75: STA 3 Z3,2 ; Z3.ZONE:=LINK;
20 00622*034103 LDA 3 .1B2 ; ACTION:=LOOKUP ENTRY;
21 00623*000414 JMP A80 ; GOTO COMMON;
22
23 ; PROCEDURE CHANGEENTRY(ZONE,STORAGE ADDRESS);
24 ; CALL: RETURN:
25 ; AC0 DESTROYED
26 ; AC1 STORAGE ADDRES DESTROYED
27 ; AC2 ZONE ZONE
28 ; AC3 LINK DESTROYED
29
30 00624*055027 A76: STA 3 Z3,2 ; Z3.ZONE:=LINK;
31 00625*034104 LDA 3 .1B3 ; ACTION:=CHANGE ENTRY;
32 00626*000411 JMP A80 ; GOTO COMMON;
33
34 ; PROCEDURE REMOVEENTRY(ZONE);
35 ; CALL: RETURN:
36 ; AC0 DESTROYED
37 ; AC1 DESTROYED
38 ; AC2 ZONE ZONE
39 ; AC3 LINK DESTROYED
40
41 00627*055027 A77: STA 3 Z3,2 ; Z3.ZONE:=LINK;
42 00630*034105 LDA 3 .1B4 ; ACTION:=REMOVE ENTRY;
43 00631*000406 JMP A80 ; GOTO COMMON;
44
45

```

```

01      ; PROCEDURE INITCAT(ZONE,NEW/OLD,DRIVENO);
02      ;          CALL:          RETURN:
03      ; AC0          NEW/OLD     DESTROYED
04      ; AC1          DRIVENO     DESTROYED
05      ; AC2          ZONE        ZONE
06      ; AC3          LINK        DESTROYED
07
08 00632*055027 A78:  STA      3   Z3,2      ; Z3.ZONE:=LINK;
09 00633*034110      LDA      3   .1B7      ; ACTION:=INIT CAT;
10 00634*000403      JMP          A80      ; GOTO COMMON;
11
12      ; PROCEDURE SETENTRY(ZONE,ADDRESS);
13      ;          CALL:          RETURN:
14      ; AC0          STORAGE ADDR DESTROYED
15      ; AC1          ZONE        DESTROYED
16      ; AC2          LINK        DESTROYED
17      ; AC3          LINK        DESTROYED
18
19 00635*055027 A79:  STA      3   Z3,2      ; Z3.ZONE:=LINK;
20 00636*034101      LDA      3   .1B0      ; ACTION:=UPDATE CAT;
21
22 00637*055030 A80:  STA      3   Z4,2      ; COMMON:
23 00640*035021      LDA      3   ZUSED,2      ;   SHARE:=ZUSED.ZONE;
24 00641*045402      STA      1   SADDRES,3      ;   MESS2.BUF:=AC1;
25 00642*041403      STA      0   SSPEC,3      ;   MESS3.BUF:=AC0;
26 00643*035012      LDA      3   ZCONV,2      ;   IF ZCONV.ZONE <> 0 THEN
27 00644*175225      MOVZR   3,3 SNR      ;   BEGIN
28 00645*000412      JMP          A800      ;
29 00646*021402      LDA      0   +2,3      ;   CHAR:= ZCONV.ZONE(6);
30 00647*024143      LDA      1   .255      ;
31 00650*123405      AND      1,0 SNR      ;   ZNAME(6).ZONE:= CHAR;
32 00651*000406      JMP          A800      ;
33 00652*035002      LDA      3   ZNAME+2,2      ;
34 00653*167400      AND      3,1      ;
35 00654*136400      SUB      1,3      ;   END;
36 00655*163000      ADD      3,0      ;
37 00656*041002      STA      0   ZNAME+2,2      ;
38 00657*025030 A800:  LDA      1   Z4,2      ;   AC1:=ACTION;
39 00660*000633      JMP          A733      ;   GOTO CHECKAREAPROCESS;

```

10020 MUB09

```
01 ;
02 ; PROCEDURE NEWCAT(ZONE);
03 ; CALL: RETURN:
04 ; AC0 KEY DESTROYED
05 ; AC1 DESTROYED
