<B>=<b> assembler directive, 260
68000, 269
68XXX, 697
80-column color text mode, 284
8-bit adapters, 263
8-bit bus cycle eater, 75-81,83,93,110,113,128,129,699,701,704,705
 interaction with DRAM refresh cycle eater, 115
 location, 78,83
 penalty for word-sized accesses, 79,80,84,110,321,376,662
 penalty for word-sized accesses doubles when memory is destination, 321
 prefetch queue cycle eater as manifestation of, 83,84
 what to do about, 80-83
 with multiple word-sized accesses, 79
8-bit data bus, 69,71,76,78,263,525,697
 address lines of, 117,121
80186, 71,698
80188, 71,698
80286, 71,142,156,160,169,184,211,263,266,306,390,537,696-707,709-722,727
 bug in <B>popf<b>, 145,721-723,725-727
 handles pushing SP differently than 8088, 141
 imbalance between processing speed and memory speed,72
80287, 728,729
80386, 30,63,71,108,142,156,160,169,185,211,266,269,306,390,696-705,707,708,
 711-718,720,721
 imbalance between processing speed and memory speed,72
80386SX, 63,71,698,699
80387, 728,729
80486, 696
8080, 69,71,134,155,266-272,285,289,290,294,295
8080 legacy, 266,269
8080-specific instructions, 270
8085, 71,266,267
8086, 69,71,72,76,78-80,263,266,390,525,616,696-699,731
8087, 291,728-731
 can coprocess with the 8088, 730,731
 ensuring synchronization with 8088, 730,731
 keep variables in data registers as much as possible, 729,730
8088, 5,6,8,11,13,14,25,26,63,65,69-73,75,76,78-80,117,118,123,169,266-272,
 274,281,289-291,295,297,304-306,309,312,316,326,332,344,346,390,396,
 508,509,514,517,519,524,535,560,592,616,694,696-701,703-705,708,
 712-721,727-731,733

16-bit processor that often performs like an 8-bit processor, 76
16-bit processor with an 8-bit data bus, 133,134
architecture, 266,267,295
bug in disabling of interrupts while loading segment registers, 144
can address no more than four 64-Kb blocks at any one time, 134
capable of addressing 1 Mb of memory, 133,134,152,154-156,160
clock speed in PC is slow, 134
compared to 8086, 69,71,72,78,80,84
data lines of, 117,121 derivation from 8086, 69,72,134
display memory access, restricted, 102,103
effect on performance, 62
externally an 8-bit processor, 67,76,78
fairly low-performance processor that's hard to program, 134
handles pushing SP differently than 80286, 141
hodgepodge of a processor, 134
influenced by 8080, 134
instruction execution and memory access relatively slow, 134,158,346
instruction set, <see instruction set>
internal parallelism (internal coprocessing), 65-68,85,92,119,122,124,125
internally a 16-bit processor, 67,75-78,83
memory architecture, 152,154-156,167
minimum memory access time, 82,130
mismatched internal and external bus sizes, 69,71,72,76,78,80,701
programming interface, 63,133
register space is limited, 134
resources, 133,135
rests on PC's hardware, 64,65,73
sets tone for PC, 133
speed of accessing the second byte of a word-sized operand, 82,219,384
two processors in one, 65,66
8237 DMA controller chip
documentation, 52
DRAM refresh, 95
8253 timer chip, 8,14,23,24,25,28,37
accuracy, 30
configuration in the PC, 24
counting rate, 30
counting resolution, 30
documentation, 52
in PS/2 computers, 36,37
interrupts, 25,27
latching a timer count, 28,36

mode 2, divide-by-N mode, 23,25,26
mode 3, square wave mode, 25,26
modes, 25
programmability, 23-26,36
stopping a timer, 25,28,36
8259 interrupt controller chip, 25
configuration in the PC, 26
documentation, 52
edge triggering, 25,26
timer interrupt, 27
<B>aaa<b> instruction, 139,147,333,335,339-342
<B>aad<b> instruction, 139,333,335-337
<B>aam<b> instruction, 139,333,335-337
<B>aas<b> instruction, 139,147,333,335,339,341,342
accumulator, 138,139,267,270,275-277,290,298,302,351,359,412,436,444
and <B>in<b> and <B>out<b>, 138,261
multiplication and division, 138 sign extension, 139,312
string instructions, 139
accumulator-specific, 278,284
AX-specific form of <B>xchg<b>, <see <B>xchg<b>>
direct-addressing form of <B>mov<b>, 219,231,271-280,282,298,299,302,343, 344,648,670
immediate-addressing form of <B>test<b>, 286
immediate-addressing instructions, 231,271,281-283,285,342,344
instructions, 139,270,284,285
<B>adc<b> instruction, 281-283,309,320,343,560,563
<B>add<b> instruction, 147,222,281-283,294,305,306,309,320,334,342,343,643, 712
addressing modes, <see memory addressing modes>
AF (auxiliary carry flag), 144,145,147,329,330,334,340-342,354
AH register, 137,138,289-291,294,307,312,315,335,341,362,440,449
not an accumulator, 138,275,276
AL register, 137,138,159,261,298,301,307,312,315,322,334,335,337,339,341,342, 343,347,351,352,354,355,359,405,407,440,449
8-bit accumulator, 138,275-277
algorithms, 59,131,482,734
<B>and<b> instruction, 146,148,229,281-285,301,302,320,335,343,561,563,643
animation, 109,404,489-492,496-499,504,505,594,596,598,668,734
block-move animation, 497-499,504,505,596
<B>xor<b> animation, 489-492,496-499,504,505,594,594
ANSI.SYS device driver, poor performance of, 61
Apple II, 135,136,497

Apple Macintosh, 697
application programs, 60,63
<B>arg<b> directive, 244
arithmetic logic unit (ALU), 65,66
arrays, comparing, 359,371,457,458,462,482,483,485
arrays, copying, 351,404,406,407
arrays, searching, 354,400,430,432,433,439,454,471-473,479,482,572,601,656
arrays, setting, 375,376,590,602,608
ASCII, 246,335,336,338,340,672,673
ASCII adjust, 333-335,340,341
ASCII arithmetic, 139,147,333,339,340
assembler code
 advantages of mixing with high level languages, 241
 and stack frames, 240-242
 no complete solution to writing, 114
 versus compiled code, 10,174,240
assembler programmer
 can think more flexibly than high level language, 174,400,533
 knows exactly what code must do, 164,166,167
 knows more about what code must do than high level language, 174,542
assembler programming, 560,589
 high-performance must be intuitive art, 131,505,560,589,604,649,694
 no hard and fast rules, 316,560
 science of, 113
assemblers, 645,646,693 and precalculation, 249,251,258,260
 manuals, importance of reading, 192
 relationship to machine language, 7,8
assembly language, 4-9,13,166,589,623,624,650,731,733
 bad code, 10
 complete control over segments, 163,623,624
 difficulty of learning, 7,8
 excellent data-definition capabilities, 258,260
 framework, 11
 good at handling multiple segments & larger than 64 KB blocks, 155,163,164,
 167,173,179-181,191
 good code, 10
 hardware, 8
 high-performance programming, 64,73,110,505,510,530,531,560,589,593,597,
 601-604,611,649
 lack of transformation loss, 7
 low-level control, 8
 need for forethought and planning, 10

