

# **Edinburgh Bootstrap for PDP-8 paper-tape loading**

**(Smaller than  
manufacturer's)**

/OPTIMISED 8/9+1 LOCATION INITIAL LOADERS 30/10/67  
 /USED TO RESTORE BINARY LOADER AFTER A 'SCREW UP'  
 /THEY LOAD MOST OF BIN LOADER IN ARK FORMAT, THEN  
 /TRANSFER CONTROL TO IT TO FINISH OF THE JOB.

/METHOD OF USE:

/ LOAD VERSION TO USE THE REQUIRED READER  
 /USING SWITCHES, PUT ARK TAPE IN READER, START  
 /AT ADDRESS 7777. PROGRAM SHOULD STOP AT START  
 /OF TRAILER CODES ON TAPE WITH PC=7777 & AC=0000  
 /IF OPERATION WAS SUCCESSFUL. IF NOT, LOAD & READ AGAIN.

BEND=7770 ; BTEM=36 /DEFINITIONS FOR ASSEMBLER.

/VERSION TO USE HIGH SPEED READER

0010	0007	@10;	7	/LOADING POINTER
7770	6014	@BEND;	RFC	/START THINGS MOVING
7771	6011		RSF	/FLAG SET YET ?
7772	5371		JMP BEND+1	/NO, HAVE ANOTHER LOOK
7773	7430		SZL	/COMPLETE WORD ?
7774	3410		DCA I 10	/YES, STORE IT
7775	7106		CLL RTL	/ROTATE 2 PLACES LEFT
7776	6012		RRB	/OR IN NEXT CHAR.
7777	5370		JMP BEND	/REPEAT

/VERSION TO USE TELETYPE READER

0010	0007	@10	AUTO, *-1	/DON'T FORGET THIS ONE !
			@BEND-1	
7767	7430	BOOT,	SZL	/READY TO STORE ?
7770	3410		DCA I AUTO	/YES
7771	6031		KSF	/FLAG SET ?
7772	5371		JMP *-1	/NO, WAIT FOR IT !
7773	7106		CLL RTL	/MOVE READY FOR NEXT 2 BITS
7774	3036		DCA BTEM	/SAVE ACC
7775	6036		KRB KFC	/CLEAR ACC READ CHAR & FETCH NEXT
7776	1036		TAD BTEM	/ADD ACCUMULATED VALUE
7777	5367		JMP BOOT	/DO IT ALL OVER AGAIN !

AUTO	0010
BEND	7770
BOOT	7767
BTEM	0036

/BINARY LOADER WITH AUTOMATIC READER SELECTION.  
 /AND LOAD-AND-RUN FACILITY IC. 8-102-U-SYM  
 /CONTAINS INPUT SUBROUTINES AVAILABLE TO ALL PROGRAMS  
 / @\* ; IS USED TO MAKE ASSEMBLER PUNCH IN RIM FORMAT

@7650

/INPUT SUBROUTINES

7650	7402		R750,	XX	
7651	7200	@* ;	M200,F	CLA	;F=600 /TO GET M200
7652	3000	@* ;		DCA TIMER	/INITIALISE TIMER
7653	2000	@* ;		ISZ TIMER	/TIME UP ?
7654	7410	@* ;		SKP	/NO, LOOK AT FLAG
7655	5650	@* ;		JMP I R750	/YES, RETURN
7656	6011	@* ;		RSF	/FLAG SET YET ?
7657	5253	@* ;		JMP *-4	/NO, CHECK TIME AGAIN
7660	6016	@* ;		RRB RFC	/YES, READ CHAR.& FETCH NEXT
7661	5650	@* ;		JMP I R750	/THEN RETURN

7662	7402	@* ;	R33,	XX	
7663	6031	@* ;		KSF	/FLAG SET YET ?
7664	5263	@* ;		JMP *-1	/NO, CHECK AGAIN
7665	6036	@* ;		KRB KFC	/YES, READ CHAR.& FETCH NEXT
7666	5662	@* ;		JMP I R33	/THEN RETURN

7667	0012	@* ;	LOW ,	R33 - R750	
			CHAR =	R33	/SHARED VARIABLE LOCATION.

