The information in this document is subject to change without notice and should not be construed as a commitment by Thinking Machines Corporation. Thinking Machines Corporation reserves the right to make changes to any products described herein to improve functioning or design. Although the information in this document has been reviewed and is believed to be reliable, Thinking Machines Corporation does not assume responsibility or liability for any errors that may appear in this document. Thinking Machines Corporation does not assume any liability arising from the application or use of any information or product described herein.

Connection Machine is a registered trademark of Thinking Machines Corporation. CM-1, CM-2, CM, and DataVault are trademarks of Thinking Machines Corporation. Paris, *Lisp, C*, and CM Fortran are trademarks of Thinking Machines Corporation. VAX, ULTRIX, and VAXBI are trademarks of Digital Equipment Corporation. Symbolics, Symbolics 3600, and Genera are trademarks of Symbolics, Inc. Sun and Sun-4 are trademarks of Sun Microsystems, Inc. UNIX is a trademark of AT&T Bell Laboratories.

Copyright © 1988 by Thinking Machines Corporation. All rights reserved.

Thinking Machines Corporation
245 First Street
Cambridge, Massachusetts 02142-1214
(617) 876-1111
Master Index

This master index combines references from each of the three *Lisp documents included in Programming in *Lisp. Letters preceding the page numbers indicate the document in which the entry may be found.

CG indicates *Lisp Compiler Guide.

!!, RM 28; RS 22, 49; CG 16, 59
+ !!, RM 28; RS 4; CG 21
− !!, RM 28; RS 4; CG 21
* !!, RM 28; RS 4; CG 21
/ !!, RM 28; RS 4; CG 21
/= !!, RM 25
= !!, RM 24; RS 4
> !!, RM 25
> = !!, RM 25
1 + !!, RM 28; RS 4
1− !!, RM 28; RS 4

alias!!, RS 32, 39, 48
aliasing, RS 32
*all, RM 19; RS 109
allocate!!, RM 11, 12
allocate-processors-for-vp-set, RS 57, 64
allocated-pvar-p, RS 106
alpha-char-p!!, RS 14
alphanumericp!!, RS 15
*and, RM 49; CG 5
and!!, RM 26
*apply, RM 33
aref!!, RS 30, 34, 106
array pvars, RS 19, 131, 140
*array-dimensions, RS 28
array-dimensions!!, RS 29
*array-dimension, RS 28
array-dimension!!, RS 28
*array-dimension-limit, RS 21
array-dimensions!!, RS 29
*array-element-type, RS 28
array-in-bounds-p!!, RS 29
array pvars, CG 12
*array-rank, RS 28
array-rank!!, RS 28
*array-rank-limit, RS 20
array-major-index!!, RS 29
array-to-pvar, RM 44
array-to-pvar-grid, RM 45
*array-total-size, RS 29
array-total-size!!, RS 29
CG: *Lisp Compiler Guide.

*array-total-size-limit, RS 21
ash!!!, RM 29
asin!!!, RS 4
asinh!!, RS 4
atan!!!, RS 4
atanh!!!, RS 4

B
backward routing, RS 170
*before-*cold-boot-initializations*, RM 55
*before-*warm-boot-initializations*, RM 55
bit-and!!!, RS 34
bit-andc1!!!, RS 35
bit-andc2!!!, RS 35
bit-eqv!!!, RS 34
bit-ior!!!, RS 34
bit-nand!!!, RS 35
bit-nor!!!, RS 35
bit-not!!!, RS 36
bit-orc1!!!, RS 35
bit-orc2!!!, RS 35
bit-xor!!!, RS 34
boole!!!, RS 116
boolean pvars, RS 130, 135; CG 12
booleanp!!!, RS 110
both–case-p!!, RS 14
byte specifier, RS 119
byte!!!, RS 119
byte–position!!!, RS 120
byte–size!!!, RS 119

