INDEX

Numbers

8-bit excess-127 exponent, 88
8-bit registers, 10
16-bit integer variables, 54
16-bit registers, 10
16-byte-aligned addresses, 606
32-bit integer variables, 54
32-bit registers, 10
32-byte alignment within a segment, 605
64-byte alignment within a segment, 605
64-byte memory alignment, 607
80x86 memory addressing modes, 105
96-bit `rcl` and `rcr` operations, 484
128-bit comparisons, 461
128-bit decimal output (conversion to string), 508
256-bit by 64-bit division, 468
8087 FPU, 317

Symbols

`%1` (batch file parameter), 34
`/c` MASM command line option, 9
`.code` section, 108
`.const` declaration section, 109
`.data` declaration section, 108
`.data?` declaration section, 110
`.data` directive, 14
`.err` CTL statement, 748
! escape operator (MASM macros), 750
#IA exception (invalid arithmetic operation), 673
.inc files (include files), 848
+infinity, 90
–infinity, 90
.lib files, 869
`$` operator, 154
`%` operator in the first column of a source line, 751
`%` operator (MASM macros), 750
– (unary negation, within a constant expression), 153
+ (within a constant expression), 153
[ ] (within a constant expression), 153
* (within a constant expression), 153
/ (within a constant expression), 153

A

ABI (application binary interface), 27, 261
ABI (Microsoft) register usage, 38
`abs` external symbol type, 851
absolute value (floating-point), 349
absolute value (SIMD), 659
access fields of a struct/record, 199
accessing
  an element of a single dimensional array, 182
  data on the stack, 142
  data via a pointer, 162
  elements of an array, 183
  elements of multidimensional arrays, 196
  elements of three- and four-dimensional arrays, 191
  fields of a struct/record via a pointer, 199
  fields of a union, 206
  local variables, 235
  record/struct fields, 199
  reference parameters, 256
  subfields of a nested structure, 200
  value parameters, 253
accumulated errors in a floating-point calculation, 315
activation record
  construction at runtime, 228
  definition, 228
`adc` instruction, 455, 716

adding 1 to a register or memory location, 149
add instruction, 21
addition (extended-precision), 454
addition (horizontal, packed), 650
addition (SIMD), 648
addition (vertical, packed), 649
addpd instruction, 669
addps instruction, 669
addresses, 9
address expressions, 130
addressing modes, 122
  indirect, 124
  indirect-plus-offset, 125
  register indirect, 124
  scaled-indexed, 126
  scaling factor, 126
address of an object, 22
addsd instruction, 371
addss instruction, 371
Advanced Vector Extensions (AVX), 596
aggregate data types, 174
AH register, 10
  copying AH to FLAGS register, 86, 350
AL/AX/EAX register usage in string instructions, 826
algorithm to convert a string to an integer, 546
aliases, 207
aliasing registers, 10, 623
align directive, 121
aligned data movement instructions (SSE/AVX), 610
aligning
  bit strings, 710
  data in a segment, 605
  data objects on the stack or heap, 607
  within a record, 204
alignment
  data alignment, 119
  variable alignment, 121
  within a record, 204
allocating storage for arrays, 194. *See also* arrays
allocating storage for uninitialized arrays, 183
AL register, 10
anatomy of a MASM program, 5
and instruction, 58, 309, 709
ANDN (and not) operation, 645
andnpd instruction, 645
AND operation, 55
AND operator, 153
andpd instruction, 645
anonymous
  unions, 208
  variables, 125
application binary interface (ABI), 27, 261
application programming interface (API), 35
arbitrary alignment within a segment, 605
arctangent, 361
arithmetic
  expressions, 299, 302
  idioms, 310
  logical systems, 310
  operators within a constant expression, 153
  shift right, 77
arithmetic shifts (SSE/AVX), 647
arrays, 191
  accessing elements of an array, 183
  accessing elements of multidimensional arrays, 196
  allocating storage for a multidimensional array, 194
  arrays of arrays, 192
  arrays of structs, 203
  base address, 182
  bubble sort, 188
  column-major ordering, 193
  declarations, 182
  definition, 181
  dup operator, 182
  four-dimensional array access (row major), 191
  indexing operator, 181
  initialized arrays, 183
  LARGEADDRESSAWARE, 183
  multidimensional, 189, 192
  row-major ordering, 190
  sorting, 185

three-dimensional array access (row major), 191
two or more dimensions, 189
uninitialized storage, 183
array variables, 182
ASCII
character set, 53, 93
codes for numeric digits, 95
groups, 94
assembly language procedures, xxviii, 22
assembly-time initialization of structures, 200
assigning, 299
constant to a variable, 299
one variable to another, 299
associativity, 302, 304
automatic allocation, 240
automatic code generation, 748
automatic (local) variables, 235
automatic variables, 234
in a procedure, 234
average computation (SIMD), 657
avoiding branches by using calculations, 409
AVX
aligned data movement instructions, 610
AVX-512 memory alignment, 607
AVX, AVX2, AVX-256, AVX-512, 596
AVX/SSE comparison synonyms, 673
extensions, 596
floating-point arithmetic (SIMD), 668
floating-point conversions, 679
instruction operands, 606
memory alignment requirements, 606
packed byte data types, 597
packed dword data types, 598
packed qword data types, 598
packed word data types, 597
programming model, 596
sign extension, 666
unaligned memory access, 606, 612
zero extension, 665
AX register, 10

B

backspace, 93
base address (of an array), 182
Base Pointer register (RBP), 230
Basic Multilingual Plane (Unicode BMP), 97
batch files, 33
BCD (binary coded decimal), 91
arithmetic, 486
numbers, 51
representation, 91, 487
BH register, 10
biased (excess) exponents, 88
big-endian data organization, 115
big-endian to little-endian conversion, 116
binary
data types, 51
digits, 44
formats, 45
numbering system, 43
point (binary fractions), 87
binary-coded decimal (BCD), 91
arithmetic, 487
numbers, 51
representation, 91
binary search, 422
bit, 45, 51
complement, 708
counting, 739
data, 707
fields, 79
inversion, 708
manipulation, 707, 708
mask, 708
offset, 708
packed data, 79
pattern search, 743
runs, 708
sets, 708
strings, 57, 708
arrays, 733
extraction, 742
merging, 741
reversal, 739
test for 1 bits, 714
bit-by-bit operations, 58
bit string alignment, 710

- bit string masking, 58
- bitwise operations, 58
- blank macro arguments, 767
- BL register, 10
- BMP (Unicode Basic Multilingual Plane), 97
- Boolean
  - evaluation
    - complete, 400
    - short-circuit, 401
  - expressions, 308
  - logical systems, 310
  - values, 51
- BP register, 10
- bracketing characters in macro parameters, 764
- branch out of range, 393
- branch-prediction hardware, 448
- `break` statement, 438
- `bsf` instruction, 737
- `bsr` instruction, 737
- `bswap` instruction, 116
- `btc` instruction, 715
- `bt` instruction, 715
- `btoStr` (byte to string) function, 493
- `btr` instruction, 715
- `bts`, `btc`, and `btr` instructions and CPU performance, 716
- `bts` instruction, 715
- bubble sort, 185
- busy bit (FPU), 324
- BX register, 10
- byte, 52
  - alignment in a segment, 605
  - data directive, 53
  - directive, 15
- byte-sized lanes, 598
- byte strings, 825
- byte vectors (packed bytes), 597

C

- C++ compiler, 4
- callee register preservation, 222
- caller register preservation, 222
- call indirect, 278
- calling assembly code from C/C++, 4
- calling C/C++ code from assembly, 4
- `call` instruction, 22, 216, 218
- carriage return, 93
- carry flag, 12, 294
  - `and`, `or`, and `xor` instruction effect, 712
  - as a bit accumulator, 716
  - setting after an arithmetic operation, 71
- case
  - labels (noncontiguous), 418
  - statement, 396, 410
- case-sensitive identifiers, 8
- `catstr` directive, 751
- `cbw` instruction, 288
- C/C++ Standard Library, 4
- `cd` command, 930
- `cdecl` calling convention, 262
- `cdqe` instruction, 288
- `cdq` instruction, 288
- central processing unit, 9
- change sign (floating-point), 349
- char
  - data type, 96
  - declaring characters in a MASM program, 96
- character
  - data type, 92
  - literal constants, 95
  - strings, 174
- `chdir` command, 930
- checking a bit to see if it is zero or one, 298
- checking to see if a macro argument is blank, 767
- checking whether a bit string contains all 1 bits, 714
- choosing an alignment value for variables, 121
- CH register, 10
- C integer types, 454
- class argument for segment directive, 605
- `clc` instruction, 86, 716
- `cld` instruction, 86
- clearing
  - bits, 708
  - clearing bits prior to comparing them, 709
  - FPU exception bits, 363