/OUTPUT SUBROUTINE

7670	7402	@* ;	P33,	XX	
7671	6041	@* ;		TSF	/FLAG SET YET ?
7672	5271	@* ;		JMP *-1	/NO, CHECK AGAIN
7673	6046	@* ;		TLS	/CLEAR FLAG & PRINT CHAR.
7674	5670	@* ;		JMP I P33	/THEN RETURN

CONTD,

/LINKS WITH PROGRAMS ON OTHER PAGES

0000	7402	@0 ;	TIMER,	XX	/R750 SETS THIS =0 FOR EOT.
0004	7670	@4 ;		P33	
0005	7662	@* ;		R33	
0006	7650	@* ;		R750	

@CONTD

/ROUTINE TO READ CHARACTERS IGNORING ERROR  
 /MESSAGES AND FIELD SETTINGS.

7675	7402		READIN,	XX	
7676	7100	@* ;		CLL	/RESET RUBOUT SWITCH
7677	7402	@* ;		XX	/JMS R750 OR JMS R33 TO GET CHAR
7700	3262	@* ;		DCA CHAR	/SAVE IT
7701	1000	@* ;		TAD TIMER	
7702	7650	@* ;		SNA CLA	/WAS IT END-OF-TAPE ?
7703	5277	@* ;		JMP *-4	/YES TRY AGAIN
7704	1262	@* ;		TAD CHAR	
7705	1337	@* ;		TAD M377	
7706	7670	@* ;		SNA SZL CLA	/RUBOUT OR BETWEEN RUBOUTS
7707	5277	@* ;		JMP READIN+2	/YES, IGNORE THIS CHAR.
7710	1262	@* ;		TAD CHAR	
7711	0315	@* ;		AND M100	
7712	1251	@* ;		TAD M200	
7713	7450	@* ;		SNA	/LEADER-TRAILER ?
7714	5675	@* ;		JMP I READIN	/YES, RETURN TO CALL+1
7715	7700	@* ;	M100,	SMA CLA	/FIELD SETTING ?
7716	5276	@* ;		JMP READIN+1	/YES, IGNORE THIS CHAR.
7717	2275	@* ;		ISZ READIN	/NO, MUST BE BINARY
7720	5675	@* ;		JMP I READIN	/RETURN TO CALL+2

