variables

top
bot   / bottom of symbol table
blk1 - points to buffer last
list - bottom of symbol table
memory - next storage location
factor - relocation factor
gws1 -
blj
stable - dispatch table for type of block
last - 104 word buffer at end of loader
blk2
first
blk3
checksum - checksum
blk4 - lio
value - cumulative value of symbols already processed
bitrc - counter for relocation bit pairs
blj - holds address of block type dispatch table
blk - jmp
codes - storage for relocation bits word

7775 - holds entry to symbol package punch
7776 - holds entry jumps to loader
7777 -

subroutines

bits - get next two relocation bits + process with word.
    Ø - take care of external symbol, reënter bits from top
    others get value of

Gword - get a word

ad - tad

Xsym — calculates value of symbol if available

Operation of Linking Loader

SS 1 ↓ print out starting position

2 ↑ read starting location from TW

3

4

5

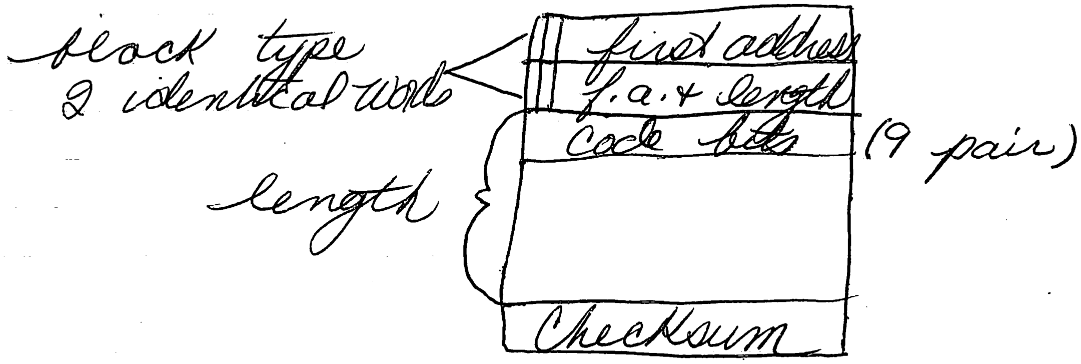
6

Linking Loader

General Tape Format

title	jmp 7776	block	. . .	block	jmp block

General Block Format



Block Types

Absolute - ∅

load at address specified (note there may be relocatable words in an absolute block- references to a non-absolute block)

Relocatable - 1

take relocation factor, add address, load there

Jump - 2

one word of data which is the starting address of the program just loaded format is the same as above 2 blocks

Block Types, cont.

Library - 3

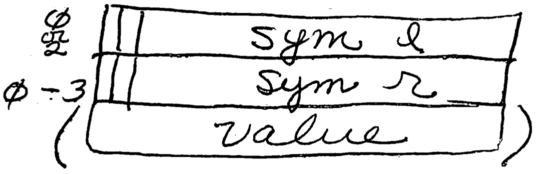
Simply a list of global symbols

if any one is needed the next data block are loaded. If not, blocks are skipped until another library or a jump block

Code Bits

External Symbol - 0

Read next two words as symbol (Sqoze code)



sym r always exists
Sym l may not

Sym r code bits

Look up in Symbol table - 0

if Sym l = 0 add value from table to current word
if Sym l = 2 subtract value in table from current word.

Use absolute value - 1

Relocate subtract - 2

Relocate add - 3

read next word as value and add factor masked to 12 bits if Sym l = 0; 16 bits if = 2

If a symbol is not in the Symbol table - add it to the table with the current memory address and whether to add or subtract

If in but not defined add current address to list of places to update

If in and currently being defined update according to list

If defined and being defined, complain if the two values don't match

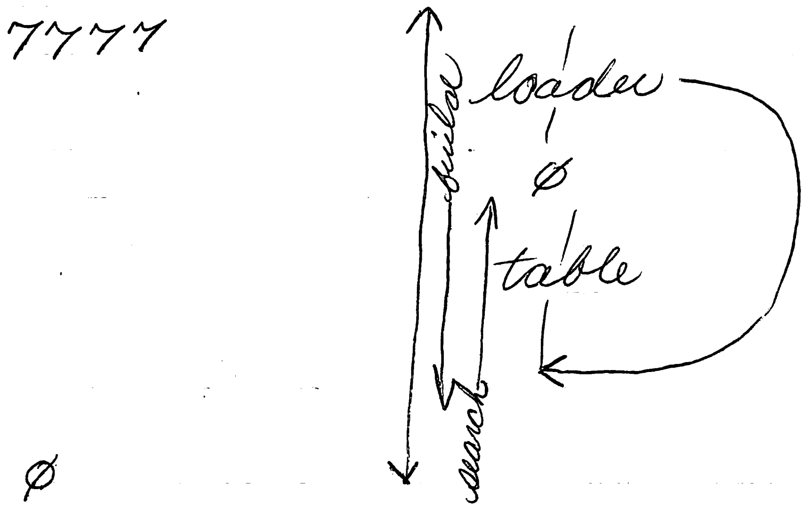
Don't Relocate - 1

Relocate subtract - 2

Relocate - 3

Take next full word, add rfactor, put in memory

Symbol Table Structure



Symbol Table Structure, cont.

Defined Entry

value
(sym l)

| | ∅ | sym r |

∅ sym l exists
1 Sym l doesn't exist

Undefined Entry

address n

⋮

address 1
n
(sym l)

| | 1 | sym r |

∅ sym l exists
1 Sym l doesn't exist