This document was produced by SDC in performance of contract DAHC15-73-C-0080

SERIES BASE NO VOL. REISSUE
TM 4310 /400/ 00

AUTHOR John Burger
TECHNICAL
RELEASE Thomas G. Williams
for M. I. Bernstein

REPRINT

TECH MEMO *a working paper*

System Development Corporation/2500 Colorado Ave./Santa Monica, California 90406

LISP 1.5 for the 360 Computer
Data Space Descriptions

Abstract

This is one in a series of documents describing SDC's LISP 1.5 system. Descriptions are given for all data spaces in terms of address boundaries, internal storage conventions, and relevant LISP functions.

1. INTRODUCTION

This document describes each of the data spaces in SDC's LISP 1.5 system (including the virtual space for small integers). For each space the following information is given where applicable: Space name; the internal space name, known as the quantum number; names of boundaries and pointers used by and for the space; addresses relative to the LISP system origin; the name of the LISP generator function for data in the space; ways in which the space adds data; descriptions of the treatment given the space by the LISP Garbage Collector; and a diagram of a typical element in the space.

The following definitions are used:

An absolute address is a number representing an S/360 address that requires no base register modification.

A LISP 24-bit pointer is a full-word address relative to the LISP system origin. (usually 1ØØØØ hex). These numbers range from Ø to 3FFFC hex.

A LISP 16-bit pointer is a LISP 24-bit pointer shifted right 2 bits. Note that the range and specification requirements on a LISP 24-bit pointer assure that no information will be lost in conversion to and from the 16-bit equivalent.

A quantum number is a name used internally by LISP to identify the various spaces. These numbers are maintained in the Quantitized Core Map (Entry QCM) which keeps track of the current size and boundary locations of all the spaces. See "Entry QCM" in TM-4310/300/00 for further details.

The names of the various boundaries and pointers (PRSO, PR20, AIB, BPP, PDO, etc.) are each an entry in Entry Space. (see TM-4310/300/00). An example best indicates how these are used. "ARP" is the name of the "current ARray Pointer", and Entry ARP contains an address relative to the LISP system origin which points to the next available computer word in Array Space.

One of the initial phases of the LISP Garbage Collector is the mark phase, which determines whether or not a given datum is, or is not necessary to the current operating system. Items that might be reclaimed by the Garbage Collector are marked if they are pointed to (directly or via a chain of pointers) by certain fundamental pointers. If the item itself points on to further information it is then marked from. Data reclamation is accomplished by the Garbage Collector by pruning, folding, or moving.

2. ENTRY SPACE

Quantum number: 1.
Lower boundary: SORG Upper space boundary: CHO
Lower boundary address: Ø Upper boundary address: 1ØØØ
Description: Contains Entrys as described in TM-4310/300/00.

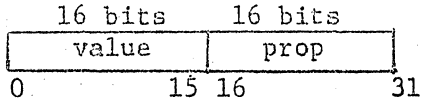
3. CHARACTER IDENTIFIER SPACE

Quantum Number: 1.

Lower boundary: CHO Upper boundary: CHE.

Lower Boundary address: 1000 Upper boundary address: 1400

Description: This space contains 256 full word items each of which corresponds to an EBCDIC character and contains the pointers to the property list and system value list for the identifier named by the character. The Garbage Collector marks from these items but does not mark them and they are not reclaimed. There is no generator function.



value is a LISP 16-bit pointer to a chain of PRS items that represent the values of the identifier in each declared section. Prop is a 16-bit pointer to the property list for the identifier.

4. PROGRAM REFERENCE SPACE (A)

Quantum number: 3

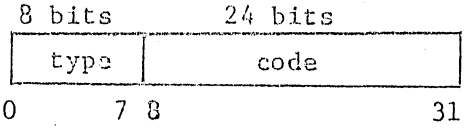
Lower boundary: PRS0 Upper boundary: PRSE

Intermediate fixed pointers at 1-page intervals: PR20, PR30, etc.

