

IDENTIFICATION

Product Code: DEC-08-LRAA-D
Product Name: Read-In-Mode Loader
Date Created: February 23, 1965
Maintainer: Software Services Group

1. ABSTRACT

The Rim Loader is a minimum-sized routine for reading and storing information contained in Read-In-Mode coded tapes via the ASR-33 perforated tape reader.

2. REQUIREMENTS

2.1 Storage

17 decimal (21 octal).

2.2 Equipment

4K PDP-8 with ASR-33.

3. USAGE

3.1 Loading

To place the Rim Loader into memory via the control console switches, proceed as follows:

3.1.1 Set 7756 in the switch register (SR).

3.1.2 Press LOAD ADDRESS.

3.1.3 Set the first instruction (6032) in the SR.

3.1.4 Press DEPOSIT.

3.1.5 Set the next instruction (6031) in the SR.

3.1.6 Press DEPOSIT.

3.1.7 Repeat steps 3.1.5 and 3.1.6 until all instructions have been deposited.

3.2 Calling Sequence

None. Cannot be called as a subroutine.

3.3 Start-up/Entry

3.3.1 Place the perforated tape which must be in RIM format in the perforated-tape reader.

3.3.2 Make sure that the ASR-33 is on line.

3.3.3 Place the starting address 7756 in the switch register.

3.3.4 Press the LOAD ADDRESS key.

3.3.5 Press console START key.

3.3.6 Move the reader control switch to START.

3.4 Errors

There are no error stops in this routine.

4. DESCRIPTION

This is a basic routine that alternately assembles an address from two successive characters on tape, then assembles data contained in the next two characters and stores this data at the associated address.

Because a tape in Rim format is twice as long as a comparable tape in Binary format, it is suggested that the Rim Loader only be used to read in the Binary Loader. After this, the Binary Loader should be used.

5. FORMAT

5.1 External Data

Tapes to be read by this program must be in Read-In-Mode coded format.

5.1.1 Leader should be about 2 feet of Leader-Trailer codes (any code with channel 8 punched; preferably code 200).

5.1.1.1 Depress ALT MODE, CTRL, and @ keys simultaneously to punch 200 Leader-Trailer.

5.1.2 Characters representing the absolute, machine language program are arranged in an alternating pattern of address, contents, address, contents, etc. Addresses have channel 7 punched, channel 8 not punched. Contents have no punch in channel 7 or 8.

5.1.3 Trailer: Same as Leader.

5.1.4 Example of Read-In-Mode coded format:

<p>Tape Channel <u>87 654 S 321</u></p> <p>10 000 . 000</p> <p>01 A1 . A2 00 A3 . A4</p> <p>00 D1 . D2 00 D3 . D4</p> <p>This pattern repeats in similar four character groups until the concluding</p> <p>10 000 . 000</p>	<p><u>Channels 8 and 7 Indicate.</u></p> <p>Leader code is always found first.</p> <p>Absolute octal Address of data in next two characters.</p> <p>Octal Data to be stored at preceding address.</p> <p>Trailer Code is encountered.</p>
--	---

5.1.4.1 Note that actually a 3-bit group (a single octal character) is designated by the notation A1 above. The "1" in this notation indicates that this particular octal character will be used as the most significant three bits in specifying the absolute address into which following data will be deposited. Correspondingly, A2, A3, and A4 designate successively less significant octal characters in the absolute address.

5.1.4.2 The remarks in 5.1.4.1 apply equally to Data as specified by the notation D1, D2, D3, and D4.

6. PROGRAM

6.1 Program Listing

<u>Abs. Addr.</u>	<u>Octal Contents</u>	<u>Tag</u>	<u>Instruction I Z</u>	<u>Comments</u>
7756,	6032	BEG,	KCC	/clear AC and flag
7757,	6031		KSF	/skip if flag = 1
7760,	5357		JMP .-1	/looking for char
7761,	6036		KRB	/read buffer

<u>Abs. Addr.</u>	<u>Octal Contents</u>	<u>Tag</u>	<u>Instruction I Z</u>	<u>Comments</u>
7762,	7106		CLL RTL	
7763,	7006		RTL	/ch8 in ACO
7764,	7510		SPA	/checking for leader
7765,	5357		JMP BEG+1	/found leader
7766,	7006		RTL	/OK, ch7 in link
7767,	6031		KSF	
7770,	5367		JMP .-1	
7771,	6034		KRS	/read, do not clear
7772,	7420		SNL	/checking for address
7773,	3776		DCA I TEMP	/store contents
7774,	3376		DCA TEMP	/store address
7775,	5356		JMP BEG	/next word
7776,	0	TEMP,	0	/temp storage
7777,	JMP start of bin loader		0	

7. ADDENDA

7.1 Other Possible RIM Loaders

Variations of the RIM Loader described may prove useful in special cases where, due to circumstances, RIM must be located in a different section of core.

On the other hand, the equipment involved may make it necessary to use a variation of RIM tailored specifically to a particular situation. As an example of this, consider a special-purpose PDP-8 system used for text editing. In this system, no ASR-33's are used. Instead several typewriters which use a different code (including provision for upper and lower case) are time-shared with respect to input and output with a central PDP-8.

Please consult the Applied Programming Department at Digital Equipment Corporation for details of other RIM Loaders currently available or for assistance in special cases.

7.2 The Use of PDP-8 System Programs

Certain system programs such as the DECTape Library System (Digital-8-8-5) require that the RIM Loader precisely as listed above (6.1) be used.

7.3 Using the RIM Loader with an Extended Memory

The RIM Loader as described and listed up to this point may actually run in any memory field provided that it is loaded into memory following a slightly different procedure than that described in 3.1 above. The Instruction Field register and the Data Field register must both be set to N (a number from 0 to 7) where N indicates the memory field in which the RIM Loader is to be placed. This is easily done.

7.3.1 Set the data field extension of the switch register to N.

7.3.2 Set the instruction field extension of the switch register to N.

7.3.3 Follow procedure in steps 3.1.1 through 3.1.7 above.

