```
0001  .MAIN
01
02                ;                                                  PAGE 1 OF 5
03                ;       RCSL:   44-RT 1557
04
05                ;       VERSION:        4.4.77  JEP
06
07                ;               PROGRAM
08                ;               *******
09                ;
10                ;       RC 3600 - INSTRUCTION TIMER TEST
11                ;
12                ;               PROGRAM SOURCE
13                ;
14                ;       S-BINARY TAPE:  RCSL 44-RT 1558 (SELFSTART AUTOLOAD HEA
15                ;       ASCII SOURCE:   RCSL 44-RT 1559 TAPE 1 - 7
16                ;       BINARY CARD:    RCSL 44-RT 1560
17                ;
18                ;0. INDEX
19                ;       1. ABSTRACT                         PAGE 1
20                ;       2. MACHINE REQUIREMENTS             PAGE 1
21                ;       3. SWITCH SETTINGS AND INIMESS      PAGE 2
22                ;       4. OPERATING PROCEDURE              PAGE 2
23                ;       5. PROGRAM DESCRIPTION              PAGE 4
24                ;       6. LIMITATIONS/MISC                 PAGE 4
25                ;       7. MESSAGES FROM THIS TEST          PAGE 5
26                ;
27                ;1. ABSTRACT
28                ;       RC 3600 INSTRUCTION TIMER TEST IS A MAINTENANCE
29                ;       PROGRAM DESIGNED TO TEST THE CPU CLOCK CIRCUITS
30                ;       BY TIMING THE INSTRUCTION SET. THE 100 MSEC RTC
31                ;       AND THE TTC CONTROLLER CLOCK ARE USED FOR CALI-
32                ;       BRATION AND ARE ASSUMED ACCURATE.
33                ;       THE PURPOSE WITH THIS PROGRAM IS TO DETECT MALFUNCTIONS
34                ;       IN TIMING, NOT ONLY IN THE CPU BUT ALSO IN RTC AND TTY-
35                ;       CONTROLLER. ERROR IN TIMING IS NOT ALLWAYS TESTED IN
36                ;       OTHER PROGRAMS BUT COULD DISTURB THE OPERATING OF OTHER
37                ;       TESTPROGRAMS AND COSTUMERPROGRAMS. AN ERROR COULD BE DU
38                ;       TO A BAD MEM, BAD CPU MIKROPROGRAM OR BAD CRYSTAL/GATES
39                ;       IN ONE OF THE 3 CLOCK CIRCUITS USED IN CPU, RTC AND TTC
40                ;       THIS ALSO CHECKS THE STATE OF FIELD CHANGE ORDERS,
41                ;       CARRIED OUT OR NOT.
42
43                ;2. MACHINE REQUIREMENTS
44                ;       RC 3600 FAMILY PROCESSOR                    CPU
45                ;       MINIMUM 8K READ/WRITE MEMORY                MEM
46                ;       REAL TIME CLOCK                             RTC
47                ;       TELETYPE OR OTHER OPERATOR CONSOLE          TTY     *NOTE
48                ;       LINEPRINTER, OPTIONALLY                     LPT     *NOTE
49                ;
50                ;       *NOTE: IF THIS DEVICES IS SET ONLINE AFTER PROGRAM
51                ;       START, THE PROGRAM MUST BE RESTARTED TO GET OUTPUT.
52
53                ;3. SWITCH SETTINGS AND INITIAL MESSAGES:
54                ;       3.1 STARTING ADDRESSES - METHOD 3.3 START !
55                ;             POWER UP RESTART IF KEY IN LOCK AND POWER MONITOR
56                ;               400 PRINT TIMES IF IN ERROR
57                ;               401 PRINT TIMES FOR ALL INSTR. TYPES
58                ;               402 (PRINT CPU IDENT CONSTANT, FOR LOCKA TABLE)
59                ;               403 RESTART OF TEST, NO QUESTIONS, LAST ANSWERS
60                ;              1400 RESTART PROGRAM AS WHEN LOADED
61                ;              2202 GET A NEW PRINTER ALPHABETH (SEE 4.1.1)
62                ;              2204 SET TO 64K WORDS MODE, MEM SIZE ?
63                ;              2206 SET TO 32K WORDS MODE, MEM SIZE ?
64                ;              2210 EXAMINE MEMORY
65                ;              2212 DEPOSIT MEMORY
66                ;              2222 START BINARY LOADER, READ FROM PTR/TTI (SW
```

```
01                ;                                                  PAGE 2 OF 5
02                ;
03                ;       3.2 CONTROL BY SWITCH SETTING, THE STATE REPORTED
04                ;                       AT TELETYPE AND LINEPRINTER
05                ;               NO SWITCHES USED, BUT SET SW0, SW10 - 15 TO ZERO.
06                ;               SW13: WAIT AFTER DIS MESSAGE.  DIS IS THE 16 CHA
07                ;               SW14: HALT AFTER DIS MESSAGE.  DISPLAY AT OPERAT.
08                ;               Sw15: CLEAR DIS AFTER MESSAGE. CONTROL PANEL. OC
09                ;
10                ;       3.3 START
11                ;               OF PROGRAM AFTER LOADING:
12                ;               SET SWITCHES TO CONTROL. (3.2).
13                ;               ANSWER START ADDRESS QUESTION.
14                ;
15                ;       3.3 A RESTART
16                ;               OF PROGRAM THE BEST WAY TO INSURE
17                ;               CORRECT SWITCH SETTINGS:
18                ;               RESET
19                ;               SET SWITCHES TO START ADDR. (3.1).
20                ;               EXAMINE
21                ;               SET SWITCHES TO CONTROL. (3.2).
22                ;               CONTINUE.
23                ;               THE START ADDR IS LISTED AT TTO, LPT AND DIS.
24                ;
25                ;       3.3 B RESTART
26                ;               OF PROGRAM IF RC 3603 CPU:
27                ;               SET DATASWITCH 0, 10 - 15 UP (1)
28                ;               SET RESET PARITY ERROR DOWN (ON)
29                ;               PRESS AUTOLOAD AND RELEASE
30                ;               SET RESET PARITY ERROR UP (OFF)
31                ;               SET SWITCHES TO CONTROL. (3.2).
32                ;               ANSWER THE QUESTION WRITTEN ON TTY/OCP.
33                ;
34                ;       3.4 LOADING MESSAGE:
35                ;               AFTER LOADING THE PROGRAM WILL WRITE FOLLOWING:
36                ;               SWITCHES: 100073 ;THIS IS THE INITIAL STATE OF S
37                ;               CPU TYPE: 000016 ;THIS IDENTIFIES WHICH CPU/MEM
38                ;               LAST LOC: 077777 ;THIS INFORMS THE USER: MEM-SIZ
39                ;               BINARY LOADER OK ;THIS IS THE TS VERSION PLACED
40                ;               AT LAST LOC (MAX 32K) BUT SLIGHTLY MODIFIED AND
41                ;               INCL BOOTSTRAP.                 ERRORHALT XX7752
42                ;               IF NOT SELFSTARTING PROG:       READYHALT XX7676
43                ;               ACTUAL PROG NAME ;THIS IS THE LOADED PROGRAM.
44                ;               SET SWITCHES TO CONTROL, (3.2), STARTADDR   400
45                ;
46                ;       3.5 START WITH BREAK OPTION (RC 3603).
47                ;               RESET
48                ;               SET SWITCHES TO HALT INSTR 063077
49                ;               SET REGISTER SELECT TO 6
50                ;               DEPOSIT INTO REGISTER
51                ;               SET SWITCHES TO ADDRESS    000003
52                ;               SET REGISTER SELECT TO 5
53                ;               DEPOSIT INTO REGISTER
54                ;               SET BREAK SWITCH TO ON
55                ;               GO TO 3.3 RESTART
56                ;
57                ;4. OPERATING PROCEDURE
58                ;       LOAD THE PROGRAM AND START IT IN ONE OF THE POSSIBLE
59                ;       START ADDRESSES.
60                ;       EVENTUALLY CHANGE THE PRINTER ALPHABETH. (SEE 4.1)
61                ;       ANSWER THE QUESTIONS GIVEN AND THE PROGRAM RUNS.
62                ;       (MORE ABOUT QUESTIONS IN 5.1 AND 7.)
63
64                ;       A PASS MESSAGE IS PRINTED, 45-75 SEC. INTERVAL. (SA 400)
65                ;       IN CASE OF ERRORS (SA 400) THE INSTRUCTION WITH BAD
66                ;       TIMING IS PRINTED. IF SA=401 THE INSTR TIMES ARE
67                ;       ALLWAYS PRINTED. THERE ARE 3 POSSIBILITIES FOR ERROR
```

```
01                ;                                                    PAGE 3 OF 5
02                ;       1.  BAD CPU/MEMORY
03                ;       2.  BAD RTC
04                ;       3.  BAD TTY
05                ;       IF ONE OF THE RESULTS IS EQUAL TO (OR WITHIN A SMALL
06                ;       TOLERENCE) THE EXPECTED TIME IT IS NOT THE CPU, BUT
07                ;       THE OTHER REFERENCE WITH BAD RESULT WHICH IS BAD.
08                ;       IF BOTH RESULTS ARE NOT CORRECT THEN THE CPU IS BAD
09                ;       (OR BOTH REFERENCES, RTC AND TTY).
10                ;       THE TOLERANCE IS SET TO 3 %. IF ANOTHER IS WANTED,
11                ;       CHANGE ITTOL (CELL 111 ?) TO ANOTHER VALUE.
12                ;       IF CPU 3603 TIMING ISN'T CORRECT, CHANGE KT 575 POS 21,
13                ;       CPU DIAGRAM 019, PIN 4, C-220 PF TO 150 OR 100 PF. CHECK
14                ;       MEMORY MODULE R 32 (AT IC Z37, 74123), CORRECT VALUE
15                ;       5,11 K. IF 6,2 K ADD A PARALLEL RESISTOR OF 47K.
16                ;
17                ;4.1 PRINTER ALPHABETH CHANGE:
18                ;AT ANY TIME AFTER LOADING THIS PROGRAM IT IS POSSIBLE TO CHANGE
19                ;THE ALPHABETH USED ON THE LINEPRINTER. THERE ARE 2 METHODS:
20                ;4.1.1                    GET ONE OF THE BUILD-IN ALPHABETHS:
21                ;       START PROGRAM IN SA 2202
22                ;       REMEMBER SWITCHES TO CONTROL. (3.2).
23                ;       ANSWER ALPHABETH #, SEE EXISTING BELOW
24                ;THE PROGRAM WILL RESTART AFTER CHANGING THE ALPHABETH.
25                ;       ANSWER NEXT START ADDRESS.
26                ;4.1.2                              GET AN -ADD ON- TAPE WITH
27                ;A NEW ALPHABETH AND LOAD THIS UPON THE PROGRAM AND RESTART. THE
28                ;TAPES WITH PRINTER TABLE ALPHABETH ARE DESCRIBED BELOW:
29                ;EXISTING: #1   44-RT 535       ASCII
30                ;          #2   44-RT 529       RC STANDARD TYPE 71/78 STARTING
31                ;          #3   44-RT 532       RC STANDARD TYPE 71/78 STARTING
32                ;                                        (SKEWED 4 POSITIONS)
33                ;          #4   44-RT 1213      PL 1, TYPE 70
34                ;HOW TO PRODUCE A NEW TABLE:
35                ;THE TABLE HAS 200 OKTAL (128 DECIMAL) BOXES. INPUT KEY
36                ;IS THE ASCII VALUE OF THE CHARACTER TO PRINT ADDED TO 2000.
37                ;THE RESULT IS ADDRESS OF A BOX. EACH BOX OCCUPIES A CORE WORD.
38                ;IT IS BUILT UP OF TWO FIGURES. THE FIRST IS THE CLASS OF THE
39                ;CHARACTER TO BE PRINTED: 0 FOR PRINT, 6 FOR BLIND. THE SECOND
40                ;IS THE CHARACTER VALUE AT THE PRESENT PRINT DRUM. BELOW
41                ;VALUE 40 OCTAL FOLLOWING CHARACTERS MAY BE USED: 11 TAB,
42                ;12 LF, 14 FF AND 15 CR. ALL OTHERS BELOW 40 WILL GIVE SPACE.
43                ;THE FIRST BOX SHOULD CONTAIN THE VALUE FOR THE NULL CHAR
44                ;AND THE LAST THE VALUE FOR THE DEL CHAR, WHICH BOTH NORMALLY
45                ;ARE BLIND. IF YOU COUNT 0,1,2, ,7,10, , THE BOX 101 SHALL
46                ;CONTAIN THE PRINT DRUM VALUE FOR AN A. IF THE DRUM DO NOT
47                ;HAVE SMALL LETTERS, FILL IN THE VALUE FOR BIG ONES. NOW PUNCH
48                ;AN ASCII TAPE LIKE THIS:
49                ;   .LOC 2000
50                ;   .RDX 8          ;WHICH RDX YOU WANT
51                ;   .TXTE?
52                ;   <6><0>          ;(2000) FIRST BOX, BOX 0
53                ;   <6><0>
54                ;   .
55                ;   .
56                ;   .
57                ;   <0><101>        ;(2101) BOX 101 FOR A. FOR ASCII DRUM
58                ;   .               ;101 IS USED, FOR TYPE 71 137 IS USED.
59                ;   .
60                ;   .
61                ;   <6><0>?         ;(2177) BOX 177, LAST
62                ;   .RDX 8
63                ;   .END 101
64                ;PRODUCE A BINARY TAPE AND LOAD THIS TO MEMORY WITH
65                ;BINARY LOADER AFTER LOADING OF MAIN PROGRAM.
```

```
↑ 0004  .MAIN
01                ;                                                  PAGE 4 OF 5
02                ;5. PROGRAM DESCRIPTION
03                ;5.1      SA 400. THE PROGRAM ASKS 3 QUESTIONS. THE FIRST TW
04                ;         ARE ABOUT WHICH TRANSMISSION RATE THE TELETYPE (OR
05                ;         OPERATOR CONSOLE) HAS, USED TO CALCULATE THE NUMBER OF
06                ;         CHARS TO TRANSMIT TO SPEND ABOUT 100 MSEC OF TIME. THE
07                ;         THIRD IS WHICH TYPE OF CPU AND MEMORY MODULE FOR TEST
08                ;         TTY SPEED  1200 ? ANSWER CR OR LF IF 1200 ELSE INPUT
09                ;                           THE CORRECT # BETWEEN 100 AND 9600.
10                ;                           ANSWER THE THEORETICAL VALUE ALLTHOUG
11                ;                           THE TTC 705 ISN'T CORRECT. THE PROGRA
12                ;                           WILL CORRECT ABOUT 0,5 % FOR YOU.
13                ;         NO OF BITS PER CHAR ? CR OR LF IF 11, ELSE INPUT THE
14                ;                           CORRECT # BETWEEN 7 AND 11, CR OR LF.
15                ;         FIRST MEM MODULE, CPU TYPE ? CR OR LF IF 16, ELSE INPUT
16                ;                           THE CORRECT NUMBER BETWEEN 11 AND 23.
17                ;         THEN THE PROGRAM CALCULATES TWO CALIBRATION CONSTANTS,
18                ;         ONE FOR RTC AND ONE FOR TTY, RUNS 1 PASS OF 10 RUNS,
19                ;         WRITES PASSMESSAGE AND STARTS AGAIN WITH NEW CALIBRATIC
20                ;         IF THE INSTRUCTION TIMES ARE NOT CORRECT, ERROR MESSAGE
21                ;         ARE GIVEN ON TTY AND LPT.
22                ;5.2      SA 401. THE OPERATION IS SIMILAR TO SA 400, BUT ONLY 2
23                ;         RUNS IN A PASS. ALL INSTRUCTION TIMES ARE PRINTED
24                ;         WITHOUT TESTING FOR CORRECTNESS. AFTER A PASS THE
25                ;         PROGRAM ASKS FOR SA, ANSWER 403 IF ANOTHER PASS IS
26                ;         WANTED.
27                ;5.3      SA 402. NO QUESTIONS ARE GIVEN. THE PROGRAM WILL PRINT
28                ;         ONE LINE WITH 3 NUMBERS. AFTER THIS THE PROGRAM ASKS
29                ;         FOR SA, ANSWER 403 IF THE ACTION IS WANTED REPEATED.
30                ;         THE 3 NUMBERS ARE THE RESULT FROM A ROUTINE INTENDED
31                ;         FOR USE IN START OF ALL PROGRAMS TO INFORM WHICH CPU.
32                ;         1. NUMBER: TTY BASE, INSTR. PROFILE CONSTANT.
33                ;         2. NUMBER: RTC BASE, INSTR. PROFILE CONSTANT.
34                ;         3. NUMBER: A SIMPLE SPEED COUNT CPU-NUMBER.
35                ;         THIS IS ONLY USEFULL IF YOU WANT TO CHANGE/CONTROL
36                ;         THE CONSTANT AFTER THE LABEL LOCK IN ANOTHER TESTPROGR
37                ;5.4      THE FOLLOWING PROCEDURE IS USED TO CALCULATE THE
38                ;         INSTRUCTION TIMES IN SA 400/401. THE RTC IS COMMANDED
39                ;         TO WORK IN 100 MSEC MODE. A "INC" INSTRUCTION THEN
40                ;         RECORDS THE NUMBER OF TIMES A SMALL LOOP IS ITERATED
41                ;         BEFORE THE RTC DONE FLAG IS ONE. THIS COUNT REPRESENTS
42                ;         100 MSEC AND IS USED FOR CALIBRATION. SIMILAR A COUNT
43                ;         IS MADE FOR TTY BUT WITH SOME COMPENSATION MADE FOR
44                ;         THE CODE TO ADMINISTRATE NUMBER OF CHARACTERS. THEN
45                ;         ALL INSTR. TYPES ARE TESTED AGAINST RTC AND TTY CL
46                ;         THIS WAY: A 1000 WORD BUFFER IS FILLED WITH THE INSTR.
47                ;         TO BE TIMED. TTY OR RTC ARE SYNCHRONIZED AND PROGRAM
48                ;         CONTROL IS TRANSFERRED TO THE BUFFER. THE BUFFER IS
49                ;         EXECUTED 10 (20 IF THE INSTR SKIPS) TIMES. WHEN THE
50                ;         INSTRUCTION IN QUESTION HAS BEEN EXECUTED 10000 TIMES
51                ;         (1000*10)((500*20 IF SKIP)) THE PROGRAM THEN TIMES THE
52                ;         REMAINDER OF THE CALIBRATION PERIOD. THE VALUE THUS
53                ;         RECORDED IS SUBTRACTED FROM THE 100 MSEC CALIBRATION
54                ;         TIME. THE DIFFERENCE REPRESENTS THE TIME FOR 10000
55                ;         INSRUCTIONS. THIS TIME IS CONVERTED TO NANO SECONDS
56                ;         BEFORE PRINTING. (SOME COMPENSATION IS MADE FOR THE
57                ;         TIME (INSTRUCTIONS) BETWEEN EACH BUFFER RUN).
58                ;
59                ;6.       LIMITATIONS/MISC
60                ;         NOTE THAT INSTRUCTION TIMING VALUES ONLY ARE VALID
61                ;         WITHIN THE SAME MEMORY MODULE. REFERENCING ANOTHER
62                ;         MODULE WILL SHORTEN THE TIME IN NOVA 2 AND EXTEND
63                ;         THE TIME IN RC 3603. THE IMPORTANT PART OF THIS
64                ;         PROGRAM RUNS IN THE FIRST 4 K OF MEMORY.
```

```
↑ 0005  .MAIN
01                ;                                                    PAGE 5 OF 5
02                ;7. MESSAGES FROM THIS TEST
03                ;       ALL MESSAGES ARE WRITTEN ON TTY AND LPT.
04                ;       ABOUT THE NUMBERS:
05                ;               0-5 DIGITS IS A DECIMAL NUMBER
06                ;                   RANGE -32768 TO -1 AND 0 TO 32767
07                ;                 6 DIGITS IS AN OCTAL NUMBER
08                ;               0-6 DIGITS IS AN OCTAL NUMBER WITH
09                ;                   LEADING ZEROES SUPPRESSED, DON'T USE.
10                ;                 8 DIGITS IS A BINARY NUMBER.
11                ;
12                ;       MESSAGES:       LPT/TTY & DIS.
13                ;
14                ;       SWITCHES: 100073        NO MESS AT DIS ABOUT SW. POSITIO
15                ;       LAST LOC. 077777
16                ;       BINARY LOADER OK
17                ;       INSTR TIMER TEST        IDENTIFIKATION
18                ;       SET SWITCHES TO CONTROL, (3,2), STARTADDR   400 ?
19                ;
20                ;       SWITCHES: 000000        NO MESS AT DIS AFTER START TEST.
21                ;       000400   STARTADDR
22                ;
23                ;       TTY SPEED  1200 ?
24                ;
25                ;       NO OF BITS PER CHAR
26                ;       1 START+X DATA+Y STOP(Y=1 OR 2)INITIALIZED TO    11?
27                ;
28                ;       FIRST MEM MODULE, CPU TYPE
29                ;       NOVA 1200       12
30                ;       NOVA 2 -  8K    16
31                ;       NOVA2 - 16K     17
32                ;       RC3603 - 16K    20
33                ;         WITH BREAK    21
34                ;       RC3603 - 32K    22
35                ;         WITH BREAK    23
36                ;       INITIALIZED TO    16?
37                ;
38                ;       INSTRUCTION EXECUTION TIMES IN NANO SECONDS
39                ;       REFERENCE:          RTC     TTY     EXPECTED
40                ;
41                ;       MOV     0,0          800     800      800
42                ;       MOVL    0,0          800     800      800
43                ;       MOVS    0,0          800     800      800
44                ;       ADD     0,0          800     800      800
45                ;       AND     0,0          800     800      800
46                ;       LDA     0,CWORK     1600    1600     1600
47                ;       STA     0,CWORK     1600    1600     1600
48                ;       ISZ     CWORK       1900    1900     1900
49                ;       DSZ     CWORK       1900    1900     1900
50                ;       JMP     .+1          800     800      800
51                ;       JSR     .+1         1100    1100     1100
52                ;       LDA     0,@CWOR     2400    2400     2400
53                ;       LDA     0,@IDX5     2900    2900     2900  ;AUTO INCR. LOC
54                ;       LDA     0,@DDX0     2900    2900     2900  ;AUTO DECR. LOC
55                ;       LDA     0,@(@W)     3200    3200     3200
56                ;       NIO     0           1600    1600     1600
57                ;       NIOS    0           1900    1900     1900
58                ;       DIA     0,0         1400    1400     1400
59                ;       DOA     0,0         1600    1600     1600
60                ;       INTA    0           1400    1400     1400
61                ;       SKPBN   0           1100    1100     1100
62                ;       SKPBZ   0           1400    1400     1400  ;WITH SKIP
63                ;       MOV     0,0,SZR     1100    1100     1100  ;WITH SKIP
64                ;       MOV     0,0,SKP     1100    1100     1100  ;WITH SKIP
65                ;
66                ;       1. PASS OF 10 RUNS      WITH/WITHOUT ERRORS (SA 400)
```

```
↑ 0006   .MAIN
01
02                ;TAPE 1
03
04
05                .EOT
```

```
  0007  .MAIN
01
02                ;TAPE 2          PAGE ZERO FOR TAPE 3,4
03
04        000000 .LOC 0
05
06 00000 003160           2*MELOC          ;MESS AFTER RDOS LOAD AND STORE PC ON I
07 00001 003301           SERINT           ;ADDR OF INTR. SERVICE ROUTINE
08 00002 001400           REBIN            ;SELFSTART ADDR FOR RDOS ETC.
09 00003 000000           0                ;0=HALT, 1=SELFSTART PROG AFTER REBIN
10 00004 000000           0                ;@ADDR FOR SELFSTART PROG AFTER REBIN
11 00005 000000           0                ;FOR LOAD RDOS, USED BY POW. INTR, FITY
12
13        000020 .LOC 20
14
15 00020 000000 IDX0:     0                ;AUTO INCREMENT LOCATION
16 00021 000000 IDX1:     0                ;AUTO INCREMENT LOCATION
17 00022 000000 IDX2:     0                ;AUTO INCREMENT LOCATION
18 00023 000000 IDX3:     0                ;AUTO INCREMENT LOCATION
19 00024 000000 IDX4:     0                ;AUTO INCREMENT LOCATION
20 00025 000000 IDX5:     0                ;AUTO INCREMENT LOCATION
21
22        000030 .LOC 30
23
24 00030 000000 DDX0:     0                ;AUTO DECREMENT LOCATION
25
26        000040 .LOC 40                   ;INDIRECT ADDRESSES
27                                         ;NOT IN AUTO INC,DEC LOC.
28 00040 000412 IMESS:    XMESS
29 00041 000664 ICHAR:    XCHAR
30 00042 000724 ITYPE:    XTYPE
31 00043 001043 ICRLF:    XCRLF
32 00044 001104 IDISP:    XDISP
33 00045 001134 IDOUT:    XDOUT
34 00046 001165 IDICL:    XDICL
35 00047 001176 IDATT:    DISATT
36 00050 001221 IHAAT:    HAATT
37 00051 001014 ITBIN:    XTBIN
38 00052 000560 ITOCT:    XTOCT
39 00053 000470 ITDEC:    XTDEC
40 00054 000550 ITZOC:    XTZOC
41 00055 001020 IDBIN:    XDBIN
42 00056 000564 IDOCT:    XDOCT
43 00057 000464 IDDEC:    XDDEC
44 00060 000554 IDZOC:    XDZOC
45 00061 006375 IWAIT:    XWAIT
46 00062 001244 IWAOP:    XWTOP
47 00063 006445 ITISK:    RTIME
48 00064 006521 ITIMS:    MSTIM
49 00065 006651 ITIRO:    XTIMS
50 00066 006621 IMULT:    XMULT
51 00067 006633 IDIVS:    XDIVS
52 00070 006634 IDIVD:    XDIVD
53 00071 011067 IQUES:    XQUES
54 00072 001736 ISAMS:    XSAMS
55 00073 001312 IRESW:    XRESW
56 00074 000000 HMEND:    0                ;TOP OF MEMORY, LOADER PROTECT
57 00075 000000 DIGIN:    0                ;INPUT BUFFER FOR INPUT ROUTINES.
58
59                ;PRINTER TABLE HANDLING AND POWER RESTART:
60
61        000076 .LOC 76
62
63 00076 002100 POWZE:    JMP     @POWRE   ;INSTRUCTION TO BE STORED IN CELL ZERO
64 00077 010620 IRESA:    SWISA            ;PROGRAM RESTART ADDR
65 00100 010546 POWRE:    POWON            ;POWER RESTART ADDR
66 00101 063077 PRINT:    HALT             ;IMPORTANT TO KEEP THIS AND NEXT IN
67 00102 002077 STOP:     JMP     @IRESA   ;101,102 BECAUSE PRINTER TABLE SELFSTAR
```

```
↑ 0008  .MAIN
01
02 00103 011731 IGTBI:  GETBI
03 00104 011610 IGTOK:  GETOK
04 00105 011305 IGTDC:  GETDC
05 00106 011522 IGTSC:  GETSC
06 00107 012040 IGTTX:  GETTX
07
08              ;  DEFINITIONS
09
10              ;TTI=10
11              ;TTO=11
12              ;RTC=14
13              ;LPT=17
14       000032 FUN=32
15       000033 FUB=33
16       000034 NUK=34
17       000035 DIS=35
18       000017 XLPT=LPT
19       000010 XTTI=TTI
20       000011 XTTO=TTO
21       000014 XRTC=RTC
22
23       006040 CMESS=JSR        @IMESS  ;EACH DEFINITION BELOW CORRESPONDS
24       006041 CCHAR=JSR        @ICHAR  ;WITH A CALL OF A ROUTINE.
25       006042 CTYPE=JSR        @ITYPE
26       006043 CCRLF=JSR        @ICRLF
27       006044 CDISP=JSR        @IDISP
28       006045 CDOUT=JSR        @IDOUT
29       006046 CDICL=JSR        @IDICL
30       006047 CDATT=JSR        @IDATT
31       006050 CHAAT=JSR        @IHAAT
32       006051 CTBIN=JSR        @ITBIN
33       006052 CTOCT=JSR        @ITOCT
34       006053 CTDEC=JSR        @ITDEC
35       006054 CTZOC=JSR        @ITZOC
36       006055 CDBIN=JSR        @IDBIN
37       006056 CDOCT=JSR        @IDOCT
38       006057 CDDEC=JSR        @IDDEC
39       006060 CDZOC=JSR        @IDZOC
40       006061 CWAIT=JSR        @IWAIT
41       006062 WATOP=JSR        @IWAOP
42       006063 TIMSK=JSR        @ITISK
43       006064 TIMMS=JSR        @ITIMS
44       006065 TIMRO=JSR        @ITIRO
45       006066 MULTI=JSR        @IMULT
46       006067 DIVIS=JSR        @IDIVS
47       006070 DIVID=JSR        @IDIVD
48       006071 CQUES=JSR        @IQUES
49       006072 CSAMS=JSR        @ISAMS
50       006073 CRESW=JSR        @IRESW
51       006103 CGTBI=JSR        @IGTBI
52       006104 CGTOK=JSR        @IGTOK
53       006105 CGTDC=JSR        @IGTDC
54       006106 CGTSC=JSR        @IGTSC
55       006107 CGTTX=JSR        @IGTTX
56
57 00110 000000 RTEST:  0               ;RETURN ADDR FROM TESTLOOPS.
58 00111 000003 ITTOL:  3               ;INSTR TIME TOLERANCE IN %
59 00112 000000 ITNSE:  0               ;TOLERANCE IN NSEC CALCULATED BY PROG.
60 00113 000000 ESWIT:  0               ;ERROR FLAG
61 00114 000000 FSWIT:  0               ;FIRST ERROR FLAG
62 00115 000000 PSWIT:  0               ;PRINT FLAG
63
64 00116 000000 DPREC:  60000.D         ;60K IN DOUBLE PRECISION
   00117 165140
65 00120 165140 D60K:   60000.          ;60K SINGLE AC
```

```
↑ 0009  .MAIN
01
02 00121 000000 CALIR:  0               ;CALIBRATION ON RTC
03 00122 000000 CALIT:  0               ;CALIBRATION ON TTO
04 00123 000000 CALIC:  0               ;CALIBRATION ON TTO BEFORE CORRECTION
05 00124 000000 TIMRT:  0               ;MESSURED TIME WITH RTC
06 00125 000000 TIMTT:  0               ;MESSURED TIME WITH TTO
07 00126 003320 CITAB:  ITABE
08 00127 004373 IBUFF:  BBUF
09 00130 006343 FBUFF:  FBUF
10 00131 000000 ADINC:  0               ;ADDRESS FOR AUTO INC LOAD
11 00132 007777 ADDEC:  7777            ;ADDRESS FOR AUTO DEC LOAD
12 00133 000012 D10:    10.
13 00134 177766 N10:    -10.
14 00135 177754 N20:    -20.
15 00136 000144 D100:   100.
16 00137 000764 D500:   500.
17 00140 001750 D1000:  1000.
18 00141 023420 D10K:   10000.
19 00142 000014 CH14:   14              ;FORM FEED
20 00143 100144 WOPNT:  @CWORK          ;CWORK POINTER
21 00144 000000 CWORK:  0               ;WORKING LOC FOR MEM REFF INSTR.
22 00145 000000 RCOTI:  0               ;RETURN CORR. TIME
23 00146 000000 RPASS:  0               ;RETURN FROM PASS
24 00147 000000 PASSC:  0               ;PASS COUNTER
25 00150 000012 PASS1:  10.             ;10 TEST RUNS
26 00151 000002 PASS2:  2.              ;2 LIST RUNS
27 00152 000000 PASSN:  0               ;PASS (RUN) NUMBER
28 00153 000000 COMPI:  0               ;INTR COMPENSATION
29 00154 000000 COMPC:  0               ;START COUNT
30 00155 000000 COMPN:  0               ;NORMAL
31 00156 000000 COMPS:  0               ;SKIP
32 00157 000000 COMPA:  0               ;ACTUAL COMPENSATION
33 00160 000000 COMPR:  0               ;REST COMP (INTR + START)
34 00161 002260 TTYSQ:  1200.           ;TTY QUESTION
35 00162 000000 TTYSA:  0               ;     SPEED ANSWER
36 00163 022600 TTYUL:  9600.           ;     UPPER LIMIT
37 00164 000144 TTYLL:  100.            ;     LOWER LIMIT
38 00165 000016 CPUTQ:  16              ;CPU TYPE QUESTION
39 00166 000000 CPUTA:  0
40 00167 000024 CPUUL:  24
41 00170 000011 CPULL:  11
42 00171 000013 BITNQ:  11.             ;BIT NUMBER QUESTION
43 00172 000000 BITNA:  0
44 00173 000013 BITUL:  11.
45 00174 000007 BITLL:  7.
46 00175 000000 HCHSP:  0               ;# USEC/0,5 CHAR
47 00176 000000 NOCHA:  0               ;-# OF CHARS TO SEND
48 00177 000000 SPFAK:  0               ;FAKTOR FOR TIME NOT EXACTLY 100 MSEC
49 00200 000000 SPFCO:  0               ;CORRECTED SPFAK FOR TTC 705
50 00201 112160 D38K:   38000.          ;CONSTANT FOR TTO TIME 76 TO 152 MSEC.
51 00202 000000 TENO:   0
52
53 00203 000204 TBLCN:  .+1             ;CPUTYP NORMAL COMPENSATION
54 00204 000000         0               ;11  VALUE FOR XTRA INSTR FOR
55 00205 000040         32.             ;12  ADMINI CODE NOT INCL IN THE
56 00206 000000         0               ;13  CALIB ROUTINE (BEFORE AND AFTER
57 00207 000000         0               ;14  THE BUFFER). IF SKIP READ
58 00210 000000         0               ;15  10X AS 20X, 9X AS 19X.
59 00211 000046         38.             ;16  VALUE = TIME ADMINI CODE/
60 00212 000050         40.             ;17  TIME CALIB CODE
61 00213 000040         32.             ;20  CALIB CODE:
62 00214 000030         24.             ;21  INC        0,0
63 00215 000041         33.             ;22  SKPDN      XRTC  WITHOUT SKIP
64 00216 000031         25.             ;23  JMP        .-2
65 00217 000050         40.             ;24
```

```
↑ 0010  .MAIN
01
02 00220 000221 TBLCS:  .+1              ;CPUTYP SKIP COMPENSATION
03 00221 000000          0               ;11  ADMINI CODE:
04 00222 000076          62.             ;12  SUB         0,0
05 00223 000000          0               ;13  LDA         3,IBUFF
06 00224 000000          0               ;14  JMP         0,3
07 00225 000000          0               ;15  LDA         2,ADINC 10X
08 00226 000113          75.             ;16  STA         2,IDX5  10X
09 00227 000116          78.             ;17  LDA         2,ADDEC 10X
10 00230 000076          62.             ;20  STA         2,IDX10 10X
11 00231 000060          48.             ;21  ISZ         TENO 9X + 1X INCL SKIP
12 00232 000100          64.             ;22  JMP         @.+2 (BBUF) 9X
13 00233 000061          49.             ;23  JMP         @.+2 (TINSR)
14 00234 000116          78.             ;24
15
16 00235 000236 TBLCI:  .+1              ;CPUTYP INTR COMPENSATION
17 00236 000000          0               ;11  VALUE WHICH CALIB COULD COUNT IN
18 00237 000004          4               ;12  THE TIME THE INTERRUPT CODE SPEND
19 00240 000000          0               ;13  STA         ADDR IN 0
20 00241 000000          0               ;14  JMP         @1
21 00242 000000          0               ;15  SKPDZ       CPU  +  SKIP
22 00243 000005          5               ;16  INC         1,1,SNR  +  SKIP
23 00244 000005          5               ;17  DOA         2,XTTO
24 00245 000004          4               ;20  NIOS        XTTO
25 00246 000004          4               ;21  INTEN
26 00247 000004          4               ;22  JMP         @0
27 00250 000004          4               ;23  VALUE=TIME INTR/TIME CALIB CODE
28 00251 000005          5               ;24
29
30 00252 000253 TBLCC:  .+1              ;CPUTYP START COUNT COMPENSATION
31 00253 000000          0               ;11  VALUE WHICH CALIB COULD COUNT IN
32 00254 000005          5               ;12  THE TIME USED BEFORE WE START CALIB
33 00255 000000          0               ;13  LDA         1,NOCHA
34 00256 000000          0               ;14  DOA         2,XTTO
35 00257 000000          0               ;15  INTEN
36 00260 000006          6               ;16  STA         ADDR IN 0
37 00261 000006          6               ;17  JMP         @1
38 00262 000005          5               ;20  SKPDZ       CPU  +  SKIP
39 00263 000004          4               ;21  INC         1,1,SNR  (WITHOUT SKIP)
40 00264 000005          5               ;22  JMP         DEVIN
41 00265 000004          4               ;23  DEVIN:      3 INSTRUCTIONS
42 00266 000006          6               ;24  VALUE=TIME SYNC INTR+START COUNT/
43                                       ;       TIME CALIB CODE
44
45 00267 000270 TBLCT:  .+1              ;CPU TYPE CORRECT TIME POINTER
46 00270 000000          0               ;11
47 00271 003432          TBLN1           ;12 NOVA 1200
48 00272 000000          0               ;13
49 00273 000000          0               ;14
50 00274 000000          0               ;15
51 00275 003462          TBLN2           ;16 NOVA 2 -  8K
52 00276 003512          TBLN3           ;17 NOVA 2 - 16K
53 00277 003572          TBLR1           ;20 RC3603 - 16K
54 00300 003622          TBLR2           ;21 RC3603 - 16K BREAK
55 00301 003652          TBLR3           ;22 RC3603 - 32K
56 00302 003702          TBLR4           ;23 RC3603 - 32K BREAK
57 00303 003512          TBLN3           ;24 NOVA 2 - DATARAM 900 NSEC
58
59 00304 000000 CTIMP:  0                ;CORRECT TIMES POINTER FOR ACTUAL CPU
60 00305 006730 ICPN:   CPUNO
61 00306 003055 IBEG:   BEG
62 00307 002427 YQUES:  WQUES
63 00310 003037 IHEAD:  PHEAD
64 00311 022600 TSTRP:  9600.            ;THEORETICALLY VALUE BAUD STRAP
65 00312 022631 CSTRP:  9625.            ;MAX FOR TTC 705, AND CORRECT VALUE.
66 00313 000012 TTTYP:  12               ;CPU TYPE WITH ACCURATE TTY CONTROLLER
```

```
↑ 0011   .MAIN
01
02          000400 .LOC 400
03
04 00400 006404          JSR      @ITEST  ;TEST PROGRAM
05 00401 006404          JSR      @ILIST  ;LIST PROGRAM
06 00402 006404          JSR      @IFRAM  ;PRINT PROFILE OF CPU, CPU#
07 00403 006404          JSR      @IANSW  ;RESTART, NO QUESTIONS, LAST ANSWERS
08
09 00404 002704 ITEST:   XTEST
10 00405 002725 ILIST:   XLIST
11 00406 006354 IFRAM:   XFRAM
12 00407 002677 IANSW:   XANSW
13
14                ;THE STANDARD ROUTINES ARE MODIFIED, ONLY TAPE 3:
15                ;DIVIDED INTO TWO PARTS AS MAIN PROGRAM + BUFFER SHOULD BE
16                ;LOCATED IN THE FIRST 4K.
17                ;THE CALL OF CPTYP IN REBIN CHANGED TO IORST BECAUSE
18                ;WE DON'T WANT TO USE INSTRUCTION TIMING.
19                ;THE CALL OF PRTYP IN REBIN CHANGED TO JMP .+1 BECAUSE
20                ;WE DON'T USE CPTYP TO FIND THE CPU TYPE.
21                ;CPUTYP ROUTINE CHANGED TO PRINT PROFILE (FITYP, SESAME), AND
22                ;TO PRINT NOT CRLF (NRTYP).
23                ;DIVIS,DIVID ROUTINE CHANGED TO ROUND UP INSTEAD OF OFF.
24                ;CQUES ROUTINE NOT USED BECAUSE ONLY TTY IS IN USE.
25                ;SA 2214, 2216, 2220 REMOVED.