C
CM–2. CG 64
CSS, RS 69, 145

ceiling!!!, RM 29; CG 21
char, RS 16
char–bit!!!, RS 17
char–bits!!!, RS 10
*char–bits–length, RS 8
*char–bits–limit, RS 8
char–code!!!, RS 10
*char–code–length, RS 8
*char–code–limit, RS 8
char–downcase!!!, RS 12
char–equal!!!, RS 17, 110
char–flipcase!!!, RS 12
char–font!!!, RS 10
*char–font–length, RS 8
*char–font–limit, RS 8
char–greaterp!!!, RS 17
char–int!!!, RS 12
char–lessp!!!, RS 17
char–not–equal!!!, RS 17
char–not–greaterp!!!, RS 17
char–not–lessp!!!, RS 17
char–upcase!!!, RS 12
char/ = !!, RS 16
char = !!, RS 16
char > !!, RS 16
char > = !!, RS 16
character–pvar, RS 131
character pvars, RS 7–18, 112, 131, 136;
CG 12
character!!!, RS 11, 112
*character–length, RS 9
characterp!!!, RS 13
cis!!!, RS 4
coerce!!!, RS 111
*cold-boot, RM 10, 53, 61; RS 56, 78
combining routing, RS 170
communication
inter–VP set, RS 87–97
communication (continued)
    inter-VP set operations, RS 91-97
    interprocessor, RM 9; RS 77-104
    interprocessor examples, RS 95
    near neighbor, RS 68
    router, RS 68
    compare!!, RS 108
    *compilation-speed*, CG 40
    *compilep*, CG 6, 27
    compiler-let, CG 25, 64
    compiler options, CG 27, 28, 29, 30, 31,
        32, 35, 36, 37, 39, 40, 41, 42
    menu, CG 22
    safety, CG 18-24, 55
    setting values of, CG 22-26
    *compiling*, CG 6
    complex canonicalization, RS 3
    complex contagion, RS 3
    complex pvars, RS 1-6, 131, 139; CG 12
    complex-pvar, RS 131
    complex!!, RS 2, 112
    complexp!!, RS 2
    *cond, RM 20; CG 5
    cond!!?, RM 32
    conjugate!!, RS 4
    *constant-fold*, CG 39
    copy!!, RS 38, 50, 86
    copy-seq!!, RS 155, 157
    cos!!, RM 30; RS 4
    cosh!!, RS 4
    count!!, RS 155, 164
    count-if!!, RS 164
    count-if-not!!, RS 164
    create-geometry, RS 58, 67
    create-segment-set!!, RS 145, 147
    create-vp-set, RS 56-58, 64
    cross-product, RS 154
    cross-product!!, RS 150

    cube address, RM 4, 62
    cube-from-grid-address, RM 50; RS 80
    cube-from-grid-address!!, RM 51; RS 81
    cube-from-vp-grid-address, RS 87, 98
    cube-from-vp-grid-address!!, RS 88, 98
    *current-cm-configuration*, RM 56;
        RS 59
    *current-send-address-length*, RS 59
    *current-grid-address-lengths*, RS 60
    *current-vp-set*, RS 59
    currently selected set, RM 5, 19;
        RS 69, 145

D
    *deallocate, RM 12
    *deallocate-’defvars, RM 12
    deallocate-vp-set, RS 66
    debugging tools, RM 33
    declare, RM 16; RS 107; CG 14, 16, 26,
        51, 53, 56, 57
    declare statement, RM 16
    def-vp-set, RS 56-58, 62
    *default-vp-set*, RS 58
    defined-float pvars, RS 130, 138; CG 12
    defining *Lisp functions, RM 9
    *defstruct, RS 23, 33, 39-54, 106
    deftype, CG 12
    *defun, RM 9, 15, 33, 34; RS 107, 109;
        CG 15, 26, 49, 52, 56, 57
    defun, CG 51, 52
    *defvar, RM 11; RS 57, 71; CG 56
    delete-initialization, RM 55
    deposit-byte!!, RM 31
    describe-pvar, RS 105
    describe-vp-set, RS 73
    digit-char!!, RS 12
    digit-char-p!!, RS 15
    dimension-address-length, RS 60
<table>
<thead>
<tr>
<th>Function</th>
<th>RM</th>
<th>RS</th>
<th>CG</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimension-size</td>
<td>56</td>
<td>80</td>
<td></td>
</tr>
<tr>
<td>do</td>
<td>56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>do-for-selected-processors</td>
<td>20</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dot-product</td>
<td>154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dot-product!!</td>
<td>150</td>
<td></td>
<td></td>
</tr>
<tr>
<td>double-complex-pvar</td>
<td>112,131</td>
<td></td>
<td></td>
</tr>
<tr>
<td>double-float pvar</td>
<td>16; 112, 131</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dph!!</td>
<td>120</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-cross-product!!</td>
<td>153</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-v+!!</td>
<td>152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-v+--constant!!</td>
<td>152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-v-!!</td>
<td>152</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-vector-normal!!</td>
<td>153</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-vsca!!</td>
<td>153</td>
<td></td>
<td></td>
</tr>
<tr>
<td>dsf-vsca-to-unit-vector!!</td>
<td>153</td>
<td></td>
<td></td>
</tr>
<tr>
<td>enumerate!!</td>
<td>32</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eq!!</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>eql!!</td>
<td>24; 110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>equalp!!</td>
<td>110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>evenp!!</td>
<td>23</td>
<td></td>
<td></td>
</tr>
<tr>
<td>every!!</td>
<td>155, 158</td>
<td></td>
<td></td>
</tr>
<tr>
<td>exp!!</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expt!!</td>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extended-float</td>
<td>131</td>
<td></td>
<td></td>
</tr>
<tr>
<td>field-pvar</td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fill</td>
<td>155, 159</td>
<td></td>
<td></td>
</tr>
<tr>
<td>find!!</td>
<td>155, 162</td>
<td></td>
<td></td>
</tr>
<tr>
<td>find-if!!</td>
<td>162</td>
<td></td>
<td></td>
</tr>
<tr>
<td>find-if-not!!</td>
<td>162</td>
<td></td>
<td></td>
</tr>
<tr>
<td>flet</td>
<td>108; 17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float!!</td>
<td>30; 112; 22</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float-epsilon!!</td>
<td>115</td>
<td></td>
<td></td>
</tr>
<tr>
<td>floatp!!</td>
<td>24</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float-pvar</td>
<td>130</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float-sign!!</td>
<td>114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>floating-point accelerator</td>
<td>171</td>
<td></td>
<td></td>
</tr>
<tr>
<td>floating-point pvars</td>
<td>114</td>
<td></td>
<td></td>
</tr>
<tr>
<td>floor!!</td>
<td>29; 21</td>
<td></td>
<td></td>
</tr>
<tr>
<td>front-end computer, data transfer</td>
<td></td>
<td></td>
<td>RM 43</td>
</tr>
<tr>
<td>front-end pvars</td>
<td>122, 135; 122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>front-end!!</td>
<td>122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>front-end-p!!</td>
<td>122</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fround!!</td>
<td>113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ftruncate!!</td>
<td>113</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ftype</td>
<td>14, 56, 57</td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>funcall</em></td>
<td>33</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**G**

