
Title:

RC3616 MAGNETIC TAPE TRANSPORT
Supplement to the Wangco Manual

 **REGNECENTRALEN**

RC SYSTEM LIBRARY: FALKONERALLE 1 DK-2000 COPENHAGEN F

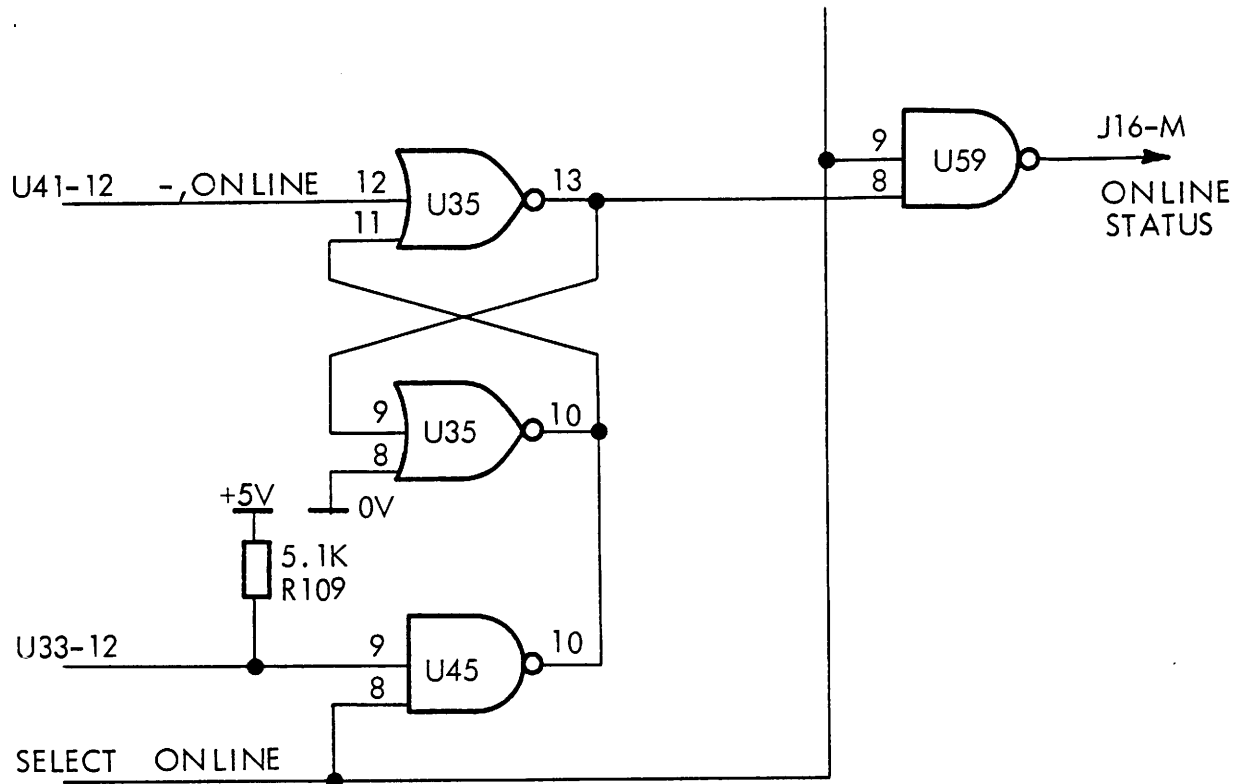
RCSL No: 44 - RT 1324
Edition: September 1976
Author: Mogens V. Pedersen

Keywords:

Magnetic Tape Transport, MTT 3

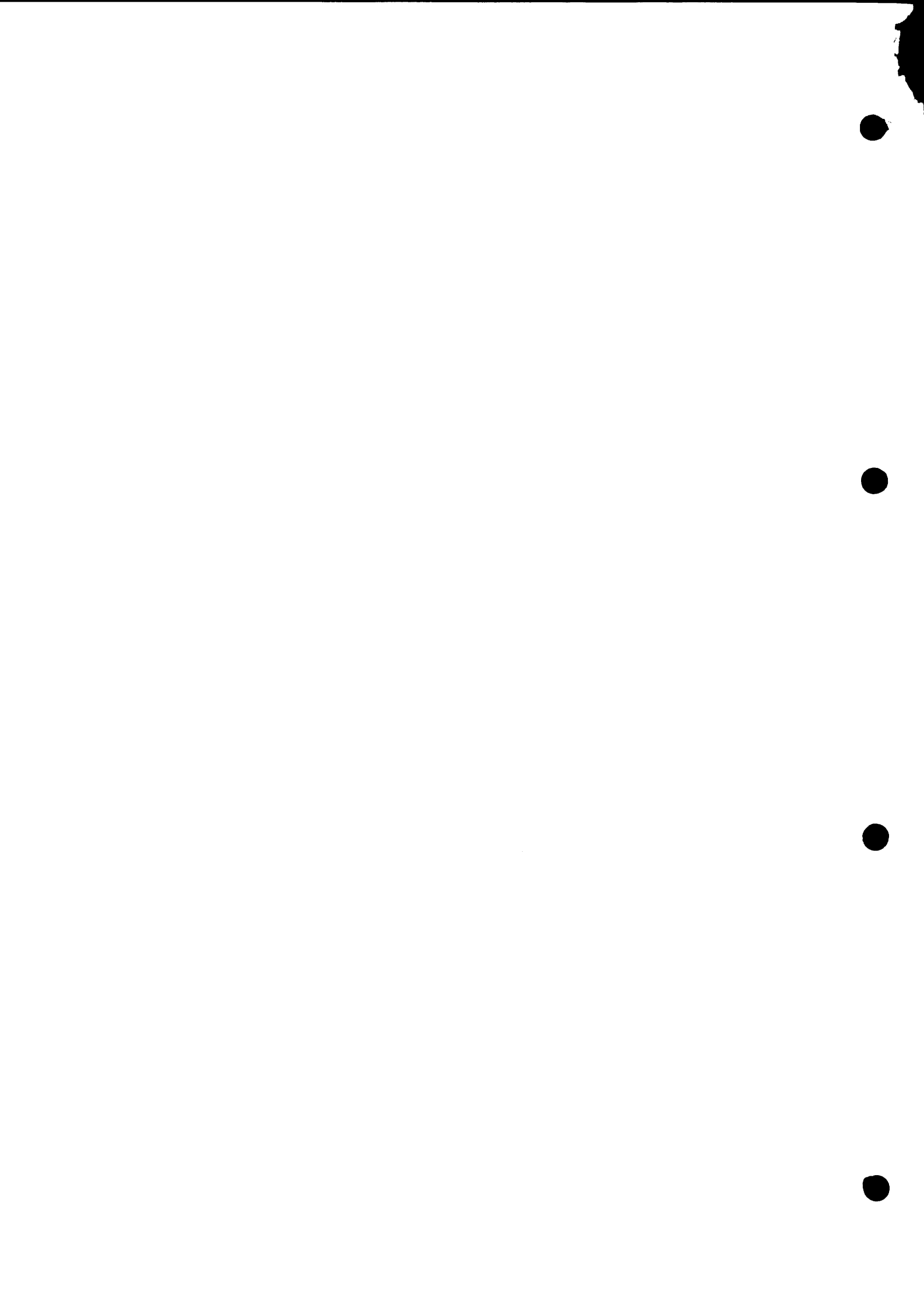
Abstract:

This paper contains a modification diagram to the Wangco 1275 PE/NRZI tape transport.



MTT 3

Modification diagram to
SCHEMATIC TRANSPORT CONTROL LOGIC



0001 ,MAIN

```
01
02 ;
03
04
05
06
07
08 ; RCSL: 44-RT 1710
09
10 ; VERSION: 15.05.78 LAB
11
12 ; PROGRAM
13 ; *****
14
15 ; RC 3600 - PROGRAM LOAD FOR PTR.
16
17 ; PROGRAM SOURCE:
18
19 ; S-BINARY TAPE: RCSL 44-RT 1711 (HEAD)
20 ; H-BINARY TAPE: RCSL 44-RT 1712 (SOLO)
21 ; ASCII SOURCE: RCSL 44-RT 1713
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
```

46 ;0. INDEX

```
47 ; 1. ABSTRACT PAGE 1
48 ; 2. MACHINE REQUIREMENTS PAGE 1
49 ; 3. DESCRIPTION OF PROGRAM PAGE 2
50 ; 4. ALTERNATIVES PAGE 5
51 ; 5. LISTING OF PROGRAM PAGE 5
52
```

53 ;1. ABSTRACT

```
54 ; THE PROGRAM IS USED TO LOAD THE ABSOLUTE
55 ; BINARY TAPE PRODUCED AS OUTPUT BY ASSEMBLER.
56 ; THE PROGRAM IS READ TO MEMORY THROUGH PTR BY
57 ; AUTOLOAD OR KEYED IN PROGRAM.
58 ; THE S-BIN TAPE SHOULD BE USED AS HEAD FOR
59 ; ABSOLUTE BINARY TAPES.
```

60 ;2. MACHINE REQUIREMENTS

```
61 ; RC 3600 FAMILY PROCESSOR CPU
62 ; MINIMUM 4K WORDS MEMORY MEM
63 ; CPU OPTION AUTOLOAD FOR PTR ROM
64 ; OR TECHNICAL CONTROL/FRONT PANEL TCP
65 ; PAPER TAPE READER PTR
```