- CLI (command line interpreter), xxx
  - `cd` command, 930
  - `del` command, 932
- `cli` instruction, 86
- clipping (saturation), 68
- `closeHandle` function, 890
- CL register, 10
  - in rotate operations, 79
  - in `shl` instruction, 75
- `cls` command, 931
- `cmc` instruction, 86, 716
- *cmd.exe* (command line interpreter), xxx
- `cmovae` instruction, 395
- `cmova` instruction, 395
- `cmovbe` instruction, 395
- `cmovb` instruction, 395
- `cmovc` instruction, 394, 716
- `cmove` instruction, 395
- `cmovge` instruction, 395
- `cmovg` instruction, 395
- `cmovnp` instruction, 395
- `cmovpe` instruction, 395
- `cmovp` instruction, 395
- `cmovle` instruction, 395
- `cmovl` instruction, 395
- `cmovnae` instruction, 395
- `cmovna` instruction, 395
- `cmovnbe` instruction, 395
- `cmovnb` instruction, 395
- `cmovnc` instruction, 394, 716
- `cmovne` instruction, 395
- `cmovnge` instruction, 395
- `cmovng` instruction, 395
- `cmovnle` instruction, 395
- `cmovnl` instruction, 395
- `cmovno` instruction, 395
- `cmovns` instruction, 394
- `cmovnz` instruction, 394
- `cmovo` instruction, 394
- `cmovpo` instruction, 395
- `cmovs` instruction, 394
- `cmovz` instruction, 394
- `cmpeqps` instruction, 674
- `cmpeqsd` instruction, 373
- `cmpeqss` instruction, 372
- `cmp` instruction, 72, 293
- `cmpleps` instruction, 674
- `cmplesd` instruction, 373
- `cmpless` instruction, 372
- `cmpltps` instruction, 674
- `cmpltsd` instruction, 373
- `cmpltss` instruction, 372
- `cmpneps` instruction, 674
- `cmpnesd` instruction, 373
- `cmpness` instruction, 372
- `cmpnleps` instruction, 674
- `cmpnless` instruction, 372
- `cmpnltps` instruction, 674
- `cmpnltsd` instruction, 373
- `cmpnltss` instruction, 372
- `cmpordps` instruction, 674
- `cmpordsd` instruction, 373
- `cmpordss` instruction, 372
- `cmppd` instruction, 671
- `cmpps` instruction, 671, 674
- `cmpsd` instruction, 372
- `cmpss` instruction, 372
- `cmps` string instruction, 832
- `cmpunordps` instruction, 674
- `cmpunordsd` instruction, 373
- `cmpunordss` instruction, 372
- coalescing bit strings, 728
- code planes (Unicode), 97
- code points (Unicode), 96
- code sections, 108
- code snippets, xxviii
- coercion, 157
- collecting disparate bits into a bit string, 728
- collecting macro parameters, 764
- column major ordering, 193
  - formula, 193
- command line, xxx
- command line assembler, 6
- command line interpreter. *See* CLI
- common C++ data type sizes, 35
- commutative operators, 307
- comparing
  - a register to zero, 298
  - bits, 708
  - dates, 85
  - strings, 825
- comparison for less than (packed/vector/SIMD), 662
- comparison operators in a constant expression, 153

comparison results (SIMD), 663, 678
comparisons
  dates, 85
  floating point, 323
  SIMD, 660
comparison synonyms (AVX/SSE), 673
compile-time
  decisions, 752
  expressions and operators, 750
  language, 748
  loops, 756
  procedures, 760
compile-time function
  sizeof, 207
compile-time language. *See* CTL
compile-time statement
  echo, 748
  else, 753
  elseif, 753
  endm, 756, 759
  .err, 748
  for, 756, 759
  forc, 756
  if, 752
  while, 756
compile-time versus runtime expressions, 155–156
complete Boolean evaluation, 400
complex arithmetic expressions, 302
complex string functions, 837
composite data types, 174
computation via table lookup, 584
computing
  arctangent, 362
  cos, 361
  cosine, 361
  $\log_2(x)$, 362
  $\log_2(x)$ plus one, 362
  sine, 361
  square root, 327, 347
  tangent, 361
  2^x minus one, 361
computing the address of a memory variable, 22
computing the length of a string at assembly time, 176
concatenation of text values in MASM, 751
conditional
  compilation, 752
  jmp aliases, 392
  jmp instructions (opposite conditions), 391–392
  statements, 396
conditional jump instructions, 70
conditional jumps
  ja, 391
  jae, 391
  jb, 391
  jbe, 391
  jc, 391, 716
  je, 391
  jg, 391
  jge, 391
  jl, 391
  jle, 391
  jna, 391
  jnae, 391
  jnb, 391
  jnbe, 391
  jnc, 391, 716
  jne, 391
  jng, 391
  jnge, 391
  jnl, 391
  jnle, 391
  jno, 391
  jnp, 391
  jns, 391
  jnz, 391
  jo, 391
  jp, 391
  jpe, 391
  jpo, 391
  js, 391
  jz, 391
conditional move (if carry), 716
conditional move instructions, 394
condition code
  flags, 12
  FPU condition codes, 322
  settings after cmp instruction, 294
conditioning inputs, 589
configuring software for several environments, 754

constant
  0.0 (FPU load instruction), 360
  expressions, 131, 152
  expressions in CTL statements, 750
  log$_2$(10), 361
  log$_2$(e), 361
  log$_{10}$(2), 361
  log$_e$(2), 361
  pi, 360
constant declarations, 18, 149
constant expression evaluation, 156
constant expressions, 164
constant values, 18
construction of an activation record, 228
`continue` statement, 438
control characters, 93
control word, 321, 363
conversions (floating-point instructions), 328
converting
  32-bit integers to floating-point, 679
  arithmetic expressions to postfix notation, 366
  ASCII digit code (0 to 9) to its corresponding integer value, 95
  BCD to floating-point, 329
  between big-endian and little-endian forms, 116
  binary to hexadecimal, 48
  binary value (0 to 9) to its ASCII character representation, 95
  `break` statements to pure assembly, 438
  complex expressions to assembly, 302
  `continue` statements to pure assembly, 439
  double-precision floating-point values to single-precision, 680
  floating-point expressions to assembly, 364
  floating-point values to a decimal string, 527
  floating-point values to an integer, 319, 679
    with truncation, 680
  floating-point values to exponential form, 537
  `forever` statements to pure assembly, 436
  `for` statements to pure assembly, 437
  hexadecimal digit to a character, 493
  hexadecimal to binary, 47
  `if` statements to pure assembly, 396
  integer to floating-point, 328
  larger integer object to a smaller one (via saturation), 667
  noncommutative arithmetic operators to assembly, 305
  numbers to strings using `fbstp`, 503
  postfix notation to assembly, 367
  `repeat..until` loop to pure assembly, 434
  simple expressions to assembly, 300
  single-precision floating-point values to double-precision, 680
  strings to integers, 546
  `while` loops to pure assembly, 433
copy command (CLI), 931
copying
  arbitrary number of bytes using the `movsd` instruction, 831
  overlapping arrays using the `movs` string instructions, 830
cosine, 361
counting bits, 739
`cpuid` instruction, 599
CPU registers, 10
`cqo` instruction, 288
creating lookup tables, 590
CTL (compile-time language), 748
  conditional assembly, 752
  decisions, 752
  `else`, 753
  `elseif`, 753
  `endif`, 753
  `endm`, 756
  `forc`, 756
  `for` loop, 756
  `if` statement, 752
  `instr` operator, 751
  loops, 756

CTL *(continued)*
  macros, 760
  ! operator, 750
  % operator, 750
  procedures (compile-time), 760
  sizestr operator, 752
  substring operator, 752
  while statement, 756
cvtdq2pd instruction, 679
cvtdq2ps instruction, 679
cvtpd2dq instruction, 679
cvtpd2ps instruction, 680
cvtps2dq instruction, 680
cvtps2pd instruction, 680
cvttpd2dq instruction, 680
cvttps2dq instruction, 680
cwde instruction, 288
cwd instruction, 288
CX register, 10

D

dangling pointers, 169
data alignment, 119
  in a segment, 605
  Microsoft ABI, 144
data declaration directives, 15
data representation, 147
data type coercion, 157
data types associated with SSE/AVX move instructions, 622
data type sizes (C++), 35
date command (CLI), 931
date comparison, 85
date/time stamp of a file in a make operation, 865
db directive, 15
dd directive, 15
debugging CTL programs, 749
debugging with conditional compilation, 755
decimal arithmetic, 453, 486, 581
decimal numbering system, 44
decimal (signed) to string conversion (extended-precision), 513
decimal string-to-integer conversion, 546
decimal string-to-numeric conversion (extended-precision), 569
decimal-to-string conversion, 500
dec instruction, 149
decisions in MASM, 397
declarations
  .code section, 108
  .const, 109
  .data, 108
  .data?, 110
  typedef, 156
declaring character variables in a MASM program, 96
declaring constants, 18
declaring parameters with the proc directive, 255
default macro parameter values, 768
default segment alignment, 605
defining read-only data in a user-defined segment, 605
definite loop, 437
del command (CLI), 932
delimiter characters, 546
delimiting macro parameters, 764
denormal exception flag (DE, SSE), 369
denormalized
  exception (FPU), 320
  floating-point values, 325
  values, 90
denormal mask (DM, SSE), 370
denormals are zero (DAZ, SSE), 370
dependencies (in a makefile), 864
destructuring, 407
determining which CPU a piece of software is running on, 599
DH register, 10
dialog box (example code), 879
differences in the imul instructions, 291
different-size operands, 485
dir command, 932
direction flag and the string instructions, 826
directives, 6
  ?, 15
  align, 121
  byte, 15, 53
  catstr, 751
  db, 15
  dd, 15
  dq, 15
  dt, 15

dw, 15
dword, 15, 55
else, 753
elseif, 753
endif, 753
endm, 756, 759, 760
endp, 216
ends (for structs), 198
equ, 18, 150
extern, 850
externdef, 24, 850
for, 756, 759
forc, 756, 760
if, 753
ifb, 767
ifdef, 849
ifdif, 767
ifdifi, 767
ifidn, 767
ifidni, 767
ifnb, 767
include, 848
instr, 751
label, 156
local (in procedures), 237
macro, 760
option, 8, 238
option epilogue, 238
option prologue, 238
oword, 15, 55
proc, 216, 255
public, 8, 850
qword, 15, 55
real4, 15
real8, 15
real10, 15
sdword, 15
sizestr, 752
sqword, 15
struct, 198
substr, 752
sword, 15
tbyte, 15
textequ, 151
typedef, 156
while, 756
word, 15, 54

direct jump instructions, 382
DI register, 10
disadvantages of macros (versus procedures), 762
displacements, 113
displaying equate values during assembly, 751
distributing bit strings, 728
div and idiv instructions, 291, 466
divide-by-zero exception (FPU), 320
divide-by-zero mask (ZM, SSE), 370
division without div or idiv, 312
divpd instruction, 670
divps instruction, 670
divsd instruction, 371
divss instruction, 371
DL register, 10
domain conditioning, 589
dot notation for accessing struct/record fields, 199
dot operator, 199
double-precision floating-point format, 88
double-precision (floating-point) lanes, 599
double-precision vector types, 597
double word, 51, 54. *See also* dword
double-word strings, 825
dq directive, 15
dt directive, 15
dtoStr (double word to string) function, 493
duplicate include files/operations (preventing), 849
duplicating data in an XMM/YMM register, 620
dup operator, 182, 195
dw directive, 15, 55
dword, 51, 54
    alignment within a segment, 605
    directive, 15, 55
    dword-sized lanes, 598
    vectors (packed dwords), 598
DX register, 10
dyadic operations, 55
dynamic
    memory allocation, 106, 166
    type systems, 209

