

Index

A

Absolute address, 274
add, 29, 30, 282, 334
Addition, 314
Addition instructions, 31
ADDR operator, 18–20
Aliasing, 160
American Standard Code for Information Interchange (ASCII), 311, 312
and, 323
And operator (&&), 62
Arithmetic instructions, 29–46
Arithmetic shift, 105–108
.asm, 291–292
Array of strings, 204–206
Arrays, 159–162
 64-bit arrays, 258–260
 floating-point arrays, 232–233
Assembler, 1
Assembly language, 1–2

B

Bankers rounding, 220
Binary Coded Decimal (BCD), 309
Binary numbers, 293–296
Bit, 294
Bit-bucket, 102
Bit manipulation
 set, 96, 300
 test, 96, 300
 toggle, 96, 300
Bit-wise, 99
Branch instructions, 47, 51, 61, 66, 76
break (C instruction), 58
byte directive, 5, 11

C

call, 123, 324
Carry flag, 102

Case structure, 58–59

cbw, 35
cdq, 35, 324
Central Processing Unit (CPU), 3, 7
Characters, 311
cld, 198, 324
cmp, 51–53, 324
cmpsb, 198, 204–208, 324
.code directive, 3
Comments, 4–6
Comparisons, 49
Conditional assembly, 138–141
Conditional assembly directives, 138
 else, 139
 endif, 139
 EQ, GE, GT, LT, LE, NE, 138
 if, 138, 142–144
 ifb, 138–141
 ifdif, 139
 ifdif, 138, 142–143
 ifdn, 138
 ifdni, 138, 142–144
 ifnb, 138
Conditional jump, 51, 57, 66
C programming language
 break, 58
 for, 78
 if, 48–50
 printf, 18–24
 scanf, 25
 switch, 58–59
 while, 73
cqo, 251
cwd, 35

D

.data directive, 3
dec, 37, 264–265, 324
Decimal numbers, 296–298

De Morgan's rules, 71
 Direction flag, 49, 198
 .686 directive, 3
 Directives. *See individual listings*
 div, 34, 324
 Division instructions, 32, 34
 Do-while loops, 76
 dup operator, 161
 dword directive, 5
 Dynamic, 169

E

eflags register, 9, 49, 66–67
 .else directive, 53
 .elseif directive, 54
 .endif directives, 54
 endm directive, 132
 endp directive, 4, 125
 .endw directive, 74
 EOD loop, 82
 Errors, 13
 Exclusive or, 96, 99, 298–300
 Execution errors, 138

F

fadd, 213
 fcomi, 226–227
 fcomip, 226–228, 230
 fdiv, 215
 fiadd, 221
 fld, 219
 First In First Out (FIFO), 175
 fistp, 219–221
 fisttp, 221
 Fixed iteration loop, 78–81
 Flags, 50
 fld, 211
 Floating-point instructions, 210–221
 Floating-point I/O, 221–226
 Floating-point numbers, 307–311
 fmul, 214
 For loop, 78
 fst, 212
 fstp, 212
 fsub, 214
 fsubr, 214

H

Hello world program, 17
 Hexadecimal numbers, 296–298
 High-level languages, 6

I

idiv, 34, 326
 IEEE standard, 309–310
 .if directive, 51–52
 If statements, 59, 64
 Immediate data, 6–7
 imul, 32, 327
 inc, 37–38, 265–267, 327
 Inclusive or, 119, 298–300, 329
 Inline assembly, 13–14, 225–226
 Input, 23–24
 64-bit input, 246
 Instructions. *See individual listings*
 Instruction timings, 275–276
 Integers, 19–20
 INVOKE directive, 18–21
 Iteration structures, 73–90, 229–230

J

ja, 60, 327
 jae, 60, 327
 jb, 60, 327
 jbe, 60, 327
 jc, 67, 328
 je, 51, 327
 jecxz, 79, 327
 jg, 51, 327
 jge, 51, 327
 jl, 51, 327
 jle, 51, 327
 jmp, 53, 54, 274–275, 327
 jna, 60, 327
 jnae, 60, 327
 jnb, 60, 327
 jnbe, 60, 327
 jnc, 67, 328
 jne, 51, 327
 jng, 51, 327
 jnge, 51, 327
 jnl, 51, 327
 jnle, 51, 327
 jno, 67, 328
 jnp, 67, 328
 jns, 67, 328
 jnz, 66, 328
 jo, 66, 328
 jp, 66, 328
 js, 66, 328
 Jump instructions, 51, 60, 66, 327–328
 jz, 66, 328