26
27
28                ;TAPE 2
29
30                .EOT
```

```
  0012  .MAIN
01
02                ;TAPE 3          OUTPUT ROUTINES AND OTHERS.
03
04        000412  .LOC 412         ;DO NOT MOVE UP.
05
06                                 ;FOR THE USE SEE EACH ROUTINE.
07
08                ;MESS    TYPE A TEXT MESSAGE ON TTO AND LPT
09                ;CHAR    TYPE A CHAR ON TTO AND LPT, CALCULATE P-BIT
10                ;TYPE    TYPE A CHAR ON TTO AND LPT
11                ;CRLF    TYPE A CR AND A LF ON TTO AND LPT
12                ;DISP    DISPLAY A TEXT MESSAGE ON DIS
13                ;DOUT    DISPLAY A CHAR ON DIS
14                ;DCL     CLEAR DIS
15                ;DISATT  DISPLAY ATTENTION: BEEP AND WAIT 3 SECONDS
16                ;HAATT   HALT ATTENTION: BEEP
17                ;TBIN    TYPE BINARY NUMBER ON TTO AND LPT
18                ;TOCT    TYPE OCTAL NUMBER ON TTO AND LPT
19                ;TDEC    TYPE DECIMAL NUMBER ON TTO AND LPT
20                ;TZOC    TYPE OCTAL NUMBER WITHOUT LEADING ZEROES.
21                ;DBIN    DISPLAY BINARY NUMBER ON DIS
22                ;DOCT    DISPLAY OCTAL NUMBER ON DIS
23                ;DDEC    DISPLAY DECIMAL NUMBER ON DIS
24                ;DZOC    DISPLAY OCTAL NUMBER WITHOUT LEADING ZEROES.
25                ;WAIT    WAIT SOME MILLISECONDS
26                ;WATOP   WAIT FOR OPERATOR ACTION
27                ;TIMSK   TIMER FOR SKIP INSTRUCTION
28                ;TIMRO   TIMER FOR ROUTINE
29                ;TIMMS   TIME MESSURE ROUTINE
30                ;MULTI   MULTIPLY
31                ;DIVIS   DIVIDE SINGLE
32                ;DIVID   DIVIDE DOUBLE
33                ;QUES    OUTPUT QUESTIONS
34                ;SAMS    START ADDR MESSAGE
35                ;RESW    READ SWITCHES, REPORT CHANGES
36                ;SAPTB   MOVE PRINTER TABLE
37                ;SAMEX   EXTEND TO 64K WORDS MEM MODE
38                ;SAMNM   RESET TO 32K WORDS MEM MODE
39                ;REBIN   INITIALIZE, RESTORE BIN LOADER
40
41
42                ;AN ADD ON PRINTERTABLE SHOULD HAVE THIS FORMAT:
43                ;    .LOC 2000
44                ;    .RDX Y
45                ;    .TXTE?
46                ;    <6><0>       ;TOTALLY 200 BOXES WITH
47                ;                 ;<TYPE OF ACTION><CHAR TO PRINT>
48                ;    <6><0>?
49                ;    .END 101
50                ;                 ;TYPE OF ACTION:        0=PRINT
51                ;                                         6=BLIND
```

```
↑ 0013  .MAIN
01
02              ;LPT, TTO AND DIS NON INTERRUPT PACKAGE
03              ;IF THE DEVICE(S) ARE NOT PRESENT THE ROUTINES CONTINUES.
04              ;IF A DEVICE IS CONNECTED (SET TO ONLINE) THE PROGRAM
05              ;MUST BE RESTARTED IN THE WANTED STARTADDRESS.
06
07              ;LPT AND TTO ROUTINES:
08              ;>MESS< PRINTS ASCII MESSAGES AS SPECIFIED BY ASSEMBLER.
09              ;>CHAR< PRINTS ASCII CHARACTOR, AC0-R.
10              ;AC0-L MUST BE 0. CORRECTS THE PARITY, 11 SIMULATE TAB.
11              ;>TYPE< PRINTS AC0-R. MUST HAVE PROPER PARITY. RETURN IS
12              ;TO CALL+1.REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF DESIRED.
13              ;>CRLF< PRINTS A CARRIAGE RETURN FOLLOWED BY A LINE FEED.
14              ;>TOCT< PRINTS AC1 IN OCTAL, 6 DIGITS AND 6 PRINTPOSITIONS.
15              ;>TDEC< PRINTS AC1 IN DECIMAL, LEADING 0'S SUPPRESSED, WITH SIGN
16              ;RANGE -32768 TO -1 AND +0 TO +32767, 6 PRINTPOS, UP TO 5 DIGITS.
17              ;>TOCT< PRINTS AC1 IN OCTAL, LEADING 0'S SUPPRESSED.
18              ;6 PRINTPOSITIONS AND UP TO 6 DIGITS.
19              ;THE ROUTINES TOCT, TZOC AND TDEC ARE FOLLOWED BY A TAB
20              ;BUT ONLY IF NOT ALLREADY AT A TABPOINT.
21              ;>TBIN< PRINTS AC1 IN BINARY, AC1-R 8 DIGITS ONLY, 8 PRINTPOS.
22
23              ;DIS ROUTINES:
24              ;>DISP< PRINTS ACSII MESSAGES AS SPECIFIED BY ASSEMBLER.
25              ;>DOUT< PRINTS ASSII CHARACTOR, AC0-R. AC0-L MUST BE 0. RETURN
26              ;TO CALL+1. REPLACE THIS ROUTINE WITH INTERRUPT TYPE IF DESIRED.
27              ;>DICL< CLEARS THE DISPLAY.
28              ;>DISATT< ATTENTION DISPLAY, SEE ROUTINE
29              ;>HAATT<  ATTENTION HALT, SEE ROUTINE
30              ;>DOCT< DISPLAYS AC1 IN OCTAL, 6 DIGITS AND 6 PRINTPOSITIONS.
31              ;>DDEC< DISPLAYS AC1 IN DECIMAL LEADING 0'S SUPPRESSED, WITH SIG
32              ;RANGE -32768 TO -1 AND +0 TO +32767, 6 PRINTPOS, UP TO 5 DIGITS.
33              ;>DZOC< DISPLAYS AC1 IN OCTAL, LEADING 0'S SUPPRESSED,
34              ;6 PRINTPOSITIONS AND UP TO 6 DIGITS.
35              ;>DBIN< DISPLAYS AC1 IN BINARY, AC1-R 8 DIGITS ONLY, 8 PRINTPOS.
36
37              ;       LDA     1,NUMBER ;ALL NUMBER ROUTINES RESTORE AC1
38              ;CALL   CTDEC
39              ;       CTOCT
40              ;       CTZOC
41              ;       CDDEC
42              ;       CDOCT
43              ;       CDZOC
44              ;       CTBIN
45              ;       CDBIN
46              ;       MOVS    1,1     ;FOR THE OTHER 8 BITS
47              ;       CTBIN
48              ;       CDBIN
49              ;       LDA     0,CHAR  ;ALL CHARACTER ROUTINES RESTORE AC0
50              ;       CTYPE
51              ;       CCHAR
52              ;       CDOUT
53              ;       CDICL
54              ;       CDISP
55              ;       TEXTLABEL
56              ;       CDATT
57              ;       CCRLF
58              ;       CMESS
59              ;       LABELTEXT
60              ;       CHAAT
61              ;       HALT
```

```
↑ 0014  .MAIN
01
02 00412 054450 XMESS:  STA     3,RPOUT ;PRINT A TEXT MESSAGE
03 00413 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
04 00414 010446         ISZ     RPOUT
05 00415 004556         JSR     PINHI
06 00416 002444         JMP     @RPOUT  ;NO PRINT
07 00417 034443         LDA     3,RPOUT ;AC3 POINTS TO MESSAGEPOINTER+1
08 00420 031777         LDA     2,-1,3  ;AC2 POINTS TO MESSAGE
09 00421 024440         LDA     1,CHMAS ;A 8 BIT MASK
10 00422 021000 MESSA:  LDA     0,0,2   ;AC0=DATA WORD
11 00423 125112         MOVL#   1,1,SZC
12 00424 123701         ANDS    1,0,SKP
13 00425 123401         AND     1,0,SKP ;AC0=DATA CHARACTOR RIGHT
14 00426 151400         INC     2,2     ;INC TO NEXT WORD
15 00427 124000         COM     1,1     ;FLIP MASK
16 00430 004404         JSR     MESCH   ;PRINT
17 00431 000771         JMP     MESSA   ;ANOTHER
18 00432 004405         JSR     BZOUT   ;TERMINATE MESS
19 00433 002427         JMP     @RPOUT  ;EXIT
20
21 00434 101015 MESCH:  MOV#    0,0,SNR ;TEST LAST CHAR
22 00435 001401         JMP     1,3     ;RETURN +2 IF NULL
23 00436 002420         JMP     @ZCHAR  ;TYPE CHAR
24
25 00437 054412 BZOUT:  STA     3,RBZOT
26 00440 006063         TIMSK           ;WAIT IF LPT/TTO BUSY
27 00441 001750 XLPTT:  1000.           ;MAX 1 SEC LPT
28 00442 063517         SKPBZ   XLPT
29 00443 044776         STA     1,XLPTT ;REMOVE WAITING, LPT NOT CONNECTED
30 00444 006063         TIMSK
31 00445 000454 XTTOT:  300.            ;MAX 300 MSEC TTO
32 00446 063511         SKPBZ   XTTO
33 00447 044776         STA     1,XTTOT ;REMOVE WAITING, TTO NOT CONNECTED
34 00450 002401         JMP     @RBZOT
35
36 00451 000000 RBZOT:  0
37 00452 000000 RINHI:  0
38 00453 000000 SADIG:  0
39 00454 000000 RXDEC:  0
40 00455 000040 CHINH:  40
41 00456 000672 ZCHAR:  YCHAR
42 00457 001142 ZDOUT:  YDOUT
43 00460 000712 ZTAB3:  YTAB3
44 00461 000377 CHMAS:  377
45 00462 000000 RPOUT:  0
46 00463 000000 PDECR:  0
47
48 00464 054776 XDDEC:  STA     3,RPOUT ;DISPLAY DECIMAL NUMBER
49 00465 004415         JSR     YPDEC
50 00466 000502         JMP     NXDIS   ;DISPLAY NEXT DIGIT
51 00467 000411         JMP     EXDIS   ;EXIT
52
53 00470 054772 XTDEC:  STA     3,RPOUT ;TYPE DECIMAL NUMBER
54 00471 004411         JSR     YPDEC
55 00472 000402         JMP     TYPNX   ;TYPE NEXT DIGIT
56 00473 000404         JMP     EXTYP   ;EXIT
57
58 00474 054760 TYPNX:  STA     3,RXDEC
59 00475 004575         JSR     YCHAR   ;TYPE DIGIT
60 00476 002756         JMP     @RXDEC  ;NEXT DIGIT, SIGPR OR DECPR
61
62 00477 006761 EXTYP:  JSR     @ZTAB3  ;YPDEC/ZOCT/POCT FINISH RETURN, TYPE TAB
63 00500 024753 EXDIS:  LDA     1,SADIG ;DISPLAY FINISHED, RESTORE PARAM
64 00501 002761         JMP     @RPOUT  ;EXIT
```

```
↑ 0015  .MAIN
01
02 00502 044751 YPDEC:  STA     1,SADIG ;SAVE PARAM FOR REPEAT
03 00503 054760          STA     3,PDECR
04 00504 030756          LDA     2,RPOUT ;SPEC RETURN IF PRINT SW
05 00505 004466          JSR     PINHI
06 00506 002754          JMP     @RPOUT  ;NO PRINT
07 00507 020436          LDA     0,CHPLU ;SETUP PLUS
08 00510 040437          STA     0,CHFLG ;SIGN FLAG
09 00511 020552          LDA     0,CHSPA ;OR PLUS: CHPLU
10 00512 040432          STA     0,CHSIG ;PRINT SPACE OR PLUS
11 00513 102620          SUBZR   0,0     ;AC0:=100000
12 00514 106415          SUB#    0,1,SNR ;TEST FOR -32768
13 00515 000404          JMP     PDEC2
14 00516 107415          AND#    0,1,SNR ;TEST FOR NEGATIVE
15 00517 000404          JMP     PDEC3   ;POS
16 00520 124400          NEG     1,1     ;NEG
17 00521 020425 PDEC2:   LDA     0,CHMIN ;SETUP MINUS
18 00522 040422          STA     0,CHSIG
19 00523 020537 PDEC3:   LDA     0,CHSP  ;SUPPRESS LEADING ZEROES
20 00524 030516          LDA     2,DECTB ;PRINT AC1 IN DECIMAL
21 00525 000466          JMP     PDEC1
22
23 00526 054415 SIGN:    STA     3,SIGNR
24 00527 034420          LDA     3,CHFLG
25 00530 030415          LDA     2,CHPLU
26 00531 172414          SUB#    3,2,SZR ;TEST PRINT SIGN
27 00532 002411          JMP     @SIGNR  ;IF NO SIGN, RETURN.
28 00533 034527          LDA     3,CHSP
29 00534 116415          SUB#    0,3,SNR ;TEST FIRST DIGIT
30 00535 002406          JMP     @SIGNR  ;IF LEADING SPACE, RETURN
31 00536 040411          STA     0,CHFLG ;STORE ASCII AND DESTROY
32 00537 020405          LDA     0,CHSIG ;PRINT SIGN FLAG
33 00540 006723          JSR     @PDECR  ;OUTPUT SIGN
34 00541 020406          LDA     0,CHFLG ;RESTORE ASCII DIGIT
35 00542 002401          JMP     @SIGNR  ;PRINT FIRST DIGIT
36
37 00543 000000 SIGNR:   0
38 00544 000000 CHSIG:   0
39 00545 000053 CHPLU:   53
40 00546 000055 CHMIN:   55
41 00547 000000 CHFLG:   0
42
43 00550 054712 XTZOC:   STA     3,RPOUT ;TYPE ZERO OCTAL NUMBER
44 00551 004430          JSR     YZOCT
45 00552 000722          JMP     TYPNX   ;TYPE NEXT DIGIT
46 00553 000724          JMP     EXTYP   ;EXIT
47
48 00554 054706 XDZOC:   STA     3,RPOUT ;DISPLAY ZERO OCTAL NUMBER
49 00555 004424          JSR     YZOCT
50 00556 000412          JMP     NXDIS   ;DISPLAY NEXT DIGIT
51 00557 000721          JMP     EXDIS   ;EXIT
52
53 00560 054702 XTOCT:   STA     3,RPOUT ;TYPE OCTAL NUMBER
54 00561 004422          JSR     YPOCT
55 00562 000712          JMP     TYPNX   ;TYPE NEXT DIGIT
56 00563 000714          JMP     EXTYP   ;EXIT
57
58 00564 054676 XDOCT:   STA     3,RPOUT ;DISPLAY OCTAL NUMBER
59 00565 004416          JSR     YPOCT
60 00566 000402          JMP     NXDIS   ;DISPLAY NEXT DIGIT
61 00567 000711          JMP     EXDIS   ;EXIT
62
63 00570 054664 NXDIS:   STA     3,RXDEC
64 00571 006666          JSR     @ZDOUT  ;DISPLAY DIGIT
65 00572 002662          JMP     @RXDEC  ;NEXT DIGIT
```

```
↑ 0016  .MAIN
01
02 00573 054657 PINHI:  STA     3,RINHI ;AFTER ACTIVATING SETPX IN TESTLOOP
03 00574 000403 SETAC:  JMP     NINHI   ;THIS IS CHANGED TO CRESW (READS 2 K
04 00575 034660         LDA     3,CHINH ;SW 10 FOR NO PRINTING
05 00576 157405         AND     2,3,SNR ;RETURN +1 IF INHIBIT
06 00577 010653 NINHI:  ISZ     RINHI   ;RETURN+2 IF PRINTING
07 00600 002652         JMP     @RINHI  ;EXIT
08
09 00601 020461 YZOCT:  LDA     0,CHSP  ;ZERO OCTAL ROUTINE
10 00602 101001         MOV     0,0,SKP
11
12 00603 020565 YPOCT:  LDA     0,CHAR0 ;OCTAL ROUTINE
13 00604 044647         STA     1,SADIG ;SAVE PARAM FOR REPEAT
14 00605 054656         STA     3,PDECR
15 00606 030654         LDA     2,RPOUT ;SPEC RETURN IF PRINT SW
16 00607 004764         JSR     PINHI
17 00610 002652         JMP     @RPOUT  ;NO PRINT
18 00611 030440         LDA     2,OCTAB ;PRINT AC1 IN OCTAL
19 00612 040735         STA     0,CHFLG ;NO-SIGN FLAG
20 00613 040556 PDEC1:  STA     0,ZSUPP ;BOTH ENTRYS PRINT NUMBER
21 00614 050401         STA     2,.+1   ;THEN TAB TO NEXT POSITION
22 00615 000000 DECOCT: 0               ;A LDA 2,TABLE INSTRUCTION
23 00616 010777         ISZ     .-1
24 00617 151005         MOV     2,2,SNR ;IF TABLE ENTRY=0 THEN
25 00620 000420         JMP     DECEX   ;EXIT WITH TAB SPECIAL IF TYPE ROU.
26 00621 034550         LDA     3,ZSUPP ;ZEROS SUPPRESS STUF
27 00622 102400         SUB     0,0
28 00623 146512 DECOT:  SUBL#   2,1,SZC
29 00624 000405         JMP     DECP
30 00625 146400         SUB     2,1     ;FORM THE DIGIT
31 00626 034542         LDA     3,CHAR0 ;DO NOT SUPPRESS
32 00627 101400         INC     0,0     ;FOLLOWING ZEROES
33 00630 000773         JMP     DECOT
34 00631 151235 DECP:   MOVZR#  2,2,SNR ;IF LAST DIGIT THEN
35 00632 034536         LDA     3,CHAR0 ;AC3=ZERO, NOT SUPPR CHAR
36 00633 054536         STA     3,ZSUPP ;AC0=DIGIT
37 00634 163000         ADD     3,0     ;MAKE ASCII
38 00635 004671         JSR     SIGN    ;TEST SIGN
39 00636 006625         JSR     @PDECR  ;OUTPUT DIGIT
40 00637 000756         JMP     DECOCT  ;GET NEXT DIGIT
41 00640 010623 DECEX:  ISZ     PDECR   ;RETURN ADDR FOR JSR Y-ROUTINE
42 00641 002622         JMP     @PDECR  ;EXIT WITH TAB IF TYPING ROUTINE
43
44 00642 030426 DECTB:  LDA     2,.+1+.-DECOCT
45       000012 .RDX 10
46 00643 023420         10000
47 00644 001750         1000
48 00645 000144         100
49 00646 000012         10
50 00647 000001         1
51 00650 000000         0
52       000010 .RDX 8
53
54 00651 030435 OCTAB:  LDA     2,.+1+.-DECOCT
55 00652 100000         100000
56 00653 010000         10000
57 00654 001000         1000
58 00655 000100         100
59 00656 000010         10
60 00657 000001         1
61 00660 000000         0
62
63 00661 000000 CHRET:  0
64 00662 000240 CHSP:   240             ;LEADING ZERO SUPPRESS CHAR + TAB SIMU
65 00663 000240 CHSPA:  240             ;A SPACE FOR + IN PDEC
```

```
↑ 0017  .MAIN
01
02 00664 040507 XCHAR:  STA     0,SACHA ;SAVE PARAM FOR REPEAT
03 00665 054774         STA     3,CHRET
04 00666 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
05 00667 004704         JSR     PINHI
06 00670 002771         JMP     @CHRET  ;NO TYPE
07 00671 000402         JMP     QCHAR
08
09 00672 054767 YCHAR:  STA     3,CHRET ;PRINT AC0 RIGHT
10 00673 101320 QCHAR:  MOVZS   0,0
11 00674 040500         STA     0,CHSAV
12 00675 176000 CHAR2:  ADC     3,3     ;COMPUTE THE PARITY
13 00676 117000         ADD     0,3     ;FOR EVEN
14 00677 163404         AND     3,0,SZR
15 00700 000775         JMP     CHAR2
16 00701 176660         SUBCR   3,3     ;COMBIND PARITY WITH CHAR
17 00702 020472         LDA     0,CHSAV
18 00703 163300         ADDS    3,0
19 00704 034463 CHAR1:  LDA     3,CHTAB ;IS THIS A TAB
20 00705 116405         SUB     0,3,SNR
21 00706 000413         JMP     CHAR4   ;YES
22 00707 004423         JSR     YTYPE   ;NO PRINT IT
23 00710 020463         LDA     0,SACHA ;RESTORE PARAM
24 00711 002750         JMP     @CHRET  ;EXIT
25
26 00712 054747 YTAB3:  STA     3,CHRET ;NUMBER ROUTINE TAB RETURN
27 00713 020462 CHAR3:  LDA     0,CHORZ ;SIMULATE A TAB
28 00714 034462         LDA     3,CHAR7 ;VIA 1 TO 8 SPACES
29 00715 117404         AND     0,3,SZR
30 00716 000403         JMP     CHAR4
31 00717 020454         LDA     0,SACHA ;RESTORE PARAM
32 00720 002741         JMP     @CHRET  ;EXIT AFTER TAB
33 00721 020741 CHAR4:  LDA     0,CHSP
34 00722 004410         JSR     YTYPE
35 00723 000770         JMP     CHAR3
36
37 00724 040554 XTYPE:  STA     0,REG0  ;SAVE PARAM FOR REPEAT
38 00725 054556         STA     3,REG3  ;SAVE RETURN
39 00726 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
40 00727 004644         JSR     PINHI
41 00730 002553         JMP     @REG3   ;NO TYPE
42 00731 000403         JMP     QTYPE
43
44 00732 054551 YTYPE:  STA     3,REG3  ;SAVE RETURN
45 00733 040545         STA     0,REG0  ;SAVE PARAM FOR TYPING
46 00734 010441 QTYPE:  ISZ     CHORZ   ;INC HORIZIONAL POSITION
47 00735 044544         STA     1,REG1  ;SAVE AC1 AND AC2 FOR NUMBER AND
48 00736 050544         STA     2,REG2  ;MESS ROUTINES
49 00737 006440         JSR     @TBZOT
50 00740 020540         LDA     0,REG0
51 00741 000437         JMP     XFORM
52 00742 061017 TYPE1:  DOA     0,XLPT  ;SEND CHAR
53 00743 060117         NIOS    XLPT    ;START LPT
54 00744 020534 TYPE2:  LDA     0,REG0
55 00745 024532         LDA     1,CHLF
56 00746 106415         SUB#    0,1,SNR ;LF ?
57 00747 000404         JMP     TYPE3   ;YES, LF
58 00750 024526         LDA     1,CHCR
59 00751 106414         SUB#    0,1,SZR ;CR ?
60 00752 000404         JMP     TYPE4   ;NO CR
61 00753 006061 TYPE3:  CWAIT           ;IF CR, LF WAIT FOR
62 00754 001237         SECM2           ;DATAPOINT 20 MSEC
63 00755 020523         LDA     0,REG0  ;RESTORE AC0
64 00756 061011 TYPE4:  DOA     0,XTTO  ;SEND CHAR
65 00757 060111         NIOS    XTTO    ;START TTO
```

```
↑ 0018  .MAIN
01
02 00760 152400 TYPE5:  SUB     2,2
03 00761 024516          LDA     1,CHLF
04 00762 106415          SUB#    0,1,SNR ;IF LF
05 00763 050412          STA     2,CHORZ ;CLEAR HORZ POS
06 00764 030516          LDA     2,REG2
07 00765 024514          LDA     1,REG1  ;ACO = REGO FOR REPEAT
08 00766 002515          JMP     @REG3   ;EXIT
09
10 00767 000011 CHTAB:   11
11 00770 000060 CHAR0:   60
12 00771 000000 ZSUPP:   0
13 00772 177770 NN10:    -10
14 00773 000000 SACHA:   0
15 00774 000000 CHSAV:   0
16 00775 000000 CHORZ:   0
17 00776 000007 CHAR7:   7
18 00777 000437 TBZOT:   BZOUT
19
20 01000 030561 XFORM:   LDA     2,RMSK  ;TRANSFORM CHAR TO
21 01001 143400          AND     2,0     ;PRINTER ALPHABET
22 01002 030556          LDA     2,PTAB
23 01003 113000          ADD     0,2     ;AC2:=CHAR+TABLE
24 01004 025000          LDA     1,0,2
25 01005 030554          LDA     2,RMSK
26 01006 133400          AND     1,2     ;AC2:=CLASS
27 01007 020554          LDA     0,LMSK
28 01010 123700          ANDS    1,0     ;ACO:=CHAR
29 01011 024443          LDA     1,ACTN
30 01012 133000          ADD     1,2     ;AC2:=ACTN+1+CLASS
31 01013 003000          JMP     @0,2    ;GO TO ACTION
32
33 01014 054461 XTBIN:   STA     3,CRBIR
34 01015 004407          JSR     YPBIN
35 01016 004654          JSR     YCHAR   ;TYPE DIGIT
36 01017 000420          JMP     PBINN   ;NEXT DIGIT
37
38 01020 054455 XDBIN:   STA     3,CRBIR
39 01021 004403          JSR     YPBIN
40 01022 004520          JSR     YDOUT   ;DISPLAY DIGIT
41 01023 000414          JMP     PBINN   ;NEXT DIGIT
42
43 01024 044447 YPBIN:   STA     1,SABIN ;SAVE PARAM FOR REPEAT
44 01025 054447          STA     3,PBINR
45 01026 030447          LDA     2,CRBIR ;SPEC RETURN IF PRINT SW
46 01027 006535          JSR     @TINHI
47 01030 002445          JMP     @CRBIR  ;NO PRINT
48 01031 030741          LDA     2,NN10  ;8 TIMES
49 01032 125300          MOVS    1,1
50 01033 020735 PBINC:   LDA     0,CHAR0
51 01034 125102          MOVL    1,1,SZC
52 01035 101400          INC     0,0     ;ACO:="CHAR1"
53 01036 002436          JMP     @PBINR  ;OUTPUT DIGIT
54 01037 151404 PBINN:   INC     2,2,SZR
55 01040 000773          JMP     PBINC
56 01041 024432          LDA     1,SABIN ;RESTORE PARAM
57 01042 002433          JMP     @CRBIR
58
59 01043 054432 XCRLF:   STA     3,CRBIR ;SAVE RETURN
60 01044 171000          MOV     3,2     ;SPEC RETURN IF PRINT SW
61 01045 006517          JSR     @TINHI
62 01046 002427          JMP     @CRBIR  ;NO TYPE
63 01047 020427          LDA     0,CHCR
64 01050 004622          JSR     YCHAR
65 01051 020426          LDA     0,CHLF
66 01052 004620          JSR     YCHAR   ;PRINT CR,LF
67 01053 002422          JMP     @CRBIR  ;EXIT
```

```
↑ 0019  .MAIN
01
02 01054 001055 ACTN:   .+1              ;ACTION ENTRY TABLE
03 01055 001071         ACT0             ;NORMAL ACTION
04 01056 001067         ILL
05 01057 001067         ILL
06 01060 001067         ILL
07 01061 001067         ILL
08 01062 001067         ILL
09 01063 001072         ACT6             ;BLIND
10 01064 001067         ILL
11 01065 001067         ILL
12 01066 001067         ILL
13
14 01067 063077 ILL:    HALT
15 01070 000777         JMP     .-1      ;ILLEGAL
16 01071 000651 ACT0:   JMP     TYPE1
17 01072 000652 ACT6:   JMP     TYPE2
18
19 01073 000000 SABIN:  0
20 01074 000000 PBINR:  0
21 01075 000000 CRBIR:  0
22 01076 000215 CHCR:   215
23 01077 000012 CHLF:   12
24 01100 000000 REG0:   0
25 01101 000000 REG1:   0
26 01102 000000 REG2:   0
27 01103 000000 REG3:   0
28
29 01104 054771 XDISP:  STA     3,CRBIR ;DISPLAY MESSAGE
30 01105 171000         MOV     3,2     ;SPEC RETURN IF PRINT SW
31 01106 010767         ISZ     CRBIR
32 01107 006455         JSR     @TINHI
33 01110 002765         JMP     @CRBIR  ;NO PRINT
34 01111 034764         LDA     3,CRBIR ;AC3=POINTS TO MESSAGEPOINTER+1
35 01112 031777         LDA     2,-1,3  ;AC2 POINTS TO MESSAGE
36 01113 024447         LDA     1,CMSK
37 01114 021000 DISP1:  LDA     0,0,2   ;AC0=DATAWORD
38 01115 125112         MOVL#   1,1,SZC
39 01116 123701         ANDS    1,0,SKP
40 01117 123401         AND     1,0,SKP ;AC0=CHAR. RIGHT
41 01120 151400         INC     2,2     ;INC TO NEXT WORD
42 01121 124000         COM     1,1     ;FLIP MASK
43 01122 004407         JSR     DDICH   ;GO DISPLAY
44 01123 000771         JMP     DISP1   ;ANOTHER
45 01124 006063         TIMSK           ;TERMINATE DISP
46 01125 000050 FDIST:  40.             ;MAX 40 MSEC
47 01126 063535         SKPBZ   DIS
48 01127 044776         STA     1,FDIST ;REMOVE WAITING, DIS NOT CONNECTED
49 01130 002745         JMP     @CRBIR  ;RETURN
50
51 01131 101015 DDICH:  MOV#    0,0,SNR ;TEST LAST CHAR
52 01132 001401         JMP     1,3     ;RETURN +2 IF NULL
53 01133 000407         JMP     YDOUT   ;DISPLAY
```

```
↑ 0020  .MAIN
01
02 01134 040744 XDOUT:  STA     0,REG0  ;SAVE PARAM FOR REPEAT
03 01135 054746          STA     3,REG3  ;SAVE RETURN
04 01136 171000          MOV     3,2     ;SPEC RETURN IF PRINT SW
05 01137 006425          JSR     @TINHI
06 01140 002743          JMP     @REG3   ;NO DISPLAY
07 01141 000403          JMP     QDOUT
08
09 01142 054741 YDOUT:  STA     3,REG3  ;SAVE RETURN
10 01143 040735          STA     0,REG0  ;SAVE PARAM FOR DISPLAYING
11 01144 044735 QDOUT:  STA     1,REG1  ;SAVE AC1 AND AC2 FOR NUMBER AND
12 01145 050735          STA     2,REG2  ;MESS ROUTINES
13 01146 006063          TIMSK           ;WAIT IF DIS BUSY
14 01147 000050 SDIST:  40.             ;MAX 40 MSEC
15 01150 063535          SKPBZ   DIS
16 01151 044776          STA     1,SDIST ;REMOVE WAITING, DIS NOT CONNECTED
17 01152 020726          LDA     0,REG0
18 01153 061035          DOA     0,DIS   ;SEND CHAR
19 01154 060135          NIOS    DIS     ;START DIS
20 01155 024724          LDA     1,REG1
21 01156 030724          LDA     2,REG2  ;AC0 = REG0 FOR REPEAT
22 01157 002724          JMP     @REG3   ;EXIT
23
24 01160 002000 PTAB:   TABLE
25 01161 000177 RMSK:   177
26 01162 000377 CMSK:   377
27 01163 077400 LMSK:   077400
28 01164 000573 TINHI:  PINHI
29
30 01165 054710 XDICL:  STA     3,CRBIR ;SAVE RETURN
31 01166 171000          MOV     3,2     ;SPEC RETURN IF PRINT SW
32 01167 006775          JSR     @TINHI
33 01170 002705          JMP     @CRBIR  ;NO DISPLAY
34 01171 000402          JMP     QDICL
35
36 01172 054703 YDICL:  STA     3,CRBIR
37 01173 020704 QDICL:  LDA     0,CHLF
38 01174 004746          JSR     YDOUT   ;DISPLAY CLEAR
39 01175 002700          JMP     @CRBIR  ;RETURN
```

```
↑ 0021  .MAIN
01
02                          ;RC 3600, ATTENTION DISPLAY OUTPUT
03                          ;BY MEANS OF ACOUSTIC ALARM FOR 50 MS
04                          ;AND WAIT FOR 3 SECONDS TO ALLOW
05                          ;THE OPERATOR TO READ THE MESSAGE.
06                          ;
07                          ;CALL   CDATT
08                          ;
09                          ;RC 3600, ATTENTION HALT BY MEANS OF
10                          ;ACOUSTIC ALARM FOR 50 MS BEFORE HALT.
11                          ;
12                          ;CALL   CHAAT
13                          ;
14 01176 054676 DISATT:  STA     3,PBINR
15 01177 171000          MOV     3,2        ;SPEC RETURN IF PRINT SW
16 01200 006764          JSR     @TINHI
17 01201 002673          JMP     @PBINR     ;NO PRINT
18 01202 004424          JSR     YHAAT
19 01203 006073          CRESW               ;READS 2 ROUTINE
20 01204 024436          LDA     1,SECS4    ;SW13 WAITING DIS
21 01205 133415          AND#    1,2,SNR
22 01206 000403          JMP     DISSW
23 01207 006061          CWAIT
24 01210 001236          SEC3
25
26 01211 006073 DISSW:   CRESW               ;READS 2 ROUTINE
27 01212 024427          LDA     1,SECS2
28 01213 133414          AND#    1,2,SZR
29 01214 063077          HALT                ;HALT IF SWITCH 14
30 01215 126520          SUBZL   1,1         ;AC1:=1
31 01216 133414          AND#    1,2,SZR
32 01217 004753          JSR     YDICL      ;CLEAR DIS IF SWITCH 15
33 01220 002654          JMP     @PBINR
34
35 01221 054654 HAATT:   STA     3,CRBIR
36 01222 171000          MOV     3,2        ;SPEC RETURN IF PRINT SW
37 01223 006741          JSR     @TINHI
38 01224 002651          JMP     @CRBIR     ;NO "PRINT"
39 01225 000402          JMP     QHAAT
40
41 01226 054647 YHAAT:   STA     3,CRBIR
42 01227 020414 QHAAT:   LDA     0,SEC4K
43 01230 061032          DOA     0,FUN
44 01231 006061          CWAIT
45 01232 001240          SECM5
46 01233 102400          SUB     0,0
47 01234 061032          DOA     0,FUN
48 01235 002640          JMP     @CRBIR
49
50       000012          .RDX 10            ;CHANGING SEC3 AFFECTS CRESW AND POWON.
51 01236 005670 SEC3:    3000               ;CHANGE TO 1000(0) FOR 1(0) SEC WAITING.
52 01237 000024 SECM2:   20                 ;20 MSEC FOR TYPE ROUTINE DATAPOINT DELA
53 01240 000062 SECM5:   50
54       000010          .RDX 8
55 01241 000002 SECS2:   2
56 01242 000004 SECS4:   4
57 01243 004000 SEC4K:   4000
```

```
↑ 0022  .MAIN
01
02                ;PROCEDURE WAIT OPERATOR
03                ;TURNS ON THE FUNCTION INDICATOR FROM ARG,
04                ;WAITS FOR THE OPERATOR TO PRESS BUTTON FROM ARG,
05                ;OR TO PRESS KEY AT TTY
06                ;AND TURNS OFF THE FUNCTION INDICATOR FROM ARG.
07                ;AND WRITES PROG NAME AT DIS (SEE LABEL: PROG),
08                ;OR TYPE $ AT TTY
09                ;UNCHANGED:     AC0
10                ;DESTROYED:     AC1,AC2,AC3,CARRY
11                ;CALL:  WATOP
12                ;       ARG
13
14 01244 025400 XWTOP:  LDA     1,0,3   ; FUNCTION MASK FROM ARGUMENT
15 01245 175400         INC     3,3     ; FOR INDICATORS AND BUTTONS
16 01246 054437         STA     3,WTORE ; SAVE RETURN
17 01247 034440         LDA     3,WHIGH
18 01250 040441         STA     0,WACSAV
19 01251 020435         LDA     0,WLOWL
20 01252 065032         DOA     1,FUN   ; TURN ON INDICATOR ARG
21 01253 063610 WTNOK:  SKPDN   XTTI
22 01254 000411         JMP     WTFUB
23 01255 070410         DIA     2,XTTI
24 01256 151300         MOVS    2,2     ; REMOVE P-BIT BY
25 01257 151120         MOVZL   2,2     ; MULTIPLYING WITH 512 DEC.
26 01260 060210         NIOC    XTTI
27 01261 172033         ADCZ#   3,2,SNC ; IS KEY BETWEEN OR
28 01262 142032         ADCZ#   2,0,SZC ; EQUAL TO LIMITS ?
29 01263 000402         JMP     WTFUB   ; NO
30 01264 000404         JMP     WTOK    ; YES
31 01265 070433 WTFUB:  DIA     2,FUB   ; SENSE BUTTON ARG
32 01266 147415         AND#    2,1,SNR ; IS ARG PRESSED?
33 01267 000764         JMP     WTNOK   ; NO - SENSE AGAIN
34 01270 152400 WTOK:   SUB     2,2
35 01271 071032         DOA     2,FUN   ; YES - TURN OFF INDICATOR ARG
36 01272 070433         DIA     2,FUB   ; WAIT TILL THE OPERATOR
37 01273 147414         AND#    2,1,SZR ; GETS HIS DIRTY FINGER
38 01274 000774         JMP     .-2     ; OFF THE BUTTON
39 01275 020413         LDA     0,WCH44
40 01276 006041         CCHAR
41 01277 006046         CDICL
42 01300 006044         CDISP
43 01301 004064         PROG            ; "ACTUAL PROGRAM NAME"
44 01302 006050         CHAAT
45 01303 020406         LDA     0,WACSAV
46 01304 002401         JMP     @WTORE  ; RETURN
47 01305 000000 WTORE:  0
48 01306 011000 WLOWL:  11*1000         ; LOW LIMIT CHAR * 2↑9
49 01307 040000 WHIGH:  40*1000         ; HIGH LIMIT CHAR * 2↑9
50 01310 000044 WCH44:  44
51 01311 000000 WACSAV: 0
52
53                ;INSERT ACTUAL PROGRAM NAME AFTER A LABEL "PROG:"
```

```
↑ 0023  .MAIN
01
02                ;ROUTINE READ SWITCHES.
03                ;SWITCHES ARE READ TO AC2. AC0 AND AC1 ARE SAVED.
04                ;THERE WILL BE A PAUSE FOR 3 SEC TO SEE IF MORE SWITCHES
05                ;ARE CHANGED. LAST SW REG CONTENT IN LSTSW.
06                ;SWITCH SETTINGS ARE REPORTED AT TTY/LPT.
07                ;CALL   CRESW
08                ;       RETURN
09
10 01312 040451 XRESW:  STA     0,SSWR0 ;SAVE AC'S
11 01313 044451         STA     1,SSWR1
12 01314 050451         STA     2,SSWR2 ;FOR PRINT INHIBIT RETURN
13 01315 054445         STA     3,RRESW ;SAVE RETURN
14 01316 024450         LDA     1,LSTSW ;LAST SW REG
15 01317 070477         READS   2       ;NEW SW REG
16 01320 132415         SUB#    1,2,SNR ;CHANGED ?
17 01321 000431         JMP     NRESW   ;NO, RETURN
18 01322 050444 ARESW:  STA     2,LSTSW ;YES, SEE IF MORE CHANGE
19 01323 006061         CWAIT           ;AFTER 3 SECONDS
20 01324 001236         SEC3
21 01325 024441         LDA     1,LSTSW
22 01326 070477         READS   2
23 01327 132414         SUB#    1,2,SZR ;CHANGED ?
