

Index

- 2005 (Puzzle Benchmark), 208.
- =, 21.
- RESET (MacLisp), 11.
- 8AREF (InterLisp), 221.
- 8ASET (InterLisp), 221.
- 16AREF (InterLisp), 209.
- 16ASET (InterLisp), 209.
- A register (MacLisp), 32.
- A-list lambda-binding, 6.
- abstract data structure, 156.
- Accent (SPICE Operating System), 61.
- address space, 1.
- analysis, 1.
- AOS/V5 (Data General Operating System), 76.
- Apollo, 66.
- Apollo Dn160, 70.
- Apollo DN300, 70.
- Apollo DN600, 70.
- APPEND, 18.
- APPLY, 12, 177.
- AR1 register (MacLisp), 32.
- AR2A register (MacLisp), 32.
- areas, 36.
- arguments in registers, 8.
- arithmetic, 16, 57.
- arrays, 12, 13, 31, 48, 52, 59, 64, 71, 74, 78, 195, 206, 218.
- array headers, 31.
- array references, 195, 206, 218.
- assembly language, 31.
- ASSOC, 18.
- atoms, 5.
- backtracking, 207.
- Baskett, 206.
- benchmarking, 1, 2, 81, 23.
- benchmarks, 1, 2, 81.
- Berkeley, 51.
- BIBOP, 15, 31, 51, 64, 77.
- Big Bag Of Pages, 15, 31, 51, 64, 77.
- bigfloat, 17.
- bignums, 17, 36, 43, 52, 55, 57, 64, 245.
- binding, 5, 6, 38, 93.
- binding stack, 38.
- black art (Benchmarking), 23.
- block compilation, 9, 11.
- block-packing puzzle, 206.
- Bob Boyer, 116.
- bottleneck registers (RTA, RTB, S-1), 46.
- boxed numbers, 9, 16, 23, 33.
- Boyer (Benchmark), 45, 139.
- Boyer (Human), 116.
- Browse (Benchmark), 136.
- buckets, 22.
- buddy-block system, 31.
- bus size, 4.
- cache memory, 2, 3, 4, 6, 25, 40, 43, 47, 61, 75, 110, 140.
- CADR (Lisp Machine), 36, 40, 42.
- CALLS instruction (Vax), 10.
- CAR, 12, 13.
- Carnegie-Mellon University, 58.
- CATCH (Lisp primitive), 9, 44, 60, 79, 99.
- CDR, 12.
- CDR-coding, 14, 37, 39, 43, 73.
- CHAOSNET, 22.
- circular lists, 156.
- circular queue, 3.
- clock rate, 3.
- closed-coding, 17.
- closed-compiling, 17.
- closing a file, 227.
- closures, 7, 79.
- CMACROs (PSL), 67.
- Common Lisp, 8, 19, 54, 58.
- common-subexpression elimination, 23.
- compilation, 44.
- compiler, 8, 20, 22.
- complex numbers, 49.
- computer algebra, 66.
- CONS, 12.
- constant-folding, 23.
- constants, 5.
- Conway, 206.
- contagion code, 57.