1 0002 .MAIN

```
01
02 ;3. DESCRIPTION OF PROGRAM
03
04 ; THE MAIN PURPOSE WITH THIS PROGRAM IS TO BE ABLE TO
05 ; HAVE ONLY ONE TAPE TO LOAD WHEN LOADING A PROGRAM,
06 ; THE S-BIN HEAD IS ABLE TO AUTOLOAD OR MANUAL LOAD
07 ; AN ATTACHED ABSOLUTE BINARY TAPE.
08
09 ; A. HOW TO USE:
10
11 ; LOAD PTR (H-BIN OR S-BIN)
12
13 ; IF NO AUTOLOAD ROMS FOR PTR: KEY IN
14 ; THROUGH SWITCHES THE PROGRAM "SELF" (PAGE 7), START.
15
16 ; IF AUTOLOAD ROMS FOR PTR: AUTOLOAD
17 ; (FOR RC 3603 AND NOVA 1200:
18 ; ROM 007/008 RC PART # 47116/47117
19 ; (FOR NOVA 2:
20 ; ROM 022/023 RC PART # 55614/55615
21
22 ; COMMON:
23 ; NOW THE BINARY LOADER IS READ INTO MEMORY, THE SAME
24 ; VERSION AS IN ASM-TESTPROGRAMS AND IN BINARY LOADER TS.
25
26 ; IF THE USED TAPE WAS
27 ; H-BIN, HALTING - SPECIAL FORMAT
28 ; (IF THE OLD 44-RT 550, GO TO 4.2 PAGE 5)
29 ; (BOTH NORMALLY USED ALONE):
30 ; THE CPU HALTS IN LOC 121.
31 ; LOAD PTR WITH ABSOLUTE BINARY TAPE (A-BIN, AB, ABS-BIN)
32 ; PRESS CONTINUE OR START IN 9X7777,
33 ; WHERE S IS DON'T CARE (FORMERLY PTR/TTI)
34 ; THE PROGRAM IS LOADED.
35
36 ; IF THE USED TAPE WAS
37 ; S-BIN, SELFSTARTING - SPECIAL FORMAT
38 ; (NORMALLY ATTACHED TO AN A-BIN PROGRAM AS HEAD):
39 ; THE PROGRAM IS LOADED.
40
41 ; COMMON:
42 ; FURTHER ACTION DEPENDS OF THE PROGRAM:
43 ; SELFSTARTING OR HALTING IN BINARY LOADER IN
44 ; LOC 0X7676 WHEN READY, IF NOT CHECKSUM ERROR OR
45 ; OVERWRITING ERROR HALT IN 0X7751.
46
47 ; IF READYHALT: SET SWITCHES TO STARTADDR, RESET, START.
48 ; IF ERRORHALT: FIND THE ERROR, TAPE OR MACHINE, RESTART.
49 ; IF CHECKSUM ERROR,
50 ; AC3 = LAST READ BYTE, AND
51 ; AC0 = CHECKSUM, SHOULD BE ZERO.
52 ; IF OVERWRITING ERROR,
53 ; AC3 = LAST ADDR IN WHICH THE DATABLOCK
54 ; WANTS TO STORE. (ADDR + WC)
55 ; THE WHOLE DATABLOCK IS NOT STORED.
56
57 ; X IS THE NUMBER OF 4K MEMORY ARRAYS - 1.
58
59 ; B. TAPE TYPES:
60 ; S-BIN SELFSTARTING AUTOLOAD HEAD, PROGRAM READY.
61 ; H-BIN HALTING AUTOLOAD TAPE, BINARY LOADER READY.
62 ; A-BIN ABSOLUTE BINARY TAPE CONTAINING A PROGRAM,
63 ; 1. LOADED BY BINARY LOADER.
64 ; 2. LOADED BY, ATTACHED TO S-BIN TAPE.
65 ; F-BIN FUNNY FORMAT TAPE, BINARY LOADER READY.
66 ; (READ BY BOOTSTRAP, A MANUAL INSERTED PROGRAM.
```

```

| 0003 .MAIN
01
02 ; C. TAPE FORMATS:
03
04 ; ** SELF OR AUTOLOAD PROGRAM IN ROMS READS: S-BIN OR H-BIN,
05
06 ; PREAMBLE PROGRAM TO MEMORY FROM PTR.
07
08 ; PRECEDING: ALL ZEROES TAPE
09
10 ; WORD CONSTRUCTION: BYTE 1 + BYTE 2 = WORD
11
12 ; DATABLOCK:
13
14 ; TAPESYNC ;(ALL ZEROES BYTE +)ALL ONE BYTE
15 ; -WORD COUNT -1
16 ; DATAWORD 1
17 ; DATAWORD 2
18 ; .
19 ; .
20 ; .
21 ; DATAWORD N
22
23 ; ONE BIG BLOCK ABOUT 15 CM LONG FOLLOWED BY
24 ; BINARY LOADER PROGRAM BLOCK READ BY PREAMBLE
25 ; WITHOUT ANY SPACING.
26
27
28 ; COMMENT: IT WAS POSSIBLE FOR SELF PROGRAM TO USE SAME
29 ; FORMAT AS PREAMBLE AND BINARY LOADER, I. E.
30 ; BYTE 2 + BYTE 1 = WORD, BUT THEN ALL TAPES
31 ; SHOULD EXIST IN TWO VERSIONS OR SELF
32 ; PROGRAM SHOULD BE CHANGED IF THE TAPE AT HAND
33 ; WAS THE OLD FORMAT, WHICH STILL IS USED FOR
34 ; AUTOLOADING.
35
36 ; IF SELF PROGRAM SHOULD USE FORMAT BYTE 2 +
37 ; BYTE 1, THE PROGRAM WILL REMAIN ONE INSTRUCTION
38 ; SHORTER, THE MOVS 1,1 USED IN GET ROUTINE, BUT
39 ; NOT USED IN READ ROUTINE IN PREAMBLE PROGRAM.
40
41
42 ; ** PREAMBLE PROGRAM READS: LAST PART OF S-BIN OR H-BIN,
43 ; BINARY LOADER TO MEMORY FROM PTR.
44
45 ; PRECEDING: PREAMBLE PROGRAM ITSELF WITHOUT
46 ; ANY SPACING.
47
48 ; WORD CONSTRUCTION: BYTE 2 + BYTE 1 = WORD
49
50 ; DATABLOCK:
51
52 ; -WORD COUNT
53 ; DATAWORD 1
54 ; DATAWORD 2
55 ; .
56 ; .
57 ; .
58 ; DATAWORD N
59
60 ; ONE BIG BLOCK FOLLOWED BY ALL ZEROES TO
61 ; THE END OF TAPE OR TO ATTACHED PROGRAM.
62
63
64 ; (OLD PREAMBLE PROGRAM, RCSL 44-RT 550, READS
65 ; (THIS FORMAT: BYTE 1 + BYTE 2 = WORD,
66 ; (BECAUSE IT USES GET ROUTINE IN AUTOLOAD PROGRAM. )))

