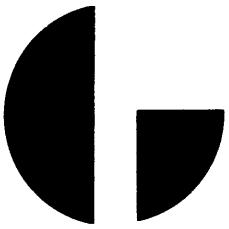


DATA GENERAL
CORPORATION
Southboro,
Massachusetts 01772
(617) 485-9100

PROGRAM

Relocatable Loader

TAPES

091-000016

October, 1969

093-000039-00

RELOCATABLE LOADER

INTRODUCTION

The NOVA relocatable loader is used to load binary tapes produced as output by the relocatable assembler. The loader accepts any number of relocatable binary tapes as input, resolves external displacements and normal externals, and maintains an entry symbol table that can be printed on demand. Extensive error detection logic is provided to prevent various fatal and non-fatal errors.

1 OPERATING PROCEDURE

The relocatable loader is almost exclusively keyboard directed, *ie*, actions are initiated by transmitting numeric codes from the teletype. The loader itself is provided in absolute binary format and is loaded using the binary loader.

The routine will self-start and type

SAFE =

on the teletype. The program is requesting the user to specify the number of words (in octal) that he wants to preserve from destruction by the loader at the high end of memory. The user may respond with a carriage return (the default case) and the loader will save 200 words, thus preserving the bootstrap and binary loaders. Otherwise, an octal number can be input (terminated by a carriage return) to explictly determine the number of words to be "safe." Errors in this input will be indicated by repeating the original message. The error cases are:

1. A character other than an octal digit or a carriage return is input.
2. More than five octal digits are input.
3. The number specified is too large for the memory configuration, *ie*, no load space remains.

Once SAFE has been correctly specified, its value is fixed for the entire duration of the load process.

The loader requests directives from the user by sending

*

to the teletype. The user should respond with a single digit code. This code will determine the action to be taken by the loader. If an illegal code is input, a "?" will be printed and no action results. The legitimate responses are discussed in the remainder of this manual and summarized in Appendix A.

2 THE RELOCATABLE LOAD MODE

The loader is initialized to assume two relocation variables. The first is used for loading page zero relocatable code and the second is used for loading normal relocatable code. These variables will be referred to as ZMAX and NMAX respectively. Initially they assume the following values:

Variable	Value (octal)
ZMAX	50
NMAX	400

During loading, these variables are updated to reflect the *next available* memory cells for page zero relocatable code and normal relocatable code respectively. For example, the initial values imply that page zero relative address 00000 will be loaded into location 00050 and that normal relative address 00000 will be loaded into location 00400.

To load a single relocatable binary tape, the mode response is

1 for input from the teletype reader

or

2 for input from the paper tape reader.

The binary tape will be loaded and the loader will respond "*" after processing the *start block*. ZMAX and NMAX will have been updated to reflect the new state

of memory available for subsequent programs. The values of ZMAX and NMAX can be printed on the teletype on demand by giving a mode response of *5*. Note that these variables are also printed with the symbol table (§3).

3 THE SYMBOL TABLE

The loader initially sizes memory and requests SAFE to determine a starting address from which a symbol table is constructed downward in memory by the loader. Any number of cells at the top of memory can be preserved by initializing SAFE. The symbol table (also called *loader map*) always maintains the current values of ZMAX and NMAX. In addition, all entry symbols and undefined external symbols appear in the table. The table can be printed on demand to the teletype by giving a mode response of *6*. The format of the table is

NMAX *dddddd*
ZMAX *dddddd*

um sssss dddddd
um sssss dddddd
.
.
.

NMAX and ZMAX are printed followed by their current six digit octal values. All subsequent symbols are printed alphabetically. Each symbol appears as five letters represented above as *sssss*. The flag, *u*, will be printed as "U" if the symbol following is an external for which no entry has yet been defined, *ie*, the symbol is an unresolved external. Normally, *u* will be printed as a space. The flag, *m*, will be printed as "M" under two conditions.

1. The symbol is an entry symbol that appeared more than once during the loading procedure, *ie*, the multiply defined entry. The first occurrence determines the entry address value.
2. A normal external symbol was encountered with the same name as an external displacement or visa versa.