24 01330 000772         JMP     ARESW   ;YES, WAIT MORE CHANGING
25 01331 050435         STA     2,LSTSW ;NO, CHANGE FINISHED
26 01332 026436         LDA     1,@XSTAC
27 01333 044436         STA     1,SSTAC ;SAVE PRINT INHIBIT SITU
28 01334 030436         LDA     2,KSTAC ;GET PRINT INHI INITIAL FOR PRINT
29 01335 052433         STA     2,@XSTAC ;SW REG WITHOUT FURTHER CALL CRESW
30 01336 006043         CCRLF
31 01337 006040         CMESS
32 01340 001522         MSWRG           ;SWITCHES: XXXXXX
33 01341 024425         LDA     1,LSTSW
34 01342 006052         CTOCT
35 01343 006424         JSR     @WBZOT  ;WAIT LPT,TTO
36 01344 030425         LDA     2,SSTAC
37 01345 052423         STA     2,@XSTAC ;RESTORE PRINT INHIBIT SITU
38 01346 034414         LDA     3,RRESW ;IF RRESW IS ALMOST EQUAL
39 01347 030421         LDA     2,XSTAC ;SETAC IT WAS PRINT INHIBIT
40 01350 156645         SUBOR   2,3,SNR ;WHO CALLED. DON'T RETURN
41 01351 000405         JMP     BRESW   ;BUT REPEAT THE OUTPUT ROUTINE
42 01352 020411 NRESW:  LDA     0,SSWR0
43 01353 024411         LDA     1,SSWR1 ;RESTORE AC'S
44 01354 030412         LDA     2,LSTSW ;NEW SW POSITION
45 01355 002405         JMP     @RRESW  ;RETURN
46
47 01356 014407 BRESW:  DSZ     SSWR2   ;REPEAT CALL OF ROUTINE
48 01357 020404         LDA     0,SSWR0 ;WHICH WAS INTERRUPTED
49 01360 024404         LDA     1,SSWR1 ;IN PRINT INHIBIT BY CRESW.
50 01361 002404         JMP     @SSWR2  ;RETURN TO MAIN PROG CALL.
51
52 01362 000000 RRESW:  0
53 01363 000000 SSWR0:  0
54 01364 000000 SSWR1:  0
55 01365 000000 SSWR2:  0
56 01366 000000 LSTSW:  0
57 01367 000437 WBZOT:  BZOUT
58 01370 000574 XSTAC:  SETAC
59 01371 000000 SSTAC:  0
60 01372 000403 KSTAC:  403             ;JMP NINHI, ASM VALUE FROM SETAC
61 01373 000000 LALOC:  0               ;LAST LOC IN MEM
62 01374 077777 LAPRG:  077777          ;LAST LOC FOR PROGRAM
63 01375 077635 BINFI:  77635           ;FIRST ADDR OF BINARY LOADER IF 32K.
64 01376 000177 BINLA:  177             ;CONSTANT FOR LAST ADDR IN MEM
65 01377 006702 XFITY:  FITYP
```

```
↑ 0024  .MAIN
01
02                ;SUBROUTINE TO RELOAD BOOTSTRAP LOADER AND BINARY LOADER.
03                ;DO USE IN HEAD OF ANY PROGRAM AS MANY THINGS ARE INITIALIZED
04                ;FOR LOADERS INFO SEE LOADER BELOW.
05                ;
06                ;CALL   START IN REBIN AFTER LOADING PROGRAM SELFSTARTING.
07
08 01400 020076 REBIN:  LDA     0,POWZE ;SET RESTART ADDR TO CELL ZERO
09 01401 040000         STA     0,0     ;FOR POWER RESTART.
10 01402 006465         JSR     @XMEXT  ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
11 01403 006463         JSR     @XMEND  ;STORE MEM END LOC IN HMEND=AC2=XX7600
12 01404 024772         LDA     1,BINLA ;AC1:=177
13 01405 147000         ADD     2,1     ;AC1:=LAST LOC IN MEM=XX7777
14 01406 044765         STA     1,LALOC ;STORE IT FOR PRINT
15 01407 125112         MOVL#   1,1,SZC ;MORE THAN 32K ?
16 01410 024764         LDA     1,LAPRG ;YES, AC1:=077777, LAST LOC FOR PROG
17 01411 034764         LDA     3,BINFI ;NO, LAST LOC IN FIRST 32K
18 01412 137400         AND     1,3     ;AC3:=FIRST ADDR LOADER
19 01413 030534         LDA     2,BINAD ;AC2:=ADDR OF BIN DATA
20 01414 021000 MODAT:  LDA     0,0,2
21 01415 041400         STA     0,0,3   ;MOVE DATA
22 01416 166415         SUB#    3,1,SNR ;LAST DATA ?
23 01417 000404         JMP     GETYP   ;YES
24 01420 151400         INC     2,2
25 01421 175400         INC     3,3     ;NEXT DATA ADDR
26 01422 000772         JMP     MODAT
27 01423 126400 GETYP:  SUB     1,1
28 01424 044742         STA     1,LSTSW ;SET LAST SW REG TO ALL ZERO
29 01425 062677         IORST           ;INSTEAD OF JSR @XFITY IN ST.ROUTINE.
30 01426 006073         CRESW           ;OUTPUT OF SWITCH SETTINGS
31 01427 000401         JMP     .+1     ;INSTEAD OF JSR @XPCPT IN ST.ROUTINE.
32 01430 006044         CDISP
33 01431 001501         MLLOC
34 01432 006040         CMESS
35 01433 001501         MLLOC           ;<15><12>LAST LOC. XX7777
36 01434 024737         LDA     1,LALOC
37 01435 006052         CTOCT
38 01436 006056         CDOCT
39 01437 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
40 01440 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
41 01441 006044         CDISP
42 01442 001510         MBILO
43 01443 006040         CMESS
44 01444 001510         MBILO           ;<15><12>BINARY LOADER OK
45 01445 006061         CWAIT
46 01446 001236         SEC3
47 01447 006046         CDICL
48 01450 006044         CDISP
49 01451 004064         PROG
50 01452 006043         CCRLF
51 01453 006040         CMESS
52 01454 004064         PROG            ;ACTUAL PROGRAM NAME
53 01455 006061         CWAIT
54 01456 001236         SEC3
55 01457 062701         DICP    0,1     ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
56 01460 020003         LDA     0,3     ;CELL 3 = 0 FOR HALT
57 01461 101004         MOV     0,0,SZR ;CELL 3 = 1 FOR
58 01462 002004         JMP     @4      ;SELFSTART ADDR 4
59 01463 002401         JMP     @.+1    ;CHOOSE YOUR OWN START ADDR,
60 01464 010620         SWISA           ;NORMAL HALT, BUT IF NO SWITCH PANEL ?
61
62 01465 006731 XPCPT:  PRTYP
63 01466 001716 XMEND:  GMEND
64 01467 002340 XMEXT:  TMEND
```

```
↑ 0025  .MAIN
01
02          000001          .TXTM 1          ;RDOS TEXT PACKING MODE
03
04                  MELOC:  .TXT !LOADING UNIT OFF! ;"LOADING UNIT OFF"
    01470 046117
    01471 040504
    01472 044516
    01473 043440
    01474 052516
    01475 044524
    01476 020117
    01477 043106
    01500 000000
05
06          000000          .TXTM 0          ;NORMAL TEXT PACKING MODE
07
08                  MLLOC:  .TXT !<15><12>LAST LOC. !       ;"<15><12>LAST LOC. "
    01501 005015
    01502 040514
    01503 052123
    01504 046040
    01505 041517
    01506 020056
    01507 000000
09
10                  MBILO:  .TXT !<15><12>BINARY LOADER OK!
    01510 005015
    01511 044502
    01512 040516
    01513 054522
    01514 046040
    01515 040517
    01516 042504
    01517 020122
    01520 045517
    01521 000000
11                          ;"<15><12>BINARY LOADER OK"
12
13                  MSWRG:  .TXT !SWITCHES: !       ;"SWITCHES: "
    01522 053523
    01523 052111
    01524 044103
    01525 051505
    01526 020072
    01527 000000
14
15                  MSAMS:  .TXT ! STARTADDR!       ;" STARTADDR"
    01530 051440
    01531 040524
    01532 052122
    01533 042101
    01534 051104
    01535 000000
16
17                  UNTIM:  .TXT !MISERABLE TIMING! ;"MISERABLE TIMING"
    01536 044515
    01537 042523
    01540 040522
    01541 046102
    01542 020105
    01543 044524
    01544 044515
    01545 043516
    01546 000000
```

```
↑ 0026  .MAIN
01
02                ;BINARY LOADER TS
03                ;ERRORHALT XX7752 FOR OVERWRITE LOADER OR CHECKSUM ERROR.
04                ;READYHALT XX7676 IF LOADED PROG ISN'T SELFSTARTING.
05                ;ERRORBLOCK=IGNORE BLOCK
06                ;REPEAT BLOCK=MULTIPLE DATA BLOCK
07                ;COUNT=WORD COUNT IN BLOCK
08
09 01547 001550 BINAD:  .+1               ;ADDR OF BIN LOADER DATA
10
11                ;SUBROUTINE TO BUILD A WORD IN AC2
12 01550 054425 BUILD:  STA     3,TEMP1 ;SAVE RETURN
13 01551 004406         JSR     GTCHR   ;GET FIRST BYTE
14 01552 171300         MOVS    3,2     ;PUT INTO LH OF AC2
15 01553 004404         JSR     GTCHR   ;GET NEXT BYTE
16 01554 173300         ADDS    3,2     ;FORM WORD IN AC2
17 01555 143000         ADD     2,0     ;ADD INTO CHECKSUM
18 01556 002417         JMP     @TEMP1  ;AND RETURN
19
20                ;READ A BYTE INTO AC3
21                ;IF SWITCH0=0 USE TELETYPE ELSE USE PTR
22 01557 054417 GTCHR:  STA     3,TEMP2 ;SAVE RETURN
23 01560 034417         LDA     3,SAVE  ;TEST WHICH DEVICE
24 01561 175103         MOVL    3,3,SNC
25 01562 000406         JMP     GTTTI   ;TTI
26 01563 063612         SKPDN   PTR     ;PTR
27 01564 000777         JMP     .-1
28 01565 074412         DIA     3,PTR   ;READ AND START
29 01566 060112         NIOS    PTR
30 01567 002407         JMP     @TEMP2  ;AND RETURN
31
32 01570 063610 GTTTI:  SKPDN   TTI
33 01571 000777         JMP     .-1
34 01572 074410         DIA     3,TTI   ;READ AND START
35 01573 060110         NIOS    TTI
36 01574 002402         JMP     @TEMP2  ;AND RETURN
37 01575 000000 TEMP1:  0
38 01576 000000 TEMP2:  0
39 01577 000000 SAVE:   0
40
41                ;TEST BLOCK TYPE
42 01600 125224 BTEST:  MOVZR   1,1,SZR ;1=START BLOCK (.END XX)
43 01601 000411         JMP     IGNOR   ;NO, IGNORE BLOCK
44 01602 101004         MOV     0,0,SZR ;TEST THE CHECKSUM
45 01603 000461         JMP     CHKER   ;ERROR
46 01604 030505         LDA     2,ADDRS ;GET ADDR
47 01605 062677         IORST           ;DO A RESET
48 01606 151113         MOVL#   2,2,SNC ;TEST BIT 0
49 01607 001000         JMP     0,2     ;0=START PROGRAM
50 01610 063077         HALT            ;1=HALT
51 01611 000777         JMP     .-1     ;DON'T PROCEED
52
53                ;IGNORE BLOCK
54 01612 004745 IGNOR:  JSR     GTCHR   ;READ UNTIL AN ALL
55 01613 020404         LDA     0,BC377 ;ONES BYTE IS SEEN
56 01614 116404         SUB     0,3,SZR ;IGNORING ERROR MESS
57 01615 000775         JMP     IGNOR
58 01616 000407         JMP     BLOCK   ;OK, GO INTO BLOCK MODE
59 01617 000377 BC377:  377
```

```
↑ 0027  .MAIN
01
02                  ;START OF PROGRAM BINARY LOADER TS
03 01620 062677 START:  IORST              ;RESET
04 01621 060477         READS   0          ;READ THE SWITCH REGISTER
05 01622 040755         STA     0,SAVE     ;AND SAVE IT FOR GTCHR
06 01623 060110         NIOS TTI           ;START BOTH READERS
07 01624 060112         NIOS PTR
08
09                  ;READ IN A BLOCK
10 01625 004732 BLOCK:  JSR     GTCHR      ;GET A BYTE
11 01626 171305         MOVS    3,2,SNR    ;AND TEST FOR NUL
12 01627 000776         JMP     BLOCK      ;YES, KEEP READING
13 01630 004727         JSR     GTCHR      ;OK. GET NEXT BYTE
14 01631 173300         ADDS    3,2        ;AND FORM COUNT. AC2:=# OF WORDS
15 01632 141000         MOV     2,0        ;SET CHECKSUM. AC0:=CHECKSUM TILL NOW
16 01633 145000         MOV     2,1        ;SET COUNTER
17 01634 004714         JSR     BUILD      ;GET ADDRESS
18 01635 050454         STA     2,ADDRS
19 01636 004712         JSR     BUILD      ;ADD IN THE CHECKSUM FROM TAPE
20 01637 125113         MOVL#   1,1,SNC    ;TEST BLOCK TYPE
21 01640 000740         JMP     BTEST      ;NOT A DATABLOCK
22 01641 044427         STA     1,COUNT    ;STORE WORD COUNT
23
24                  ;READ IN THE DATA BLOCK
25 01642 030734 DATA:   LDA     2,TEMP2    ;LAST STA IN TEMP2 WAS JSR RETURN
26 01643 034423         LDA     3,DIFF     ;ADDR 4 CELLS AFTER PROGRAM START:
27 01644 172400         SUB     3,2        ;AC2:=FIRST ADDR IN LOADER
28 01645 034444         LDA     3,ADDRS    ;ADDR IN WHICH TO STORE
29 01646 136400         SUB     1,3        ;ADD NEG WC TO CHECK SPACE
30 01647 172023         ADCZ    3,2,SNC    ;FOR WHOLE BLOCK
31 01650 000414         JMP     CHKER      ;NO, HALT THE LOADER
32 01651 030416         LDA     2,BC20     ;IF WC > 20 (OCTAL, NEG)
33 01652 147033         ADDZ#   2,1,SNC    ;IT IS A REPEAT BLOCK
34 01653 010415         ISZ     COUNT      ;WHERE WC IS ONE LESS THAN COUNT
35 01654 147022         ADDZ    2,1,SZC    ;IF REPEAT BLOCK SKIP NEXT TO READ DATA
36 01655 125113 STORE:  MOVL#   1,1,SNC    ;DON'T READ IN NEW DATA IF REPEAT BLOCK
37 01656 004672         JSR     BUILD
38 01657 052432         STA     2,@ADDRS
39 01660 010431         ISZ     ADDRS      ;NEXT ADDR
40 01661 010407         ISZ     COUNT      ;TEST COUNT
41 01662 000773         JMP     STORE      ;MORE DATA
42 01663 101004         MOV     0,0,SZR    ;TEST CHECKSUM
43 01664 063077 CHKER:  HALT               ;ERROR IN CHECKSUM, AC0=VALUE
44 01665 000740         JMP     BLOCK      ;OK,GET NEXT BLOCK
45 01666 000004 DIFF:   4
46 01667 000020 BC20:   20                 ;REPEAT BLOCKS HAVE WD>20
47 01670 000000 COUNT:  0
48
49                  ;BOOTSTRAP LOADER TS FOR PTR:   ENTER AT BSTRP
50 01671 126440 GET:    SUBO    1,1
51 01672 063612         SKPDN   12         ;10 FOR TTI
52 01673 000777         JMP     .-1
53 01674 060412         DIA     0,12       ;10 FOR TTI
54 01675 060112         NIOS    12         ;10 FOR TTI
55 01676 127100         ADDL    1,1
56 01677 127100         ADDL    1,1
57 01700 107003         ADD     0,1,SNC
58 01701 000771         JMP     GET+1
59 01702 001400         JMP     0,3
60 01703 060112 BSTRP:  NIOS    12         ;10 FOR TTI
61 01704 004765         JSR     GET
62 01705 044402         STA     1,.+2
63 01706 004763         JSR     GET
64 01707 000000         0
65 01710 000000         0                  ;FOR BOOTSTRAP
66 01711 000000 ADDRS:  0
67 01712 000706 BEND:   JMP     START      ;START OF BINARY LOADER
```

```
↑ 0028  .MAIN
01
02                              ;FIND THE TOP OF MEMORY (4K SEGMENTS)
03                              ;STORE HIGHEST USEABLE ADDRESS IN
04                              ;HMEND, PROTECTING THE BIN LOADER.
05                              ;THE CONTENTS IN MEMORY ARE LEFT UNCHANGED.
06                              ;
07                              ;CALL   JSR     GMEND
08                              ;       RETURN
09 01713 010000 DMEND:  10000           ;4K CONSTANT
10 01714 014000 EMEND:  14000           ;6K+1 ADDR
11 01715 004200 FMEND:  004200          ;2K+LOADER CONSTANT
12 01716 054451 GMEND:  STA     3,RSAMS
13 01717 030775         LDA     2,EMEND ;PASS 1 AND 2K SEGMENTS
14 01720 020773         LDA     0,DMEND
15 01721 035000 AMEND:  LDA     3,0,2   ;SAVE LOC CONTENT
16 01722 051000         STA     2,0,2   ;STORE INTO UPPER
17 01723 025000         LDA     1,0,2   ;GET IT BACK
18 01724 146404         SUB     2,1,SZR
19 01725 000405         JMP     BMEND
20 01726 055000         STA     3,0,2   ;RESTORE LOC CONTENT
21 01727 113020         ADDZ    0,2
22 01730 151013 JMEND:  MOV#    2,2,SNC ;CHECK LAST 64K/32K WITH L-SHIFT
23 01731 000770         JMP     AMEND
24 01732 024763 BMEND:  LDA     1,FMEND ;WENT TOO FAR
25 01733 132400         SUB     1,2     ;PROTECT LOADERS
26 01734 050074         STA     2,HMEND ;SET END MEM
27 01735 002432         JMP     @RSAMS
28
29                              ;ROUTINE TO PRINT WHICH STARTADDR SA OF MORE
30                              ;IS USED. INPUT:  SA IN AC1.
31                              ;CALL   CSAMS
32                              ;       RETURN
33 01736 054431 XSAMS:  STA     3,RSAMS
34 01737 044431         STA     1,SSAMS
35 01740 030431         LDA     2,CXLPT ;SET OUTPUT DEVICE ROUTINES
36 01741 052433         STA     2,@XXLPT ;TO WAIT FOR ONLINE DEVICES.
37 01742 030430         LDA     2,CXTTO
38 01743 052432         STA     2,@XXTTO
39 01744 030427         LDA     2,CDIS
40 01745 052431         STA     2,@XFDIS
41 01746 052431         STA     2,@XSDIS
42 01747 006046         CDICL
43 01750 006043         CCRLF
44 01751 024417         LDA     1,SSAMS
45 01752 152520         SUBZL   2,2     ;AC2:=1
46 01753 146400         SUB     2,1     ;SUBTRACT 1 FROM JSR ADDR
47 01754 006056         CDOCT
48 01755 006052         CTOCT           ;PRINT SA
49 01756 006044         CDISP
50 01757 001530         MSAMS
51 01760 006040         CMESS
52 01761 001530         MSAMS           ;XXXXXX STARTADDR
53 01762 006073         CRESW           ;OUTPUT OF SWITCH SETTINGS OFTEN HERE
54 01763 006040         CMESS           ;PRINT CR,LF WITH MESS TO WAIT
55 01764 010621         MCRLF           ;FOR DEVICE READY BEFORE IORST
56 01765 062677         IORST
57 01766 002401         JMP     @RSAMS
58 01767 000000 RSAMS:  0
59 01770 000000 SSAMS:  0
60 01771 001750 CXLPT:  1000.           ;TIMEOUT CONSTANTS FOR
61 01772 000454 CXTTO:  300.            ;OUTPUT DEVICES
62 01773 000050 CDIS:   40.
63 01774 000441 XXLPT:  XLPTT
64 01775 000445 XXTTO:  XTTOT
65 01776 001125 XFDIS:  FDIST
66 01777 001147 XSDIS:  SDIST
```

```
↑ 0029  .MAIN
01                              ;TABLE TO BE CHANGED TO CORRECT PRINTER ALPHABET.
02
03          002000 .LOC     2000
04          000010 .RDX     8
05
06                 TABLE:
07
08                 .TXTE?
09  02000  000006 <6><0>
10  02001  000006 <6><0>
11  02002  000006 <6><0>
12  02003  000006 <6><0>
13  02004  000006 <6><0>
14  02005  000006 <6><0>
15  02006  000006 <6><0>
16  02007  000006 <6><0>
17  02010  000006 <6><0>
18  02011  004400 <0><11>
19  02012  005000 <0><12>
20  02013  000006 <6><0>
21  02014  006000 <0><14>
22  02015  106400 <0><15>
23  02016  000006 <6><0>
24  02017  000006 <6><0>
25  02020  000006 <6><0>
26  02021  000006 <6><0>
27  02022  000006 <6><0>
28  02023  000006 <6><0>
29  02024  000006 <6><0>
30  02025  000006 <6><0>
31  02026  000006 <6><0>
32  02027  000006 <6><0>
33  02030  000006 <6><0>
34  02031  000006 <6><0>
35  02032  000006 <6><0>
36  02033  000006 <6><0>
37  02034  000006 <6><0>
38  02035  000006 <6><0>
39  02036  000006 <6><0>
40  02037  000006 <6><0>
41  02040  117400 <0><37>
42  02041  020400 <0><41>
43  02042  021000 <0><42>
44  02043  121400 <0><43>
45  02044  022000 <0><44>
46  02045  122400 <0><45>
47  02046  123000 <0><46>
48  02047  023400 <0><47>
49  02050  024000 <0><50>
50  02051  124400 <0><51>
51  02052  125000 <0><52>
52  02053  025400 <0><53>
53  02054  126000 <0><54>
54  02055  026400 <0><55>
55  02056  027000 <0><56>
56  02057  127400 <0><57>
57  02060  030000 <0><60>
58  02061  130400 <0><61>
59  02062  131000 <0><62>
60  02063  031400 <0><63>
61  02064  132000 <0><64>
62  02065  032400 <0><65>
63  02066  033000 <0><66>
64  02067  133400 <0><67>
65  02070  134000 <0><70>
```

```
↑ 0030   .MAIN
02 02071 034400 <0><71>
03 02072 035000 <0><72>
04 02073 135400 <0><73>
05 02074 036000 <0><74>
06 02075 136400 <0><75>
07 02076 137000 <0><76>
08 02077 037400 <0><77>
09 02100 140000 <0><100>
10 02101 040400 <0><101>
11 02102 041000 <0><102>
12 02103 141400 <0><103>
13 02104 042000 <0><104>
14 02105 142400 <0><105>
15 02106 143000 <0><106>
16 02107 043400 <0><107>
17 02110 044000 <0><110>
18 02111 144400 <0><111>
19 02112 145000 <0><112>
20 02113 045400 <0><113>
21 02114 146000 <0><114>
22 02115 046400 <0><115>
23 02116 047000 <0><116>
24 02117 147400 <0><117>
25 02120 050000 <0><120>
26 02121 150400 <0><121>
27 02122 151000 <0><122>
28 02123 051400 <0><123>
29 02124 152000 <0><124>
30 02125 052400 <0><125>
31 02126 053000 <0><126>
32 02127 153400 <0><127>
33 02130 154000 <0><130>
34 02131 054400 <0><131>
35 02132 055000 <0><132>
36 02133 155400 <0><133>
37 02134 056000 <0><134>
38 02135 156400 <0><135>
39 02136 157000 <0><136>
40 02137 057400 <0><137>
41 02140 006000 <0><14>
42 02141 040400 <0><101>
43 02142 041000 <0><102>
44 02143 141400 <0><103>
45 02144 042000 <0><104>
46 02145 142400 <0><105>
47 02146 143000 <0><106>
48 02147 043400 <0><107>
49 02150 044000 <0><110>
50 02151 144400 <0><111>
51 02152 145000 <0><112>
52 02153 045400 <0><113>
53 02154 146000 <0><114>
54 02155 046400 <0><115>
55 02156 047000 <0><116>
56 02157 147400 <0><117>
57 02160 050000 <0><120>
58 02161 150400 <0><121>
59 02162 151000 <0><122>
60 02163 051400 <0><123>
61 02164 152000 <0><124>
62 02165 052400 <0><125>
63 02166 053000 <0><126>
64 02167 153400 <0><127>
65 02170 154000 <0><130>
```

```
↑ 0031   .MAIN
01 02171 054400 <0><131>
02 02172 055000 <0><132>
03 02173 140000 <0><100>
04 02174 121400 <0><43>
05 02175 156400 <0><135>
06 02176 004400 <0><11>
07 02177 000006 <6><0>?
   02200 000000
08       000010 .RDX 8
09                      ;START ADDR FOR "HELP" PROGRAMS.
10
11       002201 .LOC 2201
12
13 02201 002403         SAPTB
14 02202 006777         JSR     @.-1    ;START MOVE PRINTER TABLE
15 02203 002260         SAMEX
16 02204 006777         JSR     @.-1    ;START SET TO 64K MODE, MEM SIZE ?
17 02205 002265         SAMNM
18 02206 006777         JSR     @.-1    ;START SET TO 32K MODE, MEM SIZE ?
19 02207 010667         EXMEM
20 02210 006777         JSR     @.-1    ;START EXAMINE MEMORY
21 02211 011005         DPMEM
22 02212 006777         JSR     @.-1    ;START DEPOSIT MEMORY
23
24       002221 .LOC 2221
25
26 02221 002314         LOADB
27 02222 006777         JSR     @.-1    ;START BINARY LOADER, READ FROM PTR/TTI
28
29              MLPTT:  .TXT !LPT TABLE!        ;"LPT TABLE"
   02223 050114
   02224 020124
   02225 040524
   02226 046102
   02227 000105
30
31                      ;ROUTINE TO MOVE PRINTER TABLE.
32                      ;INPUT: TABLE # IN AC2
33                      ;CALL:  JSR     MOPTB
34
35 02230 054415 MOPTB:  STA     3,MOPTR ;SAVE RETIRE
36 02231 034417         LDA     3,MOGTT ;ADDR OF POINTER
37 02232 157000         ADD     2,3     ;AC3:=TABLE ENTRY
38 02233 031400         LDA     2,0,3   ;AC2:=ADDR OF TABLE, START
39 02234 024413         LDA     1,MOLAD ;LAST STORE ADDR
40 02235 034411         LDA     3,MOFID ;FIRST STORE ADDR
41 02236 021000 MOREP:  LDA     0,0,2
42 02237 041400         STA     0,0,3   ;MOVE CHAR
43 02240 166415         SUB#    3,1,SNR ;LAST CHAR ?
44 02241 002404         JMP     @MOPTR  ;YES, RETIRE
45 02242 151400         INC     2,2
46 02243 175400         INC     3,3     ;NEXT CHAR ADDR
47 02244 000772         JMP     MOREP   ;REPEAT MOVE CHAR
48 02245 000000 MOPTR:  0               ;RETURN ADDR
49 02246 002000 MOFID:  2000            ;FIRST STORE ADDR
50 02247 002200 MOLAD:  2200            ;LAST STORE ADDR
51 02250 002250 MOGTT:  .               ;GET TABLE ADDR
52 02251 007542         PTAB1           ;ASCII TABLE ADDR
53 02252 007743         PTAB2           ;RC STANDARD 71/78 START .
54 02253 010144         PTAB3           ;RC STANDARD 71/78 START 0
55 02254 010345         PTAB4           ;PL1 TYPE 70 TABLE ADDR
56
57 02255 000004 ALPTT:  4               ;SUGGESTED ANSWER TABLE #
58 02256 000004 ULPTT:  4               ;UPPER LIMIT
59 02257 000001 LLPTT:  1               ;LOWER LIMIT
```

```
↑ 0032  .MAIN
01
02                          ;PROGRAMS TO SET MAX MEMORY LOCATION. THEY WILL SET
03                          ;MAX MEM LOC IN HMEND BOTH FOR 32K AND 64K MAX MODE.
04                          ;BUT 64K MAX (MEMORY EXTENSION) MODE WILL BE RESET
05                          ;AFTER EACH IORST. THEREFORE IF THE PROGRAMS SHOULD
06                          ;USE THIS MODE IT SHOULD BE MADE TO CHECK HMEND TO SEE
07                          ;IF IT IS BIGGER THAN 32K. IF SO YOU SHOULD FIRE THE
08                          ;DICP 0,1 COMMAND BEFORE USING MEMORY LOC BEYOND 32K.
09
10                          ;PROGRAM TO SET MAX MEM LOC TO 64K WORDS MODE.
11
12 02260 165000 SAMEX:  MOV     3,1
13 02261 006072         CSAMS           ;START ADDR MESSAGE
14 02262 004456         JSR     TMEND   ;TRY TO SET MEM EXT FLAG (RC3603 ONLY)
15 02263 000411         JMP     SAMMS
16 02264 151013 MCMEX:  MOV#    2,2,SNC
17
18                          ;PROGRAM TO SET MAX MEM LOC TO 32K WORDS MODE.
19
20 02265 165000 SAMNM:  MOV     3,1
21 02266 006072         CSAMS           ;START ADDR MESSAGE
22 02267 024404         LDA     1,MCMNM
23 02270 046423         STA     1,@MIMEX
24 02271 062677         IORST           ;RESET MEM EXT FLAG (RC3603 ONLY)
25 02272 000402         JMP     SAMMS
26 02273 151113 MCMNM:  MOVL#   2,2,SNC
27
28 02274 006416 SAMMS:  JSR     @IMEND  ;AC2=HMEND=XX7600
29 02275 006044         CDISP
30 02276 001501         MLLOC
31 02277 006040         CMESS
32 02300 001501         MLLOC           ;<15><12>LAST LOC. XX7777
33 02301 020074         LDA     0,HMEND
34 02302 026407         LDA     1,@SAMCO ;AC1:=177
35 02303 107000         ADD     0,1
36 02304 006056         CDOCT
37 02305 006052         CTOCT
38 02306 006047         CDATT
39 02307 002401         JMP     @.+1    ;PROGRAM FINISHED
40 02310 010620         SWISA           ;RESTART MAIN PROGRAM.
41 02311 001376 SAMCO:  BINLA
42 02312 001716 IMEND:  GMEND
43 02313 001730 MIMEX:  JMEND
44
45                          ;ROUTINE RESTORE BINARY LOADER AND
46                          ;START LOADING PTR/TTI, DEPENDING ON SWITCH 0.
47
48 02314 165000 LOADB:  MOV     3,1
49 02315 006072         CSAMS           ;START ADDR MESSAGE
50 02316 024755         LDA     1,MCMNM
51 02317 046774         STA     1,@MIMEX ;MAX 32K MODE
52 02320 062677         IORST           ;RESET MEM EXT FLAG
53 02321 006771         JSR     @IMEND  ;AC2=HMEND=0X7600
54 02322 026767         LDA     1,@SAMCO ;AC1:=177
55 02323 147000         ADD     2,1     ;LAST LOC = 0X7777
56 02324 036412         LDA     3,@LOADF ;FIRST ADDR LOADER
57 02325 137400         AND     1,3     ;IN THIS MEM
58 02326 032411         LDA     2,@LOADA ;ADDR OF BIN DATA
59 02327 021000 LOADR:  LDA     0,0,2
60 02330 041400         STA     0,0,3   ;MOVE DATA
61 02331 166415         SUB#    3,1,SNR ;LAST DATA ?
62 02332 001400         JMP     0,3     ;YES, START BINARY LOADER
63 02333 151400         INC     2,2
64 02334 175400         INC     3,3     ;NEXT DATA ADDR
65 02335 000772         JMP     LOADR
66 02336 001375 LOADF:  BINFI           ;FIRST ADDR BIN LOADER (32K)
67 02337 001547 LOADA:  BINAD           ;ADDR POINTER TO BIN DATA
```

```
↑ 0033  .MAIN
01
02                          ;ROUTINE TO CHECK FOR MORE THAN 32K WORDS MEMORY.
03                          ;(MEMORY EXTEND OPTION).
04
05 02340 054435 TMEND:  STA     3,RMEND
06 02341 062677         IORST               ;RESET MEM EXT FLAG
07 02342 062701         DICP    0,1         ;TRY TO SET MEM EXT FLAG (RC3603 ONLY).
08 02343 030433         LDA     2,OMEND     ;DON'T USE SKPDN 1 TO CHECK FLAG
09 02344 034433         LDA     3,PMEND     ;AS NOT IMPLEMENTED IN ALL CPU'S
10 02345 021000         LDA     0,0,2       ;SAVE LOWER LOC CONTENT
11 02346 040432         STA     0,QMEND
12 02347 051000         STA     2,0,2       ;STORE LOWER ADDR
13 02350 025400         LDA     1,0,3       ;SAVE UPPER LOC CONTENT
14 02351 044430         STA     1,SMEND
15 02352 055400         STA     3,0,3       ;STORE UPPER ADDR
16 02353 021400         LDA     0,0,3       ;GET IT BACK
17 02354 116414         SUB#    0,3,SZR
18 02355 000411         JMP     UMEND       ;NOT MORE THAN 32K BUT MEM EXT OPTION
19 02356 025000         LDA     1,0,2       ;MORE THAN 32K OR NOT MEM EXT OPTION
20 02357 132415         SUB#    1,2,SNR     ;UPPER STORED IN LOWER (15 BIT ADDR) ?
21 02360 000406         JMP     UMEND       ;NO, MEM EXT OPTION
22 02361 020417         LDA     0,QMEND     ;YES, NOT MEM EXT OPTION, MAX 32K
23 02362 041000         STA     0,0,2       ;RESTORE LOWER LOC CONTENT
24 02363 024710         LDA     1,MCMNM     ;SET 32K MAX
25 02364 046727         STA     1,@MIMEX
26 02365 002410         JMP     @RMEND      ;EXIT
27 02366 020412 UMEND:  LDA     0,QMEND
28 02367 041000         STA     0,0,2       ;RESTORE LOWER LOC CONTENT
29 02370 024411         LDA     1,SMEND
30 02371 045400         STA     1,0,3       ;RESTORE UPPER LOC CONTENT
31 02372 030672         LDA     2,MCMEX     ;SET 64K MAX
32 02373 052720         STA     2,@MIMEX
33 02374 002401         JMP     @RMEND      ;EXIT
34 02375 000000 RMEND:  0                   ;RETURN ADDR
35 02376 002402 OMEND:  VMEND               ;LOWER ADDR
36 02377 102402 PMEND:  @VMEND              ;UPPER ADDR 16 BIT, NOT INDIRECT
37 02400 000000 QMEND:  0                   ;SAVE LOWER CONTENT
38 02401 000000 SMEND:  0                   ;SAVE UPPER CONTENT
39 02402 000000 VMEND:  0                   ;WORK CELL LOWER ADDR
40
41                          ;PROGRAM TO GET NEW PRINTER TABLE.
42
43 02403 165000 SAPTB:  MOV     3,1
44 02404 006072         CSAMS               ;START ADDR MESSAGE
45 02405 006071 RLPTT:  CQUES
46 02406 002223         MLPTT               ;LPT TABLE
47 02407 002223         MLPTT
48 02410 002255         ALPTT               ;SUGGESTED ANSWER
49 02411 006053         CTDEC
50 02412 006057         CDDEC
51 02413 006105         CGTDC               ;READ ANSWER
52 02414 000402         JMP     .+2         ;SUGGESTED ACCEPTED
53 02415 000770         JMP     RLPTT       ;ERROR RETURN
54 02416 030075         LDA     2,DIGIN     ;ANSWER INPUT'ED
55 02417 024637         LDA     1,ULPTT     ;UPPER LIMIT
56 02420 020637         LDA     0,LLPTT     ;LOWER LIMIT
57 02421 132033         ADCZ#   1,2,SNC
58 02422 142032         ADCZ#   2,0,SZC     ;AC0=<AC2=<AC1 ?
59 02423 000762         JMP     RLPTT       ;OUTSIDE LIMITS
60 02424 004604         JSR     MOPTB       ;INPUT ACCEPTED, MOVE TABLE
61 02425 002401         JMP     @.+1        ;PROGRAM FINISHED
62 02426 010620         SWISA               ;RESTART MAIN PROGRAM
63
64
65                  ;TAPE 3A
66
67                  .EOT
```

```
  0034  .MAIN
01
02                          ;TAPE 3B
03
04 02427 054146 WQUES:  STA     3,RPASS
05 02430 006043 RPTTY:  CCRLF
06 02431 006040         CMESS
07 02432 004107         TXTTY           ;TTY SPEED
08 02433 024161         LDA     1,TTYSQ
09 02434 044075         STA     1,DIGIN ;STORE SUGGESTED ANSWER
10 02435 006053         CTDEC
11 02436 006040         CMESS
12 02437 004163         TXQUE           ;?
13 02440 006040         CMESS
14 02441 004161         TX2SP           ;2 SPACE
15 02442 006105         CGTDC           ;READ ANSWER
16 02443 000402         JMP     .+2     ;SUGGESTED ACCEPTED
17 02444 000764         JMP     RPTTY   ;ERROR RETURN
18 02445 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
19 02446 101212         MOVR#   0,0,SZC ;EVEN ?
20 02447 000761         JMP     RPTTY   ;NO
21 02450 024163         LDA     1,TTYUL
22 02451 030164         LDA     2,TTYLL
23 02452 122033         ADCZ#   1,0,SNC
24 02453 112032         ADCZ#   0,2,SZC ;AC2=<AC0=<AC1 ?
25 02454 000754         JMP     RPTTY   ;OUTSIDE LIMITS
26 02455 040162         STA     0,TTYSA ;INPUT ACCEPTED
27
28 02456 006043 RPBIT:  CCRLF
29 02457 006040         CMESS
30 02460 004114         TXBIT           ;NUMBER OF BITS
31 02461 024171         LDA     1,BITNQ
32 02462 044075         STA     1,DIGIN ;STORE SUGGESTED ANSWER
33 02463 006053         CTDEC
34 02464 006040         CMESS
35 02465 004163         TXQUE           ;?
36 02466 006040         CMESS
37 02467 004161         TX2SP           ;2 SPACE
38 02470 006105         CGTDC           ;READ ANSWER
39 02471 000402         JMP     .+2     ;SUGGESTED ACCEPTED
40 02472 000764         JMP     RPBIT   ;ERROR RETURN
41 02473 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
42 02474 024173         LDA     1,BITUL
43 02475 030174         LDA     2,BITLL
44 02476 122033         ADCZ#   1,0,SNC
45 02477 112032         ADCZ#   0,2,SZC ;AC2=<AC0=<AC1 ?
46 02500 000756         JMP     RPBIT   ;OUTSIDE LIMITS
47 02501 040172         STA     0,BITNA ;INPUT ACCEPTED
48
49 02502 006043 RPCPU:  CCRLF
50 02503 006040         CMESS
51 02504 003732         TXCPU           ;CPU TYPE
52 02505 024165         LDA     1,CPUTQ
53 02506 044075         STA     1,DIGIN ;STORE SUGGESTED ANSWER
54 02507 006054         CTZOC
55 02510 006040         CMESS
56 02511 004163         TXQUE           ;?
57 02512 006040         CMESS
58 02513 004161         TX2SP           ;2 SPACE
59 02514 006104         CGTOK           ;READ ANSWER
60 02515 000402         JMP     .+2     ;SUGGESTED ACCEPTED
61 02516 000764         JMP     RPCPU   ;ERROR RETURN
62 02517 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
63 02520 024167         LDA     1,CPUUL
64 02521 030170         LDA     2,CPULL
65 02522 122033         ADCZ#   1,0,SNC
66 02523 112032         ADCZ#   0,2,SZC ;AC2=<AC0=<AC1 ?