objectives of good programming, 9
objectives of high-performance programming, 9
optimization, 10
performance, 10
porting 8080 code, 267,269,270,272
potential of, 5
produces best code, 10,241
programmers, 8,58
programmers and responsibility for code quality, 8
superior information and adaptability, 10
traditional programming model, 59,60
versus high level languages, 10,163,164,167,173,174,191,240,246,258,400,488,
533,542,560,729,730
when to use, 10
<B>assume<b> directive, 168,196,393
and segment override prefixes, 196,197
assume nothing, 13,250,344
assuming and related problems, 13
automatic (dynamic) variables, 233,240
AUX device driver for reporting timer results, 29
auxiliary carry flag, <see AF>
AX register, 137,138,261,267,275,298,307,312,314,315,335,337,339,343,347,351,
352,354,355,359,405,407,449,680
16-bit accumulator, 138
accessible as AH and AL, 138
barrel shifter of the 80386, 720
base addressing, 207,209,211,213,214,218,219,221
base addressing component, 200,202,207
base+displacement, 214,220,255
base+index addressing, 213,214,255
base+index+displacement addressing, 200,202,207,214,718
based relative addressing, 206
BASIC, 166 BC register pair of 8080, 267
BCD arithmetic, 139,147,334,335
beneath the programming interface, importance of understanding, 75,701
best code, rarely such a beast, 174,257
BH register, 137,139
not used as memory-addressing register, 139
BIOS, 8,25,28,35,59-61,75,133,152,194,363,637,653,712,722
adapter-select bits, 284
as an assembly-language program, 61
documentation, 52

equipment flag, 284
inadequacy of some services, 61
interrupt 17h, printer control, 29
time of day count, <see time of day count>
bit doubling, 250-254,256,257,332
BL register, 137,139
not used as memory-addressing register, 139
blocks of data, comparing, 359,457,458
blocks of data, copying, 351,413,414,416-420,424
blocks of data, setting, 365-367,384,389
Borland, 645
<B>bound<b> instruction, 715
BP register, 137,140,159,193,239,245,643
base addressing component, 207
no addressing mode that uses only BP, 207,643
normally addresses the stack segment, 140,143,192
specializes as stack frame-addressing register, 140,208,239,240,245
used as memory-addressing register, 140,200,208,223,233,239,245,246,309
branched-to in-line code, 598,599,601,602,604,607,608,679,688
branched-to partial in-line code, 605,606,608
branches, 150,192,297,346,366,471,508,509,512,517,519,521-525,527-531,533,
534,537-540,542,543,545,555-557,560-562,564,565,569,571,572,574,577,
579,581,582,586,587,589-594,596,599,601,602,604-608,610,611,615,
619-621,627,631,632,636,637,649-653,657,658,661,662,668,670,672,674,
678,679,686,688,693,701,702,705,708,710,711,717,723
absolute, 510,511
desirability of eliminating, 251,368,440,444,479,499,508,509,514,530,531,
555,560,574,577,579,582,586,592-594,596,601,602,611,619-621,658,699,
702,705,717
empty the prefetch queue, 66,67,86,294,518,519,522-525,527,528,530,531,539,
556,582,589,590,601,616,619,620,636,637,652,701,702,704,705
first prefetch after is included in execution time, 523-525,527,530
not-branching, <see not-branching>
relative, 510-515,565
bus, 62,63,65,68,70,71
bug in splitting 16-bit accesses to 8-bit adapters in certain computers, 263
effect on performance, 62,64,72,77-80,83,84
maximum data transfer rate, 67,70,73,77,78,84
sizes in 8086 family, 69
splits 16-bit accesses to 8-bit adapters with some processors, 263
transparency to programs, 62 bus interface unit (BIU), 65-72,78,158,520,524-529,616,699,701
bottleneck, 70-72,78

can coprocess with EU, 66-68,85,92,119,122,124,519,520
converts 16-bit memory accesses into 8-bit accesses, 69,78,263
features contained within, 67
inability to fetch instruction bytes as fast as EU can execute them, 75,77, 84,85,130,699
interleaves instruction fetches with memory operand accesses, 125
minimum transfer time for bytes from the prefetch queue to the EU, 130
path for all memory and I/O accesses, including instruction fetches, 69
performs all memory and I/O accesses, 66,67
potentially greatest cycle eater of all, 69
prefetches, 66,67,85,519-521,523,616,699
provides instruction bytes for EU, 66,67
provides only 8-bit access to bus, 67,78
BX register, 137,139,159,267,643
accessible as BH and BL, 139
base addressing component, 207
normally addresses the data segment, 140
used as memory-addressing register, 139,200,208,223,246,309,316
<I>Byte<i> magazine, 52
byte-sized accesses, desirability of, 80,81,110
byte-sized versus word-sized operations, 80-83
byte-to-word conversion, 312,313
bytes versus cycles, 9,160,249,255,257,471,537,554,581,588-590,592,596, 602-604,610,617,637,649,651,674,694,711
C, 6,58,166,233,240,433,488,542,589,619,623,624,715
switch, 303,646,670
cache memory, 698,702,703
<B>call<b> instruction, 141-144,517,568,582,584,620,627,630,631,633,635-627, 670,712,726
call tables, <see jump tables>
calls, subroutine, 508,517,526,581,582,584,586,588,589,592,627,628,670
direct, 620,622,626,637
far, <see far calls>
indirect, 187,620,622,623,626,670,671
moving called code into loop, 582,584,586
near, <see near calls>
replacing with macros, 582,584,630
carry flag, <see CF>
carrying results along in a flag, 310-312
<B>cbw<b> instruction, 139,140,312,313,315
CF (carry flag), 144-147,309,310,322,323,327-330,332,341,354,488,547,560,562, 563,564,655,716