- context-switching, 7.
- Cray-1, 42, 66, 69.
- Cray-XMP, 69.
- cross optimizations, 23.
- Ctak (Benchmark), 99.
- CTSS (Cray Operating System), 69.
- D-LAST (CADR Microcode), 35.
- Dandelion (Xerox), 75.
- DandTiger (Xerox), 75.
- data base, 139.
- data-driven derivative, 175, 181.
- Data General Common Lisp, 76.
- Data General MV Architecture Computers, 76.
- data paths, 3.
- data stack, 39.
- data structures, 12.
- DATATYPE (InterLisp), 12, 160.
- data types, 47, 51, 52, 54, 71, 73, 77.
- Dderiv (Benchmark), 91, 175.
- debuggably, 11.
- debugging, 1.
- DEC-20, 13, 31, 66, 69.
- deep binding, 5, 6, 20, 93.
- DEFSTRUCT, 12, 158.
- DEFSUBST, 40.
- DEFVST, 12.
- deleting a file, 227.
- depth-first search, 207.
- Deriv (Benchmark), 170.
- derivative, 170.
- Destructive (Benchmark), 146.
- destructive list operations, 246.
- DG-UX (Data General Operating System), 76.
- directed graph, 156.
- disk service, 2.
- Div2 (Benchmark), 186.
- division by 2, 186.
- Dolphin (Xerox), 74.
- Dorado (Xerox), 75.
- DRECONC, 18.
- dynamic, 5.
- dynamic binding, 93.
- dynamic variables, 6.
- EBOX, 25, 111.
- ECL logic, 65.
- ELISP, 66.
- EQ, 21.
- EQUAL, 21.
- error correction, 4, 37.
- Ethernet, 22.
- expert system, 139.
- EXTEND, 12.
- FASL (File Format), 67.
- fast fourier transform, 193.
- fast links, 11.
- Fateman, 51, 245.
- Fdderiv (Benchmark), 181.
- FFT (Benchmark), 40, 193.
- file input, 232.
- file management, 20, 21.
- file output, 227.
- file print, 227.
- file read, 232.
- file system, 20, 21.
- firmware, 18.
- fixed-point arrays, 13.
- fixnums, 35, 55, 58, 64, 78, 148.
- FIXSW, 148.
- FLAVORS, 12, 38.
- FLENGTH, 149.
- flexibility, 1.
- floating point arrays 13.
- floating point numbers, 40, 57, 58, 64, 71, 78, 195, 245.
- flonums (see Floating Point Numbers), 35.
- FLOOR, 148.
- FLPDL (MacLisp), 33.
- fluid, 5.
- Forest Baskett, 206.
- FORTTRAN-like programming style, 195.
- forwarding pointer, 14.
- Fprint (Benchmark), 227.
- frames, 139.
- Franz Lisp, 2, 11, 51.
- Fread (Benchmark), 232.
- free, 5.
- free/special lookup, 7.
- free variables, 6.
- FRPLACA, 13, 149.
- Frpoly (Benchmark), 40, 240.
- FUNCALL, 12, 176.
- function calls, 6, 8, 35, 39, 43, 50, 51, 55, 59, 63, 72, 78, 82.
- funny quotes, 92, 98, 104, 109, 115, 135, 145, 152, 169, 174, 180, 185, 192, 202, 216, 226, 231, 235, 239, 274.
- FXPDL, 33.

- G-vectors (SPICE), 59.
- garbage collection, 6, 13, 20, 28, 30, 57, 63, 67, 74, 78.
- generic arithmetic, 15, 17.
- GENSYM, 140.
- GET, 176.
- global, 5.
- global variables, 20.
- Griss, 66.

- hardware, 2.
- hardware considerations, 3.
- hardware tagging, 36.
- hash table, 21.
- hayseed hackers, 11.
- heap, 10.
- Hewlett-Packard, 66.
- history of the address space, 29.
- HP9000, Series 200, 69.
- HP-9836, 66, 69.
- HRRZ, 13.
- hunks (MacLisp), 12, 31.

- I-vector (SPICE), 59.
- IBM 370, 66, 40, 70.
- IBM 3081, 42, 70.
- IEEE proposed standard floating-point, 46.
- IEEE single precision specifications, 40.
- IFU (Symbolics), 40.
- Ikuo Takeuchi, 81.
- implementation strategies, 5.
- implementation tactics, 5.
- inner loops, 28, 30.
- instruction-counting methodology, 3.
- instruction fetch, 3.
- instruction pre-fetch unit (Symbolics), 40.
- InterLisp, 2, 9, 11, 12, 13, 15, 17, 19, 73, 160, 196, 221.
- InterLisp-10, 2, 11, 13, 17.
- InterLisp-D, 2, 15, 73.
- InterLisp-Vax, 2.
- interpreter, 7, 11, 20.
- iteration versus recursion, 186.

- J Moore, 116.
- John Conway, 206.
- John McCarthy, 81.
- JonL White, 31.
- JSB, 51.
- jumping (Triang Benchmark), 218.

- KA-10 (DEC), 6, 9.
- KL-10 (DEC), 6, 9.

- lambda binding (see Binding), 5, 20.
- lambda-list, 56.
- Lambda machine (LMI), 42.
- LAP (Lisp Assembly Program), 67.
- least recently used (LRU), 36.
- levels of Lisp system architecture, 2.
- lexical, 5, 6, 7.
- lexical binding, 6.
- lexical contour, 7.
- lexical variables, 6.
- Lisp-in-Lisp, 46, 58, 66.
- Lisp compiler, 8, 20, 22.
- Lisp instruction level, 5.
- Lisp machines, 10, 12, 15, 34, 35, 36.
- Lisp machine hardware, 35, 36.
- Lisp Machine Inc, 34, 42.
- Lisp operation level, 18.
- list utilities, 147.
- LM-2 (Lisp machine), 36.
- LMI, 34, 42.
- load average, 2, 30.
- loading compiled Lisp code, 32.
- local function call, 51.
- local variable, 5.
- locality, 3, 25.
- locative, 37.
- loop-unwinding, 23.
- LRU (Least Recently Used), 36.