Generator function: (MAKEPR . 122)

Description: Items in this space never move. They form the fixed reference to link all functions, Special and Unspecial variables, Quote cells, Macros and Instructions. Unused items are chained from Entry PRSF. The Garbage Collector marks from these items, marks them, and reclaims them by pruning. Each item is associated with a corresponding word in Program Reference Space (B). PRS(B) is not pointed to and the separation between the two spaces (difference between the addresses of the same item in each space) is the value of Entry DPRS.

For Quote cells, Special and Unspecial variables, the value in a PRS(A) item is a LISP 24-bit pointer.

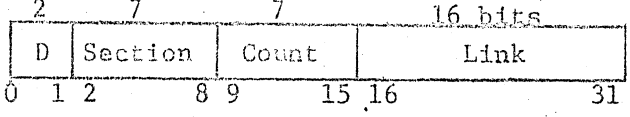


For Functions, Macros, and Instructions, the value of a PRS(A) item is in two fields: type is an 8-bit field with the value ∅ for Functions, 1 for Macros and 2 for Instructions. Code is a 24-bit absolute address pointing to the beginning of the code in Binary Program Space.

5. PROGRAM REFERENCE SPACE (B)

Lower Boundary: PGO Upper boundary: PGE

Description: Items in this space are associated with corresponding items in Program Reference Space (A). PRS(A) items contain the value of the item while PRS(B) items contain various descriptive data and the linking information.



D is a descriptor field that identifies the type of PRS data. The following values apply:

∅	Quote cell
1	Function, Macro or Instruction
2	Special Variables
3	Unspecial Variables

Section is a section number from ∅ to 127, inclusive.

Count is the number of references to this PRS item from code in Binary Program Space.

Link is a LISP 16-bit pointer that points to another item in PRS(A). All values of a given identifier that have been declared are strung together using this link. For any identifier a single value may be declared in each section. The first item on any identifier chain is pointed to by the value field of the

identifier (see sections 2., Character Identifier Space, and 9., Identifier Space). The link of the last item on a chain is Ø. This link field is also used to string all unused PRS items from Entry PRSF.

6. FLOATING POINT SPACE

Quantum Number: 4
Lower Boundary FLO Upper Boundary: FLE Pointers FLP and OFLP
Generator function: (MAKEFLOT . 122)
Description: Each item contains a 2-word double precision floating point number. New items are added contiguously from the high core end towards low core (e.g. from FLE towards FLO). Items are marked, not marked from and are reclaimed by the Garbage Collector by folding down.

64-BIT REAL NUMBER

7. INTEGER SPACE

Quantum Number: 7
Lower Boundary: INO Upper Boundary: IBB Pointers: INP and OINP
Generator function: (IN2S . 122)
Description: These items are 32-bit signed binary numbers which LISP prints in decimal. Values are in the range $-2^{31} \leq n < -4096$ and $4096 \leq n \leq 2^{31}$. New items are added contiguously from the low core end (INO) toward high core. Items are marked, not marked from, and are reclaimed by folding up. This space shares a floating boundary (IBB) with Bit Space.

32-BIT INTEGER

8. BIT SPACE

Quantum Number: 8
Lower Boundary: IBB Upper Boundary BTO Pointers: BTP and OBTP
Generator Function: (MAKEBIT . 122)

Description: These items are 32-bit logical (unsigned) binary numbers which LISP will print in either Hexidecimal or Octal. Values are in the range to ØØØØØØØØ to FFFFFFFF. New items are added contiguously from the high core end (BTO) toward low core. Items are marked, not marked from, and are reclaimed by folding down. This space shares a floating boundary (IBB) with Integer Space.

32 BIT NUMBER

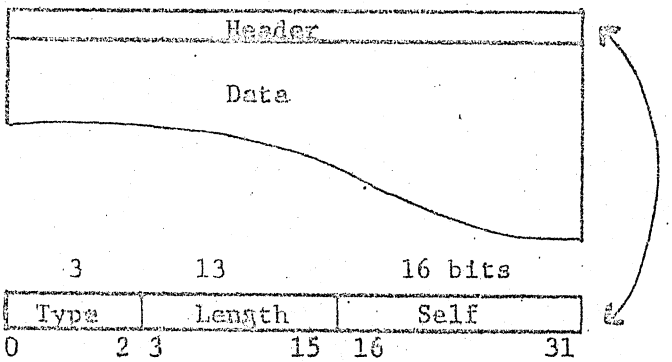
9. ARRAY SPACE

Quantum Number: 5

Lower Boundary: ARO Upper Boundary: AIB Pointers: ARP and OARP.