```

1 0004 ,MAIN

```
01
02 ; ** BINARY LOADER READS: A-BIN,
03 ; PROGRAM TO MEMORY FROM PTR,
04
05 ; PRECEDING: SOME CENTIMETERS WITH ALL ZEROES TAPE.
06
07 ; SYNCHRONIZING: FIRST BYTE DIFFERENT FROM ZERO,
08 ; USED AS BYTE 1 IN FIRST WORD.
09
10 ; WORD CONSTRUCTION: BYTE 2 + BYTE 1 = WORD
11
12 ; 4 BLOCK TYPES ARE DEFINED:
13 ; DATA, MULTIPLE DATA (REPEAT),
14 ; START AND ERROR (IGNORE).
15
16 ; A NORMAL BINARY TAPE (A-BIN, AB, ABS-BIN)
17 ; IS BUILD UP LIKE THIS:
18 ; DATABLOCK, DATABLOCK, ..., DATABLOCK,
19 ; DATABLOCK, MULTIPLE DATABLOCK, DATABLOCK,
20 ; DATABLOCK, ..., DATABLOCK, STARTBLOCK.
21
22 ; NEXT BLOCK FOLLOWS WITHOUT ANY SPACING OR
23 ; WITH TWO ALL ZEROES CHAR SPACING.
24 ; AFTER LAST BLOCK: SOME CENTIMETERS WITH
25 ; ALL ZEROES TAPE, MAYBE FOLLOWED BY A
26 ; SPECIAL SUMCHECK BLOCK FOR COPIING PURPOSE.
27
28 ; DATABLOCK FORMAT: FIRST WORD NEGATIVE.
29 ; WORD COUNT: 0 < WC < 17
30
31 ; -WORD COUNT ;TYP. 177760 (16 WORDS OF DATA)
32 ; ADDRESS ;ADDRESS FOR DATAWORD 1
33 ; CHECKSUM ;SUM OF BLOCK INCL CHECKSUM = 0
34 ; DATAWORD 1
35 ; DATAWORD 2
36 ; .
37 ; .
38 ; .
39 ; DATAWORD N ;MAX NUMBER OF DATAWORDS = 16
40
41 ; MULTIPLE DATABLOCK FORMAT: FIRST WORD NEGATIVE.
42 ; WORD COUNT: 16 < WC < 32768
43
44 ; *MINIMUM* 177757 (-17, WC=17)
45 ; -WORD COUNT ;GIVING 16 WORDS OF IDENT. DATA
46 ; ADDRESS ;ADDRESS FOR STORING FIRST TIME
47 ; CHECKSUM ;SUM OF BLOCK INCL CHECKSUM = 0
48 ; DATAWORD ;NUMBER OF IDENTICAL DATAWORDS N
49 ; ;IS GIVEN BY N = WC - 1
50
51 ; START BLOCK FORMAT: FIRST WORD: 000001
52
53 ; IDENTIFICAT. ; = 000001
54 ; F + ADDRESS ;F = FLAG BIT 0, BIT 1-15 = ADDR
55 ; CHECKSUM ;SUM OF BLOCK INCL CHECKSUM = 0
56
57 ; ;IF FLAG = 1 THE LOADER WILL HALT IN
58 ; ;READYHALT. ( .END)
59 ; ;IF FLAG = 0 THE LOADER WILL GIVE
60 ; ;CONTROL TO THE LOADED PROGRAM IN
61 ; ;THE LOCATION ADDRESS. ( .END ADDRESS)
```


1 0005 ,MAIN

```
01
02 ; IGNORE BLOCK FORMAT: FIRST WORD: GREATER THAN +1.
03
04 ; IDENTIFICAT. ; NUMBER > +1.
05 ; GARBAGE ;STORING NOTHING
06 ; GARBAGE ;IGNORING DATA, BYTE FOR
07 ; . ;BYTE UNTIL ...
08 ; .
09 ; .
10 ; GARBAGE
11 ; TERMINATOR ;AN ALL ONE BYTE IS SEEN (377)
```

12
13
14 ;4. ALTERNATIVES

15
16 ; 1. USE THE OLD BOOTSTRAP LOADER:

17
18 ; TAPE: F-BIN 44-RT 1531 (FUNNY FORMAT, F-BIN)

19
20 ; LIST: AT THE END OF THIS MANUAL,

21
22 ; TOGETHER WITH TESTPROGRAMS AND BINARY LOADER TS AND
23 ; THE BINARY LOADER DESCRIBED HERE, THE BOOTSTRAP
24 ; LOADER IS LOADED INTO MEMORY TOO.

25 ; IF NOT PRESENT, KEY IN THROUGH SWITCHES TO MEMORY
26 ; FROM LOC 0X7756 TO LOC 0X7773 INCL.
27 ; LOAD THE READER WITH F-BIN TAPE.
28 ; START IN 0X7770.

29 ; THIS WILL LOAD ANOTHER VERSION OF THE BINARY LOADER.
30 ; AFTER LOADING THE CPU HALTS IN LOC 0X7776.

31 ; LOAD THE READER WITH A-BIN TAPE.

32 ; START IN LOC SX7777,

33 ; WHERE S = 1 FOR PTR (0 FOR TTI),

34
35 ; 2. USE THE OLD AUTOMATIC PROGRAM LOAD:

36
37 ; TAPE: H-BIN 44-RT 550 (HALTING - SPECIAL FORMAT)

38
39 ; DESCRIPTION: 44-RT 551

40
41 ; THIS WILL LOAD ANOTHER VERSION OF THE BINARY LOADER,
42 ; BUT THE INPUT METHOD IS EXACTLY AS DESCRIBED EARLIER
43 ; IN THIS MANUAL, PAGE 2:

44 ; AUTOLOAD OR SELF.

45 ; AFTER LOADING THE CPU HALTS IN LOC 121.

46 ; LOAD THE READER WITH A-BIN TAPE.

47 ; PRESS CONTINUE OR START IN LOC SX7777,

48 ; WHERE S = 1 FOR PTR (0 FOR TTI),

49
50 ; 3. USE ANOTHER MEDIA, I. E. NOT PAPER TAPE.