67 02524 000756         JMP     RPCPU   ;OUTSIDE LIMITS
```

```
↑ 0035  .MAIN
01
02 02525 030203          LDA     2,TBLCN ;TABEL POINTER
03 02526 113000          ADD     0,2     ;AC0= CPU TYPE #
04 02527 025367          LDA     1,-11,2 ;NORMAL COMPENSATE CONST
05 02530 125015          MOV#    1,1,SNR
06 02531 000751          JMP     RPCPU   ;CPU NOT KNOWN
07 02532 044155          STA     1,COMPN
08 02533 030220          LDA     2,TBLCS ;TABEL POINTER
09 02534 113000          ADD     0,2
10 02535 025367          LDA     1,-11,2 ;SKIP COMPENSATE CONST
11 02536 044156          STA     1,COMPS
12 02537 030235          LDA     2,TBLCI ;TABEL POINTER
13 02540 113000          ADD     0,2
14 02541 025367          LDA     1,-11,2 ;INTR COMPENSATE CONST
15 02542 044153          STA     1,COMPI
16 02543 030252          LDA     2,TBLCC ;TABEL POINTER
17 02544 113000          ADD     0,2
18 02545 025367          LDA     1,-11,2 ;COUNT START COMPENSATE CONST
19 02546 044154          STA     1,COMPC
20 02547 030267          LDA     2,TBLCT ;TABEL POINTER
21 02550 113000          ADD     0,2
22 02551 025367          LDA     1,-11,2 ;POINTER TO CORRECT TIMES
23 02552 044304          STA     1,CTIMP
24 02553 040166          STA     0,CPUTA ;INPUT ACCEPTED
25 02554 042305          STA     0,@ICPN
26 02555 024172          LDA     1,BITNA ;# OF BITS/CHAR
27 02556 030140          LDA     2,D1000
28 02557 006066          MULTI
29 02560 030137          LDA     2,D500
30 02561 006066          MULTI           ;# OF BITS * 500000
31 02562 030162          LDA     2,TTYSA ;BAUD SPEED
32 02563 006070          DIVID
33 02564 044175          STA     1,HCHSP ;AC1= # OF USEC/0,5 CHAR
34 02565 024201          LDA     1,D38K
35 02566 030175          LDA     2,HCHSP
36 02567 006067          DIVIS           ;AC1= # OF CHAR TO SEND
37 02570 130400          NEG     1,2     ;AC2= -# OF CHAR TO SEND
38 02571 050176          STA     2,NOCHA
39 02572 131120          MOVZL   1,2     ;AC2= # OF CHAR TO SEND * 2
40 02573 024175          LDA     1,HCHSP ;# OF USEC/0,5 CHAR
41 02574 006066          MULTI           ;AC2 WAS # OF CHAR TO SEND * 2
42 02575 030133          LDA     2,D10
43 02576 006070          DIVID
44 02577 044177          STA     1,SPFAK ;FAKTOR FOR 1 CHAR TIME * # OF CHAR
45 02600 024311          LDA     1,TSTRP ;NOT EXACTLY 100 MSEC AS BY RTC.
46 02601 030162          LDA     2,TTYSA ;CALCULATE THE # IN SPEED STRAP
47 02602 006067          DIVIS           ;(SPEED=9600/N+1), #=N+1
48 02603 102520          SUBZL   0,0     ;AC0=1
49 02604 106400          SUB     0,1     ;SUBTRACT THE DIVI ROUND UP
50 02605 030162          LDA     2,TTYSA ;BAUD SPEED
51 02606 006066          MULTI           ;AC1=STRAP # * BAUD = ABOUT 9600
52 02607 030177          LDA     2,SPFAK ;100 MSEC RTC/TTY SPEED FAKTOR
53 02610 006066          MULTI           ;FOR #CHAR * THEORY CHAR SPEED
54 02611 030312          LDA     2,CSTRP ;CORRECT MAX BAUD VALUE FOR TTC 705
55 02612 006070          DIVID           ;AC1=CORRECTED SPFAK
56 02613 032305          LDA     2,@ICPN ;ACTUAL CPU #
57 02614 034313          LDA     3,TTTYP ;NOVA 1200 #, WITH CORRECT TTY XTAL.
58 02615 156415          SUB#    2,3,SNR
59 02616 024177          LDA     1,SPFAK ;USE SPFAK IF NOVA 1200
60 02617 044200          STA     1,SPFCO ;ELSE USE CORRECTED SPFAK FOR TTC 705
61 02620 024153          LDA     1,COMPI ;INTR CODE COMPENSATION
62 02621 030176          LDA     2,NOCHA ;-# OF CHAR TO SEND
63 02622 150000          COM     2,2     ;# OF CHAR - 1
64 02623 006066          MULTI           ;COMPI * # OF CHAR-1
```

```
↑ 0036  .MAIN
01
02 02624 030154          LDA     2,COMPC ;COUNT START CODE COMPENSATION
03 02625 147000          ADD     2,1     ;INTR + START CODE COMPENSATION
04 02626 044160          STA     1,COMPR
05 02627 002146          JMP     @RPASS
06
07                TXHEAD: .TXT !INSTRUCTION EXECUTION TIMES IN NANOSECONDS<15><12>
   02630 047111
   02631 052123
   02632 052522
   02633 052103
   02634 047511
   02635 020116
   02636 054105
   02637 041505
   02640 052125
   02641 047511
   02642 020116
   02643 044524
   02644 042515
   02645 020123
   02646 047111
   02647 047040
   02650 047101
   02651 051517
   02652 041505
   02653 047117
   02654 051504
   02655 005015
08 02656 042522 REFERENCE:<11>  RTC<11>  TTY<11>  EXPECTED!
   02657 042506
   02660 042522
   02661 041516
   02662 035105
   02663 020011
   02664 051040
   02665 041524
   02666 020011
   02667 052040
   02670 054524
   02671 020011
   02672 042440
   02673 050130
   02674 041505
   02675 042524
   02676 000104
09
10 02677 165000 XANSW:  MOV     3,1     ;PRINT START ADDR
11 02700 006072         CSAMS
12 02701 034402         LDA     3,RENOQ
13 02702 001401         JMP     1,3     ;RESTART, NO QUESTIONS, LAST ANSWERS
14
15 02703 002705 RENOQ:  XTEST+1 ;RESTART NO QUESTION, ASM VALUE TO POINT FOR
16                              ;MAX # OF QUES IN CASE OF START BEFORE ANY QUES
17                              ;IS ANSWERED, DON'T FORGET HOW USED: JMP 1,RENOQ
```

```
↑ 0037  .MAIN
01
02 02704 165000 XTEST:  MOV     3,1       ;START OF TESTPROGRAM
03 02705 006072         CSAMS             ;PRINT START ADDR
04 02706 006307         JSR     @YQUES    ;QUESTIONS
05 02707 004401         JSR     .+1       ;NO QUESTION RESTART ADDR
06 02710 054773         STA     3,RENOQ
07 02711 102000         ADC     0,0       ;AC0:=177777
08 02712 040114         STA     0,FSWIT   ;FIRST ERROR FLAG
09 02713 040115         STA     0,PSWIT   ;DON'T PRINT FLAG
10 02714 102400         SUB     0,0       ;AC0:=0
11 02715 040152         STA     0,PASSN   ;PASS NUMBER
12 02716 020150         LDA     0,PASS1
13 02717 040147         STA     0,PASSC   ;NUMBER OF RUNS
14 02720 004427 NCAL0:  JSR     BEGIN     ;CALIBRATE WITH RTC/TTO
15 02721 004525 NTES0:  JSR     BEG2      ;TEST INSTR TIMES
16 02722 004461         JSR     PASSA
17 02723 000776         JMP     NTES0
18 02724 000774         JMP     NCAL0     ;10 PASS FINISHED
19
20 02725 165000 XLIST:  MOV     3,1       ;START OF PRINTPROGRAM
21 02726 006072         CSAMS             ;PRINT START ADDR
22 02727 006307         JSR     @YQUES    ;QUESTIONS
23 02730 004401         JSR     .+1       ;NO QUESTION RESTART ADDR
24 02731 054752         STA     3,RENOQ
25 02732 020142         LDA     0,CH14
26 02733 006041         CCHAR             ;FF ON LPT
27 02734 102400         SUB     0,0       ;AC0:=0
28 02735 040114         STA     0,FSWIT   ;NOT FIRST ERROR FLAG
29 02736 040115         STA     0,PSWIT   ;PRINT FLAG
30 02737 020151         LDA     0,PASS2
31 02740 040147         STA     0,PASSC   ;NUMBER OF RUNS
32 02741 004406 NCAL1:  JSR     BEGIN     ;CALIBRATE WITH RTC/TTO
33 02742 004504 NTES1:  JSR     BEG2      ;PRINT INSTR TIMES
34 02743 004457         JSR     PASSB
35 02744 000776         JMP     NTES1
36 02745 002401         JMP     @.+1      ;2 PASS FINISHED
37 02746 010620         SWISA             ;RESTART MAIN PROGRAM
38
39 02747 054110 BEGIN:  STA     3,RTEST   ;NEW CALIBRATION
40 02750 063511         SKPBZ   XTTO
41 02751 000777         JMP     .-1       ;WAIT FOR TTO TO FINISH QUESTION
42 02752 062677         IORST             ;CLEAR INTR AND RTC
43 02753 152520         SUBZL   2,2       ;AC2:=1  ;RTC 100 MSEC
44 02754 071014         DOA     2,XRTC
45 02755 060114         NIOS    XRTC
46 02756 004454         JSR     TIMER     ;SYNCHRONIZE
47 02757 060114         NIOS    XRTC
48 02760 004452         JSR     TIMER     ;CALIBRATE ON
49 02761 040121         STA     0,CALIR   ;100 MS CLOCK.
50 02762 060214         NIOC    XRTC      ;CLEAR INTR
51 02763 060177         INTEN
52 02764 126000         ADC     1,1       ;AC1:=-1, -# OF CHARS TO SEND
53 02765 152400         SUB     2,2       ;AC2:=0 ;TTO NUL CHAR
54 02766 071011         DOA     2,XTTO    ;BUFFER AFTER TRANSMIT=ALL STARTBITS=1
55 02767 060111         NIOS    XTTO
56 02770 004442         JSR     TIMER     ;SYNCHRONIZE
57 02771 024176         LDA     1,NOCHA   ;-# OF CHARS TO SEND
58 02772 071011         DOA     2,XTTO    ;AC2 STILL NUL CHAR
59 02773 060111         NIOS    XTTO      ;CLEAR INTR AND START
60 02774 060177         INTEN
61 02775 004435         JSR     TIMER     ;CALIBRATE ON TTO CLOCK
62 02776 040123         STA     0,CALIC   ;RETURN HERE FROM INTERRUPT SERVICE
63 02777 024160         LDA     1,COMPR   ;INTR + START CODE COMPENSATION
64 03000 107000         ADD     0,1
65 03001 044122         STA     1,CALIT   ;CALIC COMPENSATED
66 03002 002110         JMP     @RTEST
```

```
↑ 0038  .MAIN
01
02 03003 054146 PASSA:  STA     3,RPASS
03 03004 014147         DSZ     PASSC   ;PASS COUNT DOWN
04 03005 002146         JMP     @RPASS
05 03006 010146         ISZ     RPASS
06 03007 020150         LDA     0,PASS1 ;SETUP A NEW 10 PASSES
07 03010 040147         STA     0,PASSC
08 03011 006043         CCRLF
09 03012 006040         CMESS
10 03013 004161         TX2SP           ;2 SPACE
11 03014 010152         ISZ     PASSN   ;COUNT PASS NO
12 03015 024152         LDA     1,PASSN
13 03016 006053         CTDEC
14 03017 006040         CMESS
15 03020 004075         TXPSM           ;.PASS OF 10 RUNS<15><12>
16 03021 002146         JMP     @RPASS
17
18 03022 054146 PASSB:  STA     3,RPASS
19 03023 006043         CCRLF
20 03024 014147         DSZ     PASSC   ;PASS COUNT DOWN
21 03025 002146         JMP     @RPASS
22 03026 010146         ISZ     RPASS
23 03027 020142         LDA     0,CH14
24 03030 006041         CCHAR           ;FF ON LPT
25 03031 002146         JMP     @RPASS
26
27 03032 102000 TIMER:  ADC     0,0     ;RECORD THE TIME TO FINISH
28 03033 101400         INC     0,0
29 03034 063614         SKPDN   XRTC    ;IF TIMING ON TTO
30 03035 000776         JMP     .-2     ;WAIT HERE TO LAST INTR
31 03036 001400         JMP     0,3
32
33 03037 054146 PHEAD:  STA     3,RPASS
34 03040 006043         CCRLF
35 03041 006043         CCRLF
36 03042 006040         CMESS
37 03043 002630         TXHEAD
38 03044 006043         CCRLF
39 03045 002146         JMP     @RPASS
40
41 03046 054110 BEG2:   STA     3,RTEST ;A NEW PASS WITH SAME CALIBRATION
42 03047 020115         LDA     0,PSWIT
43 03050 101004         MOV     0,0,SZR
44 03051 000402         JMP     BEG1    ;DON'T PRINT HEADER
45 03052 004765         JSR     PHEAD
46 03053 020126 BEG1:   LDA     0,CITAB ;INITIALIZE INSTR
47 03054 040024         STA     0,IDX4  ;TO BE TIMED.
48 03055 022024 BEG:    LDA     0,@IDX4
49 03056 101005         MOV     0,0,SNR ;END OF TABLE ?
50 03057 002110         JMP     @RTEST  ;YES, RETURN
51 03060 030127         LDA     2,IBUFF ;FILL A 1K BUFFER
52 03061 034130         LDA     3,FBUFF ;WITH INSTRUCTIONS.
53 03062 041000         STA     0,0,2
54 03063 151400         INC     2,2
55 03064 156404         SUB     2,3,SZR
56 03065 000774         JMP     .-4
```

```
↑ 0039  .MAIN
01
02 03066 176400 INIT1:  SUB     3,3      ;INIT ISZ, DSZ START IN 0 (10000 TIMES)
03 03067 054144          STA     3,CWORK  ;INIT FOR @ TEST, AC3=0, STOP @ CHAIN
04 03070 034131          LDA     3,ADINC
05 03071 054025          STA     3,IDX5   ;INCREMENT 4000 TIMES IS POSSIBLE
06 03072 030132          LDA     2,ADDEC
07 03073 050030          STA     2,DDX0   ;DECREMENT 4000 TIMES IS POSSIBLE
08 03074 034024          LDA     3,IDX4   ;C(3) POINTER TO INSTR UNDER TEST
09 03075 021431          LDA     0,ISKPT-ITABL,3 ;SKIP FLAG
10 03076 024134          LDA     1,N10
11 03077 101014          MOV#    0,0,SZR  ;THE INSTR WILL SKIP ?
12 03100 024135          LDA     1,N20    ;YES, DOUBLE # OF BUFFER RUNS
13 03101 044202          STA     1,TENO   ;# OF EXECUTIONS OF BUFFER
14 03102 024155          LDA     1,COMPN  ;COMPENSATE CONST FOR ADM CODE
15 03103 101014          MOV#    0,0,SZR  ;THE INSTR WILL SKIP ?
16 03104 024156          LDA     1,COMPS  ;YES, MORE ADM CODE AFTER BUFF
17 03105 044157          STA     1,COMPA  ;ACTUAL COMP FOR ADM CODE.
18 03106 060114          NIOS    XRTC
19 03107 004723          JSR     TIMER    ;SYNCHRONIZE
20 03110 102400          SUB     0,0      ;AC0:=0
21 03111 034127          LDA     3,IBUFF
22 03112 060114          NIOS    XRTC     ;ENABLE RTC
23 03113 001400          JMP     0,3      ;EXIT TO BUFFER
24
25 03114 176400 INIT2:  SUB     3,3      ;SEE INIT1 FOR COMMENTS
26 03115 054144          STA     3,CWORK
27 03116 034131          LDA     3,ADINC
28 03117 054025          STA     3,IDX5
29 03120 034132          LDA     3,ADDEC
30 03121 054030          STA     3,DDX0
31 03122 034024          LDA     3,IDX4
32 03123 021431          LDA     0,ISKPT-ITABL,3
33 03124 024134          LDA     1,N10
34 03125 101014          MOV#    0,0,SZR
35 03126 024135          LDA     1,N20
36 03127 044202          STA     1,TENO
37 03130 024155          LDA     1,COMPN
38 03131 101014          MOV#    0,0,SZR
39 03132 024156          LDA     1,COMPS
40 03133 044157          STA     1,COMPA  ;ONLY HERE IF RTC NOT USED.
41 03134 063511          SKPBZ   XTTO
42 03135 000777          JMP     .-1      ;WAIT FOR TTO TO FINISH LAST INSTR
43 03136 060211          NIOC    XTTO
44 03137 060217          NIOC    XLPT     ;CLEAR INTR
45 03140 060214          NIOC    XRTC
46 03141 060177          INTEN
47 03142 126000          ADC     1,1      ;AC1:=-1, -# OF CHARS TO SEND
48 03143 152400          SUB     2,2      ;AC2:=0 ;TTO NUL CHAR
49 03144 071011          DOA     2,XTTO
50 03145 060111          NIOS    XTTO
51 03146 004664          JSR     TIMER    ;SYNCHRONIZE
52 03147 102400          SUB     0,0      ;AC0:=0
53 03150 024176          LDA     1,NOCHA  ;-# OF CHARS TO SEND
54 03151 034127          LDA     3,IBUFF
55 03152 071011          DOA     2,XTTO   ;LOAD NUL CHAR FOR START/STOP BITS
56 03153 060111          NIOS    XTTO
57 03154 060177          INTEN
58 03155 001400          JMP     0,3      ;EXIT TO BUFFER
59
60 03156 054145 GCOTI:  STA     3,RCOTI  ;ROUTINE TO GET CORRECT TIME
61 03157 034024          LDA     3,IDX4   ;AC3=POINTER TO INSTR UNDER TEST
62 03160 030540          LDA     2,ITABE
63 03161 156400          SUB     2,3      ;AC3=INSTR # IN TABLE
64 03162 030304          LDA     2,CTIMP  ;TABLE START FOR THIS CPU
65 03163 157000          ADD     2,3      ;+ # THIS INSTR
66 03164 031400          LDA     2,0,3    ;AC2=CORRECT TIME
67 03165 002145          JMP     @RCOTI   ;RETURN
```

```
↑ 0040  .MAIN
01
02 03166 004644 TINSR:  JSR     TIMER   ;TIME RTC FOR REST TIME
03 03167 063614         SKPDN   XRTC    ;TTO OR RTC ?
04 03170 000431         JMP     TINST   ;TTO
05 03171 024121         LDA     1,CALIR ;AC0=REST TIME
06 03172 106400         SUB     0,1     ;C(1)=10K INSTR AND ADM CODE TIME
07 03173 020157         LDA     0,COMPA ;ADM CODE COMPENSATION
08 03174 106400         SUB     0,1     ;AC1=10K INSTR TIME
09 03175 030141         LDA     2,D10K
10 03176 006066         MULTI
11 03177 030121         LDA     2,CALIR ;C(1)=TIME PER INSTR
12 03200 006070         DIVID           ;IN NANO SECONDS
13
14 03201 044124 RLOOK:  STA     1,TIMRT ;AC1=ACTUAL TIME
15 03202 004754         JSR     GCOTI   ;AC2=CORRECT TIME
16 03203 024111         LDA     1,ITTOL ;TOLERANCE IN %
17 03204 006066         MULTI
18 03205 030136         LDA     2,D100
19 03206 006070         DIVID
20 03207 044112         STA     1,ITNSE ;TOLERANCE IN NSEC
21 03210 004746         JSR     GCOTI   ;AC2=CORRECT TIME
22 03211 024124         LDA     1,TIMRT ;AC1=ACTUAL TIME
23 03212 132423         SUBZ    1,2,SNC ;THEORY - ACTUAL
24 03213 150400         NEG     2,2     ;AC2= + DIFFERENCE
25 03214 020112         LDA     0,ITNSE ;AC0= XX NANO SECONDS
26 03215 112443         SUBO    0,2,SNC ;CARRY = 0 MEANS ERROR
27 03216 102400         SUB     0,0     ;AC0:=0 ;PRINT LINE
28 03217 040113         STA     0,ESWIT ;ERROR FLAG
29 03220 000674         JMP     INIT2   ;TEST WITH TTO TOO
30
31 03221 024160 TINST:  LDA     1,COMPR ;AC0=REST TIME, AC1=COMPENSATION
32 03222 123000         ADD     1,0     ;AC0=CORRECTED REST TIME
33 03223 024122         LDA     1,CALIT ;AC1=CORRECTED CALIR
34 03224 106400         SUB     0,1     ;AC1=10K INSTR AND ADM CODE TIME
35 03225 020157         LDA     0,COMPA ;ADM CODE COMPENSATION
36 03226 106400         SUB     0,1     ;AC1=10K INSTR TIME
37 03227 030200         LDA     2,SPFCO ;10000 FAKTOR COMP FOR # OF CHAR
38 03230 006066         MULTI
39 03231 030122         LDA     2,CALIT ;AC1=TIME PER INSTR
40 03232 006070         DIVID           ;IN NANO SECONDS
41
42 03233 044125 TLOOK:  STA     1,TIMTT ;C(1)=ACTUAL TIME
43 03234 004722         JSR     GCOTI   ;AC2=CORRECT TIME
44 03235 132423         SUBZ    1,2,SNC ;THEORY-ACTUAL
45 03236 150400         NEG     2,2     ;C(2)=+DIFFERENCE
46 03237 020112         LDA     0,ITNSE ;C(0)=XX NANO SECONDS
47 03240 112440         SUBO    0,2     ;CARRY = 0 MEANS ERROR
48 03241 030113         LDA     2,ESWIT ;ERROR FLAG FROM RTC PART
49 03242 151015         MOV#    2,2,SNR ;ERROR ?
50 03243 000406         JMP     PERRO   ;YES
51 03244 030115         LDA     2,PSWIT ;0 FOR PRINT IN LIST PROG
52 03245 151016         MOV#    2,2,SEZ
53 03246 000607         JMP     BEG
54 03247 151012         MOV#    2,2,SZC ;ERROR ?
55 03250 000407         JMP     PINST   ;NO, BUT LIST
56 03251 030114 PERRO:  LDA     2,FSWIT ;YES
57 03252 151015         MOV#    2,2,SNR ;FIRST ERROR ?
58 03253 000404         JMP     PINST   ;NO
59 03254 152400         SUB     2,2     ;AC2:=0
60 03255 050114         STA     2,FSWIT ;NOT FIRST ERROR FLAG
61 03256 006310         JSR     @IHEAD  ;PRINT HEADER
```

```
↑ 0041  .MAIN
01
02 03257 034024 PINST:  LDA     3,IDX4  ;AC3=POINTER TO INSTR UNDER TEST
03 03260 035461         LDA     3,IMEST-ITABL,3
04 03261 054403         STA     3,.+3
05 03262 006043         CCRLF
06 03263 006040         CMESS           ;MESSAGE ABOUT
07 03264 000000         0               ;INSTRUCTION TIMED.
08 03265 024124         LDA     1,TIMRT
09 03266 006053         CTDEC           ;PRINT THE TIME FOR RTC
10 03267 024125         LDA     1,TIMTT
11 03270 006053         CTDEC           ;PRINT THE TIME FOR TTO
12 03271 034024         LDA     3,IDX4  ;AC3=POINTER TO INSTR UNDER TEST
13 03272 030426         LDA     2,ITABE
14 03273 156400         SUB     2,3     ;INSTR # IN TABLE
15 03274 030304         LDA     2,CTIMP ;TABLE START FOR THIS CPU
16 03275 157000         ADD     2,3     ;+#
17 03276 025400         LDA     1,0,3   ;AC1=CORRECT TIME
18 03277 006053         CTDEC           ;PRINT THE CORRECT TIME.
19 03300 002306         JMP     @IBEG
20
21                      ;INTERRUPT SERVICE ROUTINE, ONLY USED FOR TTO AND POWER.
22
23 03301 063777 SERINT: SKPDZ   CPU     ;POWER INTR. ?
24 03302 000407         JMP     POWIN   ;YES
25 03303 125405         INC     1,1,SNR ;AC1:= -# OF CHARS STILL TO SEND
26 03304 000411         JMP     DEVIN   ;LAST CHAR SENT.
27 03305 071011         DOA     2,XTTO  ;AC2 STILL NUL CHAR
28 03306 060111         NIOS    XTTO    ;START NEXT CHAR, CLEAR DONE
29 03307 060177         INTEN           ;ENABLE INTERRUPT
30 03310 002000         JMP     @0      ;INTERRUPT RETURN
31 03311 020076 POWIN:  LDA     0,POWZE ;SET RESTART ADDR TO CELL ZERO
32 03312 040000         STA     0,0     ;FOR POWER RESTART MONITOR
33 03313 063077         HALT            ;POWER INTERRUPT
34 03314 000777         JMP     .-1     ;DON'T PROCEED
35 03315 024076 DEVIN:  LDA     1,POWZE ;SET RESTART ADDR TO CELL ZERO
36 03316 044000         STA     1,0     ;FOR POWER RESTART WITHOUT INTR.
37 03317 001400         JMP     0,3     ;RETURN AS FROM TIMER
38
39 03320 003321 ITABE:  .+1             ;POINTER TO TABLE
40 03321 101000 ITABL:  MOV     0,0     ;INSTR TO BE TIMED
41 03322 101100         MOVL    0,0
42 03323 101300         MOVS    0,0
43 03324 103000         ADD     0,0
44 03325 103400         AND     0,0
45 03326 020144         LDA     0,CWORK
46 03327 040144         STA     0,CWORK
47 03330 010144         ISZ     CWORK
48 03331 014144         DSZ     CWORK
49 03332 000401         JMP     .+1
50 03333 004401         JSR     .+1
51 03334 022144         LDA     0,@CWORK
52 03335 022025         LDA     0,@IDX5
53 03336 022030         LDA     0,@DDX0
54 03337 022143         LDA     0,@WOPNT
55 03340 060000         NIO     0
56 03341 060100         NIOS    0
57 03342 060400         DIA     0,0
58 03343 061000         DOA     0,0
59 03344 061477         INTA    0
60 03345 063400         SKPBN   0
61 03346 063500         SKPBZ   0
62 03347 101004         MOV     0,0,SZR
63 03350 101001         MOV     0,0,SKP
64 03351 000000         0
```

```
↑ 0042  .MAIN
01
02 03352 000000 ISKPT:  0               ;INSTR WILL SKIP (-1) OR NOT (0).
03 03353 000000          0
04 03354 000000          0
05 03355 000000          0
06 03356 000000          0
07 03357 000000          0
08 03360 000000          0
09 03361 000000          0
10 03362 000000          0
11 03363 000000          0
12 03364 000000          0
13 03365 000000          0
14 03366 000000          0
15 03367 000000          0
16 03370 000000          0
17 03371 000000          0
18 03372 000000          0
19 03373 000000          0
20 03374 000000          0
21 03375 000000          0
22 03376 000000          0
23 03377 177777          -1
24 03400 177777          -1
25 03401 177777          -1
26
27 03402 004164 IMEST:  I1              ;TEXT MESSAGE TABLE
28 03403 004326          I18
29 03404 004333          I19
30 03405 004171          I2
31 03406 004176          I3
32 03407 004203          I4
33 03410 004212          I5
34 03411 004221          I6
35 03412 004227          I7
36 03413 004235          I8
37 03414 004242          I9
38 03415 004247          I10
39 03416 004256          I11
40 03417 004265          I12
41 03420 004274          I13
42 03421 004340          I20
43 03422 004344          I21
44 03423 004303          I14
45 03424 004310          I15
46 03425 004315          I16
47 03426 004321          I17
48 03427 004350          I22
49 03430 004355          I23
50 03431 004364          I24
```

```
↑ 0043  .MAIN
01
02          000012 .RDX 10                  ;ALL FIGURES IN NANO SECONDS
03
04 03432 002506 TBLN1:   1350               ;CORRECT TIMES FOR NOVA 1200.
05 03433 002506           1350
06 03434 002506           1350
07 03435 002506           1350
08 03436 002506           1350
09 03437 004766           2550
10 03440 004766           2550
11 03441 006116           3150
12 03442 006116           3150
13 03443 002506           1350
14 03444 002506           1350
15 03445 007246           3750
16 03446 010376           4350
17 03447 010376           4350
18 03450 011526           4950
19 03451 006116           3150
20 03452 006116           3150
21 03453 004766           2550
22 03454 006116           3150
23 03455 004766           2550
24 03456 004766           2550
25 03457 004766           2550
26 03460 005214           2700
27 03461 005214           2700
28
29 03462 001440 TBLN2:    800               ;CORRECT TIMES FOR NOVA 2 - 8K.
30 03463 001440            800
31 03464 001440            800
32 03465 001440            800
33 03466 001440            800
34 03467 003100           1600
35 03470 003100           1600
36 03471 003554           1900
37 03472 003554           1900
38 03473 001440            800
39 03474 002114           1100
40 03475 004540           2400
41 03476 005524           2900
42 03477 005524           2900
43 03500 006200           3200
44 03501 003100           1600
45 03502 003554           1900
46 03503 002570           1400
47 03504 003100           1600
48 03505 002570           1400
49 03506 002114           1100
50 03507 002570           1400
51 03510 002114           1100
52 03511 002114           1100
```

```
↑ 0044   .MAIN
01
02 03512 001750 TBLN3:   1000            ;CORRECT TIMES FOR NOVA 2 - 16K.
03 03513 001750           1000
04 03514 001750           1000
05 03515 001750           1000
06 03516 001750           1000
07 03517 003720           2000
08 03520 003720           2000
09 03521 004374           2300
10 03522 004374           2300
11 03523 001750           1000
12 03524 002260           1200
13 03525 005670           3000
14 03526 006654           3500
15 03527 006654           3500
16 03530 007640           4000
17 03531 003244           1700
18 03532 003720           2000
19 03533 002734           1500
20 03534 003244           1700
21 03535 002734           1500
22 03536 002260           1200
23 03537 002734           1500
24 03540 002260           1200
25 03541 002260           1200
26
27 03542 001604 TBLN4:    900            ;CORRECT TIMES FOR NOVA 2 DATARAM 900 NS
28 03543 001604           900
29 03544 001604           900
30 03545 001604           900
31 03546 001604           900
32 03547 003410           1800
33 03550 003410           1800
34 03551 004064           2100
35 03552 004064           2100
36 03553 001604           900
37 03554 002114           1100
38 03555 005214           2700
39 03556 006200           3200
40 03557 006200           3200
41 03560 007020           3600
42 03561 003100           1600
43 03562 003554           1900
44 03563 002570           1400
45 03564 003100           1600
46 03565 002570           1400
47 03566 002114           1100
48 03567 002570           1400
49 03570 002114           1100
50 03571 002114           1100
```

```
↑ 0045  .MAIN
01
02 03572 002032 TBLR1:  1050            ;CORRECT TIMES FOR RC 3603 - 16K.
03 03573 002506          1350
04 03574 003636          1950
05 03575 002032          1050
06 03576 002032          1050
07 03577 002570          1400
08 03600 002652          1450
09 03601 004146          2150
10 03602 004146          2150
11 03603 001274           700
12 03604 002260          1200
13 03605 004146          2150
14 03606 005442          2850
15 03607 005442          2850
16 03610 005524          2900
17 03611 003636          1950
18 03612 003636          1950
19 03613 003636          1950
20 03614 004064          2100
21 03615 003636          1950
22 03616 002506          1350
23 03617 003016          1550
24 03620 002342          1250
25 03621 002342          1250
26
27 03622 003554 TBLR2:  1900            ;CORRECT TIMES FOR RC 3603 - 16K BREAK.
28 03623 004230          2200
29 03624 005360          2800
30 03625 003554          1900
31 03626 003554          1900
32 03627 003636          1950
33 03630 003720          2000
34 03631 005670          3000
35 03632 005670          3000
36 03633 003016          1550
37 03634 004002          2050
38 03635 005214          2700
39 03636 006510          3400
40 03637 006510          3400
41 03640 006572          3450
42 03641 005360          2800
43 03642 005360          2800
44 03643 005360          2800
45 03644 005606          2950
46 03645 005360          2800
47 03646 004230          2200
48 03647 004540          2400
49 03650 004064          2100
50 03651 004064          2100
```

```
↑ 0046  .MAIN
01
02 03652 002114 TBLR3:  1100          ;CORRECT TIMES FOR RC 3603 - 32K.
03 03653 002570          1400
04 03654 003720          2000
05 03655 002114          1100
06 03656 002114          1100
07 03657 003100          1600
08 03660 003100          1600
09 03661 004540          2400
10 03662 004540          2400
11 03663 001440           800
12 03664 002342          1250
13 03665 004540          2400
14 03666 006200          3200
15 03667 006200          3200
16 03670 006200          3200
17 03671 003720          2000
18 03672 003720          2000
19 03673 003720          2000
20 03674 004146          2150
21 03675 003720          2000
22 03676 002570          1400
23 03677 003100          1600
24 03700 002424          1300
25 03701 002424          1300
26
27 03702 003636 TBLR4:  1950          ;CORRECT TIMES FOR RC 3603 - 32K BREAK.
28 03703 004312          2250
29 03704 005442          2850
30 03705 003636          1950
31 03706 003636          1950
32 03707 004064          2100
33 03710 004064          2100
34 03711 006200          3200
35 03712 006200          3200
36 03713 003100          1600
37 03714 004064          2100
38 03715 005524          2900
39 03716 007164          3700
40 03717 007164          3700
41 03720 007164          3700
42 03721 005442          2850
43 03722 005442          2850
44 03723 005442          2850
45 03724 005670          3000
46 03725 005442          2850
47 03726 004312          2250
48 03727 004622          2450
49 03730 004146          2150
50 03731 004146          2150
51
52       000010 .RDX 8
```

```
↑ 0047  .MAIN
01
02                TXCPU:  .TXT !FIRST MEM MODULE, CPU TYPE<15><12>
   03732 044506
   03733 051522
   03734 020124
   03735 042515
   03736 020115
   03737 047515
   03740 052504
   03741 042514
   03742 020054
   03743 050103
   03744 020125
   03745 054524
   03746 042520
   03747 005015
03 03750 047516 NOVA 1200<11>12<15><12>
   03751 040526
   03752 030440
   03753 030062
   03754 004460
   03755 031061
   03756 005015
04 03757 047516 NOVA 2 -  8K<11>16<15><12>
   03760 040526
   03761 031040
   03762 026440
   03763 020040
   03764 045470
   03765 030411
   03766 006466
05 03767 047012 NOVA 2 - 16K<11>17<15><12>
   03770 053117
   03771 020101
   03772 020062
   03773 020055
   03774 033061
   03775 004513
   03776 033461
   03777 005015
06 04000 020040   900NS DATARAM<11>24<15><12>
   04001 030071
   04002 047060
   04003 020123
   04004 040504
   04005 040524
   04006 040522
   04007 004515
   04010 032062
   04011 005015
07 04012 041522 RC3603 - 16K<11>20<15><12>
   04013 033063
   04014 031460
   04015 026440
   04016 030440
   04017 045466
   04020 031011
   04021 006460
```

```
↑ 0048   .MAIN
02 04022 020012    WITH BREAK<11>21<15><12>
   04023 053440
   04024 052111
   04025 020110
   04026 051102
   04027 040505
   04030 004513
   04031 030462
   04032 005015
03 04033 041522 RC3603 - 32K<11>22<15><12>
   04034 033063
   04035 031460
   04036 026440
   04037 031440
   04040 045462
   04041 031011
   04042 006462
04 04043 020012    WITH BREAK<11>23<15><12>
   04044 053440
   04045 052111
   04046 020110
   04047 051102
   04050 040505
   04051 004513
   04052 031462
   04053 005015
05 04054 047111 INITIALIZED TO!
   04055 052111
   04056 040511
   04057 044514
   04060 042532
   04061 020104
   04062 047524
   04063 000000
06
07                 PROG:    .TXT !INSTR TIMER TEST!
   04064 047111
   04065 052123
   04066 020122
   04067 044524
   04070 042515
   04071 020122
   04072 042524
   04073 052123
   04074 000000
08
09                 TXPSM:   .TXT !. PASS OF 10 RUNS<15><12>!
   04075 020056
   04076 040520
   04077 051523
   04100 047440
   04101 020106
   04102 030061
   04103 051040
   04104 047125
   04105 006523
   04106 000012
10
11                 TXTTY:   .TXT !TTY SPEED!
   04107 052124
   04110 020131
   04111 050123
   04112 042505
   04113 000104
```

```
↑ 0049  .MAIN
01
02                  TXBIT:  .TXT !NO OF BITS PER CHAR<15><12>
    04114 047516
    04115 047440
    04116 020106
    04117 044502
    04120 051524
    04121 050040
    04122 051105
    04123 041440
    04124 040510
    04125 006522
03  04126 030412 1 START + X DATA + Y STOP (Y=1 OR 2)<15><12>
    04127 051440
    04130 040524
    04131 052122
    04132 025440
    04133 054040
    04134 042040
    04135 052101
    04136 020101
    04137 020053
    04140 020131
    04141 052123
    04142 050117
    04143 024040
    04144 036531
    04145 020061
    04146 051117
    04147 031040
    04150 006451
04  04151 044412 INITIALIZED TO!
    04152 044516
    04153 044524
    04154 046101
    04155 055111
    04156 042105
    04157 052040
    04160 000117
05
06                  TX2SP:  .TXT !  !                   ;"2 SPACE"
    04161 020040
    04162 000000
07
08                  TXQUE:  .TXT !?!
    04163 000077
09
10                  I1:     .TXT !MOV       0,0     !
    04164 047515
    04165 004526
    04166 026060
    04167 004460
    04170 000000
11                  I2:     .TXT !ADD       0,0     !
    04171 042101
    04172 004504
    04173 026060
    04174 004460
    04175 000000
```

```
↑ 0050  .MAIN
01
02                  I3:     .TXT !AND       0,0     !
    04176 047101
    04177 004504
    04200 026060
    04201 004460
    04202 000000
03                  I4:     .TXT !LDA       0,CWORK !
    04203 042114
    04204 004501
    04205 026060
    04206 053503
    04207 051117
    04210 004513
    04211 000000
04                  I5:     .TXT !STA       0,CWORK !
    04212 052123
    04213 004501
    04214 026060
    04215 053503
    04216 051117
    04217 004513
    04220 000000
05                  I6:     .TXT !ISZ       CWORK   !
    04221 051511
    04222 004532
    04223 053503
    04224 051117
    04225 004513
    04226 000000
06                  I7:     .TXT !DSZ       CWORK   !
    04227 051504
    04230 004532
    04231 053503
    04232 051117
    04233 004513
    04234 000000
07                  I8:     .TXT !JMP       .+1     !
    04235 046512
    04236 004520
    04237 025456
    04240 004461
    04241 000000
08                  I9:     .TXT !JSR       .+1     !
    04242 051512
    04243 004522
    04244 025456
    04245 004461
    04246 000000
09                  I10:    .TXT !LDA       0,@CWOR !
    04247 042114
    04250 004501
    04251 026060
    04252 041500
    04253 047527
    04254 004522
    04255 000000
```

```
↑ 0051  .MAIN
01
02                  I11:    .TXT !LDA       0,@IDX5 !
    04256 042114
    04257 004501
    04260 026060
    04261 044500
    04262 054104
    04263 004465
    04264 000000
03                  I12:    .TXT !LDA       0,@DDX0 !
    04265 042114
    04266 004501
    04267 026060
    04270 042100
    04271 054104
    04272 004460
    04273 000000
04                  I13:    .TXT !LDA       0,@(@W) !
    04274 042114
    04275 004501
    04276 026060
    04277 024100
    04300 055500
    04301 004451
    04302 000000
05                  I14:    .TXT !DIA       0,0     !
    04303 044504
    04304 004501
    04305 026060
    04306 004460
    04307 000000
06                  I15:    .TXT !DOA       0,0     !
    04310 047504
    04311 004501
    04312 026060
    04313 004460
    04314 000000
07                  I16:    .TXT !INTA      0       !
    04315 047111
    04316 040524
    04317 030011
    04320 000011
08                  I17:    .TXT !SKPBN     0       !
    04321 045523
    04322 041120
    04323 004516
    04324 004460
    04325 000000
09                  I18:    .TXT !MOVL      0,0     !
    04326 047515
    04327 046126
    04330 030011
    04331 030054
    04332 000011
10                  I19:    .TXT !MOVS      0,0     !
    04333 047515
    04334 051526
    04335 030011
    04336 030054
    04337 000011
11                  I20:    .TXT !NIO       0       !
    04340 044516
    04341 004517
    04342 004460
    04343 000000
```

```
↑ 0052  .MAIN
01
02                  I21:    .TXT !NIOS      0       !
    04344 044516
    04345 051517
    04346 030011
    04347 000011
03                  I22:    .TXT !SKPBZ     0       !
    04350 045523
    04351 041120
    04352 004532
    04353 004460
    04354 000000
04                  I23:    .TXT !MOV       0,0,SZR !
    04355 047515
    04356 004526
    04357 026060
    04360 026060
    04361 055123
    04362 004522
    04363 000000
05                  I24:    .TXT !MOV       0,0,SKP !
    04364 047515
    04365 004526
    04366 026060
    04367 026060
    04370 045523
    04371 004520
    04372 000000
06
07 04373 000000 BBUF:   0                   ;1000 WORD BUFFER
08       000012 .RDX 10
09       006343 .LOC .+999
10       000010 .RDX 8
11
12 06343 034131 FBUF:   LDA     3,ADINC
13 06344 054025         STA     3,IDX5  ;RESET AUTO INC LOC
14 06345 034132         LDA     3,ADDEC
15 06346 054030         STA     3,DDX0  ;RESET AUTO DEC LOC
16 06347 010202         ISZ     TEN0    ;BUFFER EXECUTED LAST TIME ?