L

Label field, 4
Last In First Out (LIFO), 111
lea, 272–273, 328
Lengthof operator, 174–175
Little endian order, 286
Load operation, 10
lods, 191, 194–195, 328
Logic, 298–300
Logical shifts, 100–104
Logic errors, 125
Logic instructions, 96–100
loop, 79–80, 328
Loop instructions, 78–87
Low-level languages, 2
.lst file, 52, 292

M

Machine language, 2, 266
macro directive, 132
Macros, 132–135, 138
 64-bit, 256–258
 definition, 134
 expansion, 134
 invocation, 133
 parameters, 135–136
Masks, 96, 299
Mnemonics, 2
model directive, 3
mov, 6–7, 14–16, 266–271, 328
movsb, 192–193, 328
movsxd, 239
mul, 32, 329
Multiplication instructions, 32–36

N

neg, 38, 39, 328
Nested if statements, 54–58
Nested loops, 85–87
nop, 52, 329
Normalization, 310
not, 97, 329
Not operator (!), 60
Number systems, 293–296

O

offset operator, 168–169, 272–274
One's complement, 303–304
Opcode field, 4
Operand field, 4
Operator precedence, 39–41
Operators. *See individual listings*
Or, 97–100
Or operator (||), 61

Output, 19–21

 64-bit output, 241–245
 Overflow flag, 50, 305–306

P

Parity flag, 50
pop, 111–112, 329
popad, 131, 329
Postfix, 215–219
Post-test loops, 76–78
Precedence, 41–44
Pre-test loops, 73–76
printf (C instruction), 20–24
proc directive, 3, 124
Procedures, 123–127
PROTO directive, 20–23
push, 111–113, 330
pushad, 130–131, 330

Q

Queues, 175–179

R

Random Access Memory (RAM), 7
Registers, 7–8
 eax, 8–11
 ebp, 9
 ebx, 8, 164–165
 ecx, 8–9, 78–81
 edi, 9, 168–173, 190–194
 edx, 8–9, 33–35
 eflags, 9, 49, 66
 eip, 9
 esi, 9, 168–173, 190–191, 195
 esp, 9
 r8–r15, 241
 rax, 239
 rbx, 238
 rcx, 238
 rdx, 238
rep, 193, 330
repe, 193, 330
 .repeat directive, 78–80
 .repeat-.untilcxz directives, 78–81
Repeat–until instructions, 76–77
repne, 194, 330
ret, 4, 124–129, 330
rol, 109–110, 330
ror, 109, 330
Rotate instructions, 108–110, 330

S

sal, 105–107, 331
sar, 105–108, 331

scanf (C instruction), 21–24
scasb, 191, 194–196, 331
sdword directive, 11
Search (sequential), 166
Selection sort, 180–183
Selection structures, 47–70
 floating-point, 226–229
Sentinel-controlled loop, 83
Sequential search, 166
Shift instructions, 100–104, 331
shl, 102–103, 331
shr, 102, 104, 331
Sign bit, 305
Signed numbers, 304–307
Sign flag, 50
Sign magnitude, 304
SIGN? operator, 50
sizeof operator, 173–175
Sort (selection), 180–185
sqword directive, 242
Stacks, 95–121
 .stack directive, 3, 111
Static, 169
std, 192, 331
Store operation, 7
stosb, 194–196, 331
Strings, 191–193
sub, 31, 271–272, 332
Subprograms, 123
Subtraction instructions, 31
Swap, 114

switch (C instruction), 58–59
sword directive, 5
Syntax errors, 13

T

test, 104–105, 332
Two's complement, 303

U

Unary operations, 36–39
Unconditional jump, 54
Unsigned numbers, 300–302
 .untilcxz directive, 78–80, 82
 .until directive, 76–77

V

Variables, 1–6

W

 .while directive, 74–75
While loops, 73–74
word directive, 5

X

xchg, 113–115, 332
xor, 96, 99, 332

Z

Zero flag, 50
ZERO? operator, 50