51
52
53 ;5. LISTNING OF PROGRAM

54
55 ; NEXT PAGES: AUTOLOAD PAGE 6
56 ; SELF PAGE 7
57 ; PREAMBLE PAGE 8
58 ; BINARY LOADER TS PAGE 9
59 ; BOOTSTRAP LOADER PAGE 11

1 0006 .MAIN

```
01
02 ;AUTOLOAD, A PROGRAM FOR AUTOMATIC PROGRAM LOAD, CONTAINED
03 ;IN TWO ROMS, SEE 3A IN ABOVE DESCRIPTION FOR NUMBERS.
04 ;THIS VERSION WILL HANDLE PTR (AND MTC (AND DKP DEV 73
05 ;IF ALLREADY AT TRACK ZERO, I. E. AFTER LOADING DISCPACK)).
06 ;THIS PROGRAM STARTS IN MEMORY LOC 0 AFTER AN AUTOLOAD.
07
08
09 000000 ,LOC 0
10
11 000000 060477 ABEG: READS 0 ;READ SWITCHES INTO AC0
12 000001 105120 MOVZL 0,1 ;ISOLATE DEVICE CODE
13 000002 124240 COMOR 1,1 ; -(DEVICE CODE - 1)
14
15 000003 010011 ALOOP: ISZ AOP1 ;COUNT DEVICE CODE IN ALL
16 000004 010031 ISZ AOP2 ;I/O INSTRUCTIONS TO ACTUAL.
17 000005 010033 ISZ AOP3 ;
18 000006 010014 ISZ AOP4 ;
19 000007 125404 INC 1,1,SZR ;DONE ?
20 00010 000003 JMP ALOOP ;NO, INCREMENT AGAIN
21 ;YES, START DEVICE
22 00011 060077 AOP1: 060077 ;(NIOS 0)-1, MTC=READ COMMAND (0)
23 ;AFTER RESET.
24 00012 030017 LDA 2,AC377 ;PUT JMP 377 INTO LOC 377 FOR LATER
25 00013 050377 STA 2,377 ;LOOPING, WAITING FOR MTC, DATACHANNEL
26 00014 063377 AOP4: 063377 ;(SKPBN 0)-1, BUSY ? IF MTC AND NOT
27 ;ONLINE: TERMINATED, NONBUSY.
28 00015 000011 JMP AOP1 ;START AGAIN UNTIL MTC ONLINE.
29
30 00016 101102 MOVL 0,0,SZC ;LOW SPEED DEVICE ? (SWITCH 0 = 0)
31 00017 000377 AC377: JMP 377 ;NO, GO TO 377 AND WAIT FOR CHANNEL
32 ;NOW: CARRY = 0, AC1 = 0
33 00020 004031 ALOP2: JSR AGET1 ;GET A FRAME, AFTER RETURN: CARRY = 1
34 00021 101065 MOVC 0,0,SNR ;IS IT NONZERO ? (SWAP CARRY TO 0)
35 00022 000020 JMP ALOP2 ;NO, IGNORE AND GET ANOTHER BYTE.
36 ;YES, SYNCHRONIZED,
37 00023 004030 ALOP4: JSR AGET ;GET A FULL WORD (CARRY = 0)
38 00024 046027 STA 1,AC77 ;STORE STARTING AT 100
39 ;FIRST WORD = 2'S COMPL OF WORD COUNT
40 00025 010100 ISZ 100 ;COUNT WORD, DONE ?
41 00026 000023 JMP ALOP4 ;NO, GET ANOTHER WORD
42 00027 000077 AC77: JMP 77 ;YES, FINISHED, JMP TO LAST WORD LOADED
43 ;LOC 27, LOCATION COUNTER AUTO INCRE.
44
45 00030 126420 AGET: SUBZ 1,1 ;CLEAR AC1, SET CARRY
46 AGET1:
47 ALOP3:
48 00031 063577 AOP2: 063577 ;(SKPDN 0)-1, DONE ?
49 00032 000031 JMP ALOP3 ;NO, WAIT
50 00033 060477 AOP3: 060477 ;(DIAS 0,0)-1, YES, READ INTO AC0
51 00034 107363 ADDCS 0,1,SNC ;ADD 2 FRAMES SWAPPED, GOT SECOND ?
52 00035 000031 JMP ALOP3 ;NO, GO BACK AFTER IT (CARRY = 0)
53 00036 125300 MOVS 1,1 ;YES, SWAP AC1 (FOR TAPE FORMAT)
54 00037 001400 JMP 0,3 ;RETURN WITH WORD = BYTE 1 + BYTE 2
55 ;OR WORD = BYTE SYNCHRONIZER.
56
57 ;ALL LABELS CHANGED FROM ORIGINAL TO AXXXX TO
58 ;ASSEMBLE TOGETHER WITH REST OF PROGRAM.
```

1 0007 ,MAIN

01
02 ;SELF, A PROGRAM SIMILAR TO AUTOLOAD PROM FOR PTR, BUT
03 ;READ TO MEMORY BY THE OPERATOR THROUGH FRONT PANEL SWITCHES.

04
05 ;FIRST: IN LOC 30 DEPOSIT 000064 (JMP 64),
06 ; (ONLY USED IF AN OLD VERSION OF S-BIN OR H-BIN
07 ; (IS LOADED. PROGRAMS ISSUED BEFORE JUNE 1978 ARE
08 ; (WITHOUT READ ROUTINE IN PREAMBLE PROGRAM AND
09 ; (USES GET ROUTINE IN AUTOLOAD INSTEAD, ASSUMED
10 ; (PRESENT IN LOC 30, THEREFORE JSR 30.))))
11 ; OLD VERSIONS: FIRST CHAR ON TAPE = 1
12 ; THIS VERSION: FIRST CHAR ON TAPE = 377
13 ;THEN DEPOSIT SELF PROGRAM.
14 ;START IN LOCATION 50, NOT 10050 AS IN LISTNING.