 can sometimes be used to avoid branching, 561-564
CGA, <see Color/Graphics Adapter>
CH register, 137,139
 not used as counting register, 139
"Chips in Transition," 721
CL register, 137,139,324-327,329,331
 used as counting register, 139 <B>clc<b> instruction, 146
<B>cld<b> instruction, 149,362,363,420
clearing the screen, 58
<B>cli<b> instruction, 149
<B>cmc<b> instruction, 146
<B>cmp<b> instruction, 277,278,281-283,301-304,321,343,354,356,359,436,444,
          540,643
<B>cmps<b> instruction, 140,359-361,368,371,372,377,381,391,393,457,462,466,
          471,482,483,485,499
 repeated, <see <B>repz cmps<b> and <B>repnz cmps<b>>
<B>cmpsb<b>, <see <B>cmps<b>>
<B>cmpsw<b>, <see <B>cmps<b>>
COBOL, 166,589
code
 as data that 8088 interprets as instructions, 65,68-70,73,615,617
 desirability of keeping short, <see shorter is better>
 equivalence with data, 65
 seemingly similar sequences can perform quite differently, 289
code maintenance, 9
code segment, <see CS register>
Codeview
 run from a remote terminal, 29
cold boot and system clock, 26,27
Color/Graphics Adapter (CGA), 14,106,262
 and display adapter cycle eater, 105
Commodore Amiga, 697
common exit code, <see sharing code>
compilers, 6,240-242,362,400,533,542,623
 as data transformation programs, 65,533
 information available to, 10,542
<I>Complete Turbo Pascal<i>, 735
conditional jumps, <see jumps, conditional>
constants, handling efficiently, 297,298
context, 58,85,91,92,116,129,130,174,517,542,554,563
coprocessors, 728,731
CP/M, 266

CS register, 137,142,158,159,169,192,193,396,509-512,514,619,620,625,626,636,
722,725
can't be used for temporary storage, 170
default access to, 142,159
must be loaded, together with IP, in a single instruction, 142,170
<B>CS:<b> segment override prefix, 193
cursor, 58
<B>cwd<b> instruction, 139,312-314,338
CX register, 137,139,267,363,364,371,373-375,377,378,382,452
accessible as CH and CL, 139
used as counting register, 139,316
cycle counting, 14
cycle eaters, 62-64,70,75-77,110,111,504,510,517,530,597,701,717
8-bit bus <see 8-bit bus cycle eater>
assembler programming can't be reduced to a science because of, 113 can't be eliminated, 111
cause performance to vary with context and time, 116,119,126,130,517,697
display adapter <see display adapter cycle eater>
don't change the effects of code, but do change performance, 129
dynamic RAM (DRAM) refresh, <see dynamic RAM refresh cycle eater>
foundation of Zen of assembler, 131
importance, 62,63,65
interaction between, 111,114-116,122
interaction with instruction execution, 117,123,128
live outside execution unit, 75
locations in PCs, 77
must understand to be able to interpret Zen timer results, 110
no compiler can deal with as well as good assembler programmer, 116
no need to understand all interactions between, 115
part of knowledge aspect of Zen of assembly language, 131
prefetch queue, <see prefetch queue cycle eater>
underlie programming interface, 131,701
wait state, <see wait state cycle eater>
cycles versus bytes, <see bytes versus cycles>
<B>daa<b> instruction, 139,147,333,334,341,344
<B>das<b> instruction, 139,147,333,334,341
data, equivalence with code, 65,615,617
data alignment, 696,699,705-711
code, 708-711,717,721
doubleword, 390,711
word, 390,525,705-707,709-711,717,721
data alignment cycle eater, 699,701,705-707,711,717
dealing with, 707,708,710,711,717,721

data bus, <see bus>
data segment, <see DS register>
Date command, 25,35
<B>db<b> directive, 168
DE register pair of 8080, 267
debugger, 28
 using to see actual code generated, 197,208,244,284,693,709
<B>dec<b> instruction, 5,80,81,147,303,304-309,318,342,590,597,662
 16-bit register-only form, 305-308,342,662
 doesn't affect CF, 309
decimal adjust, 333,334
development environment, 9
device drivers, 59-61
 extending DOS's capabilities, 61
 poor performance of, 61
DF (direction flag), 144,145,149,291,294,347,348,351-355,359,360,362,363,379,
 416-425,427,445
 must be in a known state before using string instructions, 362
DH register, 137
DI register, 137,140,149,159,351-356,358-360,380,643
 index addressing component, 207
 specializes as memory-addressing register for string instructions, 140,192,
 193,351,354,359 used as memory-addressing register, 140,200,208,223,246,309,391,396
direct addressing, 207,208,213,218,219,270,272,274-277,280,298,302,343,648
 accumulator-specific, <see accumulator-specific direct-addressing>
 always uses a 16-bit displacement, 208,271
direct indexed addressing, 206
direct memory access, <see DMA>
direction flag, <see DF>
directives, 168
disk, 60,64,133
disk-backup software, 61
displacements, branch/call/jump, 510,512-515,517,522,531,564,565,601,615,616,
 629,643,646,679-680,683,688
 forward references, 643-645
displacements, memory addressing, 187,200,202,205,208,213,255,271,643-645
 and addressing mode naming, 207
 can always reach anywhere in a segment, 224
 forward references, 643-645
 impact on size and performance, 205,206,209,214,218,264,643
 must be numbers or symbols equated to numbers in order to sign extend, 206
 negative, 224

sign extension, 205,206,231,643
use 1-byte displacements whenever possible, 214,243,263,601,628
display adapter, 8,29,60,64,133,136,669,713,714
8088 may get as little as 10 percent of display memory accesses, 102,103
allows same access speed regardless of speed of computer, 104,108,713,714
may stretch out access time based on pixel clock, 102,104
must guarantee video circuitry all needed accesses, 102
not very fast compared to system memory, 105
provides memory accesses only at certain intervals, 102,104,713,714
serves two masters, 101
slows up CPU, 101-105
usually 8-bit devices, 713
video circuitry may take up to 90 percent of display memory accesses, 102, 103
"Display Adapter Bottleneck" 110
display adapter cycle eater, 62,76,77,101,110,113,114,122,129,499,597,703,704, 712-714
can more than double the execution time of 8088 code, 107,110
effect on performance, 105-110
effect on performance varies with display mode, 105
effect on performance varies with type of adapter, 105
effects tend to even out over time, 129
impact is proportional to time spent accessing display memory, 107
impact varies with intensity of access to display memory, 108-110
impact varies with time, 114,128
in text mode, 106
interaction with prefetch queue cycle eater, 115
makes exclusive-or animation less-than-ideal, 109,499
matters most in high-resolution EGA and VGA graphics modes, 106
much worse on ATs and 80386 machines, 108
rarely a factor in text mode, 105 what to do about, 108,109
worst with EGA and VGA, 105,106
display memory, 107,152,401,489,491,496-498,594,597,713,714
access as little as possible, 108-110,594,714,717
access speed in the EGA's high-resolution graphics mode, 107
and word-sized string instructions, 390,391,392
avoid read/modify/write accesses to, 109
compared to system memory, 107
don't put code in, 105,115
perform multiple accesses to very rapidly, 109
performance of, 63
set all pixels in a byte at once for speed, 109