E

e10toStr function, 537
EAX, EBX, ECX, EDX, ESI, EDI, EBP, and ESP registers, 10
echo CTL statement, 748
effective address, 125
EFLAGS register, 12
else compile-time statement, 753
else directive, 753
elseif compile-time statement, 753
elseif directive, 753
else statement, 397
empty macro arguments, 767
endian byte organization, 114
endian conversions, 116
endif directive, 753
endm compile-time statement, 756, 759
endm directive, 756, 759, 760
endp directive, 216
ends directive (for structs), 198
ends (end segment) directive, 604
enumerated data constants in MASM, 156
epiloguedef option, 239
epilogue (operand for option directive), 238
eq operator, 153
equality (macro arguments), 767
equates, 149
equ directive, 18, 150
erase command (CLI), 932
escape character in MASM expressions, 750
exception-handling in C++, 30
exceptions
    divide by zero (FPU), 320
    flags (FPU), 322
    FPU exception bits, 363
    masks (FPU), 320
    overflow (FPU), 320
excess-127 exponent, 87, 88
excess-1023 exponent, 88
excess (biased) exponents, 88
exclusive-or operation, 55, 57
executing a loop backward, 445
exponent of a floating-point number, 88
expressions, 302
    and temporary values, 307
extended-precision
    addition, 454
    AND, 479
    arithmetic, 453
    comparisons, 458
    conversions
        decimal-to-string (signed), 513
        decimal-to-string (unsigned), 566
        string-to-numeric, 555
        unsigned integer-to-string, 508
    division, 466
    floating-point format, 89
    formatted I/O, 514
    I/O, 491
    multiplication, 461
    neg, 477
    NOT, 480
    numeric conversion routines, 546
    OR, 479
    rotates, 484
    shifts, 480
    shifts and the flags, 482
    XOR, 480
external directives, 849
external symbols, 850
external symbol types, 851
externdef directive, 24, 849, 851
extern directive, 849, 851
extracting
    bits, 708
    bit strings, 742
    sign bits from SSE/AVX floating-point values, 676
extractps instruction, 643

F

f2xm1 instruction, 361
fabs instruction, 349
facade code, 27
fadd instruction, 330
faddp instruction, 330
false precision, 315
false (representation), 308
FASTCALL calling convention, 263
fbld instruction, 329, 488, 566
fbstp instruction, 329, 488, 503, 566
fchs instruction, 349

- fclex instruction, 363
- fcomi instruction, 357
- fcom instruction, 322, 350
- fcomip instruction, 357
- fcomp instruction, 322, 350
- fcompp instruction, 322, 350
- fcos instruction, 361
- fdiv instruction, 343
- fdivp instruction, 343
- fdivr instruction, 343
- fdivrp instruction, 343
- ficom instruction, 322
- ficomp instruction, 322
- field, 197
- field access (of a record/struct) via a pointer, 199
- field alignment within a record, 204
- fild instruction, 328
- finit instruction, 363
- first clear bit, 708, 736
- first set bit, 708, 736
- fist instruction, 328
- fistp instruction, 328
- fisttp instruction, 328
- flags, 12
  - and instruction, 712
  - carry, 12, 294
  - cmp instruction effect on flags, 293
  - copying AH register to flags, 86, 350
  - direction, 826
  - lahf instruction, 86
  - or instruction, 712
  - overflow, 293
  - sign, 293
  - xor instruction, 712
  - zero, 293
- flag settings for the logical instructions (and, or, xor, and not), 71
- FLAGS register, 12
- fld1 instruction, 360
- fldcw instruction, 321, 363
- fld instruction, 326
- fldl2e instruction, 361
- fldl2t instruction, 361
- fldlg2 instruction, 361
- fldln2 instruction, 361
- fldpi instruction, 360
- fldz instruction, 360
- floating-point
  - arithmetic, 317
  - calculations, 317
  - comparisons, 323, 350
    - SIMD, 671
  - control register, 317
  - control word, 321, 363
  - conversion to integer, 319, 328
  - conversion to string, 519, 527
    - exponential form, 537
  - data registers, 317
  - data types, 324
  - division, 343
  - exchange registers, 327
  - FPU (floating-point unit), 11, 317
  - multiplication, 339
  - negation, 349
  - normalized format, 325
  - overflow, 316
  - overflow exception, 320
  - partial remainder, 348
  - precision control, 320
  - pushing a value onto the FPU stack, 326
  - pushing the constant 1.0 onto the FPU stack, 360
  - registers, 11, 317
  - remainder, 348
  - rounding control, 319
  - status register, 317
  - string conversion (to real), 570
  - string output, 519
  - subtraction, 334
  - test for zero, 322, 360
  - underflow, 316
  - unordered comparisons, 357, 360
  - unit. *See* FPU
  - values, 54
    - as parameters, 244
- flush to zero (FZ, SSE), 370
- fmul instruction, 339
- fmulp instruction, 339
- fnclex instruction, 363
- fninit instruction, 363
- fnstsw instruction, 364
- forc directive, 756, 759

forcing
a zero result, 56
bits to one, 58
bits to zero, 58
for directive, 756, 759
for and endm compile-time statement, 756, 759
for loops, 437
format specifiers (printf), 24
formatted numeric-to-string conversions, 514
formula for two-dimensional row-major access, 191
FORTRAN programming language, 424
four-dimensional array element access, 191
fpatan instruction, 362
fprem1 instruction, 348
fprem instruction, 348
fptan instruction, 361
FPU (floating-point unit), 11, 317
busy bit, 324
condition code bits, 322
control register, 318
control word, 321, 363
data movement instructions, 326
data registers, 317
data types, 324
denormalized result exception, 320
divide-by-zero exception, 320
exception bits, 363
exception flags, 322
exception masks, 320
floating-point unit, 317
invalid operation exception, 320
overflow exception, 320
popping the FPU stack, 326
precision exception, 321
registers, 317
rounding control, 319
round-up and round-down, 319
stack fault flag, 322
status register, 321, 364
status word, 321
top of stack pointer, 324
truncate during computations, 319
underflow exception, 321
free (memory deallocation) function, 170
frndint instruction, 349
fsincos instruction, 361
fsin instruction, 361
fsqrt instruction, 327, 347
fstcw instruction, 321, 363
fst instruction, 326
fstp instruction, 326
fstsw instruction, 321, 350, 364
fsub instruction, 334
fsubp instruction, 334
fsubr instruction, 334
fsubrp instruction, 334
ftst instruction, 322, 360
fucom instruction, 323
fucomp instruction, 323
fucompp instruction, 323
function
computation via table lookup, 584
results, 270
fxam instruction, 323
fxch instruction, 327
fyl2x instruction, 362
fyl2xp1 instruction, 362

G

general protection fault, 107
general purpose registers, 10, 12
ge operator, 153
getLastError function, 891
getStdErrHandle function, 883
GetStdHandle (Win32 API function), 875
getStdInHandle function, 884
getStdOutHandle function, 883
getting the address of a variable, 22
granularity (MMU pages), 111
greater-than comparisons on SSE CPUs, 673
GT operator, 153
guard digits/bits, 314

H

haddpd instruction, 671
haddps instruction, 671
handling SIMD comparisons, 663
header files, 849, 852

- heap variable address alignment, 607
- Hello, world!
  - compile-time program, 748
  - MASM program, 6
  - stand-alone version, 874
- hexadecimal
  - digit-to-character conversion, 493
  - hexadecimal-to-string conversion, 492
    - using table lookup, 497
  - numbering system, 43, 46
  - numbers, 51
  - output (extended-precision), 499
  - string-to-numeric conversion, 556
- `high32` operator, 153
- `high` operator, 153
- high-order (HO), 46
  - bit, 46, 52
  - byte, 53
  - nibble, 52
  - word, 54
- `highword` operator, 153
- HO (high-order), 46
- horizontal addition, 650
  - and subtraction (floating-point), 671
- `hsubpd` instruction, 671
- `hsubps` instruction, 671
- hybrid programs (assembly and C/C++), 7

I

- `i128toStr` function, 513
- identifiers, 8
- idiom, 685
  - machine idiosyncrasies, 310
- `idiv` instruction, 291, 407, 466
- IEEE
  - floating-point standard, 86, 318, 320
- `ifb` directive, 767
- `if` compile-time statement, 752
- `if` conditional statement, 396
- `ifdef` directive, 849
- `ifdif` directive, 767
- `ifdifi` directive, 767
- `if` directive, 753
- `ifidn` directive, 767
- `ifidni` directive, 767
- `ifnb` directive, 767
- `imul` instruction, 148, 289, 461
- `inc` instruction, 149
- `include` directive, 848
- inclusive-or operation, 56
- indirect
  - addressing modes, 124
    - indirect and scaled-indexed addressing modes, 106
    - indirect-plus-offset addressing mode, 125
  - calls, 278
  - `jump` instructions, 383
  - jumps, 396, 424
    - through a memory pointer, 389
- induction variables, 449
- infinite loops, 433
- infinite-precision arithmetic, 313
- infinity (IEEE representation), 90
- infix notation, 364
- initialized arrays, 183
- initializing struct fields, 200
- initializing the FPU, 363
- input conditioning, 589
- input/output (I/O), 9
- input redirection, 927
- inserting
  - a bit into a bit array, 734
  - a bit set into another bit string, 710
  - a bit string into a larger bit string, 718
- `insertps` instruction, 643
- `instr` directive, 751
- instructions
  - `adc`, 455, 716
  - `add`, 21
  - `addpd`, 669
  - `addps`, 669
  - `addsd`, 371
  - `adss`, 371
  - `and`, 58, 309, 709
  - `andnpd`, 645
  - `andpd`, 645
  - `bsf`, 737
  - `bsr`, 737
  - `bswap`, 116
  - `bt`, 715
  - `btc`, 715
  - `btr`, 715