15
16
17 010050 .LOC 10050 ;ONLY FOR ASM, ACTUAL LOC = 50
18

19 10050 060112 BEGIN: NIOS PTR ;START READER
20 10051 126440 SUBO 1,1 ;CLEAR AC1, CLEAR CARRY
21 10052 004413 LOOP: JSR GET1 ;GET A BYTE
22 10053 101065 MOVC 0,0,SNR ;IS IT ZERO ?
23 10054 000776 JMP LOOP ;YES, IGNORE AND GET NEXT
24 10055 030420 LDA 2,SAPRE ;NO, IT WAS TAPE SYNCRONIZER, DROP IT
25 ;AND SET AC2 TO FIRST MEM LOC FOR PREAM
26 10056 004406 LOOP1: JSR GET ;GET A FULL WORD, FIRST = WORD COUNT
27 10057 045000 STA 1,0,2 ;STORE INTO MEMORY FROM COUNT
28 10060 151400 INC 2,2 ;INCRMENT ADDR TO NEXT
29 10061 010417 ISZ COUNT ;BUMP WORD COUNT, DONE ?
30 10062 000774 JMP LOCP1 ;NO, REPEAT, STILL DATA
31 10063 000416 JMP PREAM ;YES, FINISHED, GIVE CONTROL TO
32 ;FIRST WORD IN PREAM PROGRAM
33 10064 126420 GET: SUBZ 1,1 ;CLEAR AC1, SET CARRY
34 10065 063512 GET1: SKPBZ PTR ;
35 10066 000777 JMP -1 ;WAIT NON-BUSY
36 10067 060412 DIA 0,PTR ;READ A BYTE TO AC0
37 10070 060112 NIOS PTR ;START READER FOR NEXT BYTE
38 10071 107363 ADDCS 0,1,SNR ;ADD 2 BYTES SWAPPED, GOT SECOND ?
39 10072 000773 JMP GET1 ;NO, GO BACK AFTER IT
40 10073 125300 MOVS 1,1 ;YES, SWAP AC1
41 10074 001400 JMP 0,3 ;RETURN WITH FULL WORD
42 10075 000100 SAPRE: +3-10000 ;START ADDR FOR LOADING PREAM
43 10076 000050 SADDR: 50 ;SPARE, NOT USED ,(START ADDR FOR SELF)

44
45 ;AFTER DEPOSITING ABOVE PROGRAM, SET SWITCHES TO 000050 AND
46 ;LOAD THE PAPER TAPE READER WITH SUITABLE TAPE (S-BIN OR H-BIN).
47 ;PRESS RESET, START. NOW THE BINARY LOADER IN NORMAL OR
48 ;SELFSTART VERSION IS READ IN, THEN THE MAIN PROGRAM.

```

01 0008 ,MAIN
02
03 ;PREAMBLE PROGRAM READ TO MEMORY FROM PTR By
04 ;
05 ; OR 1, AUTOLOAD PTR
06 ; 2, PRIM PROGRAM
07
08 010077 ,LOC 10077 ;ONLY FOR ASM, FIRST LOC ACTUAL
09 ;USED IS 100, AS THE TAPE SYNC CHAR
10 ;READ FROM TAPE NOT IS STORED,
11
12 10077 000377 SYNCH: 000377 ;TAPE SYNC CHAR
13
14 10100 177743 COUNT: PREAM-END-2 ;NEG WORD COUNT FOR PREAMBLE-1
15
16 10101 020421 PREAM: LDA 0,C4K ;MEMORY SIZING INCREMENT
17 10102 176221 ADCZR 3,3,SKP ;FORM HIGHEST ADDR = 077777
18 10103 116400 LOOP2: SUB 0,3 ;DECREMENT ADDR (SKIPPED FIRST TIME)
19 10104 055400 STA 3,0,3 ;STORE ADDR
20 10105 031400 LDA 2,0,3 ;GET IT BACK
21 10106 172414 SUB# 3,2,SZR ;SAME ?
22 10107 000774 JMP LOOP2 ;NO, NO MEMORY
23 10110 004414 JSR READ ;YES, READ FIRST DATA
24 10111 044412 STA 1,LENG ;SAVE LENGTH OF BINARY LOADER
25 10112 133000 ADD 1,2 ;FORM FIRST ADDRESS-1
26 10113 151400 LOOP3: INC 2,2 ;INCREMENT ADDR TO NEXT
27 10114 004410 JSR READ ;READ BINARY LOADER DATA
28 10115 045000 STA 1,0,2 ;STORE INTO MEMORY
29 10116 010405 ISZ LENG ;BUMP LENGTH COUNT
30 10117 000774 JMP LOOP3 ;REPEAT IF STILL DATA
31
32 10120 000401 JMP .+1 ;;;ONLY TO USE SAME LOC'S AS THE
33 ;NOT SELFSTARTING VERSION:
34
35 ;;10120 063077 ;;OLD LOC, CONTENT
36 ;; HALT ;LOAD PTR, PRESS CONTINUE OR
37 ;; ;START BINARY LOADER IN LAST
38 ;; ;LOC: 0X7777 OR 1X7777, SWITCH 0
39 ;; ;NO LONGER USED (PTR/TTI).
40
41 10121 001000 JMP 0,2 ;FINISHED, GIVE CONTROL TO LAST
42 ;USED MEM LOC, LAST LOC IN
43 ;BINARY LOADER (AND IN MEMORY).
44
45 10122 004000 C4K: 4000 ;MEMORY SIZING STEP
46 10123 000000 LENG: 0 ;LENGTH COUNT
47
48 10124 126420 READ: SUBZ 1,1 ;CLEAR AC1, SET CARRY
49 10125 063512 READ1: SKPBZ PTR ;
50 10126 000777 JMP .-1 ;WAIT NON-BUSY
51 10127 060412 DIA 0,PTR ;READ A BYTE
52 10130 060112 NIOS PTR ;START READER FOR NEXT BYTE
53 10131 107363 ADDCS 0,1,SNC ;ADD 2 BYTES SWAPPED, GOT SECOND ?
54 10132 000773 JMP READ1 ;NO, GO BACK AFTER IT
55 10133 001400 JMP 0,3 ;YES, RETURN WITH FULL WORD
56
57 10134 000745 END: JMP PREAM ;GETS CONTROL HERE IF AUTOLOADED.
58
59 ;NEXT LOCATION ONLY FOR ASM TO HAVE DATA ON PAPER TAPE:
60 10135 177635 BUILD-BEND-1 ;NEG WORD COUNT FOR BINARY LOADER
;LOADED FROM TAPE TO LABEL LENG
;IN PREAM PROGRAM