- MacLisp, 10, 11, 12, 15, 16, 19, 31, 50, 148, 181.
- MACRO32 (Assembly language), 57.
- MACSYMA, 28, 51.
- MACSYMA-like system, 28.
- MAPCAR, 18.
- Martin Griss, 66.
- MBOX, 111.
- MC68000, 25, 66, 69.
- McCarthy, 81.
- MDL (Language), 15.
- memory bandwidth, 4.
- methodology, 2.
- microcode, 3, 15, 18, 20, 58, 73.
- microcoded machines, 18.
- microcompiler (LMI), 44.
- micro-micro function call, 44.
- mircoSTACK (LMI), 44.
- MIT Artificial Intelligence Laboratory, 34.
- MIT CADR, 34.
- monitor calls, 32.
- Moore, 116.
- Multics-like ring protection, 46.
- multiple values, 9, 45, 60.

- multi-processing Lisp, 7.
- MV4000 (Data General), 80.
- MV8000 (Data General), 80.
- MV10000 (Data General), 80.
- MV memory structure (Data General), 76.
- MV-UX (Data General Operating System), 76.
- natural benchmarks, 2.
- NCONC, 18.
- New Implementation of Lisp, 54.
- NIL, 15, 54.
- NOUVO (MacLisp), 11.
- NuBus, 42.
- NULL, 15.
- number-CONSing, 16, 23, 33, 49, 52, 55.
- number-format sizes, 17.
- obarray, 21.
- object oriented programming, 12.
- oblist, 21.
- one-dimensional arrays, 218.
- open-coding, 17, 22.
- open-compiling 17, 22.
- opening a file, 227, 232.
- operand fetch and decode, 6.
- &optional, 56, 63.
- order of evaluation, 23.
- P0 space (Vax), 55.
- P1 space (Vax), 55.
- page boundaries, 30.
- paging, 2, 14, 30, 34, 36, 38, 42, 61, 70, 73, 140.
- PASCAL, 5.
- pattern matching, 139.
- PDL buffer, 3.
- PDL numbers, 10, 16, 27, 33, 50.
- PDP-6, 31.
- PDP-10 MacLisp, 2.
- PDP-10, 9, 16, 31.
- peephole optimization, 23.
- performance, 1.
- performance evaluation defined, 1.
- performance profile, 23.
- PERQ Systems Corporation, 58.
- PERQ T2, 62.
- personal machines, 29.
- pipeline, 4, 44, 47, 50, 65.
- polynomial manipulation, 240.
- Portable Standard Lisp (PSL), 8, 16, 63, 66, 67, 68, 69.
- position-independent code, 56.
- powers of polynomials, 245.
- pre-fetch units, 4.
- Prettyprint, 21.
- PRINT, 21, 227, 236.
- programming style, 23, 245.
- property lists, 139, 246.
- PSL (Portable Standard Lisp), 8, 16, 63, 66, 67, 68, 69.
- PSL compiler, 67.
- PSL versions, 68.
- PSL-HP200, 68.
- PSL-HP-UX (Hewlit-Packard Operating System), 69.
- Puzzle (Benchmark), 59, 203, 218.
- QUOTIENT, 148.
- random graph, 156.
- random numbers, 140.
- random number generator, 140, 156.
- range checking of arrays, 13.
- raw-data tables, 81.
- READ, 21, 232.
- real-benchmark methodology, 3.
- recursion versus iteration, 186.
- reference count (Garbage Collection), 14, 74.
- register A (MacLisp), 32.
- register allocation, 6, 9, 23, 39.
- register AR1 (MacLisp), 32.
- register AR2A (MacLisp), 32.
- register B (MacLisp), 32.
- register C (MacLisp), 32.
- register optimization, 23.
- registers, 6.
- &rest, 44, 55, 56.
- &rest arguments, 39, 44, 55, 56.
- RETFROM (InterLisp), 9, 100.
- REVERSE, 18.
- Richard Fateman, 51, 245.
- rings, 76.
- Rlisp (PSL), 68.
- roots of Common Lisp, 33.
- RPLACA, 12, 13, 147.
- RPLACD, 12, 140.
- RTA (S-1), 46.
- RTB (S-1), 46.
- runtime typing, 14, 17.
- runtime type checking, 14, 17.
- S-1 Lisp, 2, 11, 15, 16, 46, 93.
- S-1 Mark IIA, 46.
- SAIL (Operating System), 25, 29, 33, 110.
- search problem, 206.
- segmented memory, 76
- SEUS (Language), 10.