Generator Function (GETARRAY . 122)

Description: Items in this space are variable length collections (or tables) of numbers, EBCDIC characters, or symbolic data (LISP 16-bit pointers). The first computer word in an array is a header containing descriptive information for the LISP system and the rest of the array consists of 1 or more 1-, 2-, 4-, or 8-byte elements. Arrays are marked and reclaimed by moving up. Arrays of type "symbol" and "table" are marked from. New entrys are added contiguously from the low core end of the space (ARO) towards high core. Entry Space shares a floating boundary (AIB) with Identifier Space.



Ø	STRING	EBCDIC byte	8191
1	INTEGER	full word number	2Ø47
2	BIT	full word number	2Ø47
3	ID	EBCDIC byte	8191
4	FLOAT	double word real number	1Ø23
5	not used		
6	SYMBOL	LISP 16-bit pointer	4Ø95
7	TABLE	LISP 16-bit pointer	4Ø95

Length is an unsigned 13-bit number indicating the length of the array in bytes, not including the header. If type is STRING, ID, SYMBOL, or TABLE, the core space allotted to the array is rounded up to the next full-word boundary.

Self is a LISP 16-bit pointer pointing to the header of the array. This field is used only by the Garbage Collector.

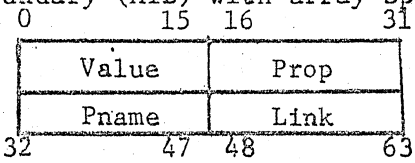
10. IDENTIFIER SPACE

Quantum Number: 6

Lower Boundary: AIB Upper Boundary IDE Pointers IDP and OIDP

Generator Function: (MAKEID . 122)

Description: Each item in this space represents a single identifier which either has a unique print name or has no name at all. (GENSYM). Each item is two computer words long and is marked, marked from in the property list and print name and the Garbage Collector reclaims by folding down. Identifier space shares a floating boundary (AIB) with array space.



Value is a LISP 16-bit pointer to a chain of PRS items that represent the values of the identifier in each declared section. Prop is a LISP 16-bit pointer to the property list of the identifier. Pname is a LISP 16-bit pointer to an item in Array Space containing the print (and read) name of the identifier. This array will be of type ID if the character string can be symmetrically printed without the $$ artifact mechanism; otherwise, the array is of type STRING, Pname will be all zeros (pointer to NIL) if the identifier is a Gensym (i.e. has no name). Link is a LISP 16-bit pointer to the next identifier in the same Oblist bucket. These buckets are each strung from a word in the 137-word OBLS in Entry space, The last identifier in the bucket points to NIL.

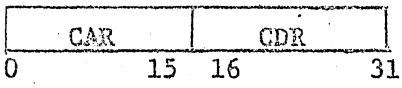
11. NODE SPACE

Quantum Number: 9

Lower Boundary: LSO Upper Boundary: LBB Pointers: LSP and OLSP.

Generator Function: (CONS . Ø)

Description: Each item in Node Space contains two LISP 16-bit pointers, the CAR and the CDR. A pointer to any space other than Node Space is considered to point to an atom. Nodes are therefore non-atomic and are the basic structure-building element in the LISP system and language. These items are marked and marked from in both the CAR and CDR, and are reclaimed by folding up. Node space shares a floating boundary (LBB) with Binary Program Space.