```

```

1 0009 .MAIN
01
02 ;BINARY LOADER TS CHANGED TO ONLY PTR, SEE ;;;
03 ;ERRORHALT XX7751 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 062677 ALRST=IORST
10 060477 ,DIAC RDSWI=HEADS 0
11
12 017635 ,LOC 17635
13
14 ;SUBROUTINE TO BUILD A WORD IN AC2
15 17635 054425 BUILD: STA 3,BTMP1 ;SAVE RETURN
16 17636 004406 JSR GTCHR ;GET FIRST BYTE
17 17637 171300 MOVS 3,2 ;PUT INTO LH OF AC2
18 17640 004404 JSR GTCHR ;GET NEXT BYTE
19 17641 173300 ADDS 3,2 ;FORM WORD IN AC2
20 17642 143000 ADD 2,0 ;ADD INTO CHECKSUM
21 17643 002417 JMP 0BTMP1 ;AND RETURN
22
23 ;READ A BYTE INTO AC3
24 ;IF SWITCH0=0 USE TELETYPE ELSE USE PTR
25 17644 054417 GTCHR: STA 3,BTMP2 ;SAVE RETURN
26 17645 034417 LDA 3,BSAVE ;TEST WHICH DEVICE
27 17646 175103 MOVL 3,3,SNC
28 17647 000406 JMP GTTI ;TTI
29 17650 063612 SKPDN PTR ;PTR
30 17651 000777 JMP .-1
31 17652 074412 DIA 3,PTR ;READ AND START
32 17653 060112 NIOS PTR
33 17654 002407 JMP 0BTMP2 ;AND RETURN
34
35 17655 063610 GTTI: SKPDN TTI
36 17656 000777 JMP .-1
37 17657 074410 DIA 3,TTI ;READ AND START
38 17660 060110 NIOS TTI
39 17661 002402 JMP 0BTMP2 ;AND RETURN
40 17662 000000 BTMP1: 0
41 17663 000000 BTMP2: 0
42 17664 000000 BSAVE: 0
43
44 ;TEST BLOCK TYPE
45 17665 125224 TEST: MOVZR 1,1,SZR ;1=START BLOCK (.END XX)
46 17666 000411 JMP IGNOR ;NO, IGNORE BLOCK
47 17667 101004 MOV 0,0,SZR ;TEST THE CHECKSUM
48 17670 000460 JMP CHKER ;ERROR
49 17671 030505 LDA 2,ADDRS ;GET ADDR
50 17672 062677 ALRST ;DO A RESET (IORST)
51 17673 151113 MOVL# 2,2,SNC ;TEST BIT 0
52 17674 001000 JMP 0,2 ;0=START PROGRAM
53 17675 063077 HALT ;1=HALT
54 17676 000777 JMP .-1 ;DON'T PROCEED
55
56 ;IGNORE BLOCK
57 17677 004745 IGNOR: JSR GTCHR ;READ UNTIL AN ALL
58 17700 020404 LDA 0,BC377 ;ONES BYTE IS SEEN
59 17701 116404 SUB 0,3,SZR ;IGNORING ERROR MESS
60 17702 000775 JMP IGNOR
61 17703 000407 JMP BLOCK ;OK, GO INTO BLOCK MODE
62 17704 000377 BC377: 377