/MAIN LOADER ROUTINE

7721	6014	@*;	BEGN,	RFC	/ENTRY POINT
7722	4250	@*;	CJMSH,	JMS R750	
7723	1000	@*;		TAD TIMER	
7724	7650	@*;		SNA CLA	/WHICH READER ?
7725	1267	@*;		TAD LOW	
7726	1322	@*;		TAD CJMSH	
7727	3277	@*;		DCA READIN+2	/SET JUMP TO INPUT SR.
7730	3214	@*;		DCA CHKSUM	/RESET CHECKSUM
7731	2000	@*;		ISZ TIMER	/SET NON-ZERO IN CASE OF ITY
7732	4275	@*;		JMS READIN	
7733	5332	@*;		JMP *-1	/IGNORE LEADER
7734	1262	@*;	GO,	TAD CHAR	
7735	3215	@*;		DCA WORD	/SAVE FIRST CHAR OF PAIR
7736	4275	@*;		JMS READIN	
7737	7401	@*;	M377,	-377	/EAE NOP, NORMALLY SKIPPED
7740	1215	@*;		TAD WORD	
7741	1262	@*;		TAD CHAR	
7742	3213	@*;		DCA CKT	/ADD PAIR TOGETHER FOR CHECKSUM
7743	1215	@*;		TAD WORD	
7744	7106	@*;		CLL RTL	
7745	7006	@*;		RTL	
7746	7006	@*;		RTL	/ROTATE FIRST ONE OF PAIR
7747	1262	@*;		TAD CHAR	/ADD SECOND ONE TO IT
7750	3215	@*;		DCA WORD	/TO FORM COMPLETE WORD
7751	7250	@*;		STA RAR	
7752	3212	@*;		DCA MEM	/STORE -1 IF LINK SET
7753	4275	@*;		JMS READIN	
7754	5370	@*;		JMP BEND	/TRAILER, END OF BLOCK
7755	1213	@*;		TAD CKT	
7756	1214	@*;		TAD CHKSUM	
7757	3214	@*;		DCA CHKSUM	/UPDATE CHECKSUM
7760	1215	@*;		TAD WORD	
7761	2212	@*;		ISZ MEM	/WAS IT ORIGIN SETTING ?
7762	5365	@*;		JMP *+3	
7763	3216	@*;		DCA ORIGIN	/RESET ORIGIN
7764	5334	@*;		JMP GO	
7765	3616	@*;		DCA I ORIGIN	/LOAD IT
7766	2216	@*;		ISZ ORIGIN	
7767	5334	@*;		JMP GO	
7770	1215	@*;	BEND,	TAD WORD	/BLOCK END
7771	7041	@*;		CMA IAC	
7772	1214	@*;		TAD CHKSUM	
7773	7450	@*;		SNA	/CHECKSUM OK IF AC=0
7774	2212	@*;		ISZ MEM	/LOAD AND RUN ?
7775	7412	@*;		SKP HLT	/NO, DISPLAY CHECKSUM ERROR
7776	5616	@*;		JMP I ORIGIN	/YES, GO EXECUTE PROGRAM
7777	5321	@*;		JMP BEGN	

@7612 /VARIABLE STORAGE

7612	7402		MEM,	XX
7613	7402	@*;	CKT,	XX
7614	7402	@*;	CHKSUM,	XX
7615	7402	@*;	WORD,	XX
7616	7402	@*;	ORIGIN,	XX

/END OF PROGRAM  
PAUSE

BEGN	7721
BEND	7770
CHAR	7662
CHKSUM	7614
CJMSH	7722
CKT	7613
CONTD	7675

## A Note on the New System for Loading the PDP-8 Binary Loader

### Introduction

The PDP-8 has no hardware read-in or bootstrap facilities to get the normal binary loader into core when it has been accidentally overwritten. DEC supply a 16 instruction loader-loader which will read a tape in RIM format (this is used to load the BIN loader and some maintenance programs). A shorter (9 + 1 locations) loader has been developed which will load the binary loader in 'ARK' format. This note describes the techniques used.

### Description

A paper tape loader must include the following functions:

- a) Differentiate between the leader/trailer (needed to thread the reader) and the body of the tape.
- b) Assemble one or more tape characters to form a complete computer word.
- c) Know where in memory to store each word.
- d) Stop when loading is complete.

It may also test for a correct checksum or parity etc.

The methods of meeting these requirements with the simplest possible program are described below.

a) and b) To produce a 12 bit word needs at least 2, 8 bit tape characters. In BIN and RIM formats 6 bits of each tape character are used, the first being rotated six places left before being 'ORed' with the second to get the complete word. If however 2 bits of each character, <sup>except the first</sup> are used (needing ~~3~~ 4 characters to form a 12 bit word) they can be combined using

a single RTL instruction each time. If this is combined in a loop with the read instructions we need to count  $\$ 4$  iterations of the loop to know when a complete word has been assembled. This can be done neatly by punching the  $100_2$  ~~third~~ bit of the first character of the group of 4, this will be shifted into the link when the complete word is assembled; it also enables us to ignore blank leader.

