February, 1989

Movement of Guaranteed Type Numbers, Addition of hashing MISCN subops.

Summary

The type numbers of several Lisp datatypes were moved down into the range that is "known to the microcode," to allow me to write C support for hashing.

Overview

Certain lisp type numbers must be known to the underlying implementation (microcode on 1186’s, the C emulator on Suns). One obvious example is that the emulator must be able to detect SMALLPs, so it can do arithmetic quickly.

There is a range of type numbers that are allocated very early in the load-up process, so that they are assigned known numbers. After those "well-known" types, the type numbers for hunked storage are allocated. After that, type numbers are allocated to types as the loadup progresses, in whatever order they are defined.

The New Requirement

I implemented MISCN sub-opcodes for CL:SXHASH, CL::EQLHASHBITSFN, IL:STRINGHASHBITS, and IL:STRING-EQUAL-HASHBITS. CL:SXHASH has special-case code for several data types that were not in the "well-known" range: RATIONAL, COMPLEX, PATHNAME, and BIGNUM. I needed to move the type numbers for those types down.

The New "Well-Known" Type Numbers

The well-known type numbers are defined from the list \BUILD-IN-SYSTEM-TYPES, which is defined in the file LLDATATYPE. Listed below are the old and new type number assignments:

Pre-existing	Newly-Added
1 SMALLP	
2 FIXP	
3 FLOATP	
4 LITATOM	
5 LISTP	
6 ARRAYP	
7 STRINGP	
8 STACKP	
9 CHARACTER	
10 VMEMPAGEP	
11 STREAM	
12 BITMAP	
13 COMPILED-CLOSURE	
14 ONED-ARRAY	
15 TWOD-ARRAY	
16 GENERAL-ARRAY	
	17 BIGNUM
	18 RATIO
	19 COMPLEX
	20 PATHNAME

Changes to the C emulator

Moving those four types into the well-known range had the effect of moving the array hunk type numbers up. The C coded garbage collector had several hunk type numbers hard-coded into it (as hex constants!!). I added definitions for all the type numbers in use (both well-known and hunk) to LISPTYPES.H, and changed the GC code (in GCRECLAIMCELL.C, as I recall) to accomodate it.

The new hashing MISCN sub-opcodes

The source for the new opcodes is in sxhash.c. The document {Eris}<Lispcore>Internal>Doc>Opcodes.TEdit has been updated to reflect their addition. I have also reserved opcodes for CL:VALUES and CL:VALUES-LIST.

The Lisp definitions for the functions CL:SXHASH, CL::EQLHASHBITSFN, IL:STRINGHASHBITS, AND IL:STRING-EQUAL-HASHBITS are now written to use the MISCN sub-opcode appropriate; the UFNs are defined in the same files as the original functions, CMLHASH and LLARRAYELT.