display memory wait states, 85,100-104,106,107,109
<B>div<b> instruction, 335-339,720
 divide-by-0 interrupt, 339
division, 138,140,312,329-331,335,336,338-340,620,668
 effect on EU/BIU coprocessing, 68
 rounding with <B>sar<b>, 331
DL register, 137
DMA, 8,24,95
<I>Doctor Dobb's Journal<i> magazine, 344
documentation, 8
don't assume intended instruction uses are always best, 304,315,333
don't jump, 508,530,531,693,705
DOS, 25,59-61,75,133,169,363,637,638,641,642,700,712,717,721
 as an assembly-language program, 61
 calls part of programming interface, 131
 file system, 59
 function 4, auxiliary output, 29
 function 5, printer output, 29
 function 9, print string, 28
 functions 2Ah to 2Dh, time and date, 25,35
 inadequacy of some services, 61
 interrupts, use to obtain blocks, not single characters, 582,637,638,641-643
dot operator, 245
double duty from a single instruction, 656
doubleword alignment, <see data alignment>
doublewords, loading, 184,185,189,700
DS register, 137,142,143,158,159,173,192,193,347,351,358,359,396,400
 can be used for temporary storage when free, 171
 default access to, 142,143,159,208,239,246,247,347,351,359,381,391,395
 loading with <B>lds<b>, 176,177
<B>DS:<b> segment override prefix, 193
dual 8/16-bit registers, 137
Duntemann, Jeff, 135,635,735
duplicating code, 568-572,574,579,588
<B>dw<b> directive, 168
DX register, 137,139,267,312,314,315,337,338
 accessible as DH and DL, 139
 least specialized general-purpose register, 139
 used as I/O-addressing register, 261 dynamic RAM (DRAM), 94
dynamic RAM (DRAM) refresh, 24,25,75,76,83,85,94,95,113,114,251,294,368,531,
703,707,712
 act of God, 94,99

can scarcely be addressed at all, 111
can't make less frequent, 99,712
code can't directly control, 94
doesn't necessarily stop 8088, 97
holds up PC via READY line, 117
in PC, 95,96
lowest level, 93,94
dynamic RAM (DRAM) refresh cycle eater, 62,76,77,79,87,110,122,275,316,517,
530,712
affects high-performance assembler code most, 98
affects performance of every program, 93
affects slower instructions least, 99
can cause fractional Zen timer counts, 99
can't structure code to avoid, 99
effect on performance, 95-98,110,123
effect on Zen timer accuracy, 30
effects tend to even out over time, 129
external to 8088, 94
impact varies with time, 114,126,128
interaction with 8-bit bus cycle eater, 115
what to do about, 98-99
EA calculations, <see effective address calculations>
EAX, 716
EBCDIC, 248
EBX, 716
effective address (EA) calculations, 210-214,219-221,531,645,676,718,720
only <I>mod-reg-rm<i> memory operands require, 212,217
EGA, <see Enhanced Graphics Adapter>
<B>ends<b> directive, 168
Enhanced Graphics Adapter (EGA), 62,262,263,492,499,713,714
and display adapter cycle eater, 105,713
and word-sized string instructions, 390-392
<B>enter<b> instruction, 715
ES register, 137,142,143,158,159,173,192,193,351,354,358,359,396,400,401
and loading doublewords, 184
can be used for temporary storage, 171-173
default access to, 143,159,351,354,359,381,391,393,396
loading with <B>les<b>, 176,177,737
<B>ES:<b> segment override prefix, 193
<B>even<b> directive, 707,710
event timeline, 118
execution

as two parallel chains of execution, 122,124
as three interleaved streams of events, 119,122,123,125,126
execution unit (EU), 65-70,72,75,77,78,117,520,522,524,525,527,528,530,699, 701
ability to execute instruction bytes faster than BIU can fetch them, 75,77, 84,85,130,702-704
can coprocess with BIU, 66-68,85,92,119,122,124,519,520
can process during DRAM refresh unless bus access needed, 97
can process while wait states are inserted unless bus access needed, 101
does not perform direct memory and I/O accesses, 65-67
don't double-count execution time overlapped with instruction fetching, 92, 124
execution times, 85,210,251,279,325,364,619,652,699,718,720
features contained within, 65
fully 16-bit, 67
gets instruction bytes from BIU, 66,67,519-521,523
minimum transfer time for bytes from the prefetch queue to the EU, 130
transfers to from prefetch queue, 118
usually stopped by wait states, 101
extra segment, <see ES register>
far calls, 187,517,619-621,623-626,636,670,723,725,726
emulating, 625,725
far data, 397,398,400,413
far jumps, 514,517,619-621,623-626,670
ending subroutines with, 567,568
far pointers, organization in memory, 187
far returns, 568,619-621,623-627,722
filling the screen, <see screen filling>
flags, 65,66,145,289-291,293,294,298,307,310,359,372,375,378,380,432,448,449, 482,488,560,627,636,653,654,657,722,723
double duty, 656
multiple tests, 619,653-655
used by interrupt handler, 318
flags register, 137,144,145,267,289,310,322,636,657,722
flexible mind, 9-12,109,131,471,504,506,508,635,733-735,741,742
developing skill, 10
<B>for<b> statement, 58
Forth, 712
forward references, 244,643-645,646,692,693
front-end entry points, <see multiple entry points>
further reading, 52,53
Geary, Michael, 129

general-purpose registers, 136-138,170,270,316,343,349
 and <I>mod-reg-rm<i> byte, 204,205
 loading segment registers via, 174
 use as operands, 138
global variables, 193
graphics support by DOS and BIOS, 61
<B>group<b> directive, 199
hardware, 8,59,60,62,133
 documentation, 52
 foundation for 8088, 63,65
 independence, 59
 transparency to programs, 62
Heinlein, Robert, 380
Hercules Graphics Card (HGC), and display adapter cycle eater, 105 hexadecimal, 8,156-158
high level languages, 6,9,10,58,163,164,167,173,179,191,193,240,241,244,258,
 400,488,535,542,560,622,623,651,729,730
 advantages of mixing with assembler, 241
 and segment sharing, 192,199,623
 interfacing assembler to, 241,242,623
 limitations, 10
 low-level control, 8
 transformation inefficiencies, 6,10
HL register pair of 8080, 267
horizontal retrace, 106
Hoyt, Michael, 249-251
IBM, 72,154,269,637
 AT, 30,53,63,101,108,616,698,702-704,707,708,712-714,717-719
 Model 30, 30,53
 Models 50 and 60, 53
 Model 80, 53,616
 PC, 5,8,9,53,58,72,101,113,114,152-155,489,616,694,697,698,704,712-714,717,
 719-721,733,734
 PC<I>jr<i>, all memory was display memory, 105
 PS/2 computers, 30,37
 technical reference manuals, 53
 XT, 30,53
<B>idiv<b> instruction, 338
IF (interrupt flag), 143-145,148,149,291,294,636,721,723
<B>if<b> directive, 680,692,693
<B>if...then...else<b>, 560
Illowsky, Dan, 332,497,545
immediate addressing, 208,224,226-232,281,284,298,302,343