instructions *(continued)*
- bts, 715
- call, 22, 216, 218
- cbw, 288
- cdq, 288
- cdqe, 288
- clc, 86, 716
- cld, 86
- cli, 86
- cmc, 86, 716
- cmova, 395
- cmovae, 395
- cmovb, 395
- cmovbe, 395
- cmovc, 394, 716
- cmove, 395
- cmovg, 395
- cmovge, 395
- cmovl, 395
- cmovle, 395
- cmovna, 395
- cmovnae, 395
- cmovnb, 395
- cmovnbe, 395
- cmovnc, 394, 716
- cmovne, 395
- cmovng, 395
- cmovnge, 395
- cmovnl, 395
- cmovnle, 395
- cmovno, 395
- cmovnp, 395
- cmovns, 394
- cmovnz, 394
- cmovo, 394
- cmovp, 395
- cmovpe, 395
- cmovpo, 395
- cmovs, 394
- cmovz, 394
- cmp, 72, 293
- cmpeqps, 674
- cmpeqsd, 373
- cmpeqss, 372
- cmpleps, 674
- cmplesd, 373
- cmpless, 372
- cmpltps, 674
- cmpltsd, 373
- cmpltss, 372
- cmpneps, 674
- cmpnesd, 373
- cmpness, 372
- cmpnleps, 674
- cmpnless, 373
- cmpnltps, 674
- cmpnltsd, 373
- cmpnltss, 372
- cmpordps, 674
- cmpordsd, 373
- cmpordss, 373
- cmppd, 671
- cmpps, 671, 674
- cmps, 832
- cmpsd, 372
- cmpss, 372
- cmpunordps, 674
- cmpunordsd, 373
- cmpunordss, 372
- cqo, 288
- cvtdq2pd, 679
- cvtdq2ps, 679
- cvtpd2dq, 679
- cvtpd2ps, 680
- cvtps2dq, 680
- cvtps2pd, 680
- cvttpd2dq, 680
- cvttps2dq, 680
- cwd, 288
- cwde, 288
- dec, 149
- div, 291, 466
- divpd, 670
- divps, 670
- divsd, 371
- divss, 371
- extractps, 643
- f2xm1, 361
- fabs, 349
- fadd, 330
- faddp, 330
- fbld, 329, 488, 503
- fbstp, 329, 488, 503, 566
- fchs, 349
- fclex, 363
- fcom, 322, 350

fcomi, 357
fcomip, 357
fcomp, 322, 350
fcompp, 322, 350
fcos, 361
fdiv, 343
fdivp, 343
fdivr, 343
fdivrp, 343
ficom, 322
ficomp, 322
fild, 328
finit, 363
fist, 328
fistp, 328
fisttp, 328
fld, 326
fld1, 360
fld2e, 361
fldcw, 321, 363
fldl2t, 361
fldlg2, 361
fldln2, 361
fldpi, 360
fldz, 360
floating-point comparisons, 350
floating-point conversions, 328
fmul, 339
fmulp, 339
fnclex, 363
fninit, 363
fnstsw, 364
fpatan, 362
fprem, 348
fprem1, 348
fptan, 361
FPU data movement, 326
frndint, 349
fsin, 361
fsincos, 361
fsqrt, 327, 347
fst, 326
fstcw, 321, 363
fstp, 326
fstsw, 321, 350, 364
fsub, 334
fsubp, 334
fsubr, 334
fsubrp, 334
ftst, 322, 360
fucom, 323
fucomp, 323
fxam, 323
fxch, 327
fyl2x, 362
fyl2xp1, 362
haddpd, 671
haddps, 671
hsubpd, 671
hsubps, 671
idiv, 291, 407, 466
imul, 148, 289, 461
inc, 149
indirect jumps, 383
insertps, 643
intmul, 291
ja, 73, 391
jae, 73, 391
jb, 73, 391
jbe, 73, 391
jc, 70, 74, 391, 716
je, 72, 74, 391–392
jg, 73, 391
jge, 73, 391–392
jl, 73, 391–392
jle, 73, 391–392
jmp, 69, 382
jna, 74, 391–392
jnae, 74, 391
jnb, 74, 391
jnbe, 74, 391
jnc, 70, 74, 391, 716
jne, 72, 74, 391–392
jng, 74, 391–392
jnge, 74, 391–392
jnl, 74, 391–392
jnle, 74, 391
jno, 70, 391
jnp, 70, 391
jns, 70, 391
jnz, 70, 74, 298, 391
jo, 70, 391
jp, 391
jpe, 391
jpo, 391
js, 70, 391

instructions *(continued)*
- jz, 70, 74, 298, 391
- lahf, 86
- lddqu, 622
- ldmxcsr, 370
- lea, 22, 125, 378
- leave, 234
- lods, 836
- maxpd, 670
- maxps, 670
- maxsd, 371
- maxss, 371
- minpd, 670
- minps, 670
- minsd, 371
- minss, 371
- mov, 18, 122
- movapd, 610
- movaps, 610
- movd, 371, 609
- movddup, 621
- movdqa, 610
- movdqu, 612
- movhlps, 619
- movhpd, 617
- movhps, 617
- movlhps, 619
- movlpd, 615
- movlps, 615
- movmskpd, 676
- movmskps, 676
- movq, 371, 609
- movs, 826
- movsb, 826
- movsd, 370, 826
- movshdup, 620
- movsldup, 620
- movss, 370
- movsw, 826
- movupd, 612
- movups, 612
- mul, 289, 461
- mulpd, 670
- mulps, 670
- mulsd, 371
- mulss, 371
- neg, 478
- not, 58, 309, 709
- or, 58, 309, 709
- orpd, 645
- pabsb, 659
- pabsd, 659
- pabsw, 659
- packssdw, 667
- packsswb, 667
- packusdw, 667
- packuswb, 667
- paddb, 648
- paddd, 649
- paddq, 649
- paddw, 648–649
- pavgb, 657
- pavgw, 657
- pclmulqdq, 656
- pcmpeqb, 660
- pcmpeqd, 660
- pcmpeqq, 660
- pcmpeqw, 660
- pcmpgtb, 660
- pcmpgtd, 660
- pcmpgtq, 660
- pcmpgtw, 660
- pextrb, 641
- pextrd, 642
- pextrq, 642
- pextrw, 642
- phaddd, 650
- phaddw, 650
- pinsrd, 642
- pinsrq, 642
- pinsrw, 642
- pmaxsb, 657
- pmaxsd, 658
- pmaxsq, 658
- pmaxsw, 657
- pmaxub, 658
- pmaxud, 658
- pmaxuq, 658
- pmaxuw, 658
- pminsb, 658
- pminsd, 658
- pminsw, 658
- pminub, 658
- pminud, 658
- pminuq, 658
- pminuw, 658

pmovmskb, 662
pmovsxbd, 666
pmovsxbq, 666
pmovsxbw, 666
pmovsxdq, 666
pmovsxwd, 666
pmovsxwq, 666
pmovzxbd, 665
pmovzxbq, 665
pmovzxbw, 665
pmovzxdq, 665
pmovzxwd, 665
pmovzxwq, 665
pmuldq, 656
pmulld, 655
pmuludq, 656
pop, 135, 222
popf, 140
popfd, 140
pshufb, 625
pshufd, 626
pshufhw, 628
pshuflw, 628
psignb, 659
psignd, 660
psignw, 659
pslldq, 647
psllw, 647
psrldq, 647
psubb, 654
psubd, 653
psubq, 653
psubw, 654
ptest, 646
punpckhbw, 637
punpckhdq, 637
punpckhqdq, 637
punpcklbw, 637
punpckldq, 637
punpcklqdq, 637
punpcklwd, 637
push, 134, 222
pushf, 140
pushfq, 140
pushw, 134
rcl, 79, 716
rcpss, 372
rcr, 79, 716
repe prefix on cmpsb, cmpsw, cmpsd, and cmpsq, 827
repne prefix on cmpsb, cmpsw, cmpsd, and cmpsq, 827
rep prefix on movsb, movsw, movsd, and movsq, 826
ret, 22, 218
rol, 78
ror, 78
rsqrtps, 670
rsqrtss, 372
sahf, 86, 350
sar, 77, 312
sbb, 457, 716
scas, 835
seta, 296
setae, 296
setb, 296
setbe, 296
setc, 295, 716
sete, 296
setg, 296
setge, 297
setl, 297
setna, 296
setnae, 296
setnb, 296
setnbe, 296
setnc, 295, 716
setne, 296
setng, 297
setnge, 297
setnl, 297
setnle, 296
setno, 295
setnp, 295
setns, 295
setnz, 295, 298
seto, 295
setp, 295
setpe, 295
setpo, 295
sets, 295
setz, 295, 298
shl, 75, 310
shld, 482
shr, 76, 312
shrd, 482

instructions *(continued)*
- shufpd, 630
- shufps, 630
- sqrtpd, 670
- sqrtps, 670
- sqrtsd, 372
- sqrtss, 372
- stc, 716
- std, 86
- sti, 86
- stmxcsr, 370
- stos, 835
- sub, 21
- subpd, 669
- subps, 669
- subsd, 371
- subss, 371
- test, 297, 709
- unpckhpd, 633
- unpckhps, 633
- unpcklpd, 633
- unpcklps, 633
- vaddpd, 669
- vaddps, 669
- vandnpd, 645
- vandpd, 645
- vcmppd, 671, 674
- vcmpps, 671, 674
- vcvtdq2pd, 679
- vcvtdq2ps, 679
- vcvtpd2dq, 679
- vcvtpd2ps, 680
- vcvtps2dq, 680
- vcvtps2pd, 680
- vcvttpd2dq, 680
- vcvttps2dq, 680
- vdivpd, 670
- vdivps, 670
- vextractps, 643
- vhaddpd, 671
- vhaddps, 671
- vhsubpd, 671
- vhsubps, 671
- vinsertps, 643
- vlddqu, 622
- vmaxpd, 670
- vmaxps, 670
- vminpd, 670
- vminps, 670
- vmovapd, 610
- vmovaps, 610
- vmovd, 609
- vmovddup, 621
- vmovdqa, 610
- vmovdqu, 612
- vmovhlps, 619
- vmovhpd, 618
- vmovhps, 618
- vmovlhps, 619
- vmovlpd, 615
- vmovlps, 615
- vmovmskpd, 676
- vmovmskps, 676
- vmovq, 609
- vmovshdup, 620
- vmovsldup, 620
- vmovupd, 612
- vmovups, 612
- vmulpd, 670
- vmulps, 670
- vorpd, 645
- vpabsb, 659
- vpabsd, 659
- vpabsw, 659
- vpackssdw, 667
- vpacksswb, 667
- vpackusdw, 667
- vpackuswb, 667
- vpaddb, 649
- vpaddd, 649
- vpaddq, 649
- vpaddw, 648–649
- vpavgb, 657
- vpavgw, 657
- vpclmulqdq, 656
- vpcmpeqb, 661
- vpcmpeqd, 661
- vpcmpeqq, 661
- vpcmpeqw, 661
- vpcmpgtb, 661
- vpcmpgtd, 661
- vpcmpgtq, 661
- vpcmpgtw, 661
- vpextrb, 642
- vpextrd, 642
- vpextrq, 642