gcd!!                          | 118|     |      |
| general mutable pvars         | 5, 12| |      |
| general pvars                 | 122, 130, 132; 5, 12| |      |
| and type conversion           | 135|     |      |
| *generate-comments*           | 42|     |      |
| graphic-char-p!!              | 14|     |      |
| grey-code-from-integer!!      | 121|     |      |
| grid                           | 97, 98| |      |
| grid address                   | 4, 62; 60| |      |
grid!!, RS 97, 98
grid–from–cube-address, RM 50; RS 81
grid–from–cube-address!!, RM 51; RS 82
grid–from–vp–cube–address, RS 89
grid–from–vp–cube–address!!, RS 90
grid–relative!!, RS 98

H
help, RS 105

I
*if, RM 20; CG 5
if!!, RM 31
imagpart!!, RS 4
*immediate error if location*, CG 19
*immediate–error–if–location*, CG 41
*inconsistency–action*, CG 28
indirect addressing, RS 31, 33, 170
initialize–character, RS 9
int–char!!, RS 13, 112
*integer–length, CG 5
integer pvar, RS 112, 118
integer–from–grey–code!!, RS 121
integer–length!!, RS 117
integer–reverse!!, RS 109
integerp!!, RM 24
*interpreter–safety*, RS 61, 123–125
interprocessor, RM 37–51
interprocessor communication,
   RS 68, 77–104
irrational functions, and complex pvars,
   RS 4
isqrt!!, RM 29; CG 21

L
labels, RS 108; CG 17
ldb!!, RS 120
ldb–test!!, RS 120
least–negative–float!!, RS 114
least–positive–float!!, RS 114
length!!, RS 155, 157
*let, RM 13, 15; RS 23, 42, 107, 109;
   CG 5, 6, 15, 26, 49, 56, 57
let, CG 56
*let*, RM 13, 15; RS 23, 42, 107, 109;
   CG 5, 15, 26, 49, 56
let–vp–set, RS 65
list–of–active–processors, RM 35
load–byte!!, RM 30
*locally, RS 107, 108; CG 15, 16, 26, 52,
   53, 56, 60
*log–number–of–processors–limit*,
   RM 56
log!!!, RM 30; RS 4
*logand, RM 49; CG 5
logand!!, RM 27; RS 116
logandc!!!, RS 116
logandc2!!!, RS 116
logbitp!!, RS 117
logcount!!, RS 117
logeqv!!!, RM 27
logical operations, RM 26–27
*logior, RM 49; CG 5
logior!!!, RM 27; RS 116
lognot!!, RM 27; CG 21
logorc1!!!, RS 116
logorc2!!!, RS 116
logtest!!!, RS 117
*logxor, CG 5
logxor!!!, RM 27
long–complex–pvar, RS 131
long-float pvar, RM 16; RS 131
lower-case-p!!, RS 14

