## For Immediate Action:

-----

The following opcodes are not emitted, have no microcode, and can be immediately recycled: (Someone should volunteer to do this and announce to xclispcore)

Opcode#(octal)	Name	
007	CDDR	;; unwind
036	PUTHASH	;; findkey
041	BOUT	;; still seems to be emitted
043	LIST1	;; restlist
044	DOCOLLECT	;; there is apparently a holdout that emits this.
045	ENDCOLLECT	;; ditto.
177	AUDIO	;; retcall
374	RESERVED for	Dolphin

The following opcodes are emitted, but have no microcode -- and may be recycled after recompilation of all sources. (Someone should volunteer to remove all optimizers -- Macros, Dopvals, Dopcodes, etc. -- associated with these opcodes and announce to xclispcore)

Opcode#(octal)	Name	
033		GETPROP (?)
035		GETHASH
050		ELT
051		NTHCHC
052		SETA
053		RPLCHARCODE
055		EVALV
160		ATOMNUMBER
313		GETBASEFIXP
314		PUTBASEFIXP

Recycling these opcodes demands recompilation, but this might be an additional reason for declaring .DCOM files not readable in Lute.

## For Discussion and Design:

\_\_\_\_\_

## Uncoordinated changes:

(that is, changes which do not alternate the meaning of existing constructs)

(Order of tasks is random -- not priority order)

Approx. time	Task	
Day		Add opcode for Read-Char (like NTHCHC, BIN) convert (byte/word) to (Characterp/Smallp)
Day		Add opcode for = Compare two numbers for equality does coercions
Day		Add opcode for ASH (arithmetic shift) Arithmeticly shift integer X bits to left if X pos. or X bits to right if X neg.
Day		Port LISTGET to 1186 8K microcode
Day		CL versions for ASSOC, FMEMB, EQUAL, (MEMBER) Differ from IL versions in that EQL is used rather than EQ microcode may only do EQ test on Symbols and punt otherwise
Day		Debug EQL, ARGO $\operatorname{}$ insure correct and efficient algorithm
Week		Complete port of unboxed scalar F.P. opcodes to 1186 (includes basic arith., and comparison)
Week		Microcode more cases for RECLAIMCELL RECLAIMCELL currently only reclaims Cons's should be extended to FIXP and Floatp boxes.

Microcode additional CL arith. operators Month

(Truncate and friends)

Spec. here not yet clear -- will appear after

benchmarking

? CL: EVAL (like EVAL but for commonlisp)

? REF

Finish array microcode Month

Coordinated changes:

(that is, changes which require changes both to lisp and microcode)

(Order of tasks is random -- not priority order)

Approx. time Task

Week CREATECELL countdown

Modify GC so reclaim occurs after

countdown to zero from some settable start - Important for tighter control over GC -

Month

GC hash algorithm changes Address issues of GC hash table overflow - Important for Cons'y benchmarks -

Subtyping in TYPEP Make TYPEP more useful May render TYPEMASK.N redundant Week

Week

Replace DTEST and TYPECHECK to be more TYPEP like (two type bytes rather than one?)

Month +

FN call changes for &optional, &key, &rest, closures and arg # checking, field descriptors, multiple values, descriminators (Larry, Bill and Pavel have volunteered to work

on this)