forward references, 645
no sign extension for <B>mov<b>, 231,232,299
sign extension, 231,282-284,643
sign extension of operands to logical instructions, 284
implementation, <see program implementation>
implied addressing, 206
<B>imul<b> instruction, 146,337
<B>in<b> instruction, 138,140,261,262
in-line assembler (in high level language), 241,242
in-line code, 5,155,160,440,471,504,535,572,589,592-596,598,599,601-605,607,
608,613-615,679-680,683,685,688,692,693
branched-to, <see branched-to in-line code>
handling blocks of varying size, 679-680,683,688
labels in, <see labels in in-line code>
partial, <see partial in-line code>
pure, <see pure in-line code>
<B>inc<b> instruction, 145,147,182,226,302,304-310,340,342,349,356,366,547,
556,563,699,712
16-bit register-only form, 231,305-307,342
and byte order in memory, 190
doesn't affect CF, 309
incrementing 32-bit values, 652 index addressing, 209,212-214,218,221
index addressing component, 200,202,207
index registers, 137
index+displacement addressing, 207,214
indirect addressing, 208
indirect branching, 208
indivisible doubleword reads, 185,186,700
input/output (I/O)
not performed directly by EU, 65,66
performed directly by BIU, 66
input/output (I/O) addressing, 140,261,262
input/output (I/O) instructions, 261,262
input/output (I/O) ports, 62,261
input/output to the real world
needed by all useful programs, 61
<B>ins<b> instruction, 715
instruction execution time, 13,279
assumes instruction already prefetched, 68,85
inaccuracy due to prefetch queue cycle eater, 84,85
individual instruction times less useful than overall performance, 129
minimum of 4 cycles times the number of all memory accesses, 90,92,175,471

must be measured for working code sequences, not individual instructions, 91
no exact interval during which one and only one instruction executes, 92, 122-124
no such beast as true, 86,90,91,116,126
only meaningful in context, 91,92,129,130,517
shown in an event timeline, 126
varies depending on preceding code, 85,86,90
working definition of, 92,123,126,523
instruction fetch time, 13,251,517,524
benefits of using short instructions, <see shorter is better>
controls execution time when prefetch queue is empty, 68,300,719,720
counts as part of execution time when not overlapped with execution, 68
determines maximum execution speed, 84,369,719,720
don't double-count when overlapped with EU execution time, 92
varies depending on context, 85,524
instruction fetching, 14,53,84,346,519-525,528,529,538,557,589,601,702,703, 708-710,718
can proceed during shifts and rotates by CL, 325,538
effect on performance, 85,88-90,209,210,213,214,251,369,444,471,479,701,719, 720
maximum rate, 84
often subject to wait states in AT, 101
rarely subject to wait states in PC, 101
subject to cycle eaters, 69,70,75,77
instruction mix, effect on performance, 68,85,88-90,542
instruction pointer, <see IP register>
instruction prefetch queue, <see prefetch queue>
instruction prefetching, 75,262,519-521,523,703
determines execution time of sequence of register-only instructions, 88
most important coprocessing BIU performs, 68
instruction set, 8,59,75,76,134,150,169,263,266,269,270,274,285,295,297,344, 346,535,563,616,617,627,630,635,670,694,717,727,733
view as capable of being used, not as intended to be used, 304,315,333,449, 733,740
instructions, 297,305
desirability of keeping short, <see shorter is better>
instruction selection often matters, 289
part of programming interface, 131
int 0, 339
int 1, 149
int 4, 148
<B>int<b> instruction, 141-145,149,636,637,641,657,712

flags affected by, 657
Intel, 13,23,25,34,156,187,190,267,295,363,519,524,721
<I>Microsystem Components Handbook<i>, 52
interleaved memory, 698,702,703
interleaving of memory accesses and instruction execution, 119
interrupt flag, <see IF>
interrupt handlers, 193,318,363,626,637
interrupt vectors, 187,636,637
interrupts, 148,319,401,508,517,581,622,636,641,653,657,712,721-723
can use the stack at any time, 237,238
disabling, 143,144,149,319,401
divide-by-0, 339
don't change the effects of code, but do change performance, 129
flush the prefetch queue and change execution patterns, 129
loading vectors is only memory access that doesn't involve a segment, 159
should be disabled for as short a time as possible, 149,401
<B>into<b> instruction, 148
INTR pin, 148
IP register (instruction pointer), 66,67,137,142,144,226,227,509-511,513-515,
517,519,531,582,619,620,625,626,635,636,722
complications arising from prefetching handled internally by 8088, 144
<B>iret<b> instruction, 141-145,362,401,627,636,641,712,722-724,726
all flags are affected by, 657
IRQ0, 24-26
<B>ja<b> instruction, 146
<B>jae<b> instruction, 146
<B>jb<b> instruction, 146
<B>jbe<b> instruction, 146
<B>jc<b> instruction, 146
<B>jcxz<b> instruction, 139,374,432,448,449,462,543,544,608,657,668,669
doesn't affect any flags, 657
<B>je<b> instruction, 147
<B>jg<b> instruction, 147,148
<B>jge<b> instruction, 148
<B>jl<b> instruction, 148
<B>jle<b> instruction, 148
<B>jmp<b> instruction, 142,144,509-511,513,514,517-519,524-529,531,568-570,
572,582,598,615,627,630-633,643,645,680
<B>jmp short<b>, 509,512,522,525,569,570,572,628,630,643-645,648,679,726 <B>jna<b>
instruction, 146
<B>jnae<b> instruction, 146
<B>jnb<b> instruction, 146