06 ; AC2 ZONE ZONE
07 ; AC3 LINK DESTROYED
08 ;
09 00661*024404 A81: LDA 1 A7390 ; ACTION:= CREATE CATALOG PROCESS;
10 00662*000677 JMP A738 ; CATPROC(ACTION,ZONE);
11 ;
12 ; PROCEDURE FREECAT(ZONE);
13 ; CALL: RETURN:
14 ; AC0 DESTROYED
15 ; AC1 DESTROYED
16 ; AC2 ZONE ZONE
17 ; AC3 LINK DESTROYED
18 ;
19 00663*024732 A82: LDA 1 A739 ; ACTION:= REMOVE CATALOG PROCESS;
20 00664*000675 JMP A738 ; CATPROC(ACTION,ZONE);
21 ;
22 ;
23 00665*002400 A7390:5B7 ; ACTION(CREATE-CATALOG-PROCESS);
```

10021 MUB09

```
01      000202 .LOC WAITTRANS-GOS
02 00202 000102*      A52      ; WAIT TRANSFER
03      000203 .LOC .REPEATSHARE-GOT
04 00203 000000*      A528     ;
05      000210 .LOC FREESHARE-GOS
06 00210 000157*      A523     ; FREESHARE
07      000204 .LOC TRANSFER-GOS
08 00204 000275*      A53      ; TRANSFER
09      000205 .LOC INBLOCK-GOS
10 00205 000331*      A54      ; INBLOCK
11      000206 .LOC OUTBLOCK-GOS
12 00206 000342*      A55      ; OUTBLOCK
13      000222 .LOC WAITZONE-GOS
14 00222 000360*      A700     ; WAITZONE
15      000217 .LOC SETPOS-GOS
16 00217 000416*      A70      ; SET POSITION
17      000220 .LOC CLOSE-GOS
18 00220 000431*      A71      ; CLOSE
19      000221 .LOC OPEN-GOS
20 00221 000466*      A72      ; OPEN
21      000346 .LOC CREATEENTRY-GOS
22 00346 000616*      A74      ; CREATE ENTRY
23      000347 .LOC LOOKUPENTRY-GOS
24 00347 000621*      A75      ; LOOKUP ENTRY
25      000350 .LOC CHANGEENTRY-GOS
26 00350 000624*      A76      ; CHANGE ENTRY
27      000351 .LOC REMOVEENTRY-GOS
28 00351 000627*      A77      ; REMOVE ENTRY
29      000352 .LOC INITCAT-GOS
30 00352 000632*      A78      ; INIT CAT
31      000353 .LOC SETENTRY-GOS
32 00353 000635*      A79      ; SETENTRY
33      000332 .LOC NEWCAT-GOS
34 00332 000661*      A81      ; CREATE CATALOG PROCESS
35      000333 .LOC FREECAT-GOS
36 00333 000663*      A82      ; REMOVE CATALOG PROCESS
37
38      ; ***** END BASIC I/O PROCEDURES *****
39
40      .END
0000 SOURCE LINES IN ERROR
```

| | | | | | | |
|-------|---------|-------|-------|-------|-------|-------|
| A52 | 000102* | 5/15 | 21/02 | | | |
| A520 | 000105* | 4/31 | 5/18 | | | |
| A5201 | 000106* | 4/23 | 5/19 | | | |
| A5206 | 000121* | 5/27 | 5/30 | | | |
| A5207 | 000132* | 5/36 | 6/02 | | | |
| A521 | 000103* | 5/16 | 11/16 | | | |
| A5220 | 000036* | 3/18 | 3/27 | 3/31 | 4/02 | |
| A5221 | 000064* | 4/24 | | | | |
| A5222 | 000065* | 4/25 | 4/35 | | | |
| A5223 | 000066* | 4/26 | 4/33 | | | |
| A5224 | 000004* | 3/11 | 5/38 | | | |
| A5225 | 000101* | 3/36 | 4/37 | | | |
| A5226 | 000014* | 3/16 | 3/19 | | | |
| A523 | 000157* | 7/19 | 21/06 | | | |
| A5232 | 000223* | 8/06 | 8/14 | | | |
| A5233 | 000225* | 7/26 | 8/16 | | | |