M
*machine-type*, CG 32
make-array!!, RS 21
make-char!!, RS 11
*map, RS 36
*max, RM 49; CG 5
max!!, RM 28
mask-field!!, RS 121
memory management. RM 58
*min, RM 49; CG 5
min!!, RM 28
*minimum-size-for-vp-set*, RS 59
minusp!!, RM 24
mod!!, RM 29; CG 21
most-negative-float!!?, RS 114
most-positive-float!!, RS 114
mutable general pvars, RS 133
mutable pvars, RS 132; CG 12, 61
multiple values, RM 61

N
N-D NEWS, RS 77—104
NEWS address, RS 68
near neighbor communication, RS 68
*news, RS 85
news!!, RM 10; RS 84
news-order, 68
nil!!, RM 7, 25
next-power-of-two–> =, RS 108
not!!, RM 26
notany!!, RS 155, 158
notevery!!, RS 155, 158
*nreverse, RS 155, 157
nsubstitute–if!!, RS 155, 161
nsubstitute–if–not!!, RS 161
null!!, RS 110
numberp!!, RM 24
*number–of–dimensions*, RM 56;
RS 59
*number–of–processors–limit*, RM 56;
RS 59
numberp!!, RS 4

O
odd!!, RM 23
off–grid–border–p!!, RM 83
off–grid–border–relative–p!!, RM 51;
RS 83
off–vp–grid–border–p!!, RS 90
*optimize, RS 107; CG 16, 25
optimize, RS 107; CG 16, 25
*optimize–bindings*, CG 30
*optimize–check–stack*, CG 42
*optimize–peephole*, CG 30
*or, RM 49; CG 5
or!!, RM 26

P
Paris, called from *Lisp, RM 57
phase!!, RS 4
plusp!!, RM 23
position!!, RS 155, 163
position–if!!, RS 163
position–if–not!!, RS 163
power–of–two–p, RS 108
ppp!!, RS 125
ppp–address–object, RS 126
*ppp–default–end*, RM 34
*ppp–default–format*, RM 34
CG: *Lisp Compiler Guide.

*ppp-default-mode*, RM 34
*ppp-default-per-line*, RM 34
*ppp-default-start*, RM 34
pppdbg!!, RS 125
predicate operations, RM 23
*pref!!, RM 10
pref, RM 6, 14; RS 37, 94, 98, 106; CG 5
pref!!, RM 37, 61; RS 33, 40, 91, 93, 98, 106, 170; CG 22
pref-grid, RM 14
pref-grid!!, RM 38, 61
pref-grid-relative!!, RM 39, 61
pretty-print-pvar-in-currently
selected-set, RM 35
processor selection, RM 7
processors, RM 4
   non-selected, RM 59
*proclaim, RM 15; RS 43; CG 13, 53, 56, 57
proclaim, CG 13
*psys, RM 40; RS 33, 91, 92, 170; CG 5, 21, 22
*psys-grid, RM 41
*psys-grid-relative, RM 42
*pull-out-subexpressions*, CG 31
pvar, RM 4, 6, 11-17
pvar *, CG 12
(pvar *), RS 133; CG 5
pvar t, CG 12
(pvar t), RS 132; CG 5
pvar-to-array, RM 44
pvar-to-array-grid, RM 44
pvar type declaration, RM 15-17
pvar types, RS 129-142
pvar-vp-set, RS 73
pvarp, RM 12

pvars, extent, RM 59

R
random!!, RM 29
rank!!, RM 32
realpart!!, RS 4
reduce, RS 145
reduce!!, RS 155, 159
reduce-and-spread!!, RS 86
rem!!, RS 110; CG 21
return-pvar-p, RS 109
reverse!!, RS 155, 158
rot!!, RM 30
round!!, RM 29; CG 21
router communication, RS 68
routing
   backward, RS 170
   combining, RS 170
   sprint, RS 169

S
*safety*, CG 29
scale-float!!, RS 113
scan!!, RS 45; RS 79, 87
scan-grid!!, RM 48
scanning, RS 68, 145
segment sets, RS 145
segment-set-scan!!, RS 145, 146
selection, of processors, RM 19-21
self!!, RS 100
self-address, RM 7
self-address!!, RM 50
self-address-grid!!, RM 50; RS 80
send address, RS 59, 68, 101, 145
send-order, RS 68
sequence pvar, RS 155