<B>jnbe<b> instruction, 146
<B>jnc<b> instruction, 146,552
<B>jne<b> instruction, 147
<B>jng<b> instruction, 147,483
<B>jnge<b> instruction, 148
<B>jnl<b> instruction, 148
<B>jnle<b> instruction, 148
<B>jno<b> instruction, 148
<B>jnp<b> instruction, 147
<B>jns<b> instruction, 148,563
<B>jnz<b> instruction, 5,147,309,318,567,590,597,662,680
<B>jo<b> instruction, 148
<B>jp<b> instruction, 147
<B>jpe<b> instruction, 147
<B>jpo<b> instruction, 147
<B>js<b> instruction, 148
jump tables, 193,194,257,450,452,607,619,620,623,625,630,670-676,678,679,
683-686,688
letting assembler do the work of creating, 685
partial jump tables, <see partial jump tables>
pure jump tables, <see pure jump tables>
jumps, 508-514,563,565,581,615,616,627-629,632,635,644,646,670
around jumps, 457,563,567,611,672,679,683,688
backward references, 643,646
conditional, 510,512,539-541,545,560,562-565,567,569,577,586,601,610,611,
646,652,672,679-680,683,688,692
direct, 619,630
far, <see far jumps>
forward references, 643-646
indirect, 619,623,626,670,671
near, <see near jumps>
short, <see <B>jmp short<b>>
replacing calls, 628
to other subroutines in place of <B>ret<b>, 567,568,630,633
unconditional, 563,565,567-569,572,611,679,688,693
<B>jz<b> instruction, 147,540,565,567,660,680
keyboard, 8,35,59,60,64,129,401,675,676,712
interrupt, 136,149
macro software, 61
knowledge, 8,9,11,58,59,131,471,504,508,733,734,741,742
labels in in-line code, 613,614,684
<B>lahf<b> instruction, 139,145,267,289-291,294,309,312,727

latches, EGA/VGA, 391,392
layered system software, 59,60,63,64
<B>lds<b> instruction, 143,176,177,182
<B>lea<b> instruction, 198,214,221-223
<B>leave<b> instruction, 715
<B>les<b> instruction, 143,170,176-179,182,184,737 and byte order in memory, 189
 and loading doublewords, 184-186,189,700,736
 and performance, 179
libraries, subroutine and macro, 589
link, 32
listing file, using to see actual code generated, 197,244,284,588,693
<B>local<b> directive, 244,614
 bug when used in <B>rept<b> blocks, 614
local variables, 239,240,242-244
<B>lods<b> instruction, 82,139,140,182,288,347-351,357,359,361,377,381,391,
404-407,412,413,420,421,435,436,439,440,442-444,458,464,466,534,610,
613
 doesn't affect any flags, 375
 repeated, <see <B>rep lods<b>>
 synergy with other instructions, 404-407,412,413,464,613
<B>lodsb<b>, <see <B>lods<b>>
<B>lodsw<b>, <see <B>lods<b>>
look-up tables, 14,153,154,160,249-252,254,256-258,260,297,303,332,333,404,
450,452,456,457,468,471,538,539,607,615,626,673,676-678,686,693
<B>loop<b> instruction, 139,179,192,309,310,318,366,368,375,378,483,533,534,
537,538,572,574,589-592,594,597,598,601,604,608,657,658,660-662,688,
709
 doesn't affect any flags, 310,657
<B>loope<b> instruction, 658
<B>loopne<b> instruction, 658
<B>loopnz<b> instruction, 139,657,658,660
 doesn't affect any flags, 657
loops, 508,534,535,537,538,542,565,567,569,572,574,577,579,581,582,586-599,
602-606,608,611,613,632,637,653,656-658,660-662,668,669,685,693,709,
711
<B>loopz<b> instruction, 139,657,658
 doesn't affect any flags, 657
<B>lss<b> instruction, doesn't exist, 190
LZTEST.ASM long-period Zen timer test bed program, 49
LZTEST.EXE long-period Zen timer executable program, 50
LZTIME.BAT long-period Zen timer batch file, 50,51,164,639
LZTIMER.ASM long-period Zen timer source code, 37,50

machine language, 6,8,10,645
 ultimate form of all source code, 63-65,73
macros, 582,584,586,588,589,615,620,630,646,679-680,683,684,685,688,692,693,
725,727
 text substitution, 683,684,685
<I>MAD<i> magazine, 168
many ways to approach any task in assembler, 231,254,257,292,404,448,458,470,
545,554
maskable interrupts, <see interrupts>
MASM (Microsoft Macro Assembler), 32,50,52,168,170,190,192,196,197,198,242,
253,284,372,373,393-396,614,645,646,693,730
memory, 8,9,60,64,133,152,219
 8-bit as a cycle eater in AT, 62,713
 accessed directly by BIU, 66 avoid accessing whenever possible for performance,
92,153,182,210,218,263,
316,440,471,699,705,720
 constant access by programs, 62
 effect on performance, 62,279
 initializing, 258,259,301
 initializing when defined (at assembly time), desirability of, 260
 is slow, 210,346,697
 not accessed directly by EU, 65
 part of the PC's hardware, 61
 should not be destination operand whenever that can be avoided, 319-321
 tradeoff for performance, 160
 use to improve performance, 9,154,160
 versus processing horsepower, 9
memory accessing instructions, effect on instruction prefetching, 68,90
 are slow, 152,158,211,246,704
 never faster than register-only instructions, 217
 often slower than 4 cycles per memory access, 90
 suffer less from the prefetch queue than register-only instructions, 217
memory addresses, calculate outside loops, 214,220-222
memory addressing, 6,59,149,150,154,157,198,263
 can only address memory pointed to by at least one segment register, 160,161
 difficulty of dealing with blocks larger than 64 Kb, 161,162,167
 difficulty of handling blocks that cross a segment boundary, 162,167
 part of programming interface, 131
 use of square brackets to denote, 154,246
memory addressing modes, 152,153,159,198-200,202,211,212,232,715-717,729
 16 completely distinct memory-addressing modes, 205,206,271,349
 24 distinct ways to generate a memory offset, 204

immediate addressing, <see immediate addressing>
<I>mod-reg-rm<i> addressing, <see <I>mod-reg-rm<i> addressing>
naming <I>mod-reg-rm<i> addressing modes, 206,207,209
specifying <I>mod-reg-rm<i> addressing modes, 245
stack addressing, <see stack addressing>
memory architecture in 80286- and 80386-based computers, 698,702-704,721
memory resident programs, 254,637
MEMR line, 117,121,123,124
MEMW line, 117,121,123,124
microcode, 211
Microsoft Linker, 29,32,50
Miller, Dave, 129
mini-interpreters, 260,653,734
<I>mod-reg-rm<i> addressing, 199,200,203,205,208,213,218,221,223-226,231-233,
239,245,246,255,272,274-279,281-286,288,306,307,342,343,344,648,676,
740
allow only word-sized registers to be used to address memory, 248
and code size, 209,210,217,218,234,237
and performance, 209-214,216-218,235,237
avoid whenever possible for performance, 210,263
can be forced to access any segment, 208
can select a memory offset or register as an operand in any of 16 ways, 205,
271,349 can specify 256 possible source/destination combinations, 203,204
defaults to accessing DS except when BP is used, 208,239
direct addressing as anomaly, 207
doesn't work with all instructions, 209
has two performance components, 211
naming modes, 206,207,209
no addressing mode that uses only BP, 207,208,245
no inherent support for constant operands, 205
slow, but less slow than you might think relative to registers, 216
some instructions can only use <I>mod-reg-rm<i> addressing, 210
specifying, 245,246
suffer less from the prefetch queue than register-only instructions, 217
use displacement-free modes whenever you can, 214
used by many register-only instructions, 210
varies in performance due to EA calculation time,211-213
very flexible, 203,205,209,224
workhorse memory addressing mode of the 8088, 210,224
<I>mod-reg-rm<i> bytes, 199,201-204,207,208,213,224,226,227,245,270,271,343,
393,511,643
time required to calculate addresses from (EA calculation time), 209,210,