Normally, *m* will be printed as a space. A six digit octal number, *dddddd*, will follow the symbol. If the symbol is defined (*ie* no *u* flag), this number is the absolute value of the entry point. If the symbol is undefined, this number is the absolute value of the last loaded reference to the symbol. (If no references to the external symbol have been made, this value will be 177777 for a normal external or 77777 for an external displacement.)

4 INITIATING EXECUTION

The loading procedure *must* be terminated by giving a mode response of *8*. Only after performing the above will the programs loaded reside at the absolute addresses indicated by the loader map. This is true because the loader itself resides in low core. Until the load procedure is terminated, all programs are loaded assuming a pseudo address for location 00000 which exists *above* the loader itself. Memory is shuffled down to reflect the true addresses only after a mode response of *8* (§Appendix D).

At this time the loader also passes information to the loaded routines that may be useful for their execution. To do this, absolute location 00044 is *unconditionally overwritten*, to contain a pointer to this information. This pointer can be used to gain access to the following:

1. The last values of ZMAX and NMAX.
2. The address of the start of the loader's symbol table.
3. The address of the first memory cell not used below the symbol table.
4. The starting address of the loaded system or 177777.

If this pointer is loaded into an index register, e.g.

```
LDA     2,44
```

the following displacements can be used to access the information specified.

Displacement	Information
0	ZMAX
1	Address of start of the symbol table
2	Address of the first memory cell not used by the symbol table
3	NMAX
4	Starting address or 177777

NMAX is useful in determining the area of memory available, for example, for dynamic storage allocation. The symbol table pointers will normally be used by the symbolic debugger. The user, however, may also wish to use the symbol table. Note that the symbol table is built downward in memory and its length is equal to the starting address minus the first available address. The example below illustrates how this length can be computed.

```
LDA  3,44
LDA  0,1,3
LDA  1,2,3
SUB  1,0     ; LENGTH IN AC0
```

After shuffling memory, the loader will HALT. Upon pressing continue, the loader will HALT again if no starting address had been specified by *any* binary tape loaded, or start execution of the loaded system at the starting address *last* specified by a loaded binary tape. For example, if only one of a number of binary tapes loaded specified a starting address, this address will receive control regardless of what order the tapes were loaded.

5 SPECIAL LOAD MODES

Three special action load modes can be initiated using the relocatable loader. The first enables the user to force a value for NMAX and thus determine the absolute load address for normally relocatable code. The second mode enables

selective loading of *library* routines. The third mode causes the loader to store *title* and *local* symbols in the loader map as well as entry and extern symbols.

To force a value for NMAX,

1. Enter the octal value desired for NMAX in the data switches, bits 1-15 (bit 0 is ignored).
2. Give a mode response of *3*.

Caution. If a value of NMAX less than the previous value is specified the user must be certain no overwrites of previously loaded programs will occur (§6). Further, NMAX will no longer reflect the first available-memory location for dynamic storage allocation.

The library load mode is initiated by a special block type preceding binary library tapes (§Library File Editor write-up). Library tapes are provided by Data General as part of their standard software package. A library tape is loaded in a manner identical to that described in §2. However, the loader performs its functions differently. First, the loader does not request a new mode after start blocks, but only after encountering a special library *end block*. Second, individual programs will be loaded only if at least one entry symbol defined by the program is currently called for by an unresolved external. For example, if a library tape that itself calls for no external symbols is the first tape loaded, none of its routines will be loaded.

The loader maintains a switch to determine whether or not to load local and title symbols. Initially (and after reinitialization (§7)), this switch is reset, *ie* local and title symbols will be ignored. It can be *complemented* by a mode response of *4*. The loader will respond *R* or *S* to indicate the new state of the switch, reset or set. The user should force the loading of local and

title symbols only if he intends to use the symbolic debugger. Otherwise, this mode uses symbol table storage unnecessarily.

6 ERROR DETECTION

The loader detects errors of two types--fatal and non-fatal. Fatal errors prevent further loader action unless the user reinitializes (§7). Non-fatal errors do not stop the loading process, but, of course, may change the intended state of the user's loaded systems. All errors are indicated by a two letter error code sent to the teletype. This code may be followed by a symbol name and will always be followed by a six digit octal number. The general form is thus

ee ssssss dddddd.