vpextrw, 642
vphaddd, 650
vphaddw, 650
vpinsrd, 643
vpinsrq, 643
vpinsrw, 643
vpmaxsb, 657
vpmaxsd, 658
vpmaxsq, 658
vpmaxsw, 657
vpmaxub, 658
vpmaxud, 658
vpmaxuq, 658
vpmaxuw, 658
vpminsb, 658
vpminsd, 658
vpminsw, 658
vpminub, 658
vpminud, 658
vpminuq, 658
vpminuw, 658
vpmovmskb, 662
vpmovsxbd, 666
vpmovsxbq, 666
vpmovsxbw, 666
vpmovsxdq, 666
vpmovsxwd, 666
vpmovsxwq, 666
vpmovzxbd, 665
vpmovzxbq, 665
vpmovzxbw, 665
vpmovzxdq, 665
vpmovzxwd, 665
vpmovzxwq, 665
vpmuldq, 656
vpmulld, 655
vpmuludq, 656
vpshufb, 625
vpshufd, 626
vpshufhw, 628
vpshuflw, 628
vpshufps, 632
vpsignb, 659
vpsignd, 660
vpsignw, 659
vpslldq, 647
vpsllw, 647
vpsrldq, 647
vpsubb, 654
vpsubd, 653
vpsubq, 653
vpsubw, 654
vptest, 646
vpunpckhbw, 640
vpunpckhdq, 641
vpunpckhqdq, 641
vpunpckhwd, 640
vpunpcklbw, 640
vpunpckldq, 640
vpunpcklqdq, 641
vrsqrtps, 670
vshufpd, 632
vsqrtpd, 670
vsqrtps, 670
vsubpd, 669
vsubps, 669
vunpckhpd, 633
vunpckhps, 633
vunpcklpd, 633
vunpcklps, 633
vxorpd, 645
xchg, 116
xlat, 584
xor, 58, 309, 709, 712
xorpd, 645

integer
- addition (SIMD), 648
- arithmetic (SIMD), 648
- average computation (SIMD), 657
- comparisons (SIMD), 660
- conversions (SIMD), 664
- integer portion of a floating-point number, 349
- integer-to-floating-point conversion, 328
- integer-to-string conversion (extended precision, unsigned), 508
- integer-to-string conversion (signed), 507
- less-than comparison (SIMD), 662
- multiplication (SIMD), 654
- signed remainder/modulo, 407
- subtraction (SIMD), 653

integer types in C, 454

integer unpack instructions (SSE/AVX), 637
interleaving comparison results (SIMD), 664
imul instruction, 291
invalid arithmetic operation (IA), 673
invalid operation exception flag (IE, SSE), 369
invalid operation exception (FPU), 320
invalid operation mask (IM, SSE), 370
invariant computations, 446
inverting
  bits, 58, 708
  bits in a bit string, 57
  selected bits in a bit set, 712
I/O (input/output), 9
iSize function, 516
itoStrSize function, 517–518

J

jae instruction, 73, 391
ja instruction, 73, 390
jbe instruction, 73, 390
jb instruction, 73, 390
jc instruction, 70, 74, 390, 716
je instruction, 72, 74, 390, 390–391
jge instruction, 73, 390, 392
jg instruction, 73, 391
jle instruction, 73, 390, 392
jl instruction, 73, 390, 392
jmp instruction, 69, 382
jnae instruction, 74, 390
jna instruction, 74, 390
jnbe instruction, 74, 390
jnb instruction, 74, 390
jnc instruction, 70, 74, 390, 716
jne instruction, 72, 74, 390, 390–391
jnge instruction, 74, 390, 392
jng instruction, 74, 390, 392
jnle instruction, 74, 390
jnl instruction, 74, 390, 392
jno instruction, 70, 390
jnp instruction, 390
jns instruction, 70, 390
jnz instruction, 70, 74, 298, 390
jo instruction, 70, 390
jpe instruction, 390
jp instruction, 390
jpo instruction, 390
js instruction, 70, 390
jump instructions, 382
jz instruction, 70, 74, 298, 390

K

KCS Floating-Point Standard, 87

L

label declaration, 114
label directive, 156
labels, 378
  in a procedure, 219
lahf instruction, 86
lanes (elements of an SSE/AVX packed array), 598
LARGEADDRESSAWARE, 127
  and arrays, 183
large address unaware applications, 127
large parameters, 258
last clear bit, 708, 736
last-in, first-out (LIFO) data structures, 137
last set bit, 736
lddqu instruction, 622
ldmxcsr instruction, 370
leaf function, 278
lea instruction, 22, 125, 378
least significant bit, 46, 52
leave instruction, 234
left
  rotates, 78
  shifts, 75
left-associative operators, 304
lengthof operator, 153
length of text string in MASM textual constants, 752
length-prefixed strings, 175
le operator, 153
less-than comparison (SIMD), 662
lexical scope, 378
lexicographical ordering, 833
library file, 869
library module, 853
lifetime of a local variable, 234
LIFO (last in, first out), 137
linear search, 422
line feed, 93

listings, xxviii
literal constant, 18
little-endian data organization, 114
little-endian to big-endian conversion, 116
LO (low-order), 46
load effective address, 378
  instruction, 22
loading data into an SSE/AVX register, 610
loading single-precision vectors into SSE/AVX registers, 612
loading the flags register from AH, 86
loading the FPU control word, 363
local directive (in procedures), 237
local symbols in procedures, 378
local symbols (statement labels) in a procedure, 219
local variable access, 235
local variable address alignment, 607
local variables, 234
location counter, 113, 154
`lods` instruction, 836
$\log_2(e)$, 361
$\log_2(x)$, 362
logical
  AND operation, 55, 309
  exclusive-or operation, 55, 57
  NOT operation, 55, 57
  operations on binary numbers, 57
  operations on bits, 55
  operators within a constant expression, 153
  OR operation, 55, 309
  shift right, 77
  XOR operation, 55, 309
logical systems
  arithmetic, 310
  Boolean, 310
loops, 433, 437
  invariant computations, 446
  loop-control variables, 433
  register usage, 442
  termination, 443
  unraveling/unrolling, 447
loops in the MASM compile-time language, 756
`low32` operator, 154
low-level control structures, 378
`low` operator, 153
low-order (LO), 46
  bit, 46, 52
  byte, 53
  nibble, 52
  word, 54
`lowword` operator, 153
`lt` operator, 153

M

machine code encoding, 73
machine idioms, 310
machine state (preservation), 220
machine state, saving the, 220
macro
  default parameter values, 768
  optional parameters, 766
  parameter delimiters, 764
  parameter expansion, 762
  parameter expansion issues, 765
  parameters, 762
  required parameters, 766
macroarchitecture, 622
`macro` directive, 760
macros, 760
make dependencies, 864
makefiles, 34
makefile syntax, 863
making symbols case-sensitive in MASM, 8
`malloc` (C Standard Library function), 166
manifest constants, 18, 149
manipulating bits in memory, 707
mantissa, 87
mask (bits), 708
masking
  bit strings, 58
  masking in bits, 58
  masking out bits, 58
MASM (Microsoft Macro Assembler)
  `dup` operator in a data declaration, 31
  enumerated constants, 156
  pointers, 162
  procedures, 22
  structures (`struct`), 198

MASM *(continued)*
support for ASCII characters, 95
variables, 14
masm32.com website, 874
MASM /c command line option, 9
MASM/C++ hybrid programs, 7
maximum instructions (SIMD), 657
maxpd instruction, 670
maxps instruction, 670
maxsd instruction, 372
maxss instruction, 371
memory, 9
addressing modes, 105, 122
allocation, 105
indirect jump through memory, 389
organization, 106
read operation, 14
subsystem, 13
write operation, 13
memory access violation exception, 169
memory addresses, 9
memory alignment requirements (SSE/AVX/SIMD), 606
memory leaks, 171
memory management unit (MMU), 111
merging bit strings, 741
merging source files during assembly, 848
microarchitecture, 622
Microsoft ABI, 35
data alignment boundary, 144
register usage, 38
volatile registers, 38
Microsoft Macro Assembler. *See* MASM
Microsoft Visual C++ (MSVC), 9, 920
minimal procedures, 218
minimum instructions (SIMD), 657
minpd instruction, 670
minps instruction, 670
minsd instruction, 371
minss instruction, 371
misaligned data and the system cache, 121
mkActRec (macro), 882
MMU (memory management unit), 111
MMX (Multimedia Extensions), 624
MMX register set, 11
mnemonic, 289
modulo
floating-point remainder, 348
integer remainder, 407
modulo-*n* counters, 312
mod (within a constant expression), 153
monadic operations, 57
more command (CLI), 932
most significant bit, 46, 52
movapd instruction, 610
movapd operands (MASM), 611
movaps instruction, 610
movaps operands (MASM), 611
movddup instruction, 621
movd instruction, 371, 609
movdqa instruction, 610
movdqa operands (MASM), 611
movdqu instruction, 612
move command (CLI), 933
movhlps instruction, 619
movhpd instruction, 617
movhps instruction, 617
moving string data, 825
mov instruction, 18, 122
mov instruction operands, 20
movlhps instruction, 619
movlpd instruction, 615
movlps instruction, 615
movmskpd instruction, 676
movmskps instruction, 676
movq instruction, 371, 609
movsb instruction, 827
movsd instruction, 370, 827
movshdup instruction, 620
movs instruction, 827
movs instruction performance, 831
movsldup instruction, 620
movss instruction, 370
movsw instruction, 827
movsx instruction, 288
movupd instruction, 612
movups instruction, 612
MSVC (Microsoft Visual C++), 9, 920
mul instruction, 289, 461
mulpd instruction, 670
mulps instruction, 670
mulsd instruction, 371
mulss instruction, 371

multi-byte data structure organization (in memory), 114
multilingual planes (Unicode), 97
Multimedia Extensions (MMX), 624
multiple data values in a single data declaration, 16
multiplication, 148, 289, 291, 461
- floating-point, 339

multiplying
- by a reciprocal to simulate division, 312
- register value by ten, 311
- without mul or imul, 310

multiprecision
- addition, 454
- comparisons, 458
- operations, 454, 703
- subtraction, 457