c) There are several methods of defining where each word should be stored - successive words may be stored in sequential addresses starting at a defined one, or each address may be defined individually (as in RIM). In the first case the 'origin' may be defined in the loader or from special codes on the tape (e.g. the  $100_2$  bit punched as in RIM-BIN). The method chosen for ARK format is to make the first word on the tape the address definition and to load in successive locations thereafter. This can be handled by storing via an auto index register and initialising so that the first word is stored in the auto index register itself.

d) There are several ways of deciding when loading is complete - the program may read to the physical end of tape (as in RIM) - it may detect trailer words (as in BIN) - it may count the words it is loading, or it may overwrite itself with the program it is loading. The latter is the one chosen as it results in the simplest possible program. This means that the program must be located at the end of the area occupied by the BINary loader, and can therefore use the same starting address which is convenient.

There is of course no checksum or parity check in this format.

The basic loop is therefore:

Wait for reader and or char into acc.

SZL

DCA I AUTO

CLL RTL

JMP \* - 4

~~10~~ 10 AUTO, \* - 1

For the high speed reader version the RRB instruction will or the character into the ACC, however reader motion needs to be initiated with a RFC instruction.

For the teletype reader version tape motion is initiated by pressing start, however KFC (needed to fetch next character) clears the accumulator, so the acc. needs to be saved while it is issued.

Because of these considerations <sup>the</sup> both versions use § 9 instructions <sup>respectively</sup> and the auto index register.

The versions shown in the listing have been optimised for minimum switch juggling when loading.

#### Details of operation

The first word of the ARK tape is the address - 1 of the start of the area occupied by the BINary loader. The loader then loads the main part of BIN up to 7766; in 7767 and 7770 it loads SZLSPA and JMP BGN+1 respectively. When the JMP is executed control is transferred to the binary loader which loads the rest of itself (in BIN format) into 7767 - 7777 (JMP GO is initially duplicated in 7770 to enable loading into 7777, then 7770 is altered to normal) and the links in 0004 - 0006. This completes the operation. The program should halt PC = 7777 AC = 0, ready for use of the binary loader.

/BINARY LOADER WITH AUTOMATIC READER SELECTION.  
 /AND LOAD-AND-RUN FACILITY IC. 8-102-U-SYM  
 /CONTAINS INPUT SUBROUTINES AVAILABLE TO ALL PROGRAMS  
 /VERSION ARRANGED FOR ARK FORMAT INITIAL LOADER

@7650

/INPUT SUBROUTINES

7650	7402	R750,	XX	
7651	7600	M200,F	CLA	;F=400 /TO GET M200
7652	3000		DCA TIMER	/INITIALISE TIMER
7653	2000		ISZ TIMER	/TIME UP ?
7654	7410		SKP	/NO, LOOK AT FLAG
7655	5650		JMP I R750	/YES, RETURN
7656	6011		RSF	/FLAG SET YET ?
7657	5253		JMP *-4	/NO, CHECK TIME AGAIN
7660	6016		RRB RFC	/YES, READ CHAR.& FETCH NEXT
7661	5650		JMP I R750	/THEN RETURN

7662	7402	R33,	XX	
7663	6031		KSF	/FLAG SET YET ?
7664	5263		JMP *-1	/NO, CHECK AGAIN
7665	6036		KRB KFC	/YES, READ CHAR.& FETCH NEXT
7666	5662		JMP I R33	/THEN RETURN

7667	0012	LOW ,	R33 - R750,	
		CHAR =	R33	/SHARED VARIABLE LOCATION

/OUTPUT SUBROUTINE

7670	7402	P33,	XX	
7671	6041		TSF	/FLAG SET YET ?
7672	5271		JMP *-1	/NO, CHECK AGAIN
7673	6046		TLS	/CLEAR FLAG & PRINT CHAR.
7674	5670		JMP I P33	/THEN RETURN

/ROUTINE TO READ CHARACTERS IGNORING ERROR  
 /MESSAGES AND FIELD SETTINGS.