17 06350 002402         JMP     @.+2
18 06351 002402         JMP     @.+2
19 06352 004373         BBUF
20 06353 003166         TINSR
21
22 06354 165000 XFRAM:  MOV     3,1     ;PROGRAM TO PRINT CPU PROFILE.
23 06355 006072         CSAMS           ;PRINT START ADDR
24 06356 004401         JSR     .+1
25 06357 056415         STA     3,@XTES5 ;NO QUESTION RESTART ADDR
26 06360 024150 NTES2:  LDA     1,PASS1
27 06361 044147         STA     1,PASSC
28 06362 006046 NTES3:  CDICL
29 06363 006040         CMESS           ;PRINT CR,LF WITH MESS TO WAIT
30 06364 010621         MCRLF           ;FOR DEVICE READY BEFORE IORST
31 06365 006406         JSR     @XTES4  ;MODIFIED TO PRINT
32 06366 006043         CCRLF
33 06367 014147         DSZ     PASSC
34 06370 000772         JMP     NTES3
35 06371 002401         JMP     @.+1    ;PROGRAM FINISHED
36 06372 010620         SWISA           ;RESTART MAIN PROGRAM
37 06373 006702 XTES4:  FITYP
38 06374 002703 XTES5:  RENOQ
39
40
41                  ;TAPE 3B
42
43                  .EOT
```

```
 0053  .MAIN
01
02                ;TAPE 3C
03
04
05                        ;DELAY SUBROUTINE
06                        ;ARGUMENT FOLLOWING CALL IS ADDRESS
07                        ;OF DELAY CONSTANT.
08                        ;DELAY IN INCPEMENTS OF 1 MSEC
09                        ;AC2 & 3 ARE USED, DEFINE WAIT, ARG
10                        ;
11                        ;CALL    CWAIT
12                        ;        ARG
13                        ;
14 06375 054416 XWAIT:    STA      3,WARET ;SAVE RETURN
15 06376 010415           ISZ      WARET   ;PASS ARG
16 06377 033400           LDA      2,@0,3  ;FETCH ARG
17 06400 050414           STA      2,WIRET ;STORE ARG
18 06401 030414           LDA      2,KINDI ;FETCH KNOVA ADDRESS
19 06402 036537           LDA      3,@XCPN ;FETCH CPUNO
20 06403 157000           ADD      2,3     ;COMPUTE KADDRESS
21 06404 031400           LDA      2,0,3   ;FETCH CPU CONSTANT
22 06405 050437 NWAIT:    STA      2,KINC  ;STORE IT FOR INC OF 1 MSEC
23 06406 014436           DSZ      KINC    ;X USEC LOOP
24 06407 000777           JMP      .-1     ;X USEC
25 06410 014404           DSZ      WIRET   ;NO. OF MS STILL TO WAIT
26 06411 000774           JMP      NWAIT   ;COUNT ARG NOT ENDED
27 06412 002401           JMP      @WARET  ;COUNT ARG ENDED, RETURN
28 06413 000000 WARET:    0
29 06414 000000 WIRET:    0
30
31 06415 006416 KINDI:    .+1               ;ADDRESS OF KNOVA
32       000012           .RDX 10
33 06416 000175 KCP0:     125               ;NOVA
34 06417 000001 KCP1:     1                 ;
35 06420 000341 KCP2:     225               ;1200
36 06421 000567 KCP3:     375               ;SUPER
37 06422 000505 KCP4:     325               ;SUPER SC/830
38 06423 000536 KCP5:     350               ;800/NOVA 2-16K
39 06424 000620 KCP6:     400               ;NOVA 2-8K
40 06425 000001 KCP7:     1                 ;CONSTANTS ABOVE ARE FOR GUESSED CPU TYPE
41 06426 000001 KCP10:    1                 ;CONSTANTS BELOW ARE EXACTLY.
42 06427 000200 KCP11:    128               ;NOVA    7,8 USEC
43 06430 000336 KCP12:    222               ;1200    4,5 USEC
44 06431 000601 KCP13:    385               ;800     2,6 USEC
45 06432 000601 KCP14:    385               ;SUPER   2,6 USEC
46 06433 000764 KCP15:    500               ;SUPER SC  2,0 USEC
47 06434 000620 KCP16:    400               ;NOVA2/8K  2,5 USEC
48 06435 000502 KCP17:    322               ;NOVA2/16K 3,1 USEC
49 06436 000531 KCP20:    345               ;RC3603/RC3609           2,90 USEC
50 06437 000334 KCP21:    220               ;RC3603/RC3609/BREAK ON  4,55 USEC
51 06440 000463 KCP22:    307               ;RC3603/RC3608           3,26 USEC
52 06441 000314 KCP23:    204               ;RC3603/RC3608/BREAK ON  4,90 USEC
53 06442 000515 KCP24:    333               ;NOVA2/DATARAM 900 NSEC  3,00 USEC
54 06443 000001 KCP25:    1                 ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
55       000010           .RDX 8
56 06444 000000 KINC:     0
```

```
↑ 0054  .MAIN
01
02
03                          ;PROCEDURE TIMER ON SKP
04                          ;CALL:  TIMSK
05                          ;       MSEC (>0) TO WAIT MAX FOR
06                          ;       SKP INSTR.
07                          ;       TIME OUT RETURN
08                          ;       NORMAL RETURN
09
10 06445 025400 RTIME:  LDA     1,0,3   ;AC1:=TIME
11 06446 125015         MOV#    1,1,SNR ;IF TIME = ZERO
12 06447 125400         INC     1,1     ;GIVE IT A CHANGE
13 06450 021401         LDA     0,1,3   ;AC0:=INSTR
14 06451 040410         STA     0,STSKP ;STORE INSTR
15 06452 124400         NEG     1,1     ;AC1:=-TIME
16 06453 020417         LDA     0,CPU00 ;FETCH CPU0-ADDR.
17 06454 032465         LDA     2,@XCPN ;FETCH CPUNO
18 06455 113000         ADD     0,2     ;COMPUTE CPU-ADDR.
19 06456 021000         LDA     0,0,2   ;FETCH CPU-CONSTANT
20 06457 040412         STA     0,CPUINC;STORE IT FOR INC OF 1MS
21 06460 020411 STINC:  LDA     0,CPUINC;# OF LOOPS FOR 1 MSEC
22 06461 000000 STSKP:  0               ; X USEC (SKP INSTR.)
23 06462 000402         JMP     .+2     ; X USEC
24 06463 001403         JMP     3,3     ;NORMAL RETURN
25 06464 101404         INC     0,0,SZR ; X USEC
26 06465 000774         JMP     STSKP   ; X USEC
27 06466 125404         INC     1,1,SZR ;COUNT # OF MSEC
28 06467 000771         JMP     STINC   ;MORE MSEC
29 06470 001402         JMP     2,3     ;TIME OUT RETURN, AC0=AC1=0
30
31 06471 000000 CPUINC: 0
32 06472 006473 CPU00:  .+1             ;ADDR. OF CPU0
33       000012         .RDX 10
34 06473 177677 CPU0:   -65             ;NOVA
35 06474 177777 CPU1:   -1
36 06475 177552 CPU2:   -150            ;1200
37 06476 177470 CPU3:   -200            ;SUPER
38 06477 177437 CPU4:   -225            ;SUPER SC/830
39 06500 177406 CPU5:   -250            ;800/NOVA 2-16K
40 06501 177324 CPU6:   -300            ;NOVA 2-8K
41 06502 177777 CPU7:   -1              ;
42 06503 177777 CPU10:  -1              ;
43 06504 177676 CPU11:  -66             ;NOVA   15,2 USEC
44 06505 177550 CPU12:  -152            ;1200    6,6 USEC
45 06506 177371 CPU13:  -263            ;800     3,8 USEC
46 06507 177504 CPU14:  -188            ;SUPER   5,3 USEC
47 06510 177427 CPU15:  -233            ;SUPER SC  4,3 USEC
48 06511 177343 CPU16:  -285            ;NOVA2/8K  3,5 USEC
49 06512 177422 CPU17:  -238            ;NOVA2/16K 4,2 USEC
50 06513 177371 CPU20:  -263            ;RC3603/RC3609          3,80 USEC
51 06514 177565 CPU21:  -139            ;RC3603/RC3609/BREAK ON 7,20 USEC
52 06515 177416 CPU22:  -242            ;RC3603/RC3608          4,14 USEC
53 06516 177571 CPU23:  -135            ;RC3603/RC3608/BREAK ON 7,40 USEC
54 06517 177371 CPU24:  -263            ;NOVA2/DATARAM 900 NSEC 3,80 USEC
55 06520 177777 CPU25:  -1              ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
56       000010         .RDX 8
```

```
↑ 0055  .MAIN
01
02                          ;ROUTINE TO MESSURE TIME.
03                          ;CALL    TIMMS
04                          ;        SKP INSTR. TO BE MESSURED
05                          ;        TIMEOUTRETURN    TIMEM:=MAX TIME
06                          ;        NORMAL RETURN    TIMEM:=MESSURED TIME
07                          ;THE TIME IS MESSURED IN TENS OF USEC.
08                          ;LOOPS IN THE INSTRUCTIONS MARKED X USEC UNTIL
09                          ;BIT 0 IS SET (TIMEOUT, MAX MESSUREABLE TIME)
10                          ;COUNTING FROM -1 IN THE INC INSTRUCTION OR UNTIL
11                          ;SKP INSTRUCTION TO BE MESSURED IS EFFECTIVE.
12                          ;COULD BE EXTENDED TO TWICE THE TIME POSSIBLE HERE
13                          ;IF CARRY IS USED INSTEAD OF BIT 0, BUT NO PRINT ROUT.
14
15 06521 054477 MSTIM:  STA     3,MSTIR ;SAVE RETURN
16 06522 021400         LDA     0,0,3   ;GET SKP INSTR.
17 06523 040405         STA     0,MSTIB ;TO BE MESSURED.
18 06524 102040         ADCO    0,0     ;AC0:=177777, C:=1
19 06525 101400 MSTIA:  INC     0,0     ;TO AVOID OVERFLOW       X USEC
20 06526 101112         MOVL#   0,0,SZC ;INCREASE LOOP TIME      X USEC
21 06527 000443         JMP     MSTIO   ;TIME OUT, BIT 0=1
22 06530 000000 MSTIB:  0               ;SKP INSTR. TO           X USEC
23 06531 000774         JMP     MSTIA   ;BE TIMED.               X USEC
24 06532 101400         INC     0,0     ;COMPENSATE INITIALIZATION 4 INSTR.
25 06533 101400         INC     0,0     ;COMPENSATE STARTING AT -1
26 06534 040402         STA     0,TIMEX ;TIMECOUNT MESSURED
27 06535 000441         JMP     MSTID   ;CALCULATE, BIT 0=0
28
29 06536 000000 TIMEX:  0               ;TIMECOUNT USED ABOVE
30 06537 000000 TIMCT:  0               ;TIMECONSTANT FROM TABLE BELOW
31 06540 000000 TIMEM:  0               ;TIME MESSURED
32 06541 006730 XCPN:   CPUNO
33
34 06542 006543 TCP00:  .+1             ;100 MSEC CONSTANTS:
35       000012         .RDX 10
36 06543 012574 TCP0:   5500            ;NOVA
37 06544 077777 TCP1:   32767           ;       WILL GIVE TIMEOUT = 100 MSEC
38 06545 030324 TCP2:   12500           ;1200
39 06546 040164 TCP3:   16500           ;SUPER
40 06547 051774 TCP4:   21500           ;SUPER SC/830
41 06550 055714 TCP5:   23500           ;800/NOVA2-16K
42 06551 063604 TCP6:   26500           ;NOVA2-8K
43 06552 077777 TCP7:   32767           ;
44 06553 077777 TCP10:  32767           ;                X USEC:
45 06554 012566 TCP11:  5494            ;NOVA            18,2 USEC
46 06555 030442 TCP12:  12578           ;1200             7,95 USEC
47 06556 060650 TCP13:  25000           ;800              4,0 USEC
48 06557 040011 TCP14:  16393           ;SUPER            6,1 USEC
49 06560 055327 TCP15:  23255           ;SUPER SC         4,6 USEC
50 06561 063314 TCP16:  26316           ;NOVA 2 - 8K      3,8 USEC
51 06562 054307 TCP17:  22727           ;NOVA 2 - 16K     4,4 SEC
52 06563 052001 TCP20:  21505           ;RC3603/RC3609          4,65 USEC
53 06564 030206 TCP21:  12422           ;RC3603/RC3609/BREAK ON 8,05 USEC
54 06565 047545 TCP22:  20325           ;RC3603/RC3608          4,92 USEC
55 06566 027531 TCP23:  12121           ;RC3603/RC3608/BREAK ON 8,25 USEC
56 06567 060650 TCP24:  25000           ;NOVA2/DATARAM 900 NSEC 4,00 USEC
57 06570 077777 TCP25:  32767           ;FOR TROUBLE, PUT HERE RELEVANT VALUE.
58
59 06571 023420 MST10:  10000
60       000010         .RDX 8
61
62 06572 102220 MSTIO:  ADCZR   0,0     ;AC0:=077777, CARRY:=1
63 06573 040743         STA     0,TIMEX ;TIMEOUT TIMECOUNT = 32767
64 06574 000402         JMP     MSTID   ;CALCULATE
```

```
↑ 0056  .MAIN
01
02 06575 000000 MSTIC:  0                ;CARRY FLAG
03 06576 101200 MSTID:  MOVR     0,0     ;
04 06577 040776         STA      0,MSTIC ;SAVE CARRY
05 06600 026741         LDA      1,@XCPN ;
06 06601 030741         LDA      2,TCP00 ;
07 06602 133000         ADD      1,2     ;
08 06603 025000         LDA      1,0,2   ;FETCH CPU CONSTANT
09 06604 044733         STA      1,TIMCT ;
10 06605 024731         LDA      1,TIMEX ;
11 06606 030763         LDA      2,MST10 ;
12 06607 006066         MULTI            ;TIMECOUNT X 10000
13 06610 030727         LDA      2,TIMCT ;
14 06611 006070         DIVID            ;DIVIDED BY CPU CONSTANT
15 06612 044726         STA      1,TIMEM ;TIME IN TENS OF USEC
16 06613 020762         LDA      0,MSTIC ;GET CARRY FLAG
17 06614 010404         ISZ      MSTIR   ;PASS SKP ARG.
18 06615 101103         MOVL     0,0,SNC ;IF CARRY THEN TIMEOUT
19 06616 010402         ISZ      MSTIR   ;PASS TIMEOUT RETURN
20 06617 002401         JMP      @MSTIR  ;RETURN
21 06620 000000 MSTIR:  0                ;RETURN ADDR.
22
23                      ;ROUTINE TO MULTIPLY
24                      ;CALL    MULTI
25                      ;        RETURN
26                      ;
27                      ;        AC0, AC1:=AC1*AC2
28                      ;
29 06621 102460 XMULT:  SUBC     0,0     ;MULTIPLY C(1)*(2)
30 06622 054426         STA      3,MSAV  ;RESULT TO C(0),(1)
31 06623 034424         LDA      3,MDCTR ;AC2 UNCHANGED
32 06624 125203 MLOOP:  MOVR     1,1,SNC ;CARRY UNCHANGED
33 06625 101201         MOVR     0,0,SKP
34 06626 143220         ADDZR    2,0
35 06627 175404         INC      3,3,SZR
36 06630 000774         JMP      MLOOP
37 06631 125260         MOVCR    1,1
38 06632 002416         JMP      @MSAV
39
40                      ;ROUTINES TO DIVIDE
41                      ;CALL    DIVIS
42                      ;        RETURN
43                      ;
44                      ;        AC0:=REMAINDER
45                      ;        AC1:=QUOTIENT FOR AC1/AC2
46                      ;
47                      ;CALL    DIVID
48                      ;        RETURN
49                      ;
50                      ;        AC0:=REMAINDER
51                      ;        AC1:=QUOTIENT FOR AC0, AC1/AC2
52                      ;
53 06633 102400 XDIVS:  SUB      0,0     ;DIVIDE      C(1)/C(2)
54 06634 054414 XDIVD:  STA      3,MSAV  ;DIVIDE C(0),C(1)/C(2)
55 06635 034412         LDA      3,MDCTR ;C(0)=REMAINDER
56 06636 125120         MOVZL    1,1     ;C(1)=QUOTIENT
57 06637 101100 DLOOP:  MOVL     0,0     ;AC2 UNCHANGED
58 06640 142412         SUB#     2,0,SZC ;CARRY...?
59 06641 142400         SUB      2,0
60 06642 125100         MOVL     1,1
61 06643 175404         INC      3,3,SZR
62 06644 000773         JMP      DLOOP
63 06645 125400         INC      1,1     ;ROUND UP NO MATTER WHAT REMAINDER IS.
64 06646 002402         JMP      @MSAV
65 06647 177760 MDCTR:  -20
66 06650 000000 MSAV:   0
```

```
↑ 0057  .MAIN
01
02                          ;SUBROUTINE TIMER ON ROUTINE
03                          ;
04                          ;CALL   TIMRO
05                          ;       ADDR. OF TIME TO WAIT MAX (MSEC)
06                          ;       JSR @PAGE ZERO (DEFINITION)     OR JMP .+1
07                          ;       ARGUMENT FOR JSR INSTR.         OR SKP INSTR.
08                          ;       TIMEOUT RETURN
09                          ;       EVENT RETURN
10
11 06651 054430 XTIMS:  STA     3,XTIMR ;
12 06652 033400         LDA     2,@0,3  ;GET TIME, # OF MSEC
13 06653 050425         STA     2,XTIMC ;TO COUNT
14 06654 010425         ISZ     XTIMR
15 06655 032424         LDA     2,@XTIMR ;GET CALL DEFINITION
16 06656 050406         STA     2,XTIMD ;TO USE FOR TIMER
17 06657 010422         ISZ     XTIMR
18 06660 032421         LDA     2,@XTIMR ;GET ARGUMENT
19 06661 050404         STA     2,XTIMA ;FOR CALLED ROUTINE
20 06662 010417         ISZ     XTIMR   ;COUNT RETURN ADDR TO PASS ARG.
21 06663 010415         ISZ     XTIMC   ;COUNT TIME FOR CORRECT USE
22 06664 000401 XTIMD:  JMP     .+1     ;JSR    XX
23 06665 000401 XTIMA:  JMP     .+1     ;ARGUMENT YY
24 06666 000403         JMP     XTIMT   ;NOT YET RETURN, TEST TIMER
25 06667 010412         ISZ     XTIMR   ;EVENT RETURN, COUNT RETURN ADDR.
26 06670 002411         JMP     @XTIMR  ;TO PASS ERROR RETURN.
27 06671 014407 XTIMT:  DSZ     XTIMC   ;TIMECOUNT STILL NOT FINISHED ?
28 06672 000402         JMP     XTIMW   ;STALL 1 MSEC
29 06673 002406         JMP     @XTIMR  ;TIMEOUT RETURN
30 06674 006061 XTIMW:  CWAIT
31 06675 006677         XTIM1
32 06676 000766         JMP     XTIMD   ;TEST EVENT
33 06677 000001 XTIM1:  1               ;1 MSEC CONSTANT
34 06700 000000 XTIMC:  0               ;X MSEC COUNTER
35 06701 000000 XTIMR:  0               ;RETURN ADDR.
36
37                          ;ROUTINE TO FIND TYPE OF CPU.
38                          ;CALLED BY REBIN
39                          ;THIS ROUTINE IS SETTING A NUMBER INDICATING
40                          ;TYPE AND SPEED OF CPU INTO CPUNO. SEE
41                          ;ROUTINE TO MESSURE TIME FOR DEFINITION OF #.
42
43 06702 054423 FITYP:  STA     3,RETYP
44 06703 062677         IORST
45 06704 004440         JSR     TYMER
46 06705 125014         MOV#    1,1,SZR ;AC1=0, SEARCH MORE
47 06706 000414         JMP     STTYP
48 06707 006417         JSR     @XOMER
49 06710 125014         MOV#    1,1,SZR ;AC1=0, SEARCH MORE
50 06711 000411         JMP     STTYP
51 06712 006415         JSR     @XWTYP
52 06713 020411         LDA     0,SES60
53 06714 123000         ADD     1,0     ;ASCII
54 06715 006045         CDOUT
55 06716 006041         CCHAR
56 06717 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
57 06720 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
58 06721 002404         JMP     @RETYP  ;EXIT
59 06722 044406 STTYP:  STA     1,CPUNO
60 06723 002402         JMP     @RETYP  ;EXIT
61 06724 000060 SES60:  60
62 06725 000000 RETYP:  0
63 06726 007210 XOMER:  TOMER
64 06727 007363 XWTYP:  NWTYP
65 06730 000006 CPUNO:  6               ;TYPE OF CPU, NOVA 2-8K SHOWN
```

```
↑ 0058  .MAIN
01
02 06731 054774 PRTYP:  STA     3,RETYP ;PRINT CPU TYPE NUMBER
03 06732 006044         CDISP
04 06733 010660         MCPUT
05 06734 006040         CMESS
06 06735 010660         MCPUT           ;<15><12>CPU TYPE:
07 06736 024772         LDA     1,CPUNO
08 06737 006052         CTOCT
09 06740 006056         CDOCT
10 06741 006061         CWAIT           ;WAIT 3 SEC TO READ MESS. DON'T
11 06742 001236         SEC3            ;USE CDATT AS SW AREN'T SET.
12 06743 002762         JMP     @RETYP  ;EXIT
13 06744 054572 TYMER:  STA     3,SVTYME
14 06745 020576         LDA     0,XTYME ;GET ADDRESS OF TYME LIST.
15 06746 040021         STA     0,IDX1  ;STORE IN AUTO POINTER.
16 06747 020575         LDA     0,XINST ;GET ADDR OF INSTRUCTION LIST
17 06750 040020         STA     0,IDX0  ;STORE IN AUTO POINTER.
18 06751 030567         LDA     2,INXW5
19 06752 050023         STA     2,IDX3  ;INITIALIZE IDX3 FOR LDA INST.
20 06753 152440         SUBO    2,2     ;AC2:=0, NULL CHAR
21 06754 071011         DOA     2,XTTO  ;SEND CHAR
22 06755 060111         NIOS    XTTO    ;STAPT TTO AND
23 06756 063511         SKPBZ   XTTO    ;SYNCHRONIZE
24 06757 000777         JMP     .-1     ;WITH TTO CLOCK
25 06760 071011         DOA     2,XTTO  ;SEND A CHAR
26 06761 060111         NIOS    XTTO    ;AND MESSURE TIME, NOT ACCURATE
27 06762 006064         TIMMS           ;CPU TYPE NOT KNOWN BUT TO GIVE AN IDEA
28 06763 063511         SKPBZ   XTTO    ;OF TTO SPEED
29 06764 000401         JMP     .+1     ;ACU IS MESSURED TIME
30 06765 022544         LDA     0,@TYMEM
31 06766 024544         LDA     1,TYLIM ;LIMIT TO DESTINGV. SPEED
32 06767 034544         LDA     3,TYTTS ;SLOW TTY CONSTANT
33 06770 106432         SUBZ#   0,1,SZC ;IS TTY FAST ?
34 06771 034543         LDA     3,TYTTF ;YES, FAST TTY CONSTANT
35 06772 054535         STA     3,TYMEN ;STORE # OF COUNTS
36 06773 152440         SUBO    2,2     ;AC2:=0, NULL CHAR
37 06774 071011         DOA     2,XTTO  ;SEND CHAR
38 06775 060111         NIOS    XTTO    ;START TTO AND
39 06776 063511         SKPBZ   XTTO    ;SYNCHRONIZE PROGRAM
40 06777 000777         JMP     .-1     ;WITH TTY CLOCK.
41 07000 152440 TYMA:   SUBO    2,2     ;CLEAR AC2
42 07001 141000         MOV     2,0     ;CLEAR AC0 ALSO.
43 07002 026020         LDA     1,@IDX0 ;GET INST. FROM LIST
44 07003 125015         MOV#    1,1,SNR ;ZERO MARKS END OF INSTR. LIST
45 07004 000455         JMP     SCORE   ;
46 07005 071011         DOA     2,XTTO  ;SEND NULL CHAR, STARTING LATER
47 07006 034451         LDA     3,CSKP  ;GET ALC-SKP MASK AND
48 07007 137400         AND     1,3     ;AND WITH INSTR.
49 07010 175123         MOVZL   3,3,SNC ;CKN BIT 0, THE ALC BIT
50 07011 000403         JMP     TYMD    ;BIT 0 = 0 MEANS NO ALC CODE.
51 07012 175004         MOV     3,3,SZR ;CKN 3 LSB'S FOR SKP CODE.
52 07013 000411         JMP     TYMB    ;FOUND ALC-SKP CODE.
53 07014 044403 TYMD:   STA     1,TYMJ  ;STORE INSTR. IN TIME LOOP.
54 07015 060111         NIOS    XTTO    ;START TTO, FOR NON ALC-SKKP INSTR.
55 07016 151400 TYMF:   INC     2,2     ;THESE **********************
56 07017 000000 TYMJ:   0               ;   INSTRUCTIONS ************
57 07020 063511         SKPBZ   XTTO    ;      FORM THE *************
58 07021 000775         JMP     TYMF    ;         TIMING LOOP *******
59 07022 052021 TYMC:   STA     2,@IDX1 ;STORE COUNT INTO TYME LIST.
60 07023 000755         JMP     TYMA    ;LOOP.
61 07024 044403 TYMB:   STA     1,TYMH  ;STORE ALC-SKP IN TIME LOOP.
62 07025 060111         NIOS    XTTO    ;START TTO
63 07026 151400 TYMG:   INC     2,2     ;THESE **********************
64 07027 000000 TYMH:   0               ;   INSTRUCTIONS ************
65 07030 000401         JMP     .+1     ;      FORM THE *************
66 07031 063511         SKPBZ   XTTO    ;         TIMING ************
67 07032 000774         JMP     TYMG    ;            LOOP ***********
68 07033 000767         JMP     TYMC    ;FOR ALC-SKP INSTRUCTIONS.
```

```
↑ 0059  .MAIN
01
02 07034 000000 INSW1:  0                ;HERE WORKS ISZ INSAD FROM TYMJ
03 07035 100005 INDW1:  @5               ;HERE WORKS LDA @INDAD FROM TYMJ
04
05 07036 101000 INSTR:  MOV      0,0     ;THIS IS THE 16 INSTRUCTION
06 07037 103000         ADD      0,0     ;LIST, SELECTED TO DEVELOPE
07 07040 103401         AND      0,0,SKP ;AN IDENTITY PROFILE OF THE
08 07041 020005         LDA      0,5     ;PROCESSOR IN THE COURSE OF
09 07042 040005         STA      0,5     ;BEING EXECUTED. THE INSTR. ARE
10 07043 010415         ISZ      INSAD   ;LATER STORED IN TYMJ,TYMH
11 07044 000401         JMP      INSTA
12 07045 004401 INSTA:  JSR      INSTR
13 07046 022005 INSTR:  LDA      0,@5
14 07047 022023         LDA      0,@IDX3
15 07050 022416         LDA      0,@INDAD ;LABEL INDAD PLACED .+16
16 07051 060400         DIA      0,0
17 07052 061000         DOA      0,0
18 07053 061477         INTA     0
19 07054 063400         SKPBN    0
20 07055 060100         NIOS     0
21 07056 000000         0                ;END OF INSTR. LIST MARKER
22 07057 100007 CSKP:   100007           ;ONLY FOR ASSEMBLING ISZ INSAD
23 07060 000000 INSAD:  0                ;TO WORK IN INSW1 AND INSW2.
24
25 07061 102520 SCORE:  SUBZL    0,0     ;SET THE
26 07062 040455         STA      0,ORDINAL ;ORDINAL COUNTER TO +1.
27 07063 020460 SCORA:  LDA      0,XTYME ;GET TYME LIST INITIAL
28 07064 040021         STA      0,IDX1  ;ADDRESS FOR AUTO INC.
29 07065 020455         LDA      0,XC20  ;SET UP THE X16 COUNTER.
30 07066 040453 INDAD:  STA      0,XX16  ;WITH A COUNT OF 16 DEC.
31 07067 026021         LDA      1,@IDX1 ;GET TYME ENTRY INTO AC1 AS FIRST
32                                       ;BIG TYME. THEN SEARCH FOR BIGGER TYME.
33 07070 022021 SCORB:  LDA      0,@IDX1 ;CK MAGNITYDE OF NX TYME ENTRY.
34 07071 106033         ADCZ#    0,1,SNC ;SKPS IF AC0 < AC1
35 07072 105000         MOV      0,1     ;AC0 > OR = AC1, AC1 = BIGGEST TYME.
36 07073 014446         DSZ      XX16    ;COUNT DOWN # OF TYMES.
37 07074 000774         JMP      SCORB   ;STILL MORE TYMES, SO LOOP.
38
39                      ;REVIEW CLEARS ALL BIG TYMES. AC1 = THE BIGGEST TYME NOW.
40                      ;LOC'S = AC1 OR (AC1-1) OR (AC1-2) ARE CLEARED AND THERE
41                      ;POS IN RANK LIST ARE SET TO THE (C) OF THE ORDINAL COUNT
42 07075 125015 REVU:   MOV#     1,1,SNR ;IF A SCORE PASS IS COMPL WITH
43 07076 000566         JMP      KEYS    ;AC1 = 0, TYME IS CLEAR, SO GO FORM KEY.
44 07077 020444         LDA      0,XTYME ;REINITIALIZE RVTMP WITH
45 07100 040435         STA      0,RVTMP ;TYME - 1.
46 07101 020444         LDA      0,XRANK ;INITIALIZE IDX2 WITH
47 07102 040022         STA      0,IDX2  ;RANK - 1.
48 07103 020437         LDA      0,XC20  ;RESET X16 COUNTER
49 07104 040435         STA      0,XX16  ;BACK TO 16 DECIMAL.
50 07105 010430 REVUA:  ISZ      RVTMP   ;INC TYMES LIST POINTER.
51 07106 030421         LDA      2,TYMEN ;GET TOLERANCE COUNT
52 07107 022426         LDA      0,@RVTMP ;GET TYME ENTRY AND
53 07110 106415 REVUD:  SUB#     0,1,SNR ;COMP WITH LARGEST TYME ENTRY.
54 07111 000411         JMP      REVUB   ;IF BIGGEST TYME, STORE ORDINAL.
55 07112 101400         INC      0,0     ;ADD +1 AND COMP AGAIN WITH BIG TYME.
56 07113 151404         INC      2,2,SZR ;TEST FOR TOLERANCE BIG TYME 0,-1,-2,...
57 07114 000774         JMP      REVUD   ;IF BIGGEST TYME -1,-2,-3...-N, STORE OR
58 07115 022022         LDA      0,@IDX2 ;ENTRY OUT OF RANGE, INC RANK POINTER.
59 07116 014423 REVUC:  DSZ      XX16    ;DECREMENT THE X16 POINTER.
60 07117 000766         JMP      REVUA   ;STILL MORE TYMES, SO LOOP.
61 07120 010417         ISZ      ORDINAL ;INC ORDINAL COUNT.
62 07121 000742         JMP      SCORA   ;GO TO SCORA TO FIND NX BIG TYME.
63 07122 030415 REVUB:  LDA      2,ORDINAL ;GET ORDINAL COUNT
64 07123 052022         STA      2,@IDX2 ;AND STORE IT IN PRESENT RANK LOC.
65 07124 176440         SUBO     3,3     ;CLEARING AC3.
66 07125 056410         STA      3,@RVTMP ;CLEARS LOC. IN TYME LIST.
67 07126 000770         JMP      REVUC   ;
```

```
↑ 0060  .MAIN
01
02 07127 177775 TYMEN:  -3              ;-# OF COUNTS, TOLERENCE FOR TYMES
03 07130 177767 TYRTC:  -9.             ;-3 WILL GIVE BIG, BIG-1, BIG-2 IS E
04 07131 006540 TYMEM:  TIMEM           ;MESSURED TIME FOR ONE TTO CHAR
05 07132 001212 TYLIM:  650.            ;LIMIT TO TELL SPEED > 1600 BAUD
06 07133 177772 TYTTS:  -6              ;COUNT TOLERANCE SLOW TTY
07 07134 177775 TYTTF:  -3              ;COUNT TOLERANCE FAST TTY
08 07135 000000 RVTMP:  0
09 07136 000000 SVTYME: 0
10 07137 000000 ORDINAL: 0
11 07140 000005 INXW5:  5               ;START LDA 0,@IDX3 IN CELL 5
12 07141 000000 XX16:   0
13 07142 000020 XC20:   20
14 07143 007145 XTYME:  TYME-1
15 07144 007035 XINST:  INSTR-1
16 07145 007166 XRANK:  RANK-1
17       000020 TYME:   .BLK 20
18 07166 000000 ENTYM:  0               ;END OF TYME LIST MARKER.
19       000020 RANK:   .BLK 20
20 07207 000000 ENRNK:  0               ;END OF RANK LIST MARKER.
21
22 07210 054726 TOMER:  STA     3,SVTYME
23 07211 020717         LDA     0,TYRTC ;GET RTC TOLERANCE
24 07212 040715         STA     0,TYMEN ;STORE # OF COUNTS
25 07213 020730         LDA     0,XTYME ;GET ADDRESS OF TYME LIST.
26 07214 040021         STA     0,IDX1  ;STORE IN AUTO POINTER.
27 07215 020727         LDA     0,XINST ;GET ADDR OF INSTRUCTION LIST
28 07216 040020         STA     0,IDX0  ;STORE IN AUTO POINTER.
29 07217 030721         LDA     2,INXW5
30 07220 050023         STA     2,IDX3  ;INITIALIZE IDX3 FOR LDA INST.
31 07221 102520         SUBZL   0,0
32 07222 101120         MOVZL   0,0     ;AC0:=2
33 07223 061014         DOA     0,XRTC  ;SET RTC FREQUENCY TO 100 HZ (10 MSEC).
34 07224 060114         NIOS    XRTC
35 07225 063514         SKPBZ   XRTC    ;SYNCHRONIZE PROGRAM
36 07226 000777         JMP     .-1     ;WITH RTC.
37 07227 152440 TOMA:   SUB0    2,2     ;CLEAR AC2
38 07230 141000         MOV     2,0     ;CLEAR AC0 ALSO.
39 07231 026020         LDA     1,@IDX0 ;GET INST. FROM LIST
40 07232 125015         MOV#    1,1,SNR ;ZERO MARKS END OF INSTR. LIST
41 07233 000626         JMP     SCORE   ;
42 07234 034623         LDA     3,CSKP  ;GET ALC-SKP MASK AND
43 07235 137400         AND     1,3     ;AND WITH INSTR.
44 07236 175123         MOVZL   3,3,SNC ;CKN BIT 0, THE ALC BIT
45 07237 000403         JMP     TOMD    ;BIT 0 = 0 MEANS NO ALC CODE.
46 07240 175004         MOV     3,3,SZR ;CKN 3 LSB'S FOR SKP CODE.
47 07241 000411         JMP     TOMB    ;FOUND ALC-SKP CODE.
48 07242 044403 TOMD:   STA     1,TOMJ  ;STORE INSTR. IN TIME LOOP.
49 07243 060114         NIOS    XRTC    ;START RTC
50 07244 151400 TOMF:   INC     2,2     ;THESE **********************
51 07245 000000 TOMJ:   0               ;   INSTRUCTIONS ************
52 07246 063514         SKPBZ   XRTC    ;      FORM THE *************
53 07247 000775         JMP     TOMF    ;         TIMING LOOP *******
54                                      ;FOR NON ALC-SKP INSTR.
55 07250 052021 TOMC:   STA     2,@IDX1 ;STORE COUNT INTO TYME LIST.
56 07251 000756         JMP     TOMA    ;LOOP.
57 07252 044403 TOMB:   STA     1,TOMH  ;STORE ALC-SKP IN TIME LOOP.
58 07253 060114         NIOS    XRTC    ;START RTC
59 07254 151400 TOMG:   INC     2,2     ;THESE **********************
60 07255 000000 TOMH:   0               ;   INSTRUCTIONS ************
61 07256 000401         JMP     .+1     ;      FORM THE *************
62 07257 063514         SKPBZ   XRTC    ;         TIMING ************
63 07260 000774         JMP     TOMG    ;            LOOP ***********
64 07261 000767         JMP     TOMC    ;FOR ALC-SKP INSTRUCTIONS.
65 07262 000000 INSW2:  0               ;HERE WORKS ISZ INSAD FROM TOMJ.
66 07263 100000 INDW2:  @0              ;HERE WORKS LDA @INDAD FROM TOMJ.
```

```
↑ 0061   .MAIN
01
02 07264 020661 KEYS:   LDA     0,XRANK ;REINITIALIZE RANK LIST
03 07265 040022         STA     0,IDX2  ;AUTO INC POINTER.
04 07266 102520         SUBZL   0,0     ;BIT 15:=1 AS END OF KEY MARKER.
05
06 07267 026022 KEYA:   LDA     1,@IDX2 ;GET ORDINAL COUNT FROM RANK LIST.
07 07270 125223         MOVZR   1,1,SNC ;CK IF # IS ODD OR EVEN.
08 07271 000404         JMP     KEYB    ;EVEN
09 07272 101122         MOVZL   0,0,SZC ;ROTATE 0 INTO BIT 15 FOR ODD ORDINAL.
10 07273 000405         JMP     SESAME  ;CARRY BIT SET MEANS END OF KEY.
11 07274 000773         JMP     KEYA    ;MORE-ON-KEY.
12 07275 101142 KEYB:   MOVOL   0,0,SZC ;ROTATE 1 INTO BIT 15 FOR EVEN ORDINAL.
13 07276 000402         JMP     SESAME  ;CARRY BIT SET MEANS END OF KEY.
14 07277 000770         JMP     KEYA    ;MORE-ON-KEY.
15
16 07300 105000 SESAME: MOV     0,1     ;AC1=KEY
17 07301 006056         CDOCT
18 07302 006052         CTOCT
19 07303 020404         LDA     0,SES40
20 07304 006045         CDOUT
21 07305 126440 SESOUT: SUBO    1,1     ;CLEAR AC1 AS KEY INDICATOR FOR NO
22 07306 002630 SESEX:  JMP     @SVTYME ;CPU FOUND, RETURN.
23 07307 000040 SES40:  40
24                                      ;IDENT TABLE FOR OTHER PROGRAMS.