N

namespace pollution, 220, 878
naming a segment, 604
NaN (not a number), 90, 296, 320
natural data alignment boundary, 144
neg128 (macro), 760
negating large values, 478
negation (floating-point), 349
neg instruction, 478
ne operator, 153
nested array constants, 195
nested dup operator, 195
nested structs, 200
nested subfield access (of a structure), 200
newLn function, 886
nibble, 51
N/No N rule, 392
noncommutative binary operators, 308
nonvolatile registers, 265
nonvolatile registers (Microsoft ABI), 39
normalized floating-point numbers, 89, 325
not a number (NaN), 90, 296
not instruction, 58, 309, 709
NOT operation, 55, 57
NOT operator, 153
NUL character, 176, 248
NULL pointer references, 107
numbering system, 44
- binary, 44
- decimal, 44
- hexadecimal, 46
- positional, 44

numeric
- conversion from string, 546
- memory addresses, 9
- numeric-to-string conversion performance, 507
- numeric-to-string conversions, 491
- representation, 48

O

octal words, 55
offset operator, 154, 378
offsets, 113
one's complement format, 87
opattr operator, 154
opcode, 123
open function, 888
openNew function, 889
operation code (opcode), 123
operations
- AND, 309
- NOT, 309
- on binary numbers, 57
- OR, 56, 309
- rotation, 74
- shift arithmetic right, 77
- shifts, 74
- XOR, 57, 309

operator precedence, 303
operators, 195
- $, 154
- AND, 153
- dot (structure/record field access), 199
- dup, 182, 195
- eq, 153
- ge, 153
- gt, 153
- high, 153
- high32, 153
- highword, 153
- le, 153
- lengthof, 153
- logical operators, 153

operators *(continued)*
- low, 153
- low32, 154
- lowword, 153
- lt, 153
- ne, 153
- NOT, 153
- offset, 154, 378
- opattr, 154
- OR, 153
- size, 154
- sizeof, 154
- this, 154
- type, 159

opposite jumps, 392
optional macro parameters, 766
option directive, 8, 238
- epilogue operand, 238
- prologue operand, 238

ordered comparison, 90, 373
or instruction, 58, 309, 709
OR operation, 55
OR operator, 153
orpd instruction, 645
output redirection (standard output), 926
overflow exception flag (OE, SSE), 369
overflow exception (FPU), 320
overflow flag, 12, 293
- setting after an arithmetic operation, 71

overflow mask (OM, SSE), 370
overlaid registers (XMM/YMM), 623
oword, 51
oword directive, 15, 55

P

pabsb instruction, 659
pabsd instruction, 659
pabsw instruction, 659
packed
- absolute value (integer), 659
- addition, 648
- arrays of bit strings, 733
- byte data types, 597
- data, 79
- decimal arithmetic, 488
- double (precision) arithmetic instructions, 668
- dword data types, 598
- floating-point arithmetic, 668
- integer comparisons, 660
- integer multiplication, 654
- memory operands (SSE/AVX), 606
- operands for SSE/AVX instructions, 606
- qword data types, 598
- shifts, 647
- sign extension, 666
- sign transfer, 659
- (SIMD) integer comparison for less than, 662
- single (precision) arithmetic instructions, 668
- word data types, 597
- zero extension, 665

packing and unpacking bit strings, 717
packssdw instruction, 667
packsswb instruction, 667
packusdw instruction, 667
packuswb instruction, 667
paddb instruction, 649
paddd instruction, 649
paddq instruction, 649
paddw instruction, 648–649
page (256-byte) alignment within a segment, 605
pages (memory management), 111
paragraph memory alignment, 606
paragraph (para/16-byte) alignment within a segment, 605
parameter declarations with the proc directive, 255
parameter expansion in macros, 762
parameters, 240
- variable length, 248

partial remainder, 348
pass by reference
- efficiency, 243

passing
- large objects as parameters, 258
- parameters by reference, 241
- parameters by value, 241
- parameters in registers, 243
- parameters in the code stream, 246
- parameters on the stack, 249

pavgb instruction, 657

pavgw instruction, 657
pclmulqdq instruction, 656
pcmpeqb instruction, 660
pcmpeqd instruction, 660
pcmpeqq instruction, 660
pcmpeqw instruction, 660
pcmpgtb instruction, 660
pcmpgtd instruction, 660
pcmpgtq instruction, 660
pcmpgtw instruction, 660
PC-relative addressing mode, 122
performance improvements for loops, 443
performance of numeric-to-string conversion, 507
performance of the string instructions, 837
pextrb instruction, 641
pextrd instruction, 641
pextrq instruction, 641
pextrw instruction, 641
phaddd instruction, 650
phaddsw instruction, 650
phaddw instruction, 650
pi (FPU load instruction), 360
pinsrb instruction, 642
pinsrd instruction, 642
pinsrq instruction, 642
pinsrw instruction, 642
pmaxsb instruction, 657
pmaxsd instruction, 658
pmaxsq instruction, 658
pmaxsw instruction, 657
pmaxub instruction, 658
pmaxud instruction, 658
pmaxuq instruction, 658
pmaxuw instruction, 658
pminsb instruction, 658
pminsd instruction, 658
pminsq instruction, 658
pminsw instruction, 658
pminub instruction, 658
pminud instruction, 658
pminuq instruction, 658
pminuw instruction, 658
pmovmskb instruction, 662
pmovmskd simulation, 663
pmovmskw simulation, 663
pmovmsq simulation, 663
pmovsxbd instruction, 666
pmovsxbq instruction, 666
pmovsxbw instruction, 666
pmovsxdq instruction, 666
pmovsxwq instruction, 666
pmovzxbd instruction, 665
pmovzxbq instruciton, 665
pmovzxbw instruction, 665
pmovzxdq instruction, 665
pmovzxwd instruction, 666
pmovzxwq instruction, 665
pmuldq instruction, 656
pmulld instruction, 655
pmuludq instruction, 656
pointer constants and pointer constant expressions, 164
pointer data access, 162
pointer problems, 167
pointers, 161
popfd instruction, 140
popf instruction, 140
pop instruction, 135, 222
popping the FPU stack, 326
postfix notation, 364
    conversion to assembly language, 367
precedence
    of arithmetic operators, 303
    rules, 303
precision, 314
    control bits (FPU), 320
    control during floating-point computations, 320
    exception (FPU), 321
    precision exception flag (PE, SSE), 369
    precision mask (PM, SSE), 370
preserving
    machine state, 220
    registers, 38, 137, 220
        in loops, 442
printf format specifiers, 24
problems with macro parameter expansion, 765
proc directive, 216, 255
    parameter declarations, 255

procedural parameters, 280
passing procedures as parameters, 280
procedure invocation, 216
procedure pointers, 278
procedures, 22, 216
effect on the stack, 278
in MASM, 22
processing SIMD comparison results, 678
proc external symbol type, 851
program counter in a section, 154
programming in the large, 847
programming language
FORTRAN, 424
program size and object/library files, 870
prolog (standard entry sequence code), 239
option, 239
prologue (operand for option directive), 238
pshufb instruction, 625
pshufd instruction, 626
pshufhw instruction, 628
pshuflw instruction, 628
psignb instruction, 659
psignd instruction, 660
psignw instruction, 659
pslldq instruction, 647
psllw instruction, 647
psrldq instruction, 647
psubb instruction, 654
psubd instruction, 653
psubq instruction, 653
psubw instruction, 654
ptest instruction, 646
public directive, 8, 849
punpckhbw instruction, 637
punpckhdq instruction, 637
punpckhqdq instruction, 637
punpckhwd instruction, 637
punpcklbw instruction, 637
punpckldq instruction, 637
punpcklqdq instruction, 637
punpcklwd instruction, 637
pushf instruction, 140
pushfq instruction, 140
pushing a value onto the floating-point stack, 326
pushing the constant 1.0 onto the FPU stack, 360
push instruction, 134, 222
pushw instruction, 134
puts function, 885

Q

qtoStr (quad word to string) function, 493
quad words, 55
quad-word strings, 825
question mark in a data declaration directive, 15
quicksort, 272
qword, 51
qword data declarations, 55
qword directive, 15
qword-sized lanes, 599
qword vectors (packed qwords), 598

R

R8B, R9B, R10B, R11B, R12B, R13B, R14B, and R15B registers, 10
R8D, R9D, R10D, R11D, R12D, R13D, R14D, and R15D registers, 10
R8W, R9W, R10W, R11W, R12W, R13W, R14W, and R15W registers, 10
r10toStr function, 527, 530
radix, 46
range of a function, 586
RAX, RBX, RCX, RDX, RSI, RDI, RBP, RSP, R8, R9, R10, R11, R12, R13, R14, and R15 registers, 10
RBP register, 13, 230
rcl instruction, 79, 716
rcpss instruction, 372
rcr instruction, 79, 716
RCX register usage in string instructions, 826
RDI register usage in string instructions, 826
rd/rmdir commands (CLI), 933
read function, 887
reading from memory, 13
readLine() function, 30
readLn function, 893