```

1 0010 .MAIN

```
01
02 ;START OF PROGRAM BINARY LOADER TS
03 17705 062677 BBEGIN: ALRST ;RESET (IORST)
04 ;;; RDSWI 0 ;READ THE SWITCH REGISTER (READS 0)
05 ;;17706 060477 ;;OLD LOC, CONTENT
06 17706 102620 SUBZR 0,0 ;;ONLY PTR, AC0:=100000
07 17707 040755 STA 0,BSAVE ;AND SAVE IT FOR GTCHR
08 17710 060110 NIOS TTI ;START BOTH READERS
09 17711 060112 NIOS PTR
10
11 ;READ IN A BLOCK
12 17712 004732 BLOCK: JSR GTCHR ;GET A BYTE
13 17713 165305 MOVS 3,1,SNR ;AND TEST FOR NUL
14 17714 000776 JMP BLOCK ;YES, KEEP READING
15 17715 004727 JSR GTCHR ;OK, GET NEXT BYTE
16 17716 167300 ADDS 3,1 ;AND FORM COUNT, AC1:=# OF WORDS
17 17717 121000 MOV 1,0 ;SET CHECKSUM, AC0:=CHECKSUM TILL NOW
18 17720 004715 JSR BUILD ;GET ADDRESS
19 17721 050455 STA 2,ADDRS
20 17722 004713 JSR BUILD ;ADD IN THE CHECKSUM FROM TAPE
21 17723 125113 MOVL# 1,1,SNC ;TEST BLOCK TYPE
22 17724 000741 JMP TEST ;NOT A DATABLOCK
23 17725 044427 STA 1,BCOUN ;STORE WORD COUNT
24
25 ;READ IN THE DATA BLOCK
26 17726 030735 BDATA: LDA 2,BTMP2 ;LAST STA IN BTMP2 WAS JSR RETURN
27 17727 034423 LDA 3,DIFF ;ADDR 4 CELLS AFTER PROGRAM START:
28 17730 172400 SUB 3,2 ;AC2:=FIRST ADDR IN LOADER
29 17731 034445 LDA 3,ADDRS ;ADDR IN WHICH TO STORE
30 17732 136400 SUB 1,3 ;ADD NEG WC TO CHECK SPACE
31 17733 172023 ADCZ 3,2,SNC ;FOR WHOLE BLOCK
32 17734 000414 JMP CHKER ;NO, HALT THE LOADER
33 17735 030416 LDA 2,BC20 ;IF WC > 20 (OCTAL, NEG) IT IS A
34 17736 147033 ADDZ# 2,1,SNC ;A REPEAT BLOCK (CARRY=ZERO, BIT0=ONE)
35 17737 010415 ISZ BCOUN ;WHERE WC IS ONE LESS THAN COUNT
36 17740 147022 ADDZ 2,1,SZC ;IF REPEAT BLOCK SKIP NEXT TO READ DATA
37 17741 125113 STORE: MOVL# 1,1,SNC ;SKIP READ IN NEW DATA IF REPEAT BLOCK
38 17742 004673 JSR BUILD ;READ DATA TO AC2
39 17743 052433 STA 2,0,ADDRS ;STORE INTO MEMORY
40 17744 010432 ISZ ADDRS ;NEXT ADDR
41 17745 010407 ISZ BCOUN ;TEST COUNT
42 17746 000773 JMP STORE ;MORE DATA
43 17747 101004 MOV 0,0,SZR ;TEST CHECKSUM, AC0=VALUE
44 17750 063077 CHKER: HALT ;OVERWRITING OR ERROR IN CHECKSUM
45 17751 000741 JMP BLOCK ;OK,GET NEXT BLOCK
46
47 17752 000004 DIFF: 4
48 17753 000020 BC20: 20 ;REPEAT BLOCKS HAVE WC>20
49 17754 000000 BCOUN: 0 ;WORD COUNT
50 17755 000000 0 ;NOT USED
```

1 0011 ,MAIN

01

02

;BOOTSTRAP LOADER IS FOR PTR:

03

;ENTER AT BSTRP

04

05 17756 126440 BBGET: SUBO 1,1

06 17757 063612 BGET1: SKPDN 12 ;10 FOR TTI

07 17760 000777 JMP 0,1

08 17761 060412 DIA 0,12 ;10 FOR TTI

09 17762 060112 NIOS 12 ;10 FOR TTI

10 17763 127100 ADDL 1,1

11 17764 127100 ADDL 1,1

12 17765 107003 ADD 0,1,SNC

13 17766 000771 JMP BGET1

14 17767 001400 JMP 0,3

15 17770 060112 BSTRP: NIOS 12 ;10 FOR TTI

16 17771 004765 JSR BBGET

17 17772 044402 STA 1,+.2

18 17773 004763 JSR BBGET

19 17774 000000 0

20 17775 000000 0

;FOR BOOTSTRAP

21 17776 000000 ADDRS: 0

22 17777 000706 BEND: JMP BBEGIN ;START OF BINARY LOADER

1 0012 .MAIN

01

02

.END

0013 .MAIN

ABEG	000000	6/11					
AC377	000017	6/24	6/31				
AC77	000027	6/38	6/42				
ADDRS	017776	9/49	10/19	10/29	10/39	10/40	11/21
AGET	000030	6/37	6/45				
AGET1	000031	6/33	6/46				
ALoop	000003	6/15	6/20				
ALOP2	000020	6/33	6/35				
ALOP3	000031	6/47	6/49	6/52			
ALOP4	000023	6/37	6/41				
ALRST	062677	9/09	9/50	10/03			
AOP1	000011	6/15	6/22	6/28			
AOP2	000031	6/16	6/48				
AOP3	000033	6/17	6/50				
AOP4	000014	6/18	6/26				
BBEGI	017705	10/03	11/22				
BBGET	017756	11/05	11/16	11/18			
BC20	017753	10/33	10/48				
BC377	017704	9/58	9/62				
BCOUN	017754	10/23	10/35	10/41	10/49		
BDATA	017726	10/26					
BEGIN	010050	7/19					
BEND	017777	8/58	11/22				
BGET1	017757	11/06	11/13				
BLOCK	017712	9/61	10/12	10/14	10/45		
BSAVE	017664	9/26	9/42	10/07			
BSTRP	017770	11/15					
BTMP1	017662	9/15	9/21	9/40			
BTMP2	017663	9/25	9/33	9/39	9/41	10/26	
BUILD	017635	8/59	9/15	10/18	10/20	10/38	
C4K	010122	8/14	8/43				
CHKER	017750	9/48	10/32	10/44			
COUNT	010100	7/29	8/12				
DIFF	017752	10/27	10/47				
END	010134	8/12	8/55				
GET	010064	7/26	7/33				
GET1	010065	7/21	7/34	7/39			
GTCHR	017644	9/16	9/18	9/25	9/57	10/12	10/15
GTTI	017655	9/28	9/35				
IGNOR	017677	9/46	9/57	9/60			
LENG	010123	8/22	8/27	8/44			
LOOP	010052	7/21	7/23				
LCOP1	010056	7/26	7/30				
LOOP2	010103	8/16	8/20				
LOOP3	010113	8/24	8/28				
PREAM	010101	7/31	8/12	8/14	8/55		
READ	010124	8/21	8/25	8/46			
READ1	010125	8/47	8/52				
SADDR	010076	7/43					
SAPRE	010075	7/24	7/42				
STORE	017741	10/37	10/42				
SYNCH	010077	8/10					
TEST	017665	9/45	10/22				

1

2