25 07310 064450 LOCKA:  064450          ;NOVA KEY.                CPU #   11
26 07311 016137         016137          ;NOVA 1200 KEY.                   12
27 07312 034174         034174          ;NOVA 800 KEY.                    13
28 07313 002544         002544          ;SUPERNOVA KEY.                   14
29 07314 007126         007126          ;SUPERNOVA SC KEY.                15
30 07315 034652         034652          ;NOVA 2 - 8K KEY.                 16
31 07316 022512         022512          ;NOVA 2 - 16K KEY.                17
32 07317 170225         170225          ;RC3603/RC3609, 16K KEY           20
33 07320 170265         170265
34 07321 170365         170365
35 07322 172225         172225
36 07323 172265         172265
37 07324 144567         144567          ;RC3603/RC3609/BREAK ON KEY       21
38 07325 146427         146427
39 07326 166610         166610
40 07327 166065         166065          ;RC3603/RC3608, 32K KEY           22
41 07330 166165         166165
42 07331 166225         166225
43 07332 166325         166325
44 07333 172065         172065
45 07334 172325         172325
46 07335 174050         174050
47 07336 174150         174150
48 07337 174210         174210
49 07340 174250         174250
50 07341 174310         174310
51 07342 176010         176010
52 07343 144352         144352          ;RC3603/RC3608/BREAK ON KEY       23
53 07344 154352         154352
54 07345 160135         160135
55 07346 162025         162025
56 07347 162202         162202
57 07350 162225         162225
58 07351 166025         166025
59 07352 170312         170312
60 07353 170352         170352
61 07354 172012         172012
62 07355 172052         172052
63 07356 172202         172202
64 07357 176025         176025
65 07360 176050         176050
66 07361 020653         020653          ;NOVA 2 - DATARAM 900 NSEC KEY    24
67 07362 000000         0               ;END OF TABLE, NO CPU FOUND, #    0
```

```
↑ 0062  .MAIN
01
02                          ;FIND TYPE OF CPU USING RTC
03                          ;PART OF CPUTYP
04 07363 054442 NWTYP:  STA     3,NIRET ;SAVE RETURN
05 07364 102520         SUBZL   0,0     ;AC0:=1
06 07365 101120         MOVZL   0,0     ;AC0:=2
07 07366 126400         SUB     1,1
08 07367 061014         DOA     0,XRTC  ;SET RTC FREQUENCY
09 07370 060114         NIOS    XRTC    ;TO 100 HZ (10 MSEC)
10 07371 063514         SKPBZ   XRTC
11 07372 000777         JMP     .-1     ;SYNCHRONIZE RTC
12 07373 060114         NIOS    XRTC    ;START RTC
13 07374 125400         INC     1,1     ;COUNTS          ;X USEC
14 07375 063514         SKPBZ   XRTC    ;                ;X USEC
15 07376 000776         JMP     .-2     ;LOOP FOR 10 MS  ;X USEC
16 07377 004463         JSR     NTTYP   ;TEST RTC STABILITY
17 07400 030516         LDA     2,NN500 ;STEP -500
18 07401 034516         LDA     3,NN9   ;8 TIMES
19 07402 141000         MOV     2,0     ;STARTING AT -500=SPEED
20 07403 175405 NXTYP:  INC     3,3,SNR ;NEXT STEP OR
21 07404 000413         JMP     NFTYP   ;TYPE OF NOVA NOT FOUND
22 07405 143020         ADDZ    2,0     ;SPEED:=SPEED-500
23 07406 107013         ADD#    0,1,SNC ;NO. OF COUNTS>=(-SPEED)
24 07407 000774         JMP     NXTYP   ;THEN GO TO NXTYP ELSE
25 07410 030510         LDA     2,NC8   ;CONVERT TIMES TO CPUNO
26 07411 173000         ADD     3,2     ;AS MENTIONED BELOW
27 07412 034425         LDA     3,NEXIS ;CHECK IN
28 07413 157000         ADD     2,3     ;THE TABEL THAT
29 07414 025400         LDA     1,0,3   ;THE FOUND CPUNO
30 07415 125004         MOV     1,1,SZR ;EXIST
31 07416 000403         JMP     NRTYP   ;YES, RETURN
32 07417 006043 NFTYP:  CCRLF           ;NO
33 07420 004430 NITYP:  JSR     NOTYP   ;GET OPERATORS CPUNO
34 07421 050403 NRTYP:  STA     2,NCTYP ;SAVE CPUNO
35 07422 024402         LDA     1,NCTYP ;RESTORE CPUNO
36 07423 002402         JMP     @NIRET  ;EXIT
37 07424 000000 NCTYP:  0
38 07425 000000 NIRET:  0
39              NOTYM:  .TXT ! SET CPUNO > AC2! ;" SET CPUNO > AC2"
   07426 051440
   07427 052105
   07430 041440
   07431 052520
   07432 047516
   07433 037040
   07434 040440
   07435 031103
   07436 000000
40                          ;RANGE          LOOPS OF X USEC CPUNO   TYPE
41 07437 007440 NEXIS:  .+1     ;   0- 500
42 07440 000001         1       ; 500-1000      7.94    12,6    0       NOVA
43 07441 000000         0       ;1000-1500                      1
44 07442 000001         1       ;1500-2000      1724     5,80   2       3608BREA
45                              ;               1770     5,65   2       3609BREA
46                              ;               1905     5,25   2       12XX
47 07443 000001         1       ;2000-2500      2222     4,5    3       SUPER
48 07444 000001         1       ;2500-3000      2702     3,7    4       SUPER SC
49                              ;               2778     3,6    4       830
50 07445 000001         1       ;3000-3500      3012     3,32   5       3603/360
51                              ;               3125     3,2    5       NOVA2-16
52                              ;               3226     3,10   5       3603/360
53                              ;               3333     3,0    5     800,820,840
54                              ;               3448     2,90   5     N-2-DATARAM
55 07446 000001         1       ;3500-4000      3703     2,7    6       NOVA2-8K
56 07447 000000         0       ;4000-4500                      7
```

```
↑ 0063  .MAIN
01
02 07450 054411 NOTYP:  STA     3,NOTRE
03 07451 006046         CDICL           ;CLEAR DIS
04 07452 006044         CDISP           ;DIS - MESSAGE
05 07453 007426         NOTYM
06 07454 006040         CMESS
07 07455 007426         NOTYM           ; SET CPUNO > AC2
08 07456 006043         CCRLF           ;NO DISATT, WAIT ROUT. NO CONSTANT
09 07457 063077         HALT            ;PUT CPUNO INTO AC2
10 07460 002401         JMP     @NOTRE  ;CONTINUE
11 07461 000000 NOTRE:  0
12
13 07462 020437 NTTYP:  LDA     0,NC125 ;TEST STABILITY
14 07463 040431         STA     0,NTDEC ;125 TIMES
15 07464 044431         STA     1,NTRES ;STORE FIRST RESULT
16 07465 054426         STA     3,NTRET
17 07466 126440 NTREP:  SUBO    1,1     ;AC1:=0
18 07467 060114         NIOS    XRTC    ;START RTC
19 07470 125400         INC     1,1     ;COUNTS
20 07471 063514         SKPBZ   XRTC    ;LOOP FOR 10 MSEC
21 07472 000776         JMP     .-2
22 07473 034422         LDA     3,NTRES ;GET FIRST COUNT
23 07474 161220         MOVZR   3,0     ;50 %
24 07475 101220         MOVZR   0,0     ;25 %
25 07476 101220         MOVZR   0,0     ;12,5 %
26 07477 101220         MOVZR   0,0     ;6,25 %
27 07500 101220         MOVZR   0,0     ;AC0:=3,12 %
28 07501 117000         ADD     0,3     ;AC3:=103,12 %
29 07502 030413         LDA     2,NTRES
30 07503 112400         SUB     0,2     ;AC2:=96,88 %
31 07504 166433         SUBZ#   3,1,SNC
32 07505 132432         SUBZ#   1,2,SZC
33 07506 000414         JMP     NYTYP   ;OUTSIDE TOLERANCE
34 07507 014405         DSZ     NTDEC   ;97 %<COUNT<103 %
35 07510 000756         JMP     NTREP   ;TRY 125 TIMES
36 07511 024404         LDA     1,NTRES ;GET FIRST RESULT
37 07512 002401         JMP     @NTRET  ;EXIT
38 07513 000000 NTRET:  0
39 07514 000000 NTDEC:  0
40 07515 000000 NTRES:  0
41       000012         .RDX 10
42 07516 177014 NN500:  -500
43 07517 177767 NN9:    -9
44 07520 000010 NC8:    8
45 07521 000175 NC125:  125
46       000010         .RDX 8
47 07522 006046 NYTYP:  CDICL
48 07523 006044         CDISP
49 07524 007531         NZTYP           ;RTC IS UNSTABLE,
50 07525 006043         CCRLF
51 07526 006040         CMESS
52 07527 007531         NZTYP           ;NO DISATT, WAIT ROUT. NO CONSTANT
53 07530 000670         JMP     NITYP
54
55               NZTYP: .TXT !RTC IS UNSTABLE,! ;"RTC IS UNSTABLE,"
   07531 052122
   07532 020103
   07533 051511
   07534 052440
   07535 051516
   07536 040524
   07537 046102
   07540 026105
   07541 000000
```

```
↑ 0064   .MAIN
01
02          000010 .RDX     8
03                 PTAB1:
04                 .TXTE?
05  07542  000006 <6><0>
06  07543  000006 <6><0>
07  07544  000006 <6><0>
08  07545  000006 <6><0>
09  07546  000006 <6><0>
10  07547  000006 <6><0>
11  07550  000006 <6><0>
12  07551  000006 <6><0>
13  07552  000006 <6><0>
14  07553  004400 <0><11>
15  07554  005000 <0><12>
16  07555  000006 <6><0>
17  07556  006000 <0><14>
18  07557  106400 <0><15>
19  07560  000006 <6><0>
20  07561  000006 <6><0>
21  07562  000006 <6><0>
22  07563  000006 <6><0>
23  07564  000006 <6><0>
24  07565  000006 <6><0>
25  07566  000006 <6><0>
26  07567  000006 <6><0>
27  07570  000006 <6><0>
28  07571  000006 <6><0>
29  07572  000006 <6><0>
30  07573  000006 <6><0>
31  07574  000006 <6><0>
32  07575  000006 <6><0>
33  07576  000006 <6><0>
34  07577  000006 <6><0>
35  07600  000006 <6><0>
36  07601  000006 <6><0>
37  07602  117400 <0><37>
38  07603  020400 <0><41>
39  07604  021000 <0><42>
40  07605  121400 <0><43>
41  07606  022000 <0><44>
42  07607  122400 <0><45>
43  07610  123000 <0><46>
44  07611  023400 <0><47>
45  07612  024000 <0><50>
46  07613  124400 <0><51>
47  07614  125000 <0><52>
48  07615  025400 <0><53>
49  07616  126000 <0><54>
50  07617  026400 <0><55>
51  07620  027000 <0><56>
52  07621  127400 <0><57>
53  07622  030000 <0><60>
54  07623  130400 <0><61>
55  07624  131000 <0><62>
56  07625  031400 <0><63>
57  07626  132000 <0><64>
58  07627  032400 <0><65>
59  07630  033000 <0><66>
60  07631  133400 <0><67>
61  07632  134000 <0><70>
62  07633  034400 <0><71>
63  07634  035000 <0><72>
64  07635  135400 <0><73>
65  07636  036000 <0><74>
66  07637  136400 <0><75>
67  07640  137000 <0><76>
68  07641  037400 <0><77>
```

```
↑ U065  .MAIN
02 07642 140000 <0><100>
03 07643 040400 <0><101>
04 07644 041000 <0><102>
05 07645 141400 <0><103>
06 07646 042000 <0><104>
07 07647 142400 <0><105>
08 07650 143000 <0><106>
09 07651 043400 <0><107>
10 07652 044000 <0><110>
11 07653 144400 <0><111>
12 07654 145000 <0><112>
13 07655 045400 <0><113>
14 07656 146000 <0><114>
15 07657 046400 <0><115>
16 07660 047000 <0><116>
17 07661 147400 <0><117>
18 07662 050000 <0><120>
19 07663 150400 <0><121>
20 07664 151000 <0><122>
21 07665 051400 <0><123>
22 07666 152000 <0><124>
23 07667 052400 <0><125>
24 07670 053000 <0><126>
25 07671 153400 <0><127>
26 07672 154000 <0><130>
27 07673 054400 <0><131>
28 07674 055000 <0><132>
29 07675 155400 <0><133>
30 07676 056000 <0><134>
31 07677 156400 <0><135>
32 07700 157000 <0><136>
33 07701 057400 <0><137>
34 07702 006000 <0><14>
35 07703 040400 <0><101>
36 07704 041000 <0><102>
37 07705 141400 <0><103>
38 07706 042000 <0><104>
39 07707 142400 <0><105>
40 07710 143000 <0><106>
41 07711 043400 <0><107>
42 07712 044000 <0><110>
43 07713 144400 <0><111>
44 07714 145000 <0><112>
45 07715 045400 <0><113>
46 07716 146000 <0><114>
47 07717 046400 <0><115>
48 07720 047000 <0><116>
49 07721 147400 <0><117>
50 07722 050000 <0><120>
51 07723 150400 <0><121>
52 07724 151000 <0><122>
53 07725 051400 <0><123>
54 07726 152000 <0><124>
55 07727 052400 <0><125>
56 07730 053000 <0><126>
57 07731 153400 <0><127>
58 07732 154000 <0><130>
59 07733 054400 <0><131>
60 07734 055000 <0><132>
61 07735 140000 <0><100>
62 07736 121400 <0><43>
63 07737 156400 <0><135>
64 07740 004400 <0><11>
65 07741 000006 <6><0>?
   07742 000000
```

```
↑ 0066   .MAIN
01
02          000012 .RDX     10
03                 PTAB2:
04                 .TXTE?
05  07743  000006 <6><0>
06  07744  000006 <6><0>
07  07745  000006 <6><0>
08  07746  000006 <6><0>
09  07747  000006 <6><0>
10  07750  000006 <6><0>
11  07751  000006 <6><0>
12  07752  000006 <6><0>
13  07753  000006 <6><0>
14  07754  004400 <0><9>
15  07755  005000 <0><10>
16  07756  000006 <6><0>
17  07757  006000 <0><12>
18  07760  106400 <0><13>
19  07761  000006 <6><0>
20  07762  000006 <6><0>
21  07763  000006 <6><0>
22  07764  000006 <6><0>
23  07765  000006 <6><0>
24  07766  000006 <6><0>
25  07767  000006 <6><0>
26  07770  000006 <6><0>
27  07771  000006 <6><0>
28  07772  000006 <6><0>
29  07773  000006 <6><0>
30  07774  000006 <6><0>
31  07775  000006 <6><0>
32  07776  000006 <6><0>
33  07777  000006 <6><0>
34  10000  000006 <6><0>
35  10001  000006 <6><0>
36  10002  000006 <6><0>
37  10003  117400 <0><31>
38  10004  036000 <0><60>
39  10005  137000 <0><62>
40  10006  040400 <0><65>
41  10007  131000 <0><50>
42  10010  034400 <0><57>
43  10011  035000 <0><58>
44  10012  136400 <0><61>
45  10013  030000 <0><48>
46  10014  130400 <0><49>
47  10015  135400 <0><59>
48  10016  027000 <0><46>
49  10017  020400 <0><33>
50  10020  127400 <0><47>
51  10021  120000 <0><32>
52  10022  133400 <0><55>
53  10023  022000 <0><36>
54  10024  122400 <0><37>
55  10025  123000 <0><38>
56  10026  023400 <0><39>
57  10027  024000 <0><40>
58  10030  124400 <0><41>
59  10031  125000 <0><42>
60  10032  025400 <0><43>
61  10033  126000 <0><44>
62  10034  026400 <0><45>
63  10035  021000 <0><34>
64  10036  121400 <0><35>
65  10037  032400 <0><53>
66  10040  132000 <0><52>
67  10041  033000 <0><54>
68  10042  037400 <0><63>
```

```
↑ 0067   .MAIN
02 10043 041000 <0><66>
03 10044 057400 <0><95>
04 10045 157000 <0><94>
05 10046 156400 <0><93>
06 10047 056000 <0><92>
07 10050 155400 <0><91>
08 10051 055000 <0><90>
09 10052 054400 <0><89>
10 10053 154000 <0><88>
11 10054 153400 <0><87>
12 10055 053000 <0><86>
13 10056 052400 <0><85>
14 10057 152000 <0><84>
15 10060 051400 <0><83>
16 10061 151000 <0><82>
17 10062 150400 <0><81>
18 10063 050000 <0><80>
19 10064 147400 <0><79>
20 10065 047000 <0><78>
21 10066 046400 <0><77>
22 10067 146000 <0><76>
23 10070 045400 <0><75>
24 10071 145000 <0><74>
25 10072 144400 <0><73>
26 10073 142400 <0><69>
27 10074 042000 <0><68>
28 10075 141400 <0><67>
29 10076 044000 <0><72>
30 10077 043400 <0><71>
31 10100 143000 <0><70>
32 10101 031400 <0><51>
33 10102 134000 <0><56>
34 10103 006000 <0><12>
35 10104 057400 <0><95>
36 10105 157000 <0><94>
37 10106 156400 <0><93>
38 10107 056000 <0><92>
39 10110 155400 <0><91>
40 10111 055000 <0><90>
41 10112 054400 <0><89>
42 10113 154000 <0><88>
43 10114 153400 <0><87>
44 10115 053000 <0><86>
45 10116 052400 <0><85>
46 10117 152000 <0><84>
47 10120 051400 <0><83>
48 10121 151000 <0><82>
49 10122 150400 <0><81>
50 10123 050000 <0><80>
51 10124 147400 <0><79>
52 10125 047000 <0><78>
53 10126 046400 <0><77>
54 10127 146000 <0><76>
55 10130 045400 <0><75>
56 10131 145000 <0><74>
57 10132 144400 <0><73>
58 10133 142400 <0><69>
59 10134 042000 <0><68>
60 10135 141400 <0><67>
61 10136 041000 <0><66>
62 10137 040400 <0><65>
63 10140 143000 <0><70>
64 10141 004400 <0><9>
65 10142 000006 <6><0>?
   10143 000000
```

```
↑ 0068   .MAIN
01
02          000012 .RDX     10
03                 PTAR3:
04                 .TXTE?
05  10144  000006 <6><0>
06  10145  000006 <6><0>
07  10146  000006 <6><0>
08  10147  000006 <6><0>
09  10150  000006 <6><0>
10  10151  000006 <6><0>
11  10152  000006 <6><0>
12  10153  000006 <6><0>
13  10154  000006 <6><0>
14  10155  004400 <0><9>
15  10156  005000 <0><10>
16  10157  000006 <6><0>
17  10160  006000 <0><12>
18  10161  106400 <0><13>
19  10162  000006 <6><0>
20  10163  000006 <6><0>
21  10164  000006 <6><0>
22  10165  000006 <6><0>
23  10166  000006 <6><0>
24  10167  000006 <6><0>
25  10170  000006 <6><0>
26  10171  000006 <6><0>
27  10172  000006 <6><0>
28  10173  000006 <6><0>
29  10174  000006 <6><0>
30  10175  000006 <6><0>
31  10176  000006 <6><0>
32  10177  000006 <6><0>
33  10200  000006 <6><0>
34  10201  000006 <6><0>
35  10202  000006 <6><0>
36  10203  000006 <6><0>
37  10204  117400 <0><31>
38  10205  134000 <0><56>
39  10206  035000 <0><58>
40  10207  136400 <0><61>
41  10210  027000 <0><46>
42  10211  032400 <0><53>
43  10212  033000 <0><54>
44  10213  034400 <0><57>
45  10214  126000 <0><44>
46  10215  026400 <0><45>
47  10216  133400 <0><55>
48  10217  125000 <0><42>
49  10220  156400 <0><93>
50  10221  025400 <0><43>
51  10222  056000 <0><92>
52  10223  031400 <0><51>
53  10224  120000 <0><32>
54  10225  020400 <0><33>
55  10226  021000 <0><34>
56  10227  121400 <0><35>
57  10230  022000 <0><36>
58  10231  122400 <0><37>
59  10232  123000 <0><38>
60  10233  023400 <0><39>
61  10234  024000 <0><40>
62  10235  124400 <0><41>
63  10236  157000 <0><94>
64  10237  057400 <0><95>
65  10240  130400 <0><49>
66  10241  030000 <0><48>
67  10242  131000 <0><50>
68  10243  135400 <0><59>
```

```
↑ 0069   .MAIN
02 10244 137000 <0><62>
03 10245 155400 <0><91>
04 10246 055000 <0><90>
05 10247 054400 <0><89>
06 10250 154000 <0><88>
07 10251 153400 <0><87>
08 10252 053000 <0><86>
09 10253 052400 <0><85>
10 10254 152000 <0><84>
11 10255 051400 <0><83>
12 10256 151000 <0><82>
13 10257 150400 <0><81>
14 10260 050000 <0><80>
15 10261 147400 <0><79>
16 10262 047000 <0><78>
17 10263 046400 <0><77>
18 10264 146000 <0><76>
19 10265 045400 <0><75>
20 10266 145000 <0><74>
21 10267 144400 <0><73>
22 10270 044000 <0><72>
23 10271 043400 <0><71>
24 10272 143000 <0><70>
25 10273 142400 <0><69>
26 10274 040400 <0><65>
27 10275 140000 <0><64>
28 10276 037400 <0><63>
29 10277 042000 <0><68>
30 10300 141400 <0><67>
31 10301 041000 <0><66>
32 10302 127400 <0><47>
33 10303 132000 <0><52>
34 10304 006000 <0><12>
35 10305 155400 <0><91>
36 10306 055000 <0><90>
37 10307 054400 <0><89>
38 10310 154000 <0><88>
39 10311 153400 <0><87>
40 10312 053000 <0><86>
41 10313 052400 <0><85>
42 10314 152000 <0><84>
43 10315 051400 <0><83>
44 10316 151000 <0><82>
45 10317 150400 <0><81>
46 10320 050000 <0><80>
47 10321 147400 <0><79>
48 10322 047000 <0><78>
49 10323 046400 <0><77>
50 10324 146000 <0><76>
51 10325 045400 <0><75>
52 10326 145000 <0><74>
53 10327 144400 <0><73>
54 10330 044000 <0><72>
55 10331 043400 <0><71>
56 10332 143000 <0><70>
57 10333 142400 <0><69>
58 10334 040400 <0><65>
59 10335 140000 <0><64>
60 10336 037400 <0><63>
61 10337 137000 <0><62>
62 10340 136400 <0><61>
63 10341 041000 <0><66>
64 10342 004400 <0><9>
65 10343 000006 <6><0>?
   10344 000000
```

```
↑ 0070  .MAIN
01
02          000010 .RDX     8
03                 PTAB4:
04                 .TXTE?
05  10345  000006 <6><0>
06  10346  000006 <6><0>
07  10347  000006 <6><0>
08  10350  000006 <6><0>
09  10351  000006 <6><0>
10  10352  000006 <6><0>
11  10353  000006 <6><0>
12  10354  000006 <6><0>
13  10355  000006 <6><0>
14  10356  004400 <0><11>
15  10357  005000 <0><12>
16  10360  000006 <6><0>
17  10361  006000 <0><14>
18  10362  106400 <0><15>
19  10363  000006 <6><0>
20  10364  000006 <6><0>
21  10365  000006 <6><0>
22  10366  000006 <6><0>
23  10367  000006 <6><0>
24  10370  000006 <6><0>
25  10371  000006 <6><0>
26  10372  000006 <6><0>
27  10373  000006 <6><0>
28  10374  000006 <6><0>
29  10375  000006 <6><0>
30  10376  000006 <6><0>
31  10377  000006 <6><0>
32  10400  000006 <6><0>
33  10401  000006 <6><0>
34  10402  000006 <6><0>
35  10403  000006 <6><0>
36  10404  000006 <6><0>
37  10405  117400 <0><37>
38  10406  022000 <0><44>
39  10407  025400 <0><53>
40  10410  157000 <0><136>
41  10411  047000 <0><116>
42  10412  156400 <0><135>
43  10413  136400 <0><75>
44  10414  127400 <0><57>
45  10415  026400 <0><55>
46  10416  027000 <0><56>
47  10417  045400 <0><113>
48  10420  137000 <0><76>
49  10421  146000 <0><114>
50  10422  145000 <0><112>
51  10423  044000 <0><110>
52  10424  046400 <0><115>
53  10425  037400 <0><77>
54  10426  140000 <0><100>
55  10427  040400 <0><101>
56  10430  041000 <0><102>
57  10431  141400 <0><103>
58  10432  042000 <0><104>
59  10433  142400 <0><105>
60  10434  143000 <0><106>
61  10435  043400 <0><107>
62  10436  144400 <0><111>
63  10437  126000 <0><54>
64  10440  122400 <0><45>
65  10441  021000 <0><42>
66  10442  030000 <0><60>
67  10443  124400 <0><51>
68  10444  125000 <0><52>
```

```
↑ 0071   .MAIN
02 10445 057400 <0><137>
03 10446 147400 <0><117>
04 10447 050000 <0><120>
05 10450 150400 <0><121>
06 10451 151000 <0><122>
07 10452 051400 <0><123>
08 10453 152000 <0><124>
09 10454 052400 <0><125>
10 10455 053000 <0><126>
11 10456 153400 <0><127>
12 10457 154000 <0><130>
13 10460 054400 <0><131>
14 10461 055000 <0><132>
15 10462 155400 <0><133>
16 10463 056000 <0><134>
17 10464 130400 <0><61>
18 10465 131000 <0><62>
19 10466 031400 <0><63>
20 10467 132000 <0><64>
21 10470 032400 <0><65>
22 10471 033000 <0><66>
23 10472 133400 <0><67>
24 10473 134000 <0><70>
25 10474 034400 <0><71>
26 10475 035000 <0><72>
27 10476 135400 <0><73>
28 10477 036000 <0><74>
29 10500 020400 <0><41>
30 10501 123000 <0><46>
31 10502 023400 <0><47>
32 10503 024000 <0><50>
33 10504 121400 <0><43>
34 10505 006000 <0><14>
35 10506 147400 <0><117>
36 10507 050000 <0><120>
37 10510 150400 <0><121>
38 10511 151000 <0><122>
39 10512 051400 <0><123>
40 10513 152000 <0><124>
41 10514 052400 <0><125>
42 10515 053000 <0><126>
43 10516 153400 <0><127>
44 10517 154000 <0><130>
45 10520 054400 <0><131>
46 10521 055000 <0><132>
47 10522 155400 <0><133>
48 10523 056000 <0><134>
49 10524 130400 <0><61>
50 10525 131000 <0><62>
51 10526 031400 <0><63>
52 10527 132000 <0><64>
53 10530 032400 <0><65>
54 10531 033000 <0><66>
55 10532 133400 <0><67>
56 10533 134000 <0><70>
57 10534 034400 <0><71>
58 10535 035000 <0><72>
59 10536 135400 <0><73>
60 10537 036000 <0><74>
61 10540 057400 <0><137>
62 10541 157000 <0><136>
63 10542 023400 <0><47>
64 10543 004400 <0><11>
65 10544 000006 <6><0>?
   10545 000000
```

```
↑ 0072  .MAIN
01
02                          ;POWER RESTART ROUTINE
03
04 10546 062677 POWON:  IORST
05 10547 006061         CWAIT           ;WAIT 3 SECONDS TO TERMINAL READY
06 10550 001236         SEC3
07 10551 020442         LDA     0,PCOTT ;PRINT 5 CR, LF FOR TTY, SILENT
08 10552 040440         STA     0,PCOUN
09 10553 006043         CCRLF
10 10554 014436         DSZ     PCOUN
11 10555 000776         JMP     .-2
12 10556 020436         LDA     0,PCH14 ;FF FOR LPT AND CLEARING SOME CRT'S
13 10557 006041         CCHAR
14 10560 006061         CWAIT           ;20 MSEC FOR CRT
15 10561 001237         SECM2
16 10562 020433         LDA     0,PCH35 ;HOME UP FOR CRT
17 10563 006041         CCHAR
18 10564 006061         CWAIT           ;20 MSEC FOR CRT
19 10565 001237         SECM2
20 10566 020430         LDA     0,PCH37 ;ERASE EOF FOR CRT
21 10567 006041         CCHAR
22 10570 006061         CWAIT           ;20 MSEC FOR CRT
23 10571 001237         SECM2
24 10572 006040         CMESS
25 10573 010623         MPOWO           ;POWER
26 10574 006043         CCRLF
27 10575 006040         CMESS
28 10576 004064         PROG            ;ACTUAL PROG NAME
29 10577 006071 RPSAQ:  CQUES
30 10600 010633         MSAQU           ;SET SWITCHES, START ADDR
31 10601 010626         DSAQU
32 10602 010617         PSAAN           ;SUGGESTED ANSWER
33 10603 006054         CTZOC
34 10604 006060         CDZOC
35 10605 006104         CGTOK           ;READ ANSWER
36 10606 000402         JMP     .+2     ;SUGGESTED ACCEPTED BY OPERATOR
37 10607 000770         JMP     RPSAQ   ;ERROR RETURN
38 10610 030075         LDA     2,DIGIN ;ANSWER INPUT'ED
39 10611 001000         JMP     0,2     ;START PROG
40
41 10612 000000 PCOUN:  0               ;COUNTER
42 10613 000005 PCOTT:  5               ;5 CR,LF
43 10614 000014 PCH14:  14              ;FF
44 10615 000035 PCH35:  35              ;HOME UP
45 10616 000037 PCH37:  37              ;ERASE EOF
46 10617 000400 PSAAN:  400             ;SUGGESTED START ADDR
47
48                          ;INITIAL START ADDRESS ROUTINE
49                          ;USED TO HELP START WITHOUT SWITCHES
50
51 10620 000757 SWISA:  JMP     RPSAQ   ;USE POWER RESTART ROUTINE
```

```
↑ 0073  .MAIN
01
02                  MCRLF:  .TXT !<15><12>! ;"<15><12>"
   10621 005015
   10622 000000
03
04                  MPOWO:  .TXT !POWER!    ;"POWER"
   10623 047520
   10624 042527
   10625 000122
05
06                  DSAGU:  .TXT !SEE3.2,SA!        ;"SEE3.2,SA"
   10626 042523
   10627 031505
   10630 031056
   10631 051454
   10632 000101
07
08                  MSAGU:  .TXT !SET SWITCHES TO CONTROL, (3.2), STARTADDR!
   10633 042523
   10634 020124
   10635 053523
   10636 052111
   10637 044103
   10640 051505
   10641 052040
   10642 020117
   10643 047503
   10644 052116
   10645 047522
   10646 026114
   10647 024040
   10650 027063
   10651 024462
   10652 020054
   10653 052123
   10654 051101
   10655 040524
   10656 042104
   10657 000122
09                          ;"SET SWITCHES TO CONTROL, (3.2), STARTADDR"
10
11                  MCPUT:  .TXT !<15><12>CPU TYPE: !       ;"<15><12>CPU TYPE: "
   10660 005015
   10661 050103
   10662 020125
   10663 054524
   10664 042520
   10665 020072
   10666 000000
```

```
↑ 0074   .MAIN
01
02                              ;ROUTINE TO EXAMINE MEMORY.
03
04 10667 165000 EXMEM:  MOV     3,1         ;
05 10670 006072         CSAMS               ;START ADDR MESSAGE
06 10671 006071 EXMMF:  CQUES
07 10672 010746         MXMMF               ;EXAMINE MEM FROM
08 10673 010757         DXMMF               ;SEE NEXT QUESTION, TO (INCL)
09 10674 010742         XFROM               ;SUGGESTED ANSWER
10 10675 006052         CTOCT
11 10676 006056         CDOCT
12 10677 006104         CGTOK               ;READ ANSWER
13 10700 000402         JMP     .+2         ;SUGGESTED ACCEPTED
14 10701 000770         JMP     EXMMF       ;ERROR RETURN
15 10702 020075         LDA     0,DIGIN     ;ANSWER INPUT'ED
16                                          ;MOVR# 0,0,SZC ;EVEN ?
17                                          ;JMP   EXMMF   ;NO, ERROR
18                                          ;LDA 1,UPPERLIMIT
19                                          ;LDA 2,LOWERLIMIT
20                                          ;ADCZ# 1,0,SNC
21                                          ;ADCZ# 0,2,SZC ;AC2=<AC0=<AC1 ?
22                                          ;JMP   EXMMF   ;OUTSIDE LIMITS
23 10703 040440         STA     0,FMADR     ;INPUT ACCEPTED
24 10704 006071 EXMMT:  CQUES
25 10705 010764         MXMMT               ;TO INCL.
26 10706 010764         MXMMT               ;USE THE SAME MESS AT DIS
27 10707 010744         XTOIN               ;SUGGESTED ANSWER
28 10710 006052         CTOCT
29 10711 006056         CDOCT
30 10712 006104         CGTOK               ;READ ANSWER
31 10713 000402         JMP     .+2         ;SUGGESTED ACCEPTED
32 10714 000770         JMP     EXMMT       ;ERROR RETURN
33 10715 020075         LDA     0,DIGIN     ;ANSWER INPUT'ED
34 10716 040427         STA     0,LMADR     ;INPUT ACCEPTED
35 10717 006043 EXPRT:  CCRLF
36 10720 006046         CDICL
37 10721 024422         LDA     1,FMADR
38 10722 006052         CTOCT
39 10723 006056         CDOCT
40 10724 030417         LDA     2,FMADR
41 10725 025000         LDA     1,0,2
42 10726 006052         CTOCT
43 10727 006056         CDOCT
44 10730 006047         CDATT
45 10731 024414         LDA     1,LMADR
46 10732 030411         LDA     2,FMADR
47 10733 010410         ISZ     FMADR       ;TO NEXT LOC
48 10734 000401         JMP     .+1
49 10735 132414         SUB#    1,2,SZR     ;LAST LOC ?
50 10736 000761         JMP     EXPRT       ;NO, NEXT
51 10737 006043         CCRLF               ;YES, PROGRAM FINISHED
52 10740 002401         JMP     @.+1
53 10741 010620         SWISA               ;RESTART MAIN PROGRAM
54
55 10742 000034 XFROM:  34                  ;FIRST MEM LOC QUES
56 10743 000000 FMADR:  0                   ;ANSWER
57 10744 000037 XTOIN:  37                  ;LAST MEM LOC QUES, INCL
58 10745 000000 LMADR:  0                   ;ANSWER
```

```
↑ 0075  .MAIN
01
02                      MXMMF:  .TXT !EXAMINE MEM FROM !        ;"EXAMINE MEM FROM "
   10746 054105
   10747 046501
   10750 047111
   10751 020105
   10752 042515
   10753 020115
   10754 051106
   10755 046517
   10756 000040
03
04                      DXMMF:  .TXT !X M FROM !        ;"X M FROM "
   10757 020130
   10760 020115
   10761 051106
   10762 046517
   10763 000040
05
06                      MXMMT:  .TXT !TO INCL. !        ;"TO INCL. "
   10764 047524
   10765 044440
   10766 041516
   10767 027114
   10770 000040
07
08                      MDMMC:  .TXT !DEPOSIT: !        ;"DEPOSIT: "
   10771 042504
   10772 047520
   10773 044523
   10774 035124
   10775 000040
09
10                      MDMMF:  .TXT !FROM LOC !        ;"FROM LOC "
   10776 051106
   10777 046517
   11000 046040
   11001 041517
   11002 000040
11
12 11003 177777 DPCON:  177777          ;DEPOSIT CONTENT QUES
13 11004 000000 CMADR:  0               ;ANSWER
```

```
↑ 0076  .MAIN
01
02                          ;ROUTINE TO DEPOSIT IN MEMORY.
03
04 11005 165000 DPMEM:  MOV     3,1
05 11006 006072         CSAMS           ;START ADDR MESSAGE
06 11007 006071 DPMMC:  CQUES
07 11010 010771         MDMMC           ;DEPOSIT:
08 11011 010771         MDMMC
09 11012 011003         DPCON           ;SUGGESTED ANSWER
10 11013 006052         CTOCT
11 11014 006056         CDOCT
12 11015 006104         CGTOK           ;READ ANSWER
13 11016 000402         JMP     .+2     ;SUGGESTED ACCEPTED
14 11017 000770         JMP     DPMMC   ;ERROR RETURN
15 11020 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
16 11021 040763         STA     0,CMADR ;INPUT ACCEPTED
17 11022 006071 DPMMF:  CQUES
18 11023 010776         MDMMF           ;FROM LOC
19 11024 010776         MDMMF
20 11025 010742         XFROM           ;SUGGESTED ANSWER
21 11026 006052         CTOCT
22 11027 006056         CDOCT
23 11030 006104         CGTOK           ;READ ANSWER
24 11031 000402         JMP     .+2     ;SUGGESTED ACCEPTED
25 11032 000770         JMP     DPMMF   ;ERROR RETURN
26 11033 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
27 11034 040707         STA     0,FMADR ;INPUT ACCEPTED
28 11035 006071 DPMMT:  CQUES
29 11036 010764         MXMMT           ;TO INCL.
30 11037 010764         MXMMT
31 11040 010744         XTOIN           ;SUGGESTED ANSWER
32 11041 006052         CTOCT
33 11042 006056         CDOCT
34 11043 006104         CGTOK           ;READ ANSWER
35 11044 000402         JMP     .+2     ;SUGGESTED ACCEPTED
36 11045 000770         JMP     DPMMT   ;ERROR RETURN
37 11046 020075         LDA     0,DIGIN ;ANSWER INPUT'ED
38 11047 040676         STA     0,LMADR ;INPUT ACCEPTED
39 11050 024734         LDA     1,CMADR
40 11051 030672 DPPRT:  LDA     2,FMADR
41 11052 045000         STA     1,0,2
42 11053 034672         LDA     3,LMADR
43 11054 010667         ISZ     FMADR   ;TO NEXT LOC
44 11055 000401         JMP     .+1
45 11056 172414         SUB#    3,2,SZR ;LAST LOC ?
46 11057 000772         JMP     DPPRT   ;NO, NEXT
47 11060 002401         JMP     @.+1    ;YES, PROGRAM FINISHED
48 11061 010620         SWISA           ;RESTART MAIN PROGRAM
```

```
↑ 0077  .MAIN
01
02 11062 000000 RQUES:  0                ;RETURN ADDR QUES ROUTINE
03 11063 000000 QUESA:  0                ;SUGG. ANSWER
04
05              MXQUE:  .TXT !?!         ;"?"
   11064 000077
06
07              MX2SP:  .TXT !  !        ;"2 SPACE"
   11065 020040
   11066 000000
08
09                      ;ROUTINE TO OUTPUT QUESTIONS.
10                      ;HOW TO USE, SEE EXMEM.
11                      ;CALL   CQUES
12                      ;       MQUES    ;LABEL TEXT TTO/LPT 1,9,17,25 LETTERS
13                      ;       DQUES    ;LABEL TEXT DIS ALLWAYS 9 LETTERS
14                      ;       AQUES    ;LABEL SUGGESTED ANSWER
15                      ;       CTOCT    ;PRINT ROUTINE FOR SUGG. ANSWER
16                      ;       CDOCT    ;DISPLAY ROUTINE FOR SUGG. ANSWER
17                      ;       RETURN   ;TO RELEVANT CALL OF INPUT ROUTINE.
18
19 11067 054773 XQUES:  STA      3,RQUES
20 11070 025400         LDA      1,0,3   ;1. PARAM
21 11071 044417         STA      1,QUESM
22 11072 025401         LDA      1,1,3   ;2. PARAM
23 11073 044413         STA      1,QUESD
24 11074 027402         LDA      1,@2,3  ;3. PARAM
25 11075 044075         STA      1,DIGIN
26 11076 044765         STA      1,QUESA
27 11077 025403         LDA      1,3,3   ;4. PARAM
28 11100 044412         STA      1,QUEST
29 11101 025404         LDA      1,4,3   ;5. PARAM
30 11102 044411         STA      1,QUESS
31 11103 006046.        CDICL
32 11104 006043         CCRLF
   11105 006044         CDISP
   11106 000000 QUESD:  0                ;DISPLAY MESSAGE
35 11107 006040         CMESS
36 11110 000000 QUESM:  0                ;PRINT MESSAGE
37 11111 024752         LDA      1,QUESA
38 11112 000000 QUEST:  0                ;NUMBER PRINT ROUTINE TTO/LPT
39 11113 000000 QUESS:  0                ;NUMBER PRINT ROUTINE DIS
40 11114 006044         CDISP
41 11115 011064         MXQUE
42 11116 006040         CMESS
43 11117 011064         MXQUE            ;?
44 11120 006040         CMESS
45 11121 011065         MX2SP            ;2 SPACE
46 11122 034740         LDA      3,RQUES
47 11123 001405         JMP      5,3     ;BYPASS PARAM., RETURN
48
49
50              ;TAPE 3
51
52              .EOT
```

```
0078  .MAIN
01
02                ;TAPE 4
03
04                ; GENERAL INPUT-ROUTINES.          VERSION  760119 BY HH
05
06                ; GETDC          GET DECIMAL NUMBER
07                ; GETOK          GET OKTAL NUMBER
08                ; GETBI          GET BINARY NUMBER
09                ; GETSC          GET SINGLE CHARACTER
10                ; GETTX          GET TEXT STRING
11
12
13                ; ABSTRACTS:
14
15                ;    AFTER CALLING DIFFERENT ROUTINES THE OPERATER CAN KEY-IN A
16                ; DEC.-, OCT.- OR A BINARY NUMBER, A CHARACTER OR A TEXT-STRING.