Non-fatal Errors

If an entry symbol having the same name as one already defined is encountered during the load process, a *multiple entry* error results. The error message has the form

ME *ssssss dddddd*

where *ssssss* represents the symbol name and *dddddd* its absolute address as previously defined. The loader proceeds by ignoring the *latest* definition.

If the loader, while attempting to resolve an external displacement, determines the displacement to be too large*, a *displacement overflow* error results. The error message has the form

DO *dddddd*

where *dddddd* represents the absolute address where the overflow occurred. The displacement is left unresolved with a value of 000.

---

*With index = 00, displacement (unsigned) > 377.
With index ≠ 00, -200 ≤ displacement < +200.

If a normal external symbol is encountered having the same name as a symbol previously defined as an external displacement, a *normal conflict with displacement* error results. The error message has the form

ND *ssss dddddd*

where *ssss* is the symbol name in conflict and *dddddd* is the absolute address of the last reference to the normal external symbol. The normal external symbol is ignored.

If the time between input characters becomes excessive, a *timeout* error occurs. The usual cause of this is a binary tape without a start block or a library tape without an end block. The error message has the form

TO *dddddd*

where *dddddd* is the location within the loader where the timeout occurred. The loader will request a new mode after a timeout.

If an illegal relocatable block type is read, an *illegal block* error results. Indication is of the form

IB *dddddd*

where *dddddd* represents the illegal block code. The loader will request a new mode after this error. If an improper tape had been mounted in the reader, it should be replaced by a relocatable binary or library tape and the load attempted again.

Fatal Errors

If the value of NMAX conflicts with the bottom of the loader's symbol table, a *memory overflow* error occurs. This implies the user programs are too large (or too numerous) to be loaded in the present memory configuration. The error indication is of the form

MO *dddddd*

where *dddddd* is the value of NMAX that caused the memory conflict.

If a checksum computed on any block differs from zero, a *checksum* error results. The error indication is of the form

CS *dddddd*

where *dddddd* is the incorrect checksum that was computed.

If an external displacement symbol is encountered having the same name as a symbol previously defined as a normal external, a *displacement conflict with normal* error results. The error message has the form

DN *ssss dddddd*

where *ssss* represents the symbol name in conflict and *dddddd* the absolute address of the last reference to the normal external symbol.

The loader does not permit memory cells, once loaded, to be overwritten by subsequent data. If an attempt to do this is made, an *overwrite* error results. Indication is of the form

OW *dddddd*

where *dddddd* represents the absolute address where the overwrite was attempted.

The loading of page zero relocatable code may overflow the page zero boundary of 377. If this occurs, a *page zero overflow* error occurs. Indication is of the form

ZO *dddddd*

where *dddddd* represents the absolute page zero address of the first word of the data block which caused the overflow.

Although the assembler restricts addresses to the range

$$0 \leq adr < 2^{15}$$

a reader error may cause bit 0 of an address word to be set. If this should occur, a *negative address* error occurs. Indication is of the form

NA *dddddd*

where *dddddd* represents the negative address.

7 REINITIALIZATION

If the load process is terminated by a fatal error or the user wishes to begin again, the loader must be reinitialized. Among other things, ZMAX and NMAX are set to 50 and 400 respectively and all symbol table entries are eliminated. To reinitialize, respond with a mode of 7.

8 LOADER PROPERTIES

The number of memory cells available for program loading is a function of the loader size, the memory configuration, SAFE, and the number of symbols entered in the loader symbol table. An *approximate* formula for determining the amount of memory available for loading relocatable programs is the following

$$MC - 2000 - SAFE - 3*NE$$

where *MC* is the machine's memory capacity and *NE* is the number of entry points defined in total by all relocatable programs to be loaded. All quantities are assumed octal.

APPENDIX A

LOADER MODES

Mode Response	Effect
1	Load a relocatable binary or a relocatable library tape from the teletype
2	Load a relocatable binary or a relocatable library tape from the paper tape reader
3	Force a load address for normally relocatable code
4	Complement the load all symbols switch
5	Print current memory limits
6	Print a loader map
7	Reinitialize the loader
8	Terminate the load process to prepare for execution