7675	7402	READIN,	XX	
7676	7100		CLL	/RESET RUBOUT SWITCH
7677	7402		XX	/JMS R750 OR JMS R33 TO READ CHAR.
7700	3262		DCA CHAR	/SAVE IT
7701	1000		TAD TIMER	
7702	7650		SNA CLA	/WAS IT END-OF-TAPE ?
7703	5277		JMP *-4	/YES TRY AGAIN
7704	1262		TAD CHAR	
7705	1337		TAD M377	
7706	7670		SNA SZL CLA	/RUBOUT OR BETWEEN RUBOUTS
7707	5277		JMP READIN+2	/YES, IGNORE THIS CHAR.
7710	1262		TAD CHAR	
7711	0315		AND M100	
7712	1251		TAD M200	
7713	7450		SNA	/LEADER-TRAILER ?
7714	5675		JMP I READIN	/YES, RETURN TO CALL+1
7715	7700	M100,	SMA CLA	/FIELD SETTING ?
7716	5276		JMP READIN+1	/YES, IGNORE THIS CHAR.
7717	2275		ISZ READIN	/NO, MUST BE BINARY
7720	5675		JMP I READIN	/RETURN TO CALL+2



```

/MMAIN LOADER ROUTINE
7721 6014 BEGN, RFC /ENTRY POINT
7722 4250 CJMSH, JMS R750
7723 1000 TAD TIMER
7724 7650 SNA CLA /WHICH READER ?
7725 1267 TAD LOW
7726 1322 TAD CJMSH
7727 3277 DCA READIN+2 /SET JUMP TO INPUT SR.
7730 3214 DCA CHKSUM /RESET CHECKSUM
7731 2000 ISZ TIMER /SET NON-ZERO IN CASE OF TTY
7732 4275 JMS READIN
7733 5332 JMP *-1 /IGNORE LEADER
7734 1262 GO, TAD CHAR
7735 3215 DCA WORD /SAVE FIRST CHAR OF PAIR
7736 4275 JMS READIN
7737 7401 M377, -377 /EAE NOP, NORMALLY SKIPPED
7740 1215 TAD WORD
7741 1262 TAD CHAR
7742 3213 DCA CKT /ADD PAIR TOGETHER FOR CHECKSUM
7743 1215 TAD WORD
7744 7106 CLL RTL ; RTL ; RTL /ROTATE FIRST ONE OF PAIR
7745 7006
7746 7006
7747 1262 TAD CHAR /ADD SECOND ONE TO IT
7750 3215 DCA WORD /TO FORM COMPLETE WORD
7751 7250 STA RAR
7752 3212 DCA MEM /STORE -1 IF LINK SET
7753 4275 JMS READIN
7754 5370 JMP BEND /TRAILER, END OF BLOCK
7755 1213 TAD CKT
7756 1214 TAD CHKSUM
7757 3214 DCA CHKSUM /UPDATE CHECKSUM
7760 1215 TAD WORD
7761 2212 ISZ MEM /WAS IT ORIGIN SETTING ?
7762 5365 JMP *+3
7763 3216 DCA ORIGIN /RESET ORIGIN
7764 5334 JMP GO
7765 3616 DCA I ORIGIN /LOAD IT
7766 2216 ISZ ORIGIN

/END OF SECTION OF BIN LOADER LOADED BY ARK INITIAL LOADER
7767 7530 SZL SPA /NEEDED FOR TTY VERSION OF ARK LOADER
7770 5321 BEND, JMP BEGN /TRANSFER FROM ARK TO BIN LOADER RTN