17                ;    THE NUMBER OR THE CHARACTER WILL AFTER A TERMINATION-INPUT
18                ; BE STORED IN "DIGIN". THE TEXT-STRING IS PACKED IN A BUFFER
19                ; CALLED "TEXIN". THE ADDRESS OF TEXIN IS STORED IN "DIGIN"
20                ; FOR INDIRECT USE.
21
22
23
24                ; DESCRIPTION:
25
26                ; "NUMBER"-INPUTROUTINES:
27
28                ; >GETDC< FOR A DECIMALNUMBER DX, WHERE -32768 <= DX <=32767.
29                ; ACCEPTED INPUTS ARE +,-,0,1 ....9.
30
31                ; >GETOK< FOR AN OCTALNUMBER OX, WHERE 0 <= OX <= 177777.
32                ; ACCEPTED INPUTS ARE 0,1, .... 7.
33
34                ; >GETBI< FOR A BINARY NUMBER BX, WHERE 0 <= BX <= 11111111.
35                ; ACCEPTED INPUTS ARE 0,1.
36
37                ; THE SPACE CHAR IS BLIND.
38
39
40                ; A TERMINATION-INPUT WILL STORE THE ENTIRE NUMBER IN THE PAGE-
41                ; ZERO-ADRESS "DIGIN".
42
43
44
45                ; "CHARACTER"-INPUTROUTINE:
46
47                ; >GETSC< FOR A SINGLE CHARACTER. ACCEPTED INPUT IS
48                ; CHARACTERS WITH THE OCTAL ASCII-CODE 11, 33, 40 - 176.
49
50                ; A TERMINATION-INPUT WILL STORE THE CHARACTER-ASCII-CODE
51                ; (WITHOUT PARITY) IN THE RIGTH HALF OF "DIGIN".
52
53
54
55                ; "TEXT"-INPUTROUTINE:
56
57                ; >GETTX< FOR A CHARACTERSTRING OF MAX. 80 CHARACTERS.
58                ; ACCEPTED INPUT IS CHARACTERS WITH THE OCTAL CODE 11, 40 - 176.
59
60                ; AFTER A TERMINATION THE STRING IS PACKED R-L AND THE 3 LAST
61                ; BYTES IN THE STRING WILL ALWAYS CONTENT CR,LF,0. THEN IT IS
62                ; STORED IN "TEXIN", FIRST UP TO 80 BYTES OF INPUT, FOLLOWED
63                ; BY THE 3 TERMINATION BYTES.
```

```
↑ 0079  .MAIN
01
02
03
04                ; OPERATING PROCEDURE:
05
06                ;    THE ROUTINES ARE CALLED BY THE INSTRUCTION: C>NAME<,
07                ; WHICH EQUALS JSR "ROUTINE". AFTER RECEIVING A LEGAL NUMBER,
08                ; CHARACTER OR TEXT-STRING FOLLOWED BY A TERMINATIOR, THE
09                ; PROGRAM RETURNS TO CALL+3. WHEN A FORMAT- OR OVERFLOWERROR
10                ; OCCURS, THE RETURN IS CALL+2. WHEN ONLY A TERMINATOR IS
11                ; INPUT, THE RETURN IS TO CALL+1.
12
13                ; CALL  CGTDC             ;OR CGTOK, CGTBI, CGTSC, CGTTX
14                ;       TERM RETURN
15                ;       ERROR RETURN
16                ;       NORMAL RETURN
17
18                ; TERMINATOR:
19                ;   IN ALL ROUTINES THE INPUTS: NL, LF, CR OR FF  WILL
20                ; TERMINATE THE MESSAGE.
21
22                ;   EVERY ERRORRETURN WILL INITIALISE THE BUFFERS, COUNTERS
23                ; ETC., AND THE RE-INPUT'ING THEN HAVE TO START AT THE LAST
24                ; TERMINATION POINT. AN * IS PRINTED.
25
26                ; CANCEL:
27                ;   IT'S POSSIBLE TO CANCEL THE LAST-KEYED DIGIT OR CHARACTER
28                ; BY INPUT'ING A RUBOUT, DEL OR CAN CHARACTER .THE PROGRAM
29                ; ECHOES THEN A "$", AND WAITS FOR A NEW INPUT. IT IS NOT
30                ; POSSIBLE TO CANCEL MORE THAN ONE CHAR. AN ATTEMT WILL
31                ; GIVE ERROR RETURN.
32
33                ; THE NULL CHAR IS BLIND.
34
35                ; IF A CHARACTER IS NOT TERMINATOR, BLIND, CANCEL OR
36                ; ACCEPTED, IT IS ILLEGAL AND CAUSES ERROR RETURN.
```

```
↑ 0080  .MAIN
01
02                ; SUBROUTINES USED BY ALL INPUTROUTINES
03
04                ; GETCH GETS A CHARACTER FROM TTI OR NUK TO AC0
05
06 11124 054456 GETCH:  STA     3,GETRE
07 11125 004443         JSR     BZNUK    ; CHECK BUSY NUK
08 11126 004431         JSR     INTTI    ; INPUTDEVICE = TTI
09 11127 004446         JSR     INNUK    ; INPUTDEVICE = NUK
10 11130 024461         LDA     1,RAZER ; READ ASCII FROM NUK
11 11131 034454         LDA     3,RALIF ; AC0=KEY
12 11132 054452         STA     3,RADYN
13 11133 101122 RANDI:  MOVZL   0,0,SZC ;
14 11134 000421         JMP     RAEND    ; DIGIT FOUND
15 11135 125400         INC     1,1      ; AC1 = DIGIT
16 11136 014446         DSZ     RADYN    ; 0-9
17 11137 000774         JMP     RANDI
18 11140 126400         SUB     1,1
19 11141 101122         MOVZL   0,0,SZC
20 11142 024445         LDA     1,RAPLU ; AC1 = +
21 11143 101122         MOVZL   0,0,SZC
22 11144 024444         LDA     1,RAMIN ; AC1 = -
23 11145 101122         MOVZL   0,0,SZC
24 11146 024437         LDA     1,RALIF ; AC1 = LF
25 11147 101122         MOVZL   0,0,SZC
26 11150 024436         LDA     1,RASPA ; AC1 = SP
27 11151 101122         MOVZL   0,0,SZC
28 11152 024440         LDA     1,RACAN ; AC1 = CAN
29 11153 127405         AND     1,1,SNR
30 11154 063077         HALT             ; NO KEYS
31 11155 121000 RAEND:  MOV     1,0      ; AC1 TO AC0
32 11156 002424         JMP     @GETRE
33
34 11157 063610 INTTI:  SKPDN   XTTI     ;IS TTI KEY PRESSED ?
35 11160 001400         JMP     0,3      ; NO, CHECK OTHER INPUT DEVICE
36 11161 024422         LDA     1,HC177 ; YES, GET CHAR
37 11162 060410         DIA     0,XTTI
38 11163 123400         AND     1,0
39 11164 060110         NIOS    XTTI
40 11165 101015         MOV#    0,0,SNR ; NULL CHAR ?
41 11166 000771         JMP     INTTI
42 11167 002413         JMP     @GETRE
43
44 11170 126400 BZNUK:  SUB     1,1      ; AC1:=0
45 11171 060434 AANUK:  DIA     0,NUK    ; TEST BUZY NUK
46 11172 122414         SUB#    1,0,SZR ; IS AC0=0  ?
47 11173 000776         JMP     AANUK    ; NO - KEYBOARD NOT READY
48 11174 001400         JMP     0,3      ; YES, RETURN.
49
50 11175 126400 INNUK:  SUB     1,1      ; AC1:= 0
51 11176 060434 BBNUK:  DIA     0,NUK    ; GET KEY BITS
52 11177 122415         SUB#    1,0,SNR ; ARE THEY ZERO ?
53 11200 001776         JMP     -2,3     ; YES, OTHER INPUT, TEST TTI
54 11201 001400         JMP     0,3      ; NO, KEY DEPRESSED
55
56 11202 000000 GETRE:  0
57 11203 000177 HC177:  177
58 11204 000000 RADYN:  0
59 11205 000012 RALIF:  12
60 11206 000040 RASPA:  40
61 11207 000053 RAPLU:  53
62 11210 000055 RAMIN:  55
63 11211 000060 RAZER:  60
64 11212 000030 RACAN:  30
```

```
↑ 0081  .MAIN
01
02 11213 030551 DELTE:  LDA     2,HC30  ; THE DELTE ROUTINE LOOKS FOR RUBOUT
03 11214 112415         SUB#    0,2,SNR ; IS AC0= 30 ?
04 11215 000405         JMP     DELOV   ; YES - RUBOUT
05 11216 030465         LDA     2,HC177 ; NO
06 11217 112415         SUB#    0,2,SNR ; IS AC0= 177 ?
07 11220 000402         JMP     DELOV   ; YES - RUBOUT
08 11221 001401         JMP     1,3     ; NO
09
10 11222 054462 DELOV:  STA     3,DELRE ; CHECK FOR TOO MANY RUBOUTS
11 11223 030460         LDA     2,LASTN ; OR THE FIRST CHAR IS RUBOUT
12 11224 151102         MOVL    2,2,SZC ; IS LASTN = 177777 ?
13 11225 000417         JMP     ILLGR   ; YES - FIRST IS RUBOUT
14 11226 151102         MOVL    2,2,SZC ; IS LASTN = 077777 ?
15 11227 000415         JMP     ILLGR   ; YES - TOO MANY RUBOUTS
16 11230 152220         ADCZR   2,2     ; NO - PLACE 077777 IN LASTN
17 11231 050452         STA     2,LASTN ; TO INDICATE RUBOUT
18 11232 020534         LDA     0,HC44  ; TYPE $
19 11233 004402         JSR     TYPIN
20 11234 002450         JMP     @DELRE  ; RETURN + 1
21
22 11235 054406 TYPIN:  STA     3,TYPRE ; TYPE THE CHAR IN AC0 ON TTY/DIS
23 11236 004430         JSR     TERMT   ; TERMINATOR ?
24 11237 002404         JMP     @TYPRE  ; YES, DO NOT COPY
25 11240 006045         CDOUT           ; TRY TO TYPE CHAR ON DISPLAY
26 11241 006041         CCHAR           ; TRY TO TYPE CHAR ON TTO
27 11242 002401         JMP     @TYPRE
28 11243 000000 TYPRE:  0
29
30 11244 020522 ILLGR:  LDA     0,HC44  ; ILLEGAL ROUTINE. TYPE $
31 11245 004770 ILLEG:  JSR     TYPIN   ; COPY LAST CHAR
32 11246 020521         LDA     0,HC52
33 11247 004766         JSR     TYPIN   ; TYPE *
34 11250 126400         SUB     1,1     ; AC1:= 0
35 11251 044075         STA     1,DIGIN ; DIGIN:= 0
36 11252 046427         STA     1,@XTXCO ; TEXIN:= CR,LF,0
37 11253 006427         JSR     @XTXND  ; FOR EMPTY TEXTSTRING
38 11254 010424         ISZ     INRET   ; ILLEGAL OR OVERFLOW RETURN
39 11255 000403         JMP     ONTER   ; IS TO CALL + 2
40
41 11256 010422 INTER:  ISZ     INRET   ; TERMINATION RETURN TO CALL+3
42 11257 010421         ISZ     INRET   ; FOR ACCEPTED INPUT
43 11260 006040 ONTER:  CMESS           ; TYPE CR,LF WITH MESS TO WAIT
44 11261 010621         MCRLF           ; FOR DEVICE READY.
45 11262 006047         CDATI           ; RETURN IS TO CALL + 1
46 11263 006044         CDISP           ; FOR NO INPUT, ONLY TERMINATION
47 11264 010621         MCRLF           ; (FOR ANSWER QUE. WITH AN OKAY).
48 11265 002413         JMP     @INRET  ; RETURN TO MAIN PROGRAM
49
50 11266 030474 TERMT:  LDA     2,HC13  ; THE TERMT ROUTINE LOOKS FOR TERM.
51 11267 112415         SUB#    0,2,SNR ; IS AC0=13 ?
52 11270 001401         JMP     1,3     ; YES - IT IS NOT A TERM
53 11271 030472         LDA     2,HC15  ; NO
54 11272 112433         SUBZ#   0,2,SNC ; IS AC0=<15 ?
55 11273 001401         JMP     1,3     ; NO, IT'S NOT A TERMINATOR
56 11274 030465         LDA     2,HC11  ; YES
57 11275 112432         SUBZ#   0,2,SZC ; IS AC0=< 11 ?
58 11276 001401         JMP     1,3     ; YES - IT IS NOT A TERMINATOR
59 11277 001400         JMP     0,3     ; NO - IT IS A TERMINATOR, RETURN
60
61 11300 000000 INRET:  0               ; RETURN ADDRESS TO MAIN PROGRAM
62 11301 012207 XTXCO:  TXCOU           ; ADDRESS OF TEXTBUFFER COUNTER
63 11302 012260 XTXND:  TXEND           ; ADDRESS OF TEXT TERM ROUTINE
64 11303 000000 LASTN:  0               ; X77777 FOR FIRST/RUBOUT, ELSE = CHAR
65 11304 000000 DELRE:  0               ; RETURN ADDRESS FOR DELTE
```

```
↑ 0082  .MAIN
01
02                ; DECIMAL-NUMBER-INPUTROUTINE.
03
04 11305 054773 GETDC:  STA     3,INRET ; INITIALIZE
05 11306 126000         ADC     1,1     ; AC1:=177777
06 11307 046540         STA     1,@LAST4 ; SET LAST4
07 11310 044534         STA     1,DSIGN ; SET DSIGN TO +
08 11311 126400         SUB     1,1     ; AC1:= 0
09 11312 044534         STA     1,NUMB4 ; CLEAR NUMBER
10 11313 004611 GETD1:  JSR     GETCH   ; GET CHARACTER TO AC0
11 11314 004752         JSR     TERMT   ; TEST FOR TERMINATOR
12 11315 000456         JMP     TERMDC  ; IT IS A TERMINATOR
13 11316 004675         JSR     DELTE   ; IT'S NOT A TERMINATOR- IS IT A DEL ?
14 11317 000774         JMP     GETD1   ; IT IS A DELETE-CHAR
15 11320 004412         JSR     LETE4   ; IT'S NOT A DELETE-CHAR
16 11321 004510         JSR     CHRAN   ; PUT CHAR IN RANGE 0-9
17 11322 004527         JSR     OFTDC   ; TEST FOR OVERFLOW
18 11323 000403         JMP     DELDC   ; FIRST
19 11324 000402         JMP     DELDC   ; LAST IS RUBOUT
20 11325 004462         JSR     PLADC   ; ADD LAST4 TO NUMB4 * 10-DEC.
21 11326 024517 DELDC:  LDA     1,DCDIG ; STORE NEW DIGIT IN LAST4
22 11327 046520         STA     1,@LAST4
23 11330 004705 ECHO4:  JSR     TYPIN   ; TYPE NEW CHARACTER
24 11331 000762         JMP     GETD1   ; GET NEXT CHARACTER
25
26                ; SUBROUTINES USED BY GETDC.
27
28 11332 054426 LETE4:  STA     3,LERE4 ; LEGAL TEST
29 11333 034432         LDA     3,HC40
30 11334 111000         MOV     0,2     ; AC2:= CHAR
31 11335 024513         LDA     1,HC60
32 11336 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
33 11337 000771         JMP     ECHO4   ; YES - THE CHAR IS A SPACE
34 11340 034427         LDA     3,HC52  ; NO
35 11341 116432         SUBZ#   0,3,SZC ; IS AC0 > 52 ?
36 11342 004703         JSR     ILLEG   ; NO - THE CHAR IS NOT LEGAL
37 11343 034427         LDA     3,HC72  ; YES
38 11344 116033         ADCZ#   0,3,SNC ; IS AC0 < 72 ?
39 11345 004700         JSR     ILLEG   ; NO - NOT LEGAL
40 11346 034422         LDA     3,HC53  ; YES
41 11347 116415         SUB#    0,3,SNR ; IS AC0 >< 53 ?
42 11350 133000         ADD     1,2     ; NO - IT'S A PLUSSIGN, ADD 60
43 11351 034420         LDA     3,HC55  ; YES
44 11352 116415         SUB#    0,3,SNR ; IS AC0 >< 55 ?
45 11353 133000         ADD     1,2     ; NO - IT'S A MINUSSIGN ADD 60
46 11354 034474         LDA     3,HC60  ; YES
47 11355 156032         ADCZ#   2,3,SZC ; IS AC0 >= 60 ?
48 11356 004667         JSR     ILLEG   ; NO NOT LEGAL
49 11357 002401         JMP     @LERE4  ; YES - IT'S A VALID DIGIT
50 11360 000000 LERE4:  0
51
52 11361 000011 HC11:   11
53 11362 000013 HC13:   13
54 11363 000015 HC15:   15
55 11364 000030 HC30:   30
56 11365 000040 HC40:   40
57 11366 000044 HC44:   44
58 11367 000052 HC52:   52
59 11370 000053 HC53:   53
60 11371 000055 HC55:   55
61 11372 000072 HC72:   72
```

```
↑ 0083  .MAIN
01
02 11373 004456 TERMDC: JSR     OFTDC   ; TERMINATION, TEST OVERFLOW
03 11374 000664         JMP     ONTER   ; TERM IS FIRST
04 11375 000402         JMP     TERDC   ; LAST IS RUBOUT
05 11376 004411         JSR     PLADC   ; ADD LAST DIGIT TO NUMB4
06 11377 024447 TERDC:  LDA     1,NUMB4
07 11400 030444         LDA     2,DSIGN
08 11401 125015         MOV#    1,1,SNR ; IS AC1 = 0 ?
09 11402 000403         JMP     TERM1   ; YES - ZERO ALLWAYS POSITIVE
10 11403 151014         MOV#    2,2,SZP ; IS DSIGN = 0 ?
11 11404 124400         NEG     1,1     ; NO - IT IS A NEGATIVE NUMBER
12 11405 044075 TERM1:  STA     1,DIGIN ; YES - IT'S A POSITIVE NUMBER
13 11406 000650         JMP     INTER   ; OUTPUT TERM CHAR
14
15 11407 054421 PLADC:  STA     3,REPL4 ; PLACE LAST DIGIT/SIGN
16 11410 026437         LDA     1,@LAST4
17 11411 152400         SUB     2,2     ; AC2:= 0
18 11412 034756         LDA     3,HC53  ; LAST = + ?
19 11413 136415         SUB#    1,3,SNR ; NO
20 11414 000412         JMP     PLASI   ; YES, SET DSIGN
21 11415 152520         SUBZL   2,2     ; AC2:= 1
22 11416 034753         LDA     3,HC55  ; LAST = - ?
23 11417 136415         SUB#    1,3,SNR ; NO
24 11420 000406         JMP     PLASI   ; YES, SET DSIGN
25 11421 030425         LDA     2,NUMB4
26 11422 004413         JSR     MULTE   ; MULTIPLY NUMB4 WITH 10-DEC.
27 11423 133000         ADD     1,2     ; ADD LAST4 TO NUMB4*10
28 11424 050422         STA     2,NUMB4 ; PLACE NEW NUMBER
29 11425 002403         JMP     @REPL4
30 11426 050416 PLASI:  STA     2,DSIGN ; PLACE NEW SIGN
31 11427 002401         JMP     @REPL4  ; RETURN
32 11430 000000 REPL4:  0
33
34 11431 024417 CHRAN:  LDA     1,HC60  ; PUT CHAR IN RANGE 0 - 9 OR
35 11432 132400         SUB     1,2     ; PUT CHAR "+" TO 53 OR
36 11433 050412         STA     2,DCDIG ; PUT CHAR "-" TO 55
37 11434 001400         JMP     0,3     ; + AND - WAS ADDED 60 IN LEGAL TEST
38
39 11435 044406 MULTE:  STA     1,MULSA ; NUMBER TO BE MULT. BY 10 IN AC2
40 11436 145120         MOVZL   2,1     ; AC2=NUM, AC1=2*NUM
41 11437 125120         MOVZL   1,1     ; AC2=NUM, AC1=4*NUM
42 11440 133120         ADDZL   1,2     ; AC2=10*NUM, AC1=4*NUM
43 11441 024402         LDA     1,MULSA
44 11442 001400         JMP     0,3     ; RESULT IN AC2
45
46 11443 000000 MULSA:  0
47 11444 000000 DSIGN:  0               ; SIGNFLAG, 0=+, 1=-, 177777=NO SIGN=+
48 11445 000000 DCDIG:  0
49 11446 000000 NUMB4:  0
50 11447 011303 LAST4:  LASTN
51 11450 000060 HC60:   60
```

```
↑ 0084  .MAIN
01
02 11451 054450 OFTDC:  STA     3,RETOF ; OVERFLOW TEST FOR SIGN AND
03 11452 026775         LDA     1,@LAST4 ; (PREV*10)+LAST < LIMIT DX
04 11453 125102         MOVL    1,1,SZC ; IS THE LAST RUBBED OUT ?
05 11454 002445         JMP     @RETOF  ; OR IS IT THE FIRST ? YES, FIRST
06 11455 010444         ISZ     RETOF
07 11456 125102         MOVL    1,1,SZC
08 11457 002442         JMP     @RETOF  ; YES, RUBOUT
09 11460 034710         LDA     3,HC53  ; NO, IS LAST A + ?
10 11461 136415         SUB#    1,3,SNR ; NO
11 11462 000432         JMP     OFTSI   ; YES
12 11463 034706         LDA     3,HC55  ; IS LAST A - ?
13 11464 136415         SUB#    1,3,SNR ; NO
14 11465 000427         JMP     OFTSI   ; YES
15 11466 176400         SUB     3,3     ; AC3:= 0
16 11467 030755         LDA     2,DSIGN ; IS DSIGN = 177777
17 11470 151112         MOVL#   2,2,SZC ; NO
18 11471 054753         STA     3,DSIGN ; YES, SET FIRST +
19 11472 030754         LDA     2,NUMB4 ; AC2:=PREVIOUS (PREV)
20 11473 155120         MOVZL   2,3     ; PREV*2
21 11474 175120         MOVZL   3,3     ; PREV*4
22 11475 175112         MOVL#   3,3,SZC ; PREV*8>=65536 IF
23 11476 006556         JSR     @XILLG  ; PREV >= 8192
24 11477 173122         ADDZL   3,2,SZC ; PREV*10>=65540 IF
25 11500 006554         JSR     @XILLG  ; PREV >= 6554
26 11501 151112         MOVL#   2,2,SZC ; PREV*10>=32770 IF
27 11502 006552         JSR     @XILLG  ; PREV >= 3277
28 11503 133000         ADD     1,2     ; (PREV*10)+LAST>32767
29 11504 151113         MOVL#   2,2,SNC ; (<=32760)+0...9>32767 ?
30 11505 000412         JMP     OFTRE   ; NO, RETURN
31 11506 034736         LDA     3,DSIGN ; YES, TEST +- 32768,32769
32 11507 175005         MOV     3,3,SNR ; TEST SIGN
33 11510 006544         JSR     @XILLG  ; +32768, +32769
34 11511 151134         MOVZL#  2,2,SZR ; -32768, OK RETURN
35 11512 006542         JSR     @XILLG  ; -32769
36 11513 000404         JMP     OFTRE   ; RETURN
37
38 11514 030730 OFTSI:  LDA     2,DSIGN ; OVERFLOW TEST SIGN
39 11515 151113         MOVL#   2,2,SNC ; IS IT FIRST SIGN ? YES
40 11516 006536         JSR     @XILLG  ; NO
41 11517 010402 OFTRE:  ISZ     RETOF   ; PASS RUBOUT/FIRST RETURN
42 11520 002401         JMP     @RETOF  ; RETURN
43
44 11521 000000 RETOF:  0               ; RETURN ADDRESS
```

```
↑ 0085  .MAIN
01
02                ; CHARACTER-INPUT-ROUTINE
03
04 11522 056537 GETSC:  STA     3,@XINRT
05 11523 126000         ADC     1,1      ; AC1:=177777
06 11524 046440         STA     1,@LAST1 ; SET LAST1
07 11525 126400         SUB     1,1      ; AC1:= 0
08 11526 044435         STA     1,NUMSC  ; CLEAR NUMSC
09 11527 006530 GETS1:  JSR     @XGTCH   ; GET CHAR TO ACO
10 11530 006526         JSR     @XTRMT   ; IS IT A TERMINATOR ?
11 11531 000450         JMP     TERMSC   ; YES
12 11532 006523         JSR     @XDLTE   ; NO - IS IT A DEL CHAR ?
13 11533 000774         JMP     GETS1    ; YES
14 11534 004410         JSR     LETE1    ; NO - IS IT LEGAL ?
15 11535 004430         JSR     OFTSC    ; IT'S A LEGAL CHAR, TEST OVERFLOW
16 11536 000403         JMP     DELSC    ; FIRST
17 11537 000402         JMP     DELSC    ; LAST IS RUBOUT
18 11540 004436         JSR     PLASC    ; PLACE LAST IN NUMSC
19 11541 042423 DELSC:  STA     0,@LAST1 ; STORE NEW CHAR IN LAST1
20 11542 006516         JSR     @XTYPN   ; TYPE NEW CHARACTER
21 11543 000764         JMP     GETS1    ; GET NEXT INPUT
22
23                ; SUBROUTINES USED BY GETSC
24
25 11544 054413 LETE1:  STA     3,LERE1  ; LEGAL TEST
26 11545 030413         LDA     2,H1C40
27 11546 034414         LDA     3,H1C11
28 11547 024412         LDA     1,H1C33
29 11550 106415         SUB#    0,1,SNR  ; IS ACO = 33 ?
30 11551 002406         JMP     @LERE1   ; YES
31 11552 116415         SUB#    0,3,SNR  ; NO - IS ACO = 11 ?
32 11553 002404         JMP     @LERE1   ; YES
33 11554 112032         ADCZ#   0,2,SZC  ; NO - IS ACO >= 40 ?
34 11555 006477         JSR     @XILLG   ; NO
35 11556 002401         JMP     @LERE1   ; YES - IT IS A LEGAL CHAR
36 11557 000000 LERE1:  0
37
38 11560 000040 H1C40:  40
39 11561 000033 H1C33:  33
40 11562 000011 H1C11:  11
41 11563 000000 NUMSC:  0
42 11564 011303 LAST1:  LASTN
43
44 11565 026777 OFTSC:  LDA     1,@LAST1 ; OVERFLOW TEST
45 11566 125102         MOVL    1,1,SZC  ; IS THE LAST CHAR RUBBED OUT ?
46 11567 001400         JMP     0,3      ; OR IS IT THE FIRST ? YES, FIRST
47 11570 125102         MOVL    1,1,SZC
48 11571 001401         JMP     1,3      ; YES, RUBOUT
49 11572 030771         LDA     2,NUMSC  ; NUMSC = 0 FOR FIRST
50 11573 151004         MOV     2,2,SZR  ; IS IT THE FIRST CHAR ?
51 11574 006460         JSR     @XILLG   ; NO, OVERFLOW
52 11575 001402         JMP     2,3      ; YES, RETURN
53
54 11576 032766 PLASC:  LDA     2,@LAST1 ; PLACE LAST CHAR
55 11577 050764         STA     2,NUMSC
56 11600 001400         JMP     0,3
57
58 11601 004764 TERMSC: JSR     OFTSC    ; TERMINATION, TEST OVERFLOW
59 11602 002461         JMP     @XONTR   ; TERM IS FIRST
60 11603 000402         JMP     TERSC    ; LAST IS RUBOUT
61 11604 004772         JSR     PLASC    ; STORE LAST CHAR IN NUMSC
62 11605 024756 TERSC:  LDA     1,NUMSC  ; MOVE CHAR TO DIGIN
63 11606 044075         STA     1,DIGIN
64 11607 002453         JMP     @XINTR   ; OUTPUT TERM CHAR
```

```
↑ 008o  .MAIN
01
02                ; OCTAL-NUMBER-INPUTROUTINE
03
04 11610 056451 GETOK:  STA     3,@XINRT
05 11611 126000         ADC     1,1     ; AC1:=177777
06 11612 046506         STA     1,@LAST2
07 11613 126400         SUB     1,1     ; AC1:= 0
08 11614 044505         STA     1,NUMB2
09 11615 006442 GET01:  JSR     @XGTCH  ; GET CHAR TO AC0
10 11616 006440         JSR     @XTRMT  ; IS IT A TERMINATOR ?
11 11617 000503         JMP     TERMOK  ; YES
12 11620 006435         JSR     @XDLTE  ; NO - IS IT A DEL CHAR ?
13 11621 000774         JMP     GET01   ; YES
14 11622 004412         JSR     LETE2   ; NO - IS IT LEGAL ?
15 11623 004441         JSR     CHRA2   ; IT IS A LEGAL DIGIT
16 11624 004445         JSR     OFTE2   ; TEST FOR OVERFLOW
17 11625 000403         JMP     DELOK   ; FIRST
18 11626 000402         JMP     DELOK   ; LAST IS RUBOUT
19 11627 004460         JSR     PLAOK   ; ADD LAST2 TO NUMB2 * 8-DEC
20 11630 024423 DELOK:  LDA     1,OKDIG ; STORE NEW DIGIT IN LAST2
21 11631 046467         STA     1,@LAST2
22 11632 006426 ECHO2:  JSR     @XTYPN  ; TYPE NEW CHARACTER
23 11633 000762         JMP     GET01   ; GET NEXT CHARACTER
24
25                ; SUBROUTINES USED BY GETOK
26
27 11634 054413 LETE2:  STA     3,LERE2 ; LEGAL TEST
28 11635 034413         LDA     3,H2C40
29 11636 116415         SUB#    0,3,SNR ; IS AC0 = 40 ?
30 11637 000773         JMP     ECHO2   ; YES - THE CHAR IS A SPACE
31 11640 034411         LDA     3,H2C60 ; NO
32 11641 116032         ADCZ#   0,3,SZC ; IS AC0 >= 60 ?
33 11642 006412         JSR     @XILLG  ; NO - NOT LEGAL
34 11643 034407         LDA     3,H2C70 ; YES
35 11644 116033         ADCZ#   0,3,SNC ; IS AC0 < 70 ?
36 11645 006407         JSR     @XILLG  ; NO - NOT LEGAL
37 11646 002401         JMP     @LERE2  ; YES - IT IS A VALID DIGIT
38 11647 000000 LERE2:  0
39
40 11650 000040 H2C40:  40
41 11651 000060 H2C60:  60
42 11652 000070 H2C70:  70
43 11653 000000 OKDIG:  0
44
45 11654 011245 XILLG:  ILLEG
46 11655 011213 XDLTE:  DELTE
47 11656 011266 XTRMT:  TERMT
48 11657 011124 XGTCH:  GETCH
49 11660 011235 XTYPN:  TYPIN
50 11661 011300 XINRT:  INRET
51 11662 011256 XINTR:  INTER
52 11663 011260 XONTR:  ONTER
53
54 11664 024765 CHRA2:  LDA     1,H2C60
55 11665 111000         MOV     0,2
56 11666 132400         SUB     1,2     ; PUT CHAR IN RANGE 0-7
57 11667 050764         STA     2,OKDIG
58 11670 001400         JMP     0,3
```

```
↑ 0087  .MAIN
01
02 11671 054415 OFTE2:  STA     3,REOF2 ; OVERFLOW TEST
03 11672 030427         LDA     2,NUMB2
04 11673 026425         LDA     1,@LAST2 ; IS THE LAST DIGIT RUBBED OUT ?
05 11674 125102         MOVL    1,1,SZC ; OR IS IT THE FIRST ?
06 11675 002411         JMP     @REOF2  ; YES, FIRST
07 11676 010410         ISZ     REOF2
08 11677 125102         MOVL    1,1,SZC
09 11700 002406         JMP     @REOF2  ; YES, RUBOUT
10 11701 034416         LDA     3,OF2CO ; NO
11 11702 156433         SUBZ#   2,3,SNC ; IS NUMB2 <= 17777 ?
12 11703 006751         JSR     @XILLG  ; NO - OVERFLOW
13 11704 010402         ISZ     REOF2   ; YES
14 11705 002401         JMP     @REOF2  ; RETURN
15 11706 000000 REOF2:  0
16
17 11707 026411 PLAOK:  LDA     1,@LAST2 ; PLACE LAST DIGIT
18 11710 030411         LDA     2,NUMB2
19 11711 151120         MOVZL   2,2     ; MULTIPLY NUMB2 WITH 8-DEC
20 11712 151120         MOVZL   2,2
21 11713 151120         MOVZL   2,2
22 11714 133000         ADD     1,2     ; ADD LAST DIGIT
23 11715 050404         STA     2,NUMB2
24 11716 001400         JMP     0,3
25
26 11717 017777 OF2CO:  17777
27 11720 011303 LAST2:  LASTN
28 11721 000000 NUMB2:  0
29
30 11722 004747 TERMOK: JSR     OFTE2   ; TERMINATION, TEST OVERFLOW
31 11723 002740         JMP     @XONTR  ;TERM IS FIRST
32 11724 000402         JMP     TEROK   ; LAST IS RUBOUT
33 11725 004762         JSR     PLAOK   ; ADD LAST DIGIT TO NUMB2
34 11726 024773 TEROK:  LDA     1,NUMB2
35 11727 044075         STA     1,DIGIN
36 11730 002732         JMP     @XINTR  ; OUTPUT TERM CHAR
```

```
↑ 0088  .MAIN
01
02                ; BINARY-NUMBER-INPUTROUTINE
03
04 11731 056730 GETBI:  STA     3,@XINRT
05 11732 126000         ADC     1,1      ; AC1:=177777
06 11733 046474         STA     1,@LAST3
07 11734 126400         SUB     1,1      ; AC1:= 0
08 11735 044473         STA     1,NUMB3
09 11736 006721 GETB1:  JSR     @XGTCH   ; GET CHAR TO AC0
10 11737 006717         JSR     @XTRMT   ; IS IT A TERMINATOR ?
11 11740 000471         JMP     TERMBI   ; YES
12 11741 006714         JSR     @XDLTE   ; NO - IS IT A DEL CHAR ?
13 11742 000774         JMP     GETB1    ; YES
14 11743 004412         JSR     LETE3    ; NO - IS IT LEGAL ?
15 11744 004431         JSR     CHRA3    ; IT IS A LEGAL DIGIT
16 11745 004435         JSR     OFTE3    ; TEST FOR OVERFLOW
17 11746 000403         JMP     DELBI    ; FIRST
18 11747 000402         JMP     DELBI    ; LAST IS RUBOUT
19 11750 004450         JSR     PLABI    ; ADD LAST3 TO NUMB3 * 2-DEC
20 11751 024423 DELBI:  LDA     1,BIDIG  ; STORE NEW DIGIT IN LAST3
21 11752 046455         STA     1,@LAST3
22 11753 006705 ECHO3:  JSR     @XTYPN   ; TYPE NEW CHARACTER
23 11754 000762         JMP     GETB1    ; GET NEXT CHARACTER
24
25                ; SUBROUTINES USED BY GETBI
26
27 11755 054413 LETE3:  STA     3,LERE3  ; LEGAL TEST
28 11756 034413         LDA     3,H3C40
29 11757 116415         SUB#    0,3,SNR  ; IS AC0 = 40 ?
30 11760 000773         JMP     ECHO3    ; YES - THE CHAR IS A SPACE
31 11761 034411         LDA     3,H3C60  ; NO
32 11762 116032         ADCZ#   0,3,SZC  ; IS AC0 >= 60 ?
33 11763 006671         JSR     @XILLG   ; NO - NOT LEGAL
34 11764 034407         LDA     3,H3C62  ; YES
35 11765 116033         ADCZ#   0,3,SNC; IS AC0 < 62 ?
36 11766 006666         JSR     @XILLG   ; NO - NOT LEGAL
37 11767 002401         JMP     @LERE3   ; YES
38 11770 000000 LERE3:  0
39
40 11771 000040 H3C40:  40
41 11772 000060 H3C60:  60
42 11773 000062 H3C62:  62
43 11774 000000 BIDIG:  0
44
45
46 11775 024775 CHRA3:  LDA     1,H3C60  ; PUT CHAR IN RANGE 0-1
47 11776 111000         MOV     0,2
48 11777 132400         SUB     1,2
49 12000 050774         STA     2,BIDIG
50 12001 001400         JMP     0,3
```

```
↑ 0089  .MAIN
01
02 12002 054415 OFTE3:  STA     3,REOF3 ; OVERFLOW TEST
03 12003 030425         LDA     2,NUMB3
04 12004 026423         LDA     1,@LAST3 ; IS THE LAST DIGIT RUBBED OUT ?
05 12005 125102         MOVL    1,1,SZC ; OR IS IT THE FIRST ?
06 12006 002411         JMP     @REOF3  ; YES, FIRST
07 12007 010410         ISZ     REOF3
08 12010 125102         MOVL    1,1,SZC
09 12011 002406         JMP     @REOF3  ; YES, RUBOUT
10 12012 034414         LDA     3,OF3CO ; NO
11 12013 156433         SUBZ#   2,3,SNC ; IS NUMB3 <= 177 ?
12 12014 006640         JSR     @XILLG  ; NO - OVERFLOW
13 12015 010402         ISZ     REOF3   ; YES
14 12016 002401         JMP     @REOF3  ; RETURN
15 12017 000000 REOF3:  0
16
17 12020 026407 PLABI:  LDA     1,@LAST3 ; PLACE LAST DIGIT
18 12021 030407         LDA     2,NUMB3
19 12022 151120         MOVZL   2,2     ; MULTIPLY WITH 2-DEC
20 12023 133000         ADD     1,2     ; ADD LAST DIGIT
21 12024 050404         STA     2,NUMB3
22 12025 001400         JMP     0,3
23
24 12026 000177 OF3CO:  177
25 12027 011303 LAST3:  LASTN
26 12030 000000 NUMB3:  0
27
28 12031 004751 TERMBI: JSR     OFTE3   ; TERMINATION, TEST OVERFLOW
29 12032 002631         JMP     @XONTR  ; TERM IS FIRST
30 12033 000402         JMP     TERBI   ; LAST IS RUBOUT
31 12034 004764         JSR     PLABI   ; ADD LAST DIGIT TO NUMB3
32 12035 024773 TERBI:  LDA     1,NUMB3 ; MOVE THE BINARY NUMBER TO DIGIN
33 12036 044075         STA     1,DIGIN
34 12037 002623         JMP     @XINTR  ; OUTPUT TERM CHAR
```

```
↑ 0090  .MAIN
01
02                ; TEXT-INPUT-ROUTINE
03
04 12040 056555 GETTX:  STA     3,@YINRT
05 12041 126000         ADC     1,1       ; AC1:=177777
06 12042 046543         STA     1,@LAST5
07 12043 126400         SUB     1,1       ; AC1:= 0
08 12044 044543         STA     1,TXCOU
09 12045 006543 GETT1:  JSR     @YGTCH    ; GET CHAR TO AC0
10 12046 006543         JSR     @YTRMT    ; IS IT A TERMINATOR ?
11 12047 000551         JMP     TERMXT    ; YES
12 12050 006542         JSR     @YDLTE    ; NO - IS IT A DEL CHAR ?
13 12051 000774         JMP     GETT1     ; YES
14 12052 004410         JSR     LETE5     ; NO - IS IT LEGAL ?
15 12053 004423         JSR     OFTTX     ; TEST FOR OVERFLOW
16 12054 000403         JMP     DELTX     ; FIRST
17 12055 000402         JMP     DELTX     ; LAST IS RUBOUT
18 12056 004505         JSR     PLATX     ; STORE LAST5 IN TEXTBUFFER
19 12057 042526 DELTX:  STA     0,@LAST5  ; STORE NEW CHAR IN LAST5
20 12060 006533         JSR     @YTYPN    ; TYPE NEW CHAR
21 12061 000764         JMP     GETT1     ; GET NEXT CHARACTER
22
23                ; SUBROUTINES USED BY GETTX
24
25 12062 054410 LETE5:  STA     3,LERE5 ; LEGAL TEST
26 12063 034410         LDA     3,H5C11
27 12064 030410         LDA     2,H5C40
28 12065 116415         SUB#    0,3,SNR ; IS AC0 = 11 ?
29 12066 002404         JMP     @LERE5  ; YES
30 12067 112032         ADCZ#   0,2,SZC ; NO - IS AC0 >= 40 ?