APPENDIX B

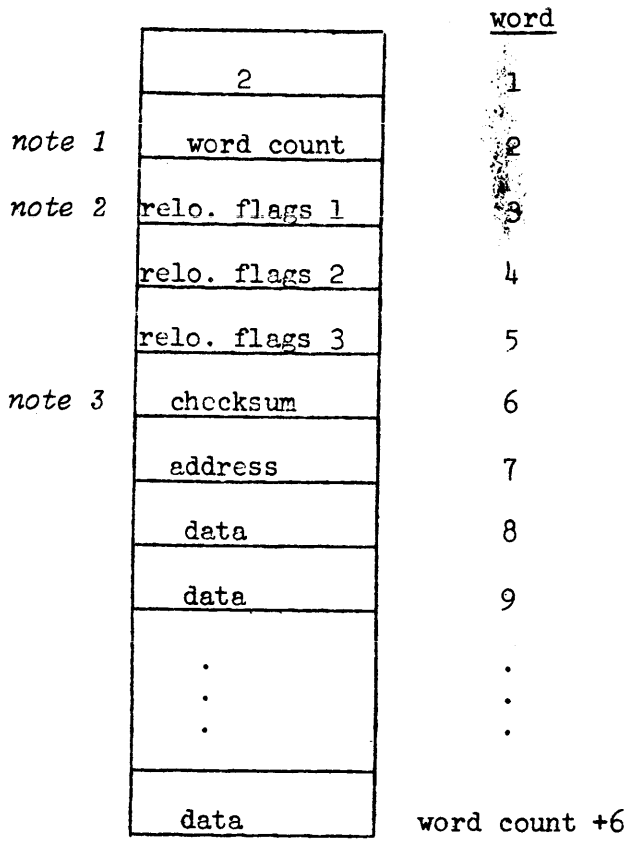
ERROR SUMMARY

Code	Error Description	Fatal
CS	Checksum error	Yes
DO	Displacement overflow	No
DN	External displacement conflict with a normal external	Yes
ME	Multiply defined entry	No
MO	Memory overflow	Yes
NA	Negative address read	Yes
ND	Normal external conflict with an external displacement	No
OW	Memory cell overwrite	Yes
TO	Input timeout	No
ZO	Page zero overflow	Yes