219,220
 with one-operand instructions, 200,204
Mode command, 29
Monochrome Display Adapter (MDA), and display adapter cycle eater, 105
"More optimizing for Speed," 251
mouse, 35,49,129,401,712
<B>mov<b> instruction, 34,143,145,170,182,221,226,232,272-277,282,285,298,
299,300,311,315,342-344,347,349-351,366,367,413,468,631,644,648,670
 accumulator-specific direct-addressing form, <see accumulator-specific>
 and byte order in memory, 187,188,190
 and segment copying, 174
 byte versus word, 79,81,84
 doesn't affect any flags, 311
 doesn't sign extend immediate operands, 231
 no non-<I>mod-reg-rm<i> memory immediate form, 344
 non-<I>mod-reg-rm<i> register immediate form, 231,232,343,344
<B>movs<b> instruction, 140,173,351,353,359,377,380,381,391,393,405,413,414,
420,428,497,499,581
 doesn't affect any flags, 375
 repeated, <see <B>rep movs<b>>
<B>movsb<b>, <see <B>movs<b>>
<B>movsd<b>, 716
<B>movsw<b>, <see <B>movs<b>>
MS_DOS, <see DOS>
<B>mul<b> instruction, 146,205,336,337,598,607,679,720
 doesn't suffer from DRAM refresh, 97
 effect on prefetch queue, 87,89,90
multi-bit shifts and rotates, 324,325,538,715
multi-byte values, storage organization in memory, 186-190,379
multi-word shifts and rotates, 328
multi-word values, addition and subtraction of, 309,310 multiple entry points, 650
 front-end entry points, 650,651
multiple tests from a single instruction, 619,653-655
multiplication, 138,140,146,329,335,336,607,673
 effect on EU/BIU coprocessing, 68
Navas, John, 244
near calls, 192,517,619,620,623-626,670,723,725
near jumps, 514,517,619,620,623-625,670
 ending subroutines with, 567,568
<B>near<b> keyword in C, 623
near returns, 568,619,620,623-625
NEC, 267

<B>neg<b> instruction, 147,321-324,546-549,716
negating 32-bit values, 322,323,544-552,554,555,652,716
negation, 322,323,544-552,554,555
NMI pin, 149
<I>non-mod-reg-rm<b> addressing, 199,208,214,218,224,225,231,343
 generally faster than <I>mod-reg-rm<i> addressing, 210,224,225
 generally less flexible than <I>mod-reg-rm<i> addressing, 210,224
 requires no EA calculation time, 224
nonmaskable interrupts, 136,149
<B>nop<b> instruction, 569,644,707,710
<B>not<b> instruction, 321-323,546,547,552
 doesn't affect any flags, 322
not-branching, 533,535,537,617,619,651,658,662
OF (overflow flag), 144-148,291,294,329-331,354
official execution times, 252,257,514,524,530,539
 can't simply add up times for individual instructions, 91
 inaccuracy of, 84-86,91,106,111,324,325
<B>offset<b> operator, 198,199
 and segment groups, 199
offsets, 155-159,198,510,511,513,514,564,607,631,671,679,686,723
 24 distinct ways to generate, 204
 calculating, 159,198,199,200,208-210,245-248,607
 just 16-bit numbers, 222,223
 loading, 198,221,736
 loading for a variable in a segment group, 199
 offsets greater than 16 bits wrap around to 0, 162,223-225,382,383,385,420
 use in addressing memory, 198
 very flexible portion of memory addressing, 198,203
OmniLab electronic test instrument, 117
"On Graphics," 244
operating system, 8,60
OPTASM, 645
optimization, 4,59,81,292,542,550,554,579,581,693,694,696-700,708,711,717,718,
          721,729,733,736,741,742
 8088 is best place to focus efforts, 696-699,715-717,731
 8088 optimizations serve well on other processors, 698,699,717,720,721
 concentrate on loops and time-critical code, 153,166,182,186,214,218,219,
          227,240,241,264,538,554,572,574,579,581,582,588,589,617,619,632,637,
          653,656,660,694,711 detailed 80286/80386 optimization, 718-721
 don't become fixated on a particular trick, 288,315
 don't use nifty tricks for their own sake, 279
 for space (size), 653,662,693,736

optimize for the common case, 449,539-545,551,552,559,560
understand conditions under which it will run, 541,542,554,688
worst-case, 541,542
"Optimizing for Speed," 249
<B>or<b> instruction, 146,148,229,281-284,301,302,320,343,412
OS/2, 696,697,700
<B>out<b> instruction, 138,140,261,262
desirability of byte-sized, 263
hazards of word-sized, 262,263
<B>outs<b> instruction, 715
overflow, 148
overflow flag, <see OF>
packed BCD, 334
paired-byte initialization, 300,301
paired jumps, <see jumps around jumps>
parameter passing, 233,239-241,244,576,650,651,675,740,741
macro parameters, 684
parity, 147
parity errors, 136
parity flag, <see PF>
partial in-line code, 537,602-606,608,610,611,613,688,693
partial jump tables, 673,674,678
Pascal, 6,242,589,623,624,635,715
Paterson, Tim, 344
PC data bus, <see bus>
<I>PC Tech Journal<i> magazine, 110,721
peripherals, 728,731
performance, 9,542,588,592,593,601-604,611,617,623,632,633,645,662,668,693,
697,698,700,710,717,733,736
can only be improved by reducing code's limiting factor, 91
can only know by measuring, 91-93,109,113,116,128,131,164,275,293,294,316,
351,517,555
can vary by up to 8.33 percent from DRAM refresh, 99
controlled by instruction fetch time, EU execution time, or both, 90
how to measure, 14
importance of measuring, 13,14
may vary for same code sequence over time because of cycle eaters, 113,114,
119,122,127,697
must be measured for working code sequences, not individual instructions, 91
pointlessness of trying to understand exactly, 113,114,116,117,121,126,128,
130,294,530
traditional assembler programming model, 60