- shallow binding, 5, 20, 93.
- small-benchmark methodology, 3.
- small-number-CONS, 27.
- small-number scheme, 16.
- Smalltalk (Language), 12, 15.
- spaghetti stack (InterLisp), 74, 99, 101.
- special, 5, 6, 10, 12, 20, 93.
- special variables, 6, 20.
- special binding, 10, 12, 83.
- SPICE Lisp byte codes, 63.
- SPICE Lisp (CMU), 58, 63, 76.
- stack-allocated number, 16.
- stack architecture, 59.
- stack buffer (Symbolics), 3, 38, 43.
- stack cache, 3, 38, 43.
- stack group (ZetaLisp), 38.
- stacks, 38.
- stack vectors, 55.
- Stak (Benchmark), 93.
- Stanford Artificial Intelligence Laboratory (SAIL), 33.
- Standard Lisp, 66.
- stock hardware, 15.
- stop-and-copy (Garbage Collection), 57, 78.
- strings, 12, 64, 74.
- SUBRCALL, (MacLisp), 182.
- SUN II, 70.
- symbolic derivative, 171.
- Symbolics 3600, 3, 36, 39, 40, 44.
- Symbolics 3600 compiler, 39.
- Symbolics 3600 data formats, 40.
- Symbolics Inc, 3, 34, 36, 39, 40, 44.
- Symbolics LM-2, 34.
- symbols, 5, 6.
- Syslisp (PSL), 66.
- systemic quantities vector, 50.
- tagged architecture, 14, 58.
- tagging, 54, 58, 64.
- tags, 14.
- tail recursion, 9, 26, 72, 111, 186.
- tail recursion removal, 186.
- Tak (Benchmark), 12, 24, 30, 42, 61, 81, 93, 99, 105.
- Takeuchi, 81.
- Takl (Benchmark), 105.
- Takr (Benchmark), 61, 110.
- TENEX (Operating System), 29.
- terminal output, 236.
- terminal print, 236.
- theorem-proving benchmark, 129.
- three operand instructions, 46.
- THROW (Lisp primitive), 9, 44, 60, 79, 99.
- time-shared machine, 29.
- top-of-stack register, 38.
- Tprint (Benchmark), 62, 236.
- trampolines, 11.
- TRANSLINK (Franz), 11, 32, 33, 51.
- Traverse (Benchmark), 153.
- Triang (benchmark), 217.
- two-dimensional arrays, 218.
- type checking facility, 15.
- types, 31, 35, 37.
- unboxing, 16.
- units, 139.
- University of California at Berkeley, 51.
- University of Utah, 66.
- unpleasant programming style, 245.
- unsnapping links, 33.
- unwinding, 9.
- user-controlled paging, 36.
- Utah, 66.
- UUO, 32.
- UUO links, 11.
- UUOlinks, 33, 51.
- value cell, 6.
- value predication, 50.
- variable lookup, 20.
- variables, 5, 6.
- Vax 11/750, 69.
- Vax 11/750 Common Lisp, 94.
- Vax 11/780, 25, 69.
- Vax 8600, 65.
- Vax Common Lisp (DEC), 63.
- Vax (DEC), 10, 25, 51, 54, 63, 65, 69, 79, 94.
- Vax NIL, 2, 11, 15, 94.
- Vax Portable Standard Lisp (PSL), 16.
- vectors, 12, 48, 52, 59, 64, 71, 74, 78.
- Venus (Dec Vax 8600), 65.
- VMS operating system (DEC), 63.
- WAITS (Operating System), 33.
- White, 31.
- wholine time, 25.
- windows, 236.
- word alignment, 4.
- working-set, 2, 14, 30.
- write-through, 4.
- XCT instruction, 32.
- Xerox, 73.
- ZEROP, 21.
- ZetaLisp, 34.