`readonly`
  segment argument, 605
  variables as constants, 150
`real4` directive, 15
`real8` directive, 15
`real10` directive, 15
real values as parameters, 244
rearranging bytes in an XMM/YMM register, 625
rearranging expressions
  in `if` statements to improve performance, 406
  to make them more efficient, 406
record, 197
  declarations, 198
  field access, 199
  field alignment, 204
record/struct field access via pointer, 200
recursion, 271
recursively converting numbers to strings, 500
reference parameters, 241, 256
register
  8-bit, 10
  16-bit, 10
  32-bit, 10
  64-bit, 10
  addressing modes, 122
  aliasing, 10, 623
  as a procedure parameters, 243
  comparison to zero, 298
  FPU, 317
  indirect addressing mode, 124
  indirect jump instruction, 383
  overlaying, 10
  preservation, 137, 220, 442
    callee, 222
    caller, 222
  usage in loops, 442
  usage in string instructions, 826
  usage in the Microsoft ABI, 38
remainder
  floating point, 348
  signed integer, 407
removing unwanted data from the stack, 140
`ren`/`rename` commands, 933
`repeat..until` loop, 433, 434
`repe` prefix on `cmpsb`, `cmpsw`, `cmpsd`, and `cmpsq` instructions, 827
repetitive compilation, 756
`repne` prefix on `cmpsb`, `cmpsw`, `cmpsd`, and `cmpsq` instructions, 827
`rep` prefix on `movsb`, `movsw`, `movsd`, and `movsq` instructions, 826
`rep`/`repe`/`repz` and `repnz`/`repne` string instruction prefixes, 826
required macro parameters, 766
restrictions in simple switch statement implementations, 414
`ret` instruction, 22, 218
return address, 218
returning a result to a C++ program from an assembly language function, 30
reverse
  division (floating-point), 343
  Polish notation (RPN), 364
  subtraction (floating-point), 334
reversing bits in a bit string, 739
RFLAGS register, 12, 140
right
  rotates, 78
  shift operation, 76, 77
  shifts, 75
right associative operators, 304
RIP-relative addressing mode, 123
`rol` instruction, 78
`ror` instruction, 78
rotate
  left, 77
  operations, 74
  right, 77
rounding
  control (FPU), 319
  control (SSE), 370
  floating-point numbers, 349
  floating-point value to an integer, 349
round-up and round-down options during floating-point computations, 319
row-major array access for three-dimensional arrays, 191
row-major ordering, 190

RPN (reverse Polish notation), 364. *See also* postfix notation
RSI register usage in string instructions, 826
rsqrtps instruction, 670
rsqrtss instruction, 372
rstrActRec (macro), 883
run of zeros bit string, 708
runtime
  language, 748
  memory organization, 106
runtime versus compile-time expressions, 155

S

sahf instruction, 86, 350
sar instruction, 77, 312
saturation addition (horizontal), 650, 652
saturation (SSE/AVX/SIMD), 667
saving the machine state, 220
sbb instruction, 457, 716
sbyte directive, 15
scalar data types, 597
scaled-indexed addressing mode, 126
scaling factor, 126
scas instruction, 835
scope, 378, 850
  of a local variable, 234
sdword directive, 15
searching
  for a bit, 736
  for a bit pattern, 743
  for a substring within another string in MASM textual constants, 751
  for the first (or last) set bit, 737
section location counter, 154
segment
  alignment option, 605
  alignment (powers of 2), 605
  class argument, 605
  declarations, 604
  directive, 604
  directive align option (for 32-byte alignment), 606
  faults, 107
  faults on unaligned memory accesses (SSE/AVX), 606
  names, 604
  registers, 10
separate assembly, 854
separate compilation, 847, 854
setae instruction, 296
seta instruction, 296
setbe instruction, 296
setb instruction, 296
setcc instructions, 295
setc instruction, 295, 716
sete instruction, 296
setge instruction, 297
setg instruction, 296
setl instruction, 297
setnae instruction, 296
setna instruction, 296
setnbe instruction, 296
setnb instruction, 296
setnc instruction, 295, 716
setne instruction, 296
setnge instruction, 297
setng instruction, 297
setnle instruction, 296
setnl instruction, 297
setno instruction, 295
setnp instruction, 295
setns instruction, 295
setnz instruction, 295, 298
seto instruction, 295
set on condition instructions, 295
setpe instruction, 295
setp instruction, 295
setpo instruction, 295
sets instruction, 295
setting bits, 708
setz instruction, 295, 298
shadow storage (for parameters), 255, 264
shift
  arithmetic right operation, 77
  left operation, 75
  operations, 74
  operations (SSE/AVX), 647
  right operation, 76
shift and rotate instructions, 709, 716
shld instruction, 482
shl instruction, 75, 310
short-circuit
  Boolean evaluation, 401

  - short-circuit versus complete Boolean evaluation, 403
- `shrd` instruction, 482
- `shr` instruction, 76, 312
- shuffle instructions, 625
- `shufpd` instruction, 630
- `shufps` instruction, 630
- side effects, 403
- sign
  - bit, 62
  - contraction, 67
  - extension, 67, 292
  - extension prior to division, 305
- sign and zero flag settings after `mul` and `imul` instructions, 291
- signed
  - comparison flag settings, 294
  - comparisons, 296
  - decimal input (extended-precision), 569
  - decimal output (extended-precision), 513
  - division, 292
  - integer remainder/modulo, 407
  - integer-to-string conversion, 507
  - multiplication, 148, 289, 291, 461
  - numbers, 62
- signed and unsigned numbers, 62
- sign extension (SIMD/SSE/AVX), 666
- sign flag, 12, 293
  - setting after an arithmetic operation, 71
- sign flag and the `and`, `or`, and `xor` instructions, 712
- significant digits, 314
- sign transfer, 659
- SIMD (single instruction, multiple data), 11, 55, 595
  - arithmetic/logical operations, 644
  - bitwise instructions, 645
  - comparison instructions (floating-point), 671
  - comparison results (processing multiple comparisons), 663
  - floating-point arithmetic operations, 668
  - floating-point conversions, 679
  - integer absolute value, 659
  - integer addition, 648
  - integer arithmetic instructions, 648
  - integer average instructions, 657
  - integer comparison instructions, 660
  - integer conversions, 664
  - integer minimum and maximum, 657
  - integer multiplication, 654
  - integer sign-transfer instructions, 659
  - integer subtraction, 653
  - memory alignment requirements, 606
  - programming model, 596
  - saturation, 667
  - SIMD string instructions, 838
  - SIMD zero-extension instructions, 665
- simple assignments (conversion to assembly language), 299
- simulating `div`, 312
- sine, 361
- single-instruction, multiple-data (SIMD) instructions. *See* SIMD
- single-instruction, single-data (SISD) instructions. *See* SISD
- single-precision floating-point format, 87
- single-precision (floating-point) lanes, 598
- single-precision vector types, 597
- SI register, 10
- SISD (single instruction, single data), 595
- `sizeof` function (applied to UNIONs), 207
- `sizeof` operator, 154
- `size` operator, 154
- `sizestr` directive, 752
- software configuration via conditional compilation, 754
- sorting, 185
  - bubble sort, 185
  - quicksort, 272
- special-purpose application-accessible registers, 10
- special-purpose kernel-mode registers, 10

specifying a variable name and type without allocating storage, 114
SP register, 10
sqrtpd instruction, 670
sqrtps instruction, 670
sqrtsd instruction, 372
sqrtss instruction, 372
square root, 327, 347
sqword directive, 15
SSE (Streaming SIMD Extensions), 596, 624
    aligned data movement instructions, 610
    denormal exception flag (DE), 369
    denormal mask (DM), 370
    denormals are zero (DAZ), 370
    divide-by-zero mask (ZM), 370
    floating-point arithmetic (SIMD), 668
    floating-point conversions, 679
    flush to zero (FZ), 370
    instruction operands, 606
    invalid operation mask (IM), 370
    memory alignment requirements, 606
    overflow exception flag (OE), 369
    overflow exception flag (UE), 369
    overflow mask (OM), 370
    packed byte data types, 597
    packed dword data types, 598
    packed qword data types, 598
    packed word data types, 597
    precision exception flag (PE), 369
    precision mask (PM), 370
    programming model, 596
    rounding control, 370
    sign extension, 666
    string instructions, 838
    unaligned memory access, 606, 612
    underflow mask (UM), 370
    zero exception flag (ZE), 369
    zero extension, 665
SSE2, SSE3, SSSE3, SSE4, SSE4.1, SSE4.2, 596
SSE/AVX comparison synonyms, 673
SSE/SSE2 instruction set, 11
ST0, 318
ST1, 318
stack, 134
stack fault flag (FPU), 322
stack manipulation by procedure calls, 224
stack operations
    pop, 135, 222
    popf, 140
    popfd, 140
    push, 134, 222
    pushf, 140
    pushfd, 140
    pushw, 134
stack pointer register, 13
stack segment, 134
stack variable address alignment, 607
standard entry sequence (to a procedure), 231
standard exit sequence (from a procedure), 233
standard input redirection, 927
standard macro parameter expansion, 762
standard macros, 760
standard output redirection, 926
state machine, 424
statement labels, 378
statements
    break, 438
    case, 396, 410
    conditional, 396
    continue, 438
    else, 397
    for, 437
    if, 396
    repeat..until, 433
    while, 433
state variable, 424
static variable declaration section, 108
status register (FPU), 321, 364
status word, 350, 364
stc instruction, 716
STDCALL calling convention, 263
stdin_getc function, 892
stdin_read function, 891
std instruction, 86
sti instruction, 86
stmxcsr instruction, 370

store data from an SSE/AVX register into memory, 610
storing AH register into flags, 86, 350
storing single-precision vectors from SSE/AVX registers to memory, 612
storing the FPU control word, 321
storing the FPU status word, 321, 350, 364
stos instruction, 835
streaming data types, 596
streaming SIMD extensions. *See* SSE
strength-reduction optimizations, 311
strfill procedure, 244
strings, 174
  comparisons, 825
  descriptors, 176
  equality test for macro/text arguments, 767
  instruction performance, 837
  instructions, 825, 836
  length, 174
  length calculated at assembly time, 176
  length operator in MASM textual constants, 752
  length-prefixed, 175
  SSE instructions, 838
  zero-terminated, 174
string-to-decimal conversion (unsigned), 563
string-to-floating-point conversion, 570
string-to-integer conversion, 546
string-to-numeric conversion (hexadecimal), 556
string-to-numeric conversions, 546
string-to-numeric conversion (signed, extended-precision), 569
strtoh128 function, 561
strtoh function, 557
strtoi function, 550
strToR10 function, 573
strtou128 function, 567
strtou function, 548, 564
struct arrays, 203
struct assembler directive, 198
struct declarations, 198
struct directive, 198
struct/record field access via pointer, 199
structs, 197
  nested, 200
structure field access, 199
structure field initialization, 200
sub instruction, 21
subpd instruction, 669
subps instruction, 669
subregisters, 623
subsd instruction, 371
subss instruction, 371
substr directive, 752
substring operator (MASM text strings), 752
substring search in MASM textual constants, 751
subtraction, 457, 716
  floating-point, 334
subtract with borrow, 457, 716
swapping bytes in a multi-byte object, 116
swapping registers on the FPU stack, 327
switch statement, 410
sword directive, 15
synthesizing
  break statements in assembly language, 438
  continue statements in assembly language, 439
  forever..endfor loops in assembly language, 436
  for statements in assembly language, 437
  repeat..until loops in assembly language, 434
  while loops in assembly language, 433
system bus, 9