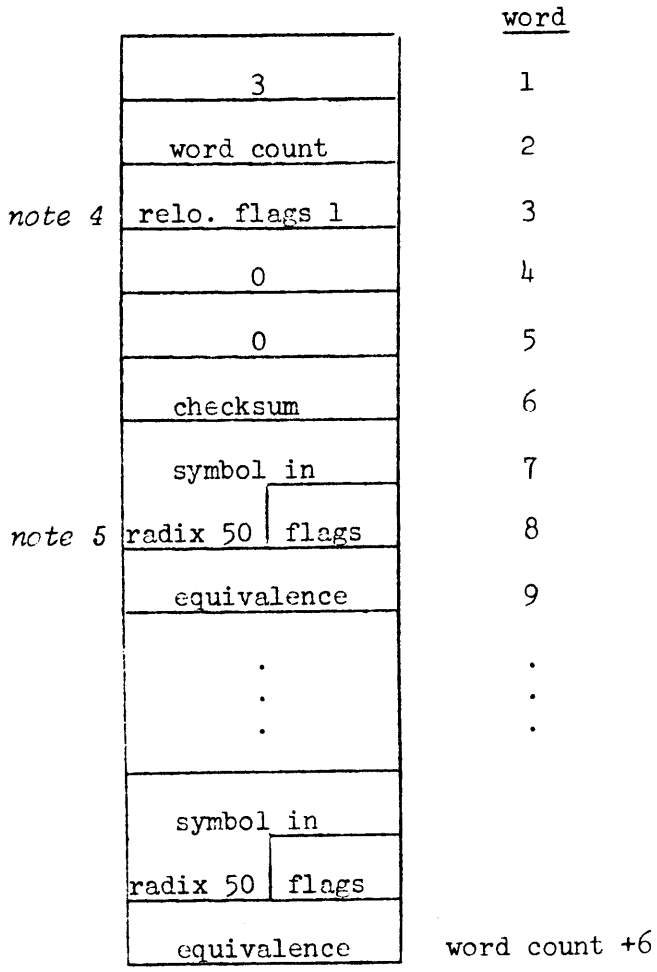
APPENDIX C

BLOCK FORMATS

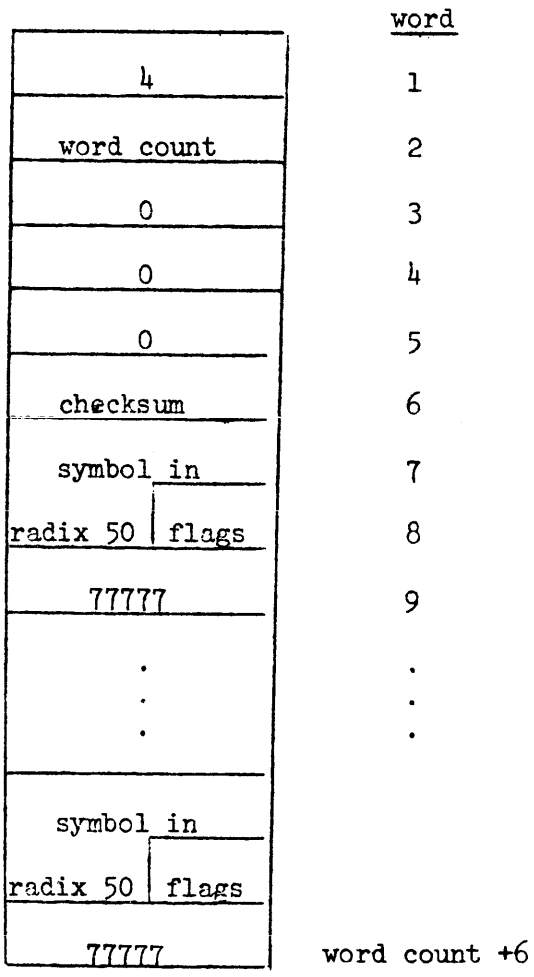
Relocatable Data Block



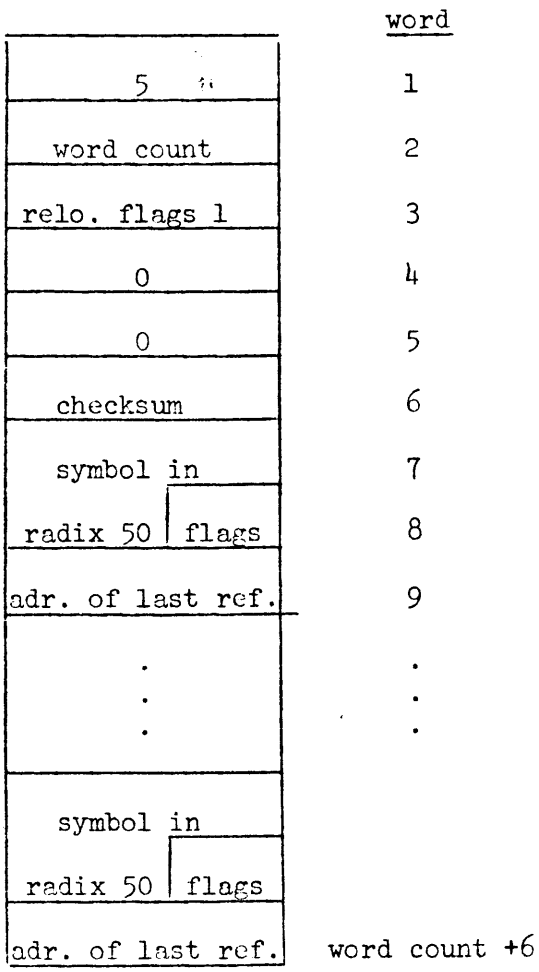
Entry Block


word
3
1
word count
2
note 4
relo. flags 1
3
0
4
0
5
checksum
6
symbol in
7
note 5
radix 50
flags
8
equivalence
9
symbol in
radix 50
flags
equivalence
word count +6