```

/SECTION TO BE LOADED BY BINARY LOADER

```

@BEND-1
7767 5334 JMP GO
7770 5334 BEND, JMP GO /THIS ENABLES LOADING INTO 7777
7771 7041 CIA
7772 1214 TAD CHKSUM
7773 7450 SNA /CHECKSUM OK IF AC.=0
7774 2212 ISZ MEM /LOAD AND RUN ?
7775 7412 SKP HLT /NO, DISPLAY CHECKSUM ERROR
7776 5616 JMP I ORIGIN /YES, GO EXECUTE PROGRAM
7777 5321 JMP BEGN
0000 7402 TIMER, XX /R750 SETS THIS =0 FOR EOT.

0004 7670 @4; P33
0005 7662 R33
0006 7650 R750 /LINKS WITH PROGRAMS ON OTHER PAGES

7770 1215 @BEND; TAD WORD /SET TO PROPER CONTENTS

```

```

7724 7650 SNA CLA /WHICH READER ?
7725 1267 TAD LOW
7726 1322 TAD CJMSH
7727 3277 DCA READIN+2 /SET JUMP TO INPUT SR.
7730 3214 DCA CHKSUM /RESET CHECKSUM
7731 2000 ISZ TIMER /SET NON-ZERO IN CASE OF TTY
7732 4275 JMS READIN
7733 5332 JMP *-1 /IGNORE LEADER
7734 1262 GO, TAD CHAR
7735 3215 DCA WORD /SAVE FIRST CHAR OF PAIR
7736 4275 JMS READIN
7737 7401 M377, -377 /EAE NOP, NORMALLY SKIPPED
7740 1215 TAD WORD
7741 1262 TAD CHAR
7742 3213 DCA CKT /ADD PAIR TOGETHER FOR CHECKSUM
7743 1215 TAD WORD
7744 7106 CLL RTL ; RTL ; RTL /ROTATE FIRST ONE OF PAIR
7745 7006
7746 7006
7747 1262 TAD CHAR /ADD SECOND ONE TO IT
7750 3215 DCA WORD /TO FORM COMPLETE WORD
7751 7250 STA RAR
7752 3212 DCA MEM /STORE -1 IF LINK SET
7753 4275 JMS READIN
7754 5370 JMP BEND /TRAILER, END OF BLOCK
7755 1213 TAD CKT
7756 1214 TAD CHKSUM
7757 3214 DCA CHKSUM /UPDATE CHECKSUM
7760 1215 TAD WORD
7761 2212 ISZ MEM /WAS IT ORIGIN SETTING ?
7762 5365 JMP **+3
7763 3216 DCA ORIGIN /RESET ORIGIN
7764 5334 JMP GO
7765 3616 DCA I ORIGIN /LOAD IT
7766 2216 ISZ ORIGIN

/END OF SECTION OF BIN LOADER LOADED BY ARK INITIAL LOADER
7767 7530 SZL SPA /NEEDED FOR TTY VERSION OF ARK LOADER
7770 5321 BEND, JMP BEGN /TRANSFER FROM ARK TO BIN LOADER RTN

```

/SECTION TO BE LOADED BY BINARY LOADER

@BEND-1

```

7767 5334 JMP GO
7770 5334 BEND, JMP GO /THIS ENABLES LOADING INTO 7777
7771 7041 CIA
7772 1214 TAD CHKSUM
7773 7450 SNA /CHECKSUM OK IF AC.=0
7774 2212 ISZ MEM /LOAD AND RUN ?
7775 7412 SKP HLT /NO, DISPLAY CHECKSUM ERROR
7776 5616 JMP I ORIGIN /YES, GO EXECUTE PROGRAM
7777 5321 JMP BEGN
0000 7402 TIMER, XX /R750 SETS THIS =0 FOR EOT.

0004 7670 @4; P33
0005 7662 R33
0006 7650 R750 /LINKS WITH PROGRAMS ON OTHER PAGES

7770 1215 @BEND; TAD WORD /SET TO PROPER CONTENTS

```

/DEFINITIONS (STORAGE NOT INITIALISED)

```

MEM = 7612
CKT = MEM+1
CHKSUM= CKT+1
WORD = CHKSUM+1
ORIGIN= WORD+1

```