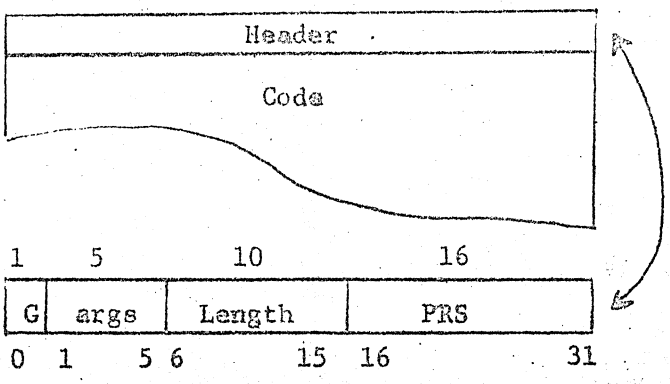
12. BINARY PROGRAM SPACE

Quantum Number: 1Ø

Lower Boundary: LBB Upper Boundary BPE Pointers: BPP and OBPP

Generator Function: (GETBPS . 122)

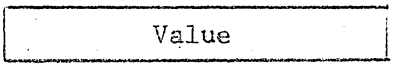
Description: These items are variable length up to 1Ø23 full words of binary code with a 1-word header. These items are marked, but not marked from, except that counts are made in PRS(B) according to BPS references. The space is reclaimed by moving down. New items are added contiguously from the high core end (BPO) toward low core.



G is a single bit used by the Garbage Collector to mark the Code.
Args is the number of arguments, and may be Ø, 1, 2, ... , 15, 16, 17, 18, or 2Ø.
Length is the length of the item in computer words, including the header.
Prs is a LISP 16-bit pointer to the PRS(A) item that points back to this BPS item.

13. PUSH DOWN STACK SPACE

Lower Boundary: PDO Upper boundary: PDE. Only pointer is register PDP.
Description: Items are 1-word long. Space is marked from, not marked and never reclaimed. These items are not pointed to by LISP pointers. The current value of register PDP is 4ØØ hex less than the absolute core address of the item pointed to. PDP is pushed (increased) and poped (decreased) by the function link routines. There are three types of PRS words as described below.



Type A. : for lambda and block variables.

Type B. : for rebound Special and Unspecial variables in PRS

Type C. : the function return pointer

Value is a LISP 24-bit pointer to any kind of data.
Prsdata and prsfn are 16-bit byte addresses of PRS items relative to PRSO.
Prsdata points to Special and Unspecial variables while Prsfn points to functions or Macros. Bvalue is a LISP 16-bit pointer to any kind of data.
Loc is a byte address relative to the beginning of the code pointed to by Prsfn and less than 4Ø96. Loc plus the contents of prsfn gives the exact return address for a function return.

14. FIXED SPACE or MARK SPACE

Lower Boundary: FXO Upper boundary: FXE
Description: This space contains 2 core pages or 65,536 (10000 hex) bits used by the Garbage Collector to mark all data (pointers) that are currently in use by the system at the time the Garbage Collector is called. Each bit in the space corresponds to a single LISP pointer.

15. INPUT/OUTPUT SPACE

Lower Boundary: IOO Upper Boundary: IOE.
Description: I/O Space was originally intended to be used to contain buffers for I/O files on the ADEPT-50 Time Sharing System. This is no longer used and the space is therefore available for LAP programmers for scratch, buffers, and otherwise useful space.

I/O Space was extensively used by the CONVERSE system for internal generation, maintenance and I/O of CONVERSE arrays, dictionary buckets, signatures, and lexicon tables; and for the automatic communication between two separate LISP programs.

16. SMALL INTEGERS

Quantum Number: 11
Description: This is a virtual space "pointed to" by LISP pointers greater than 37FFC hex. These pointers represent integers according to the formula

$$n = (p - 30000) / 4,$$

where n is the integer value "pointed to" and p is the pointer. The following table gives the limits of Small Integers, and some common values.

n (decimal)	p (hex)
-4Ø96	38ØØØ
-1	3BFFC
Ø	3CØØØ
1	3CØØ4
32	3CØ8Ø
1ØØ	3C19Ø
4Ø95	3FFFC

The pointer 3CØØØ (Ø) is the value of Entry ZERN. The pointer 3FFFC (4Ø95) is the highest possible LISP pointer. The pointer 38ØØØ is the smallest possible Small Integer pointer, and therefore, the pointer 37FFC is the largest pointer into real (non-virtual) space. 37FFC is the value of Entry TOPA.