T

tables and table lookups, 583
  table lookup computations, 584
  table lookup (hexadecimal-to-string conversion), 497
tag field, 209
taking the address of a statement label, 378
tangent, 361

tbyte directive, 15
tbyte values (BCD), 488
temporary values in an expression, 307
temporary variables, 306
test for zero (floating-point), 360
testing a floating-point operand for zero, 322, 360
testing bits, 708
testing to see if a macro argument is the empty string, 767
testing two text objects for equality, 767
test instruction, 297, 709
text delimiters, 151
textequ directive, 151
this operator, 154
three-dimensional array element access (row-major), 191
time command (CLI), 933
top of stack pointer (FPU), 324
trampoline, 393
transcendental function instructions, 361
translate arithmetic expressions into assembly language, 287
translate instruction, 585
tricky programming, 310
true (representation), 308
truncation during FPU calculations, 319
truth table, 55
try..catch statement (C++), 30
two-dimensional row-major ordered array formula (for accessing array elements), 191
two's complement
- numbering system, 54
- numeric representation, 62
- operation, 63

type checking, 20
- coercion, 157

type coercion, 157, 159
type declaration section, 156
typedef directive, 156
type operator, 159

U

unaligned loads (to XMM/YMM registers), 622
unaligned SSE/AVX data movements, 612
unaligned SSE/AVX memory accesses, 606
unary operator (conversion to assembly language), 301
unconditional jump instruction, 69
underflow, 316
underflow exception flag (UE, SSE), 369
underflow exception (FPU), 321
underflow mask (UM, SSE), 370
Unicode, 54, 96
- BMP (Basic Multilingual Plane), 97
- UTF-8 encoding, 98
- UTF-16 encoding, 98
- UTF-32 encoding, 98
- code planes, 97
- code points, 96
- encodings, 97
- multilingual planes, 97

uninitialized pointers, 168
unions, 206
- accessing fields of a union, 206
- anonymous, 208
- definition, 206
- syntax (declaration), 206

unordered comparisons, 90, 360, 373, 673
- floating-point, 357

unpacking bit strings, 717
unpack instructions, 625
unpckhpd instruction, 633
unpckhps instruction, 633
unpcklpd instruction, 633
unpcklps instruction, 633
unraveling loops, 447
unrolling loops, 448
unsigned
- comparisons, 296
- decimal input (extended-precision), 566
- decimal output, 500
- division, 291
- integer-to-string conversion (extended-precision), 508
- multiplication, 289, 461
- numbers, 62
- string-to-decimal conversion, 563

untyped reference parameters, 284
using echo to display equate values, 751

uSize function, 514
UTF-8 encoding, 98
UTF-16 encoding (Unicode), 98
UTF-32 encoding (Unicode), 98
utoStrSize function, 517

V

vaddpd instruction, 669
vaddps instruction, 669
value parameters, 241, 253
vandnpd instruction, 645
vandpd instruction, 645
variable-length parameters, 248
variable names, 14
variables in MASM, 14
variant objects, 209
variant types, 209
vcmppd instruction, 671, 674
vcmpps instruction, 671, 674
vcvtdq2pd instruction, 679
vcvtdq2ps instruction, 679
vcvtpd2dq instruction, 679
vcvtpd2ps instruction, 680
vcvtps2dq instruction, 680
vcvtps2pd instruction, 680
vcvttpd2dq instruction, 680
vcvttps2dq instruction, 680
vdivpd instruction, 670
vdivps instruction, 670
vector
  absolute value (integer), 659
  addition, 648
  data types, 597
  floating-point arithmetic, 668
  instructions, 595
  integer comparisons, 660
  integer multiplication, 654
  memory operands, 606
  operands for SSE/AVX instructions, 606
  shifts, 647
  sign extension, 666
  sign transfer, 659
  (SIMD) integer comparison for less than, 662
  zero extension, 665
vertical addition, 649
vextractps instruction, 643
vhaddpd instruction, 671
vhaddps instruction, 671
vhsubpd instruction, 671
vhsubps instruction, 671
vinsertps instruction, 643
vlddqu instruction, 622
vmaxpd instruction, 670
vmaxps instruction, 670
vminpd instruction, 670
vminps instruction, 670
vmovapd instruction, 610
vmovapd operands (MASM), 611
vmovaps instruction, 610
vmovaps operands (MASM), 611
vmovddup instruction, 621
vmovd instruction, 609
vmovdqa instruction, 610
vmovdqa operands (MASM), 611
vmovdqu instruction, 612
vmovhlps instruction, 619
vmovhpd instruction, 618
vmovhps instruction, 618
vmovlhps instruction, 619
vmovlpd instruction, 615
vmovlps instruction, 615
vmovmskpd instruction, 676
vmovmskps instruction, 676
vmovq instruction, 609
vmovshdup instruction, 620
vmovsldup instruction, 620
vmovupd instruction, 612
vmovups instruction, 612
vmulpd instruction, 670
vmulps instruction, 670
volatile registers, 265
  Microsoft ABI, 38
von Neumann architecture, 9
vorpd instruction, 645
vpabsb instruction, 659
vpabsd instruction, 659
vpabsw instruction, 659
vpackssdw instruction, 667
vpacksswb instruction, 667
vpackusdw instruction, 667
vpackuswb instruction, 667
vpaddb instruction, 649
vpaddd instruction, 649
vpaddq instruction, 649

vpaddw instruction, 648–649
vpavgb instruction, 657
vpavgw instruction, 657
vpclmulqdq instruction, 656
vpcmpeqb instruction, 661
vpcmpeqd instruction, 661
vpcmpeqq instruction, 661
vpcmpeqw instruction, 661
vpcmpgtb instruction, 661
vpcmpgtd instruction, 661
vpcmpgtq instruction, 661
vpcmpgtw instruction, 661
vpextrb instruction, 642
vpextrd instruction, 642
vpextrq instruction, 642
vpextrw instruction, 642
vphaddd instruction, 650
vphaddw instruction, 650
vpinsrb instruction, 642
vpinsrd instruction, 643
vpinsrq instruction, 643
vpinsrw instruction, 643
vpmaxsb instruction, 657
vpmaxsd instruction, 658
vpmaxsq instruction, 658
vpmaxsw instruction, 657
vpmaxub instruction, 658
vpmaxud instruction, 658
vpmaxuq instruction, 658
vpmaxuw instruction, 658
vpminsb instruction, 658
vpminsd instruction, 658
vpminsw instruction, 658
vpminub instruction, 658
vpminud instruction, 658
vpminuq instruction, 658
vpminuw instruction, 658
vpmovmskb instruction, 662
vpmovsxbd instruction, 666
vpmovsxbq instruction, 666
vpmovsxbw instruction, 666
vpmovsxdq instruction, 666
vpmovsxwd instruction, 666
vpmovsxwq instruction, 666
vpmovzxbd instruction, 665
vpmovzxbq instruction, 665
vpmovzxbw instruction, 665
vpmovzxdq instruction, 665
vpmovzxwd instruction, 665
vpmovzxwq instruction, 665
vpmuldq instruction, 656
vpmulld instruction, 655
vpmuludq instruction, 656
vpshufb instruction, 625
vpshufd instruction, 626
vpshufhw instruction, 628
vpshuflw instruction, 628
vpsignb instruction, 659
vpsignd instruction, 660
vpsignw instruction, 659
vpslldq instruction, 647
vpsllw instruction, 647
vpsrldq instruction, 647
vpsubd instruction, 653
vpsubq instruction, 653
vpsubsb instruction, 654
vpsubw instruction, 654
vptest instruction, 646
vpunpckhbw instruction, 640
vpunpckhdq instruction, 641
vpunpckhqdq instruction, 641
vpunpckhwd instruction, 640
vpunpcklbw instruction, 640
vpunpckldq instruction, 640
vpunpcklqdq instruction, 641
vpunpcklwd instruction, 640
vrsqrtps instruction, 670
vshufpd instruction, 632
vshufps instruction, 632
vsqrtpd instruction, 670
vsqrtps instruction, 670
vsubpd instruction, 670
vsubps instruction, 670
vunpckhpd instruction, 633
vunpckhps instruction, 633
vunpcklpd instruction, 633
vunpcklps instruction, 633
vxorpd instruction, 645

W

while directive, 756
while..endm compile-time statement, 756
while statement, 433
Win32 API, 876
Windows command line, xxx

word, 51, 53
- 16-bit variables, 54
- alignment in a segment, 605
- directive, 15, 54
- strings, 825
- vectors (packed words), 597
- word-sized lanes, 598

wrapper code, 882
WriteFile (Win32 API function), 875
write function, 884
wtoStr (word to string) function, 493

X

xchg instruction, 116
xlat instruction, 584
XMM registers, 11
xor instruction, 58, 309, 709, 712
XOR operation, 55, 57
xorpd instruction, 645

Y

Y2K, 85
YMM registers, 11

Z

zero and sign flag settings after mul and imul, 291
zero-divide exception (FPU), 320
zero exception flag (ZE, SSE), 369
zero-extension, 292
zero-extension (SIMD), 665
zero flag, 12, 293, 713
- setting after a multiprecision OR, 479
- setting after an arithmetic operation, 71
- settings after mul and imul instructions, 291

zero-terminated strings, 174