| A5234 | 000203* | 7/35 | 7/39 | | | |
| A5235 | 000207* | 7/32 | 7/38 | 8/02 | | |
| A5236 | 000247* | 9/16 | 9/22 | | | |
| A5237 | 000272* | 9/34 | 9/41 | | | |
| A5238 | 000231* | 4/18 | 4/34 | 7/20 | 9/08 | 12/33 |
| A5239 | 000264* | 9/14 | 9/35 | | | |
| A524 | 000150* | 6/07 | 6/21 | | | |
| A527 | 000140* | 6/10 | 6/24 | | | |
| A528 | 000000* | 3/07 | 21/04 | | | |
| A5290 | 000145* | 4/21 | 6/05 | 6/17 | | |
| A5291 | 000146* | 5/34 | 6/18 | | | |
| A5292 | 000147* | 6/19 | 7/22 | | | |
| A53 | 000275* | 10/11 | 21/08 | | | |
| A530 | 000274* | 10/10 | 11/14 | 11/38 | | |
| A531 | 000321* | 4/36 | 10/19 | 10/50 | | |
| A5310 | 000311* | 10/33 | 13/16 | 14/24 | 15/26 | |
| A532 | 000324* | 10/53 | 16/45 | | | |
| A54 | 000331* | 11/08 | 21/10 | | | |
| A541 | 000332* | 11/09 | 11/15 | | | |
| A55 | 000342* | 11/25 | 21/12 | | | |
| A551 | 000355* | 11/32 | 11/37 | | | |
| A70 | 000416* | 13/08 | 21/16 | | | |
| A700 | 000360* | 12/11 | 13/09 | 14/09 | 14/27 | 21/14 |
| A7000 | 000371* | 12/20 | 12/38 | | | |
| A7002 | 000404* | 12/29 | 12/32 | | | |
| A7004 | 000407* | 12/25 | 12/36 | | | |
| A7005 | 000412* | 12/39 | | | | |
| A7009 | 000367* | 12/18 | 12/31 | | | |
| A71 | 000431* | 14/08 | 21/18 | | | |
| A713 | 000465* | 14/18 | 14/38 | | | |
| A714 | 000441* | 14/14 | 14/16 | | | |
| A72 | 000466* | 15/08 | 21/20 | | | |
| A720 | 000501* | 15/14 | 15/19 | | | |
| A731 | 000554* | 16/21 | 16/46 | | | |
| A732 | 000512* | 14/33 | 15/18 | 16/11 | | |
| A733 | 000513* | 16/13 | 19/39 | | | |
| A734 | 000526* | 16/24 | 16/44 | | | |
| A7340 | 000523* | 16/21 | 17/22 | | | |
| A7341 | 000543* | 16/27 | 16/31 | 16/37 | | |
| A738 | 000561* | 14/36 | 17/10 | 20/10 | 20/20 | |
| A7380 | 000560* | 15/16 | 17/08 | | | |
| A7381 | 000576* | 17/23 | 17/37 | | | |
| A739 | 000615* | 14/34 | 17/38 | 20/19 | | |
| A7390 | 000665* | 15/15 | 17/25 | 20/09 | 20/23 | |

| | | | | | | | |
|-------|---------|-------|-------|-------|-------|-------|-------|
| A74 | 000616* | 18/08 | 21/22 | | | | |
| A75 | 000621* | 18/19 | 21/24 | | | | |
| A76 | 000624* | 18/30 | 21/26 | | | | |
| A77 | 000627* | 18/41 | 21/28 | | | | |
| A78 | 000632* | 19/08 | 21/30 | | | | |
| A79 | 000635* | 19/19 | 21/32 | | | | |
| A80 | 000637* | 18/10 | 18/21 | 18/32 | 18/43 | 19/10 | 19/22 |
| A800 | 000657* | 19/28 | 19/32 | 19/38 | | | |
| A81 | 000661* | 20/09 | 21/34 | | | | |
| A82 | 000663* | 20/19 | 21/36 | | | | |
| ZREPE | 000024 | 3/10 | 5/14 | 5/17 | 5/37 | | |
| .A521 | 000341* | 11/12 | 11/16 | 11/39 | | | |
| .A531 | 000100* | 4/25 | 4/36 | | | | |
| .A532 | 000553* | 16/23 | 16/45 | | | | |
| .A734 | 000552* | 16/16 | 16/44 | | | | |
| .A738 | 000614* | 17/19 | 17/37 | | | | |

00635*

19/19

21/32