31 12070 006524         JSR     @YILLG  ; NO - THE CHAR IS NOT LEGAL
32 12071 002401         JMP     @LERE5  ; YES
33 12072 000000 LERE5:  0
34
35 12073 000011 H5C11:  11
36 12074 000040 H5C40:  40
37 12075 000117 H5C79:  117             ; TEXTBUFFER LENGTH-1, OKTAL
38
39 12076 026507 OFTTX:  LDA     1,@LAST5 ; OVERFLOW TEST
40 12077 125102         MOVL    1,1,SZC ; IS THE LAST CHAR RUBBED OUT ?
41 12100 001400         JMP     0,3     ; OR IS IT THE FIRST ? YES, FIRST
42 12101 125102         MOVL    1,1,SZC
43 12102 001401         JMP     1,3     ; YES, RUBOUT
44 12103 024504         LDA     1,TXCOU ; NO
45 12104 030771         LDA     2,H5C79
46 12105 132433         SUBZ#   1,2,SNC ; IS TXCOU <= 79-DEC
47 12106 006506         JSR     @YILLG  ; NO - OVERFLOW
48 12107 001402         JMP     2,3     ; YES, RETURN
49
50 12110 012111 TEXIA:  .+1             ; ADDRESS OF TEXTBUFFER
51       000051 TEXIN:  .BLK 51         ; TEXTBUFFER 80 BYTES + CR,LF
52 12162 000000 TEXEN:  0               ; END OF TEXTBUFFER: NUL CHAR
```

```
↑ 0091  .MAIN
01
02                                              ; PLACE LAST CHAR
03 12163 054421 PLATX:  STA     3,REPL5 ; LAST5 TO CHAHA OR
04 12164 026421         LDA     1,@LAST5 ; LAST5 + CHAHA TO BUFFER
05 12165 010422         ISZ     TXCOU   ; INCREMENT CHARACTER-COUNTER
06 12166 030421         LDA     2,TXCOU
07 12167 151213         MOVR#   2,2,SNC ; IS TXCOU EVEN ?
08 12170 000403         JMP     STOTX   ; YES
09 12171 044415         STA     1,CHAHA ; NO - MOVE LAST5 TO CHAHA
10 12172 002412         JMP     @REPL5
11 12173 030413 STOTX:  LDA     2,CHAHA ; CHAHA IS CHAR HALF BUFFER
12 12174 125320         MOVZS   1,1
13 12175 147000         ADD     2,1     ; AC1 = "LAST5,CHAHA"
14 12176 030411         LDA     2,TXCOU ; CALCULATE ADRESS OF BUFFEREND+1
15 12177 151220         MOVZR   2,2
16 12200 034710         LDA     3,TEXIA
17 12201 173000         ADD     3,2     ; AC2:= TEXIN+TXCOU/2
18 12202 045377         STA     1,-1,2  ; STORE LAST TWO CHAR INTO BUFFEREND+1
19 12203 002401         JMP     @REPL5
20 12204 000000 REPL5:  0
21
22 12205 011303 LAST5:  LASTN
23 12206 000000 CHAHA:  0               ; THE CHAR BEFORE LAST5
24 12207 000000 TXCOU:  0               ; TEXT COUNTER BYTE ADDRESS
25
26 12210 011124 YGTCH:  GETCH
27 12211 011266 YTRMT:  TERMT
28 12212 011213 YDLTE:  DELTE
29 12213 011235 YTYPN:  TYPIN
30 12214 011245 YILLG:  ILLEG
31 12215 011300 YINRT:  INRET
32 12216 011256 YINTR:  INTER
33 12217 011260 YONTR:  ONTER
34
35 12220 004656 TERMXT: JSR     OFTTX   ; TERMINATION, TEST OVERFLOW
36 12221 002776         JMP     @YONTR  ; TERM IS FIRST
37 12222 000402         JMP     TERTX   ; LAST IS RUBOUT
38 12223 004740         JSR     PLATX   ; PLACE THE CHAR BEFORE TERM CHAR
39 12224 024763 TERTX:  LDA     1,TXCOU ; IN LAST5
40 12225 125213         MOVR#   1,1,SNC ; IS TXCOU EVEN ?
41 12226 000417         JMP     NBEVEN  ; YES
42 12227 020443         LDA     0,H5C15 ; NO
43 12230 101320         MOVZS   0,0
44 12231 030755         LDA     2,CHAHA
45 12232 113000         ADD     0,2     ; AC2 = "CR,CHAR"
46 12233 010754         ISZ     TXCOU   ; INCREMENT CHAR COUNT FOR CR
47 12234 004413         JSR     COUDI
48 12235 004415         JSR     STABU   ; STORE AC2 INTO BUFFER
49 12236 030435         LDA     2,H5C12 ; AC2 = " 0,LF"
50 12237 004410         JSR     COUDI
51 12240 125400         INC     1,1     ; INCREMENT BUFF ADDR FOR 0,LF
52 12241 004411         JSR     STABU   ; STORE AC2 INTO BUFFER
53 12242 030646 OUT5:   LDA     2,TEXIA
54 12243 050075         STA     2,DIGIN ; ADDRESS OF TEXTBUFFER
55 12244 002752         JMP     @YINTR  ; OUTPUT TERM CHAR
56
57 12245 004413 NBEVEN: JSR     TXEND   ; STORE LF,CR,0,0
58 12246 000774         JMP     OUT5    ; TERMINATE
```

```
↑ 0092  .MAIN
01
02 12247 024740 COUDI:  LDA     1,TXCOU
03 12250 125220          MOVZR   1,1       ; DIVIDE TXCOU WITH 2
04 12251 001400          JMP     0,3
05
06 12252 054405 STABU:   STA     3,RET5
07 12253 034635          LDA     3,TEXIA  ; CALCULATE ADRESS OF BUFFEREND
08 12254 137000          ADD     1,3
09 12255 051777          STA     2,-1,3   ; STORE AC2 INTO BUFFER
10 12256 002401          JMP     @RET5
11 12257 000000 RET5:    0
12
13 12260 054411 TXEND:   STA     3,TXNDR
14 12261 030413          LDA     2,HLFCR  ; AC2 = "LF,CR"
15 12262 004765          JSR     COUDI
16 12263 125400          INC     1,1
17 12264 004766          JSR     STABU    ; STORE AC2 INTO BUFFER
18 12265 125400          INC     1,1
19 12266 152400          SUB     2,2      ; AC2:= 0
20 12267 004763          JSR     STABU    ; STORE "0,0" INTO BUFFER
21 12270 002401          JMP     @TXNDR
22 12271 000000 TXNDR:   0
23
24 12272 000015 H5C15:   15
25 12273 000012 H5C12:   12
26 12274 005015 HLFCR:   5015
27
28               ;TAPE 4
29
30
31               .EOT
```

```
  0093  .MAIN
01
02                ;TAPE 5
03
04 12275 000000 LASTP:  0                ;LAST LOC OF PROGRAM
05
06       001400 .END REBIN
  0094  .MAIN

AANUK   011171      80/45   80/47
ACTO    001071      19/03   19/16
ACT6    001072      19/09   19/17
ACTN    001054      18/29   19/02
ADDEC   000132       9/11   39/06   39/29   52/14
ADDRS   001711      26/46   27/18   27/28   27/38   27/39   28/02
ADINC   000131       9/10   39/04   39/27   52/12
ALPTT   002255      31/57   33/48
AMEND   001721      28/15   28/23
ARESW   001322      23/18   23/24
BBNUK   011176      80/51
BBUF    004373       9/08   52/07   52/19
BC20    001667      27/32   27/46
BC377   001617      26/55   26/59
BEG     003055      10/61   38/48   40/53
BEG1    003053      38/44   38/46
BEG2    003046      37/15   37/33   38/41
BEGIN   002747      37/14   37/32   37/39
BEND    001712      28/03
BIDIG   011774      88/20   88/43   88/49
BINAD   001547      24/19   26/09   33/03
BINFI   001375      23/63   24/17   33/02
BINLA   001376      24/00   24/12   32/41
BITLL   000174       9/45   34/43
BITNA   000172       9/43   34/47   35/26
BITNO   000171       9/42   34/31
BITUL   000173       9/44   34/42
BLOCK   001625      26/58   27/10   27/12   27/44
BMEND   001732      28/19   28/24
BRESW   001356      23/41   23/47
BSTRP   001703      27/60
BTEST   001600      26/42   27/21
BUILD   001550      26/12   27/17   27/19   27/37
BZNUK   011170      80/07   80/44
BZOUT   000437      14/18   14/25   18/18   23/57
CALIC   000123       9/04   37/62
CALIR   000121       9/02   37/49   40/05   40/11
CALIT   000122       9/03   38/01   40/33   40/39
CCHAR   006041       8/24   22/40   37/26   38/24   57/55   72/13   72/17   72/21
                    81/26
CCRLF   006043       8/26   23/30   24/50   28/43   34/05   34/28   34/49   38/08
                    38/19   38/34   38/35   38/38   41/05   52/32   62/32   63/08
                    63/50   72/09   72/26   74/35   74/51   77/32
CDATT   006047       8/30   22/44   28/53   32/38   74/44   81/45
CDBIN   006055       8/36
CDDEC   006057       8/38   33/50
CDICL   006046       8/29   22/41   24/47   28/42   52/28   63/03   63/47   74/36
                    77/31
CDIS    001773      28/39   28/62
CDISP   006044       8/27   22/42   24/32   24/41   24/48   28/49   32/29   58/03
                    63/04   63/48   77/33   77/40   81/46
CDOCT   006056       8/37   24/38   28/47   32/36   58/09   61/17   74/11   74/29
                    74/39   74/43   76/11   76/22   76/33
CDOUT   006045       8/28   57/54   61/20   81/25
CDZOC   006060       8/39   72/34
CGTBI   006103       8/51
CGTDC   006105       8/53   33/51   34/15   34/38
CGTOK   006104       8/52   34/59   72/35   74/12   74/30   76/12   76/23   76/34
CGTSC   006106       8/54
```

CGTTX	006107	8/55							
CH14	000142	9/19	37/25	38/23					
CHAAT	006050	8/31							
CHAHA	012206	91/09	91/11	91/23	91/44				
CHAR0	000770	16/12	16/31	16/35	18/11	18/50			
CHAR1	000704	17/19							
CHAR2	000675	17/12	17/15						
CHAR3	000713	17/27	17/35						
CHAR4	000721	17/21	17/30	17/33					
CHAR7	000776	17/28	18/17						
CHCR	001076	17/58	18/63	19/22					
CHFLG	000547	15/08	15/24	15/31	15/34	15/41	16/19		
CHINH	000455	14/40	16/04						
CHKER	001664	26/45	27/31	27/43					
CHLF	001077	17/55	18/03	19/01	19/23	20/37			
CHMAS	000461	14/09	14/44						
CHMIN	000546	15/17	15/40						
CHORZ	000775	17/27	17/46	18/05	18/16				
CHPLU	000545	15/07	15/25	15/39					
CHRA2	011664	86/15	86/54						
CHRA3	011775	88/15	88/46						
CHRAN	011431	82/16	83/34						
CHRET	000661	16/63	17/03	17/06	17/09	17/24	17/26	17/32	
CHSAV	000774	17/11	17/17	18/15					
CHSIG	000544	15/10	15/18	15/32	15/38				
CHSP	000662	15/19	15/28	16/09	17/00	17/33			
CHSPA	000663	15/09	17/01						
CHTAB	000767	17/19	18/10						
CITAB	000126	9/07	38/46						
CMADR	011004	75/13	76/16	76/39					
CMESS	006040	8/23	23/31	24/34	24/43	24/51	28/51	28/54	32/31
		34/06	34/11	34/13	34/29	34/34	34/36	34/50	34/55
		34/57	38/09	38/14	38/36	41/06	52/29	58/05	63/06
		63/51	72/24	72/27	77/35	77/42	77/44	81/43	
CMSK	001162	19/36	20/26						
COMPA	000157	9/32	39/17	39/40	40/07	40/35			
COMPC	000154	9/29	35/19	36/02					
COMPI	000153	9/28	35/15	35/61					
COMPN	000155	9/30	35/07	39/14	39/37				
COMPR	000160	9/33	36/04	37/63	40/31				
COMPS	000156	9/31	35/11	39/16	39/39				
COUDI	012247	91/47	91/50	92/02	92/15				
COUNT	001670	27/22	27/34	27/40	27/47				
CPU0	006473	54/34							
CPU00	006472	54/16	54/32						
CPU1	006474	54/35							
CPU10	006503	54/42							
CPU11	006504	54/43							
CPU12	006505	54/44							
CPU13	006506	54/45							
CPU14	006507	54/46							
CPU15	006510	54/47							
CPU16	006511	54/48							
CPU17	006512	54/49							
CPU2	006475	54/36							
CPU20	006513	54/50							
CPU21	006514	54/51							
CPU22	006515	54/52							
CPU23	006516	54/53							

GETTX	012040	8/06	90/04				
GETYP	001423	24/23	24/27				
GMEND	001716	24/63	28/12	32/42			
GTCHR	001557	26/13	26/15	26/22	26/54	27/10	27/13
GTTTI	001570	26/25	26/32				
H1C11	011562	85/27	85/40				
H1C33	011561	85/26	85/39				
H1C40	011560	85/26	85/38				
H2C40	011650	86/28	86/40				
H2C60	011651	86/31	86/41	86/54			
H2C70	011652	86/34	86/42				
H3C40	011771	88/28	88/40				
H3C60	011772	88/31	88/41	88/46			
H3C62	011773	88/34	88/42				
H5C11	012073	90/26	90/35				
H5C12	012273	91/49	92/25				
H5C15	012272	91/42	92/24				
H5C40	012074	90/27	90/36				
H5C79	012075	90/37	90/45				
HAATT	001221	7/36	21/35				
HC11	011361	81/56	82/52				
HC13	011362	81/50	82/53				
HC15	011363	81/53	82/54				
HC177	011203	80/36	80/57	81/05			
HC30	011364	81/02	82/55				
HC40	011365	82/29	82/56				
HC44	011366	81/18	81/30	82/57			
HC52	011367	81/32	82/34	82/58			
HC53	011370	82/40	82/59	83/18	84/09		
HC55	011371	82/43	82/60	83/22	84/12		
HC60	011450	82/31	82/46	83/34	83/51		
HC72	011372	82/37	82/61				
HCHSP	000175	9/46	35/33	35/35	35/40		
HLFCR	012274	92/14	92/26				
HMEND	000074	7/56	28/26	32/33			
I1	004164	42/27	49/10				
I10	004247	42/38	50/09				
I11	004256	42/39	51/02				
I12	004265	42/40	51/03				
I13	004274	42/41	51/04				
I14	004303	42/44	51/05				
I15	004310	42/45	51/06				
I16	004315	42/46	51/07				
I17	004321	42/47	51/08				
I18	004326	42/28	51/09				
I19	004333	42/29	51/10				
I2	004171	42/30	49/11				
I20	004340	42/42	51/11				
I21	004344	42/43	52/02				
I22	004350	42/48	52/03				
I23	004355	42/49	52/04				
I24	004364	42/50	52/05				
I3	004176	42/31	50/02				
I4	004203	42/32	50/03				
I5	004212	42/33	50/04				
I6	004221	42/34	50/05				
I7	004227	42/35	50/06				
I8	004235	42/36	50/07				
I9	004242	42/37	50/08				

IANSW	000407	11/07	11/12						
IBEG	000306	10/61	41/19						
IBUFF	000127	9/08	38/51	39/21	39/54				
ICHAR	000041	7/29	8/24						
ICPN	000305	10/60	35/25	35/56					
ICRLF	000043	7/31	8/26						
IDATT	000047	7/35	8/30						
IDBIN	000055	7/41	8/36						
IDDEC	000057	7/43	8/38						
IDICL	000046	7/34	8/29						
IDISP	000044	7/32	8/27						
IDIVD	000070	7/52	8/47						
IDIVS	000067	7/51	8/46						
IDOCT	000056	7/42	8/37						
IDOUT	000045	7/33	8/28						
IDX0	000020	7/15	58/17	58/43	60/28	60/39			
IDX1	000021	7/16	58/15	58/59	59/28	59/31	59/33	60/26	60/55
IDX2	000022	7/17	59/47	59/58	60/00	61/03	61/06		
IDX3	000023	7/18	58/19	59/14	60/30				
IDX4	000024	7/19	38/47	38/48	39/08	39/31	39/61	41/02	41/12
IDX5	000025	7/20	39/05	39/28	41/52	52/13			
IDZOC	000060	7/44	8/39						
IFRAM	000406	11/06	11/11						
IGNOR	001612	26/43	26/54	26/57					
IGTBI	000103	8/02	8/51						
IGTDC	000105	8/04	8/53						
IGTOK	000104	8/03	8/52						
IGTSC	000106	8/05	8/54						
IGTTX	000107	8/06	8/55						
IHAAT	000050	7/36	8/31						
IHEAD	000310	10/63	40/61						
ILIST	000405	11/05	11/10						
ILL	001067	19/04	19/05	19/06	19/07	19/08	19/10	19/11	19/12
		19/14							
ILLEG	011245	81/31	82/36	82/39	82/48	86/45	91/30		
ILLGR	011244	81/13	81/15	81/30					
IMEND	002312	32/26	32/42	32/53					
IMESS	000040	7/28	8/23						
IMEST	003402	41/03	42/27						
IMULT	000066	7/50	8/45						
INDAD	007066	59/15	59/30						
INDW1	007035	59/03							
INDW2	007263	61/02							
INIT1	003066	39/02							
INIT2	003114	39/25	40/29						
INNUK	011175	80/09	80/50						
INRET	011300	81/38	81/41	81/42	81/48	81/61	82/04	86/50	91/31
INSAD	007060	59/10	59/23						
INSTA	007045	59/11	59/12						
INSTB	007046	59/12	59/13						
INSTR	007036	59/05	60/15						
INSW1	007034	59/02							
INSW2	007262	61/01							
INTER	011256	81/41	83/13	86/51	91/32				
INTTI	011157	80/08	80/34	80/41					
INXW5	007140	58/18	60/11	60/29					
IQUES	000071	7/53	8/48						
IRESA	000077	8/00	8/03						
IRESW	000073	7/55	8/50						

ISAMS	000072	7/54	8/49						
ISKPT	003352	39/09	39/32	42/02					
ITABE	003320	9/07	39/62	41/13	41/39				
ITABL	003321	39/09	39/32	41/03	41/40				
ITBIN	000051	7/37	8/32						
ITDEC	000053	7/39	8/34						
ITEST	000404	11/04	11/09						
ITIMS	000064	7/48	8/43						
ITIRO	000065	7/49	8/44						
ITISK	000063	7/47	8/42						
ITNSE	000112	8/59	40/20	40/25	40/46				
ITOCT	000052	7/38	8/33						
ITTOL	000111	8/58	40/16						
ITYPE	000042	7/30	8/25						
ITZOC	000054	7/40	8/35						
IWAIT	000061	7/45	8/40						
IWAOP	000062	7/46	8/41						
JMEND	001730	28/22	32/43						
KCPU	006416	53/33							
KCP1	006417	53/34							
KCP10	006426	53/41							
KCP11	006427	53/42							
KCP12	006430	53/43							
KCP13	006431	53/44							
KCP14	006432	53/45							
KCP15	006433	53/46							
KCP16	006434	53/47							
KCP17	006435	53/48							
KCP2	006420	53/35							
KCP20	006436	53/49							
KCP21	006437	53/50							
KCP22	006440	53/51							
KCP23	006441	53/52							
KCP24	006442	53/53							
KCP25	006443	53/54							
KCP3	006421	53/36							
KCP4	006422	53/37							
KCP5	006423	53/38							
KCP6	006424	53/39							
KCP7	006425	53/40							
KEYA	007267	61/06	61/11	61/14					
KEYB	007275	61/08	61/12						
KEYS	007264	59/43	61/02						
KINC	006444	53/22	53/23	53/56					
KINDI	006415	53/18	53/31						
KSTAC	001372	23/28	23/60						
LALOC	001373	23/61	24/14	24/36					
LAPRG	001374	23/62	24/16						
LAST1	011564	85/06	85/19	85/42	85/44	85/54			
LAST2	011720	86/06	86/21	87/04	87/17	87/27			
LAST3	012027	88/06	88/21	89/04	89/17	89/25			
LAST4	011447	82/06	82/22	83/16	83/50	84/03			
LAST5	012205	90/06	90/19	90/39	91/04	91/22			
LASTN	011303	81/11	81/17	82/00	83/50	85/42	87/27	89/25	91/22
LASTP	012275	93/04							
LERE1	011557	85/25	85/30	85/32	85/35	85/36			
LERE2	011647	86/27	86/37	86/38					
LERE3	011770	88/27	88/37	88/38					
LERE4	011360	82/28	82/49	82/50					

LERE5	012072	90/25	90/29	90/32	90/33				
LETE1	011544	85/14	85/25						
LETE2	011634	86/14	86/27						
LETE3	011755	88/14	88/27						
LETE4	011332	82/15	82/28						
LETE5	012062	90/14	90/25						
LLPTT	002257	31/59	33/56						
LMADR	010745	74/34	74/45	74/58	76/38	76/42			
LMSK	001163	18/27	20/27						
LOADA	002337	32/58	33/03						
LOADB	002314	31/26	32/48						
LOADF	002336	32/56	33/02						
LOADR	002327	32/59	33/01						
LOCKA	007310	61/25							
LSTSW	001366	23/14	23/18	23/21	23/25	23/33	23/44	23/56	24/28
MBILO	001510	24/42	24/44	25/10					
MCMEX	002264	32/16	33/31						
MCMNM	002273	32/22	32/26	32/50	33/24				
MCPUT	010660	58/04	58/06	73/11					
MCRLF	010621	28/55	52/30	73/02	81/44	81/47			
MDCTR	006647	56/31	56/55	57/01					
MDMMC	010771	75/08	76/07	76/08					
MDMMF	010776	75/10	76/18	76/19					
MELOC	001470	7/06	25/04						
MESCH	000434	14/16	14/21						
MESSA	000422	14/10	14/17						
MIMEX	002313	32/23	32/43	32/51	33/25	33/32			
MLLOC	001501	24/33	24/35	25/08	32/30	32/32			
MLOOP	006624	56/32	56/36						
MLPTT	002223	31/29	33/46	33/47					
MODAT	001414	24/20	24/26						
MOFID	002246	31/40	31/49						
MOGTT	002250	31/36	31/51						
MOLAD	002247	31/39	31/50						
MOPTB	002230	31/35	33/60						
MOPTR	002245	31/35	31/44	31/48					
MOREP	002236	31/41	31/47						
MPOWO	010623	72/25	73/04						
MSAMS	001530	25/15	28/50	28/52					
MSAQU	010633	72/30	73/08						
MSAV	006650	56/30	56/38	56/54	57/00	57/02			
MST10	006571	55/59	56/11						
MSTIA	006525	55/19	55/23						
MSTIB	006530	55/17	55/22						
MSTIC	006575	56/02	56/04	56/16					
MSTID	006576	55/27	56/00	56/03					
MSTIM	006521	7/48	55/15						
MSTIO	006572	55/21	55/62						
MSTIR	006620	55/15	56/17	56/19	56/20	56/21			
MSWRG	001522	23/32	25/13						
MULSA	011443	83/39	83/43	83/46					
MULTE	011435	83/26	83/39						
MULTI	006066	8/45	35/28	35/30	35/41	35/51	35/53	36/00	40/10
		40/17	40/38	56/12					
MX2SP	011065	77/07	77/45						
MXMMF	010746	74/07	75/02						
MXMMT	010764	74/25	74/26	75/06	76/29	76/30			
MXQUE	011064	77/05	77/41	77/43					
N10	000134	9/13	39/10	39/33					

PASSP	003022	37/34	38/18						
PASSC	000147	9/24	37/13	37/31	38/03	38/07	38/20	52/27	52/37
PASSN	000152	9/27	37/11	38/11	38/12				
PBINC	001033	18/50	18/55						
PBINN	001037	18/36	18/41	18/54					
PRINR	001074	18/44	18/53	19/20	21/14	21/17	21/33		
PCH14	010614	72/12	72/43						
PCH35	010615	72/16	72/44						
PCH37	010616	72/20	72/45						
PCOTT	010613	72/07	72/42						
PCOUN	010612	72/08	72/10	72/41					
PDEC1	000613	15/21	16/20						
PDEC2	000521	15/13	15/17						
PDEC3	000523	15/15	15/19						
PDECR	000463	14/46	15/03	15/33	16/14	16/39	16/41	16/42	
PERRO	003251	40/50	40/56						
PHEAD	003037	10/63	38/33	38/45					
PINHI	000573	14/05	15/05	16/02	16/16	17/05	17/40	20/28	
PINST	003257	40/55	40/58	41/02					
PLARI	012020	88/19	89/17	89/31					
PLADC	011407	82/20	83/05	83/15					
PLAOK	011707	86/19	87/17	87/33					
PLASC	011576	85/18	85/54	85/61					
PLASI	011426	83/20	83/24	83/30					
PLATX	012163	90/18	91/03	91/38					
PMEND	002377	33/09	33/36						
POWIN	003311	41/24	41/31						
POWON	010546	8/01	72/04						
POWRE	000100	7/63	8/01						
POWZE	000076	7/63	24/08	41/31	41/35				
PRINT	000101	8/02							
PROG	004064	22/43	24/49	24/52	48/07	72/28			
PRTYP	006731	24/62	58/02						
PSAAN	010617	72/32	72/46						
PSWIT	000115	8/62	37/09	37/29	38/42	40/51			
PTAB	001160	18/22	20/24						
PTAB1	007542	31/52	64/03						
PTAB2	007743	31/53	66/03						
PTAB3	010144	31/54	68/03						
PTAB4	010345	31/55	70/03						
QCHAR	000673	17/07	17/10						
QDICL	001173	20/34	20/37						
QDOUT	001144	20/07	20/11						
QHAAT	001227	21/39	21/42						
QMEND	002400	33/11	33/22	33/27	33/37				
QTYPE	000734	17/42	17/46						
QUESA	011063	77/03	77/26	77/37					
QUESD	011106	77/23	77/34						
QUESM	011110	77/21	77/36						
QUESS	011113	77/30	77/39						
QUEST	011112	77/28	77/38						
RACAN	011212	80/28	81/00						
RADYN	011204	80/12	80/16	80/58					
RAEND	011155	80/14	80/31						
RALIF	011205	80/11	80/24	80/59					
RAMIN	011210	80/22	80/62						
RANDI	011133	80/13	80/17						
RANK	007167	60/16	60/19						
RAPLU	011207	80/20	80/61						

RASPA	011206	80/26	80/60						
RAZER	011211	80/10	80/63						
RBZOT	000451	14/25	14/34	14/36					
RCOTI	000145	9/22	39/60	40/03					
REBIN	001400	7/08	24/08	93/06					
REG0	001100	17/37	17/45	17/50	17/54	17/63	19/24	20/02	20/10
		20/17							
REG1	001101	17/47	18/07	19/25	20/11	20/20			
REG2	001102	17/48	18/06	19/26	20/12	20/21			
REG3	001103	17/38	17/41	17/44	18/08	19/27	20/05	20/06	20/09
		20/22							
RENOQ	002703	36/12	36/15	37/06	37/24	52/38			
REOF2	011706	87/02	87/06	87/07	87/09	87/13	87/14	87/15	
REOF3	012017	89/02	89/06	89/07	89/09	89/13	89/14	89/15	
REPL4	011430	83/15	83/29	83/31	83/32				
REPL5	012204	91/03	91/10	91/19	91/20				
RET5	012257	92/06	92/10	92/11					
RETOF	011521	84/02	84/05	84/06	84/08	84/41	84/42	84/44	
RETYP	006725	57/43	57/56	57/60	57/62	58/02	58/12		
REVU	007075	59/42							
REVUA	007105	59/50	59/60						
REVUB	007122	59/54	59/63						
REVUC	007116	59/59	60/03						
REVUD	007110	59/53	59/57						
RINHI	000452	14/37	16/02	16/06	16/07				
RLOOK	003201	40/14							
RLPTT	002405	33/45	33/53	33/59					
RMEND	002375	33/05	33/26	33/33	33/34				
RMSK	001161	18/20	18/25	20/25					
RPASS	000146	9/23	34/04	36/05	38/02	38/04	38/05	38/16	38/18
		38/21	38/22	38/25	38/33	38/39			
RPBIT	002456	34/28	34/40	34/46					
RPCPU	002502	34/49	34/61	35/03	35/06				
RPOUT	000462	14/02	14/04	14/06	14/07	14/19	14/45	14/48	14/53
		15/00	15/04	15/06	15/43	15/48	15/53	15/58	16/15
		16/17							
RPSAQ	010577	72/29	72/37	72/51					
RPTTY	002430	34/05	34/17	34/20	34/25				
RQUES	011062	77/02	77/19	77/46					
RRESW	001362	23/13	23/38	23/45	23/52				
RSAMS	001767	28/12	28/27	28/33	28/57	28/58			
RTEST	000110	8/57	37/39	38/02	38/41	38/50			
RTIME	006445	7/47	54/10						
RVTMP	007135	59/45	59/50	59/52	60/02	60/08			
RXDEC	000454	14/39	14/58	14/60	15/63	16/01			
SABIN	001073	18/43	18/56	19/19					
SACHA	000775	17/02	17/23	17/31	18/14				
SADIG	000453	14/38	14/63	15/02	16/13				
SAMCO	002311	32/34	32/41	32/54					
SAMEX	002260	31/15	32/12						
SAMMS	002274	32/15	32/25	32/28					
SAMNM	002265	31/17	32/20						
SAPTB	002403	31/13	33/43						
SAVE	001577	26/23	26/39	27/05					
SCORA	007063	59/27	59/62						
SCORB	007070	59/33	59/37						
SCORE	007061	58/45	59/25	60/41					
SDIST	001147	20/14	20/16	29/02					
SEC3	001236	21/24	21/51	23/20	24/40	24/46	24/54	57/57	58/11

		72/06							
SEC4K	001243	21/42	21/57						
SECM2	001237	17/62	21/52	72/15	72/19	72/23			
SECM5	001240	21/45	21/53						
SECS2	001241	21/27	21/55						
SECS4	001242	21/20	21/56						
SERIN	003301	7/07	41/23						
SES40	007307	61/19	61/23						
SES60	006724	57/52	57/61						
SESAM	007300	61/10	61/13	61/16					
SESEX	007306	61/22							
SESOU	007305	61/21							
SETAC	000574	16/03	23/58						
SIGN	000526	15/23	16/38						
SIGNR	000543	15/23	15/27	15/30	15/35	15/37			
SMEND	002401	33/14	33/29	33/38					
SPFAK	000177	9/48	35/44	35/52	35/59				
SPFCO	000200	9/49	35/60	40/37					
SSAMS	001770	28/34	28/44	28/59					
SSTAC	001371	23/27	23/36	23/59					
SSWR0	001363	23/10	23/42	23/48	23/53				
SSWR1	001364	23/11	23/43	23/49	23/54				
SSWR2	001365	23/12	23/47	23/50	23/55				
STABU	012252	91/48	91/52	92/06	92/17	92/20			
START	001620	27/03	28/03						
STINC	006460	54/21	54/28						
STOP	000102	8/03							
STORE	001655	27/36	27/41						
STOTX	012173	91/08	91/11						
STSKP	006461	54/14	54/22	54/26					
STTYP	006722	57/47	57/50	57/59					
SVTYM	007136	58/13	60/09	60/22	61/22				
SWISA	010620	8/00	24/60	32/40	33/62	37/37	52/36	72/51	74/53
		76/48							
TABLE	002000	20/24	29/06						
TBLCC	000252	10/30	35/16						
TBLCI	000235	10/16	35/12						
TBLCN	000203	9/53	35/02						
TBLCS	000220	10/02	35/08						
TBLCT	000267	10/45	35/20						
TBLN1	003432	10/47	43/04						
TBLN2	003462	10/51	43/29						
TBLN3	003512	10/52	10/57	44/02					
TBLN4	003542	44/27							
TBLR1	003572	10/53	45/02						
TBLR2	003622	10/54	45/27						
TBLR3	003652	10/55	46/02						
TBLR4	003702	10/56	46/27						
TBZOT	000777	17/49	18/18						
TCP0	006543	55/36							
TCP00	006542	55/34	56/06						
TCP1	006544	55/37							
TCP10	006553	55/44							
TCP11	006554	55/45							
TCP12	006555	55/46							
TCP13	006556	55/47							
TCP14	006557	55/48							
TCP15	006560	55/49							
TCP16	006561	55/50							

TCP17	006562	55/51							
TCP2	006545	55/38							
TCP20	006563	55/52							
TCP21	006564	55/53							
TCP22	006565	55/54							
TCP23	006566	55/55							
TCP24	006567	55/56							
TCP25	006570	55/57							
TCP3	006546	55/39							
TCP4	006547	55/40							
TCP5	006550	55/41							
TCP6	006551	55/42							
TCP7	006552	55/43							
TEMP1	001575	26/12	26/18	26/37					
TEMP2	001576	26/22	26/30	26/36	26/38	27/25			
TENU	000202	9/51	39/13	39/36	52/16				
TERBI	012035	89/30	89/32						
TERDC	011377	83/04	83/06						
TERM1	011405	83/09	83/12						
TERMB	012031	88/11	89/28						
TERMD	011373	82/12	83/02						
TERMO	011722	86/11	87/30						
TERMS	011601	85/11	85/58						
TERMT	011266	81/23	81/50	82/11	86/47	91/27			
TERMX	012220	90/11	91/35						
TEROK	011726	87/32	87/34						
TERSC	011605	85/60	85/62						
TERTX	012224	91/37	91/39						
TEXEN	012162	90/52							
TEXIA	012110	90/50	91/16	91/53	92/07				
TEXIN	012111	90/51							
TIMCT	006537	55/30	56/09	56/13					
TIMEM	006540	55/31	56/15	60/04					
TIMER	003032	37/46	37/48	37/56	37/61	38/27	39/19	39/51	40/02
TIMEX	006536	55/26	55/29	55/63	56/10				
TIMMS	006064	8/43	58/27						
TIMRO	006065	8/44							
TIMRT	000124	9/05	40/14	40/22	41/08				
TIMSK	006063	8/42	14/26	14/30	19/45	20/13			
TIMTT	000125	9/06	40/42	41/10					
TINHI	001164	18/46	18/61	19/32	20/05	20/28	20/32	21/16	21/37
TINSR	003166	40/02	52/20						
TINST	003221	40/04	40/31						
TLOOK	003233	40/42							
TMEND	002340	25/00	32/14	33/05					
TOMA	007227	60/37	60/56						
TOMB	007252	60/47	60/57						
TOMC	007250	60/55	61/00						
TOMD	007242	60/45	60/48						
TOMER	007210	57/63	60/22						
TOMF	007244	60/50	60/53						
TOMG	007254	60/59	60/63						
TOMH	007255	60/57	60/60						
TOMJ	007245	60/48	60/51						
TSTRP	000311	11/00	35/45						
TTTYP	000313	11/02	35/57						
TTYLL	000164	9/37	34/22						
TTYSA	000162	9/35	34/26	35/31	35/46	35/50			
TTYSQ	000161	9/34	34/08						

XDBIN	001020	7/41	18/36						
XDDEC	000464	7/43	14/48						
XDICL	001165	7/34	20/30						
XDISP	001104	7/32	19/29						
XDIVD	006634	7/52	56/54						
XDIVS	006633	7/51	56/53						
XDLIE	011655	85/12	86/12	86/46	88/12				
XDOCT	000564	7/42	15/58						
XDOUT	001134	7/33	20/02						
XDZOC	000554	7/44	15/48						
XFDIS	001776	28/40	29/01						
XFITY	001377	24/01							
XFORM	001000	17/51	18/20						
XFRAM	006354	11/11	52/22						
XFROM	010742	74/09	74/55	76/20					
XGTCH	011657	85/09	86/09	86/48	88/09				
XILLG	011654	84/23	84/25	84/27	84/33	84/35	84/40	85/34	85/51
		86/33	86/36	86/45	87/12	88/33	88/36	89/12	
XINRT	011661	85/04	86/04	86/50	88/04				
XINST	007144	58/16	60/15	60/27					
XINTR	011662	86/00	86/51	87/36	89/34				
XLIST	002725	11/10	37/20						
XLPT	000017	8/18	14/28	17/52	17/53	39/44			
XLPTT	000441	14/27	14/29	28/63					
XMEND	001466	24/11	24/63						
XMESS	000412	7/28	14/02						
XMEXT	001467	24/10	25/00						
XMULT	006621	7/50	56/29						
XOMER	006726	57/48	57/63						
XONTR	011663	85/59	86/52	87/31	89/29				
XPCPT	001465	24/62							
XQUES	011067	7/53	77/19						
XRANK	007145	59/46	60/16	61/02					
XRESW	001312	7/55	23/10						
XRTC	000014	8/21	37/44	37/45	37/47	37/50	38/29	39/18	39/22
		39/45	40/03	60/33	60/34	60/35	60/49	60/52	60/58
		60/62	62/08	62/09	62/10	62/12	62/14	63/18	63/20
XSAMS	001736	7/54	28/33						
XSDIS	001777	28/41	29/02						
XSTAC	001370	23/26	23/29	23/37	23/39	23/58			
XTBIN	001014	7/37	18/33						
XTDEC	000470	7/39	14/53						
XTES4	006373	52/31	52/37						
XTES5	006374	52/25	52/38						
XTEST	002704	11/09	36/15	37/02					
XTIM1	006677	57/31	57/33						
XTIMA	006665	57/19	57/23						
XTIMC	006700	57/13	57/21	57/27	57/34				
XTIMD	006664	57/16	57/22	57/32					
XTIMR	006701	57/11	57/14	57/15	57/17	57/18	57/20	57/25	57/26
		57/29	57/35						
XTIMS	006651	7/49	57/11						
XTIMT	006671	57/24	57/27						
XTIMW	006674	57/28	57/30						
XTOCT	000560	7/38	15/53						
XTOIN	010744	74/27	74/57	76/31					
XTRMT	011656	85/10	86/10	86/47	88/10				
XTTI	000010	8/19	22/21	22/23	22/26	80/34	80/37	80/39	
XTTO	000011	8/20	14/32	18/00	18/01	37/40	37/54	37/55	37/58

		37/59	39/41	39/43	39/49	39/50	39/55	39/56	41/27
		41/28	56/21	58/22	58/23	58/25	58/26	58/28	5
		58/38	58/39	58/46	58/54	58/57	58/62	59/02	
XTTOT	000445	14/31	14/33	29/00					
XTXCO	011301	81/36	81/62						
XTXND	011302	81/37	81/63						
XTYME	007143	58/14	59/27	59/44	60/14	60/25			
XTYPE	000724	7/30	17/37						
XTYPN	011660	85/20	86/22	86/49	88/22				
XTZOC	000550	7/40	15/43						
XWAIT	006375	7/45	53/14						
XWTOP	001244	7/46	22/14						
XWTYP	006727	57/51	58/00						
XX16	007141	59/30	59/36	59/49	59/59	60/12			
XXLPT	001774	28/36	28/63						
XXTTO	001775	28/38	29/00						
YCHAR	000672	14/41	14/59	17/09	18/35	19/00	19/02		
YDICL	001172	20/36	21/32						
YDLTE	012212	90/12	91/28						
YDOUT	001142	14/42	18/40	19/53	20/09	20/36			
YGTCH	012210	90/09	91/26						
YHAAT	001226	21/18	21/41						
YILLG	012214	90/31	90/47	91/30					
YINRT	012215	90/04	91/31						
YINTR	012216	91/32	91/55						
YONIR	012217	91/33	91/36						
YPBIN	001024	18/34	18/39	18/43					
YPDEC	000502	14/49	14/54	15/02					
YPOCT	000603	15/54	15/59	16/12					
YQUES	000307	10/62	37/04	37/22					
YTAB3	000712	14/43	17/26						
YTRMT	012211	90/10	91/27						
YTYPE	000732	17/22	17/34	17/44					
YTYPN	012213	90/20	91/29						
YZOCT	000601	15/44	15/49	16/09					
ZCHAR	000456	14/23	14/41						
ZDOUT	000457	14/42	16/00						
ZSUPP	000771	16/20	16/26	16/36	18/12				
ZTAB3	000460	14/43	14/62						