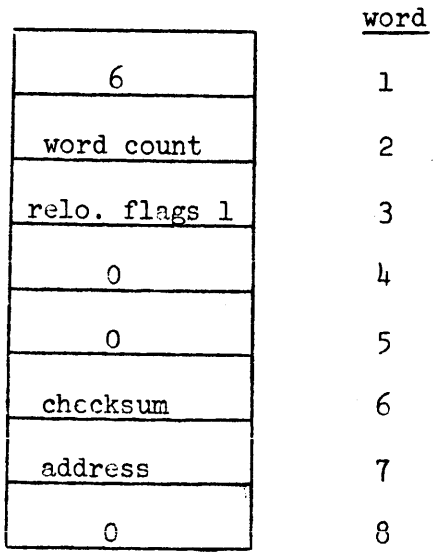
External Displacement Block



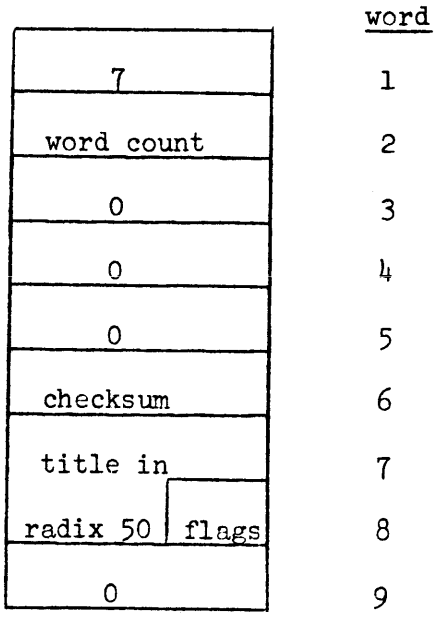
Normal External Block



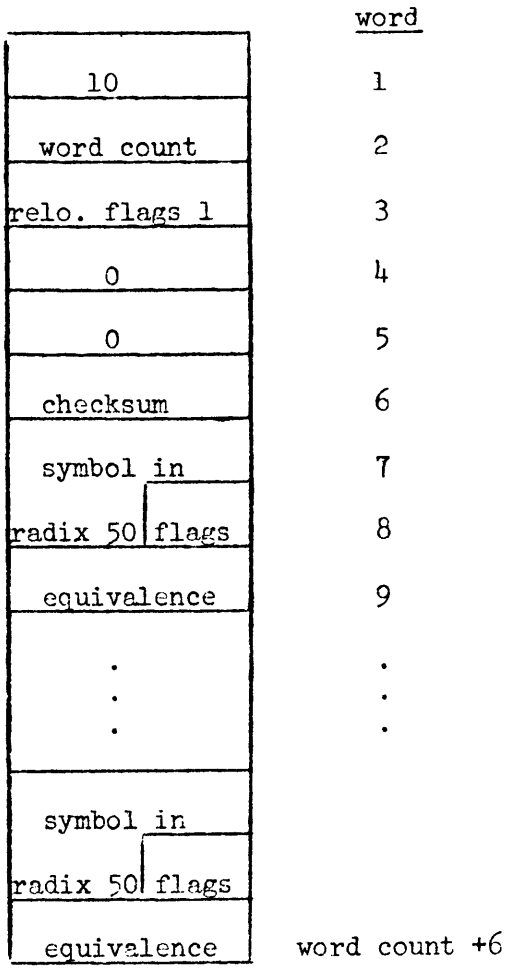
Start Block



Title Block



Local Symbol Block



Library Start Block

	word
11	1
0	2
0	3
0	4
0	5
-11	6

Library End Block

	word
12	1
0	2
0	3
0	4
0	5
-12	6

Note 1 Word counts are always two's complemented. The word count will never exceed sixteen.

Note 2 Relocation flags apply to the address and each successive data word. Three bits are allocated to each address or data word. Bits 0-2 apply to the address, bits 3-5 apply to the first data word, etc. The assigned meaning of these bits is given below.

Bit Settings	Meaning
000	Illegal
001	Absolute
010	Normal Relocatable
011	Normal Byte Relocatable
100	Page Zero Relocatable
101	Page Zero Byte Relocatable
110	Data References External Displacement
111	Illegal

Note 3 The checksum is such that the sum of all words in the block is 0.

Note 4 Relocation flags are identical to those described in *Note 2* but apply to the third word of every symbol entry (e.g. the equivalence of entry symbols).

Note 5 Symbol flags are reserved for bits 11-15 of the second word of each three-word symbol entry. Currently assigned are the following.

Bit Settings	Meaning
00000	Entry Symbol
00001	Normal External Symbol
00011	External Displacement Symbol
00100	Title Symbol
01000	Local Symbol

APPENDIX D

LOADER STORAGE ALLOCATION