 true nature of, 131
 true performance, 8,13,14
perspective, 5,315
PF (parity flag>, 144,147,329,330,354
plus operator, 246 <B>pop<b> instruction, 141,143,170,211,234-240,289,306,317,343,712,740
 and segment copying,173,174
 non-<I>mod-reg-rm<i> register form, 234,237
 <B>psw<b> on 8088, 290
<B>popa<b> instruction, 715
<B>popf<b> instruction, 145,291,294,309,312,362,721-723,725-727
 80286 bug and workaround for, 721-727
portability, 60
precalculation, desirability of, 249,253,258,260,539,610
prefetch queue, 14,66,68,71,106,300,326,368,518-525,527,528,530,531,538,539,
 542,556,582,589,590,601,616,619,620,636,637,652,701,702,709,718,720
 allowed to fill by <B>mul<b>, 89,90
 and performance of <I>mod-reg-rm<i> instructions, 216
 can add as much as 4 cycles per byte to instruction execution time, 91
 drained by register-only instruction, 88,90
 drained by short instructions, 93
 effect on Zen timer accuracy, 30,53,86
 emptied by branches, 66,67
 located in BIU, 67
 size, 67,69,616,701
 state varies with code mix, 85
 transfers from to EU, 118
prefetch queue cycle eater, 62,69,72,76,77,79,83,93,110,113,122,126,205,217,
 227,250-252,258,275,281,305,313,316,325,326,455,499,517,523,525,
 527-531,538,552,601,699,701-705,717
 as manifestation of 8-bit bus cycle eater, 83
 effect on performance, 84,85,110,123,214
 effect on register-only code, 88
 interaction with display adapter cycle eater, 115
 looms over the performance of all 8088 code, 93
 minimum transfer time for bytes from the prefetch queue to the EU, 130
 observed, 130
 undocumented and unpredictable, 84
 variation during different executions of same code sequence, 85,93
 what to do about, 92,93
prefetching, <see instruction prefetching>
prefix bytes, 363,401
 avoid multiple, 149,401

preloading values, 555-557,559,560,569,572
printer, 250
printer ports, 8
program
 commenting, 9
 conception, 5
 design, 6,9,59
 execution, true nature, 65
 implementation, 5,6,131,504,508,733
 specification and Zen of assembly language, 9
<I>Programmer's Journal<i> magazine, 244,249,251
"Programming Insight: High-Performance Software Analysis on the IBM PC," 52
programming interface, 59-65,68-70,73 beneath, 75,110,701
 component parts, 133
 part of knowledge aspect of Zen of assembly language, 131
 rests on cycle eaters, 131
protected mode, 142,169,170,184,700,701,711,715-717
<B>PS2<b> equate for assembling PS/2 version of long-period timer, 36,37
<B>public<b> option to the <B>segment<b> directive, 192
pure in-line code, 598,599,602,603,679,693
pure jump tables, 674,678
<B>push<b> instruction, 141,143,211,234-237,239,240,289,306,317,343,582,626,
          711,715
 and segment copying, 173,174
 non-<I>mod-reg-rm<i> register form, 234,237
 <B>psw<b> on 8088, 290
<B>pusha<b> instruction, 715
<B>pushf<b> instruction, 145,291,294,309,312,627,727
PZTEST.ASM precision Zen timer test bed program, 31,32,34
PZTEST.EXE precision Zen timer executable program, 32
PZTIME.BAT precision Zen timer batch file, 32,34
PZTIMER.ASM precision Zen timer source code, 15,32,34
QS0 line, 117,118
QS1 line, 117,118
RAM disk, 154
<B>rcl<b> instruction, 139,328
<B>rcr<b> instruction, 328,560
READY line, 117
real mode, 169,170,700,711,715,716,721
ReferenceZTimerOff, 29
ReferenceZTimerOn, 29
register hidden agenda, 135,136

register set, 135-137,149,267-269
register-only instructions, 217,258,263,468,534
 effect on instruction prefetching, 68,217,346
 many use <I>mod-reg-rm<i> addressing, 210,217
registers, 59,65,66,75,319,412,466,557,559,631,632,716,717
 desirability of using as much as possible, 83,93,110,135,153,210,217,218,
 227,229,230,232,234,235,237,245,246,263,305,316,321,698,699,705,717,
 720
 initializing multiple registers to same value, 299,300
 initializing two bytes with a single <B>mov<b>, 300,301
 irregularity of, 135
 part of programming interface, 131
 register selection often matters, 289
 should be destination operand whenever possible, 319-321
 use to store frequently used constants, 298,356,563,594,616,653
 use to store variables, 83,305,616,656
<B>rep cmps<b>, 372
<B>rep movs<b>, 79,173,363,373,405,413,416-420,497,596,597,707,713,714
<B>rep<b> prefix, 139,351,361,363-365,372,373,375,378,380,401,404,405,497,
 499,537,598,601,613
 don't use with segment override prefixes, 401 <B>rep lods<b>, 373
<B>rep scas<b>, 372
<B>rep stos<b>, 58,298,365-368,370,373,376,736
<B>repe<b> prefix, 371
repeated string instructions, 109,139,319,346,354,356,363-365,368,375,377,
 381,386,390,405,412,413,471,538,688,717,736
 always alter CX, 373,374
 don't use with segment override prefixes, 401
 execution time, 364
 handling 0-byte and 64 K-byte blocks, 377,380,382,384,385,389,432
 handling blocks larger than 64 K, 505,506
 much less flexible than normal instructions, 368
 no branching, 366-368,538,717
 no instruction fetching, 364,366-368,370,538,717
 operation when CX equals zero, 377,378,380,382,384,432,448,449,462
 try to use whenever possible, 368,677
 use ZF, not CX, to evaluate comparison results, 371,374,375
 word-sized operations, advantages of, 376,384,389,420,449,450,581
 word-sized operations, hazards of, 390-392,420,421
<B>repne<b> prefix, 371
<B>repnz cmps<b>, 371-375,378,448,449,482,485
<B>repnz<b> prefix, 368,371,372,433,482

<B>repnz scas<b>, 323,354,356,357,371-375,378,380,430,432-435,439,442-445,
448-450,452,454,456,457,464,471,474,479,482,658,675,677
<B>rept<b> assembler directive, 52,260,592,598,611,614,685
<B>repz cmps<b>, 371-375,378,380,448,449,458,464,471,472,474,479,482,485
<B>repz<b> prefix, 368,371,372,433,482,658
<B>repz scas<b>, 354,356,371-375,378,432-434,442-445,448-450,454,482
<B>ret<b> instruction, 141-144,517,567-569,572,582,584,620,627,630,631,633,
648,670,712
return addresses, organization in memory, 187
returns, subroutine, 508,581,582,584,586,588,589,627,628,631-633,635
far, <see far returns>
near, <see near returns>
to anywhere, 631,632
<B>rol<b> instruction, 327
ROMable code, 653
<B>ror<b> instruction, 139,327
rotate instruction affect fewer flags than you might think, 329
rotates, multi-bit, <see multi-bit shifts and rotates>
<B>sahf<b> instruction, 139,145,267,289,290,291,294,309,312,362,727
<B>shl<b> instruction, 329,563,564,655,699
<B>sar<b> instruction, 257,330-333
<B>sbb<b> instruction, 147,281-283,309,320,323,341,343,548,550-552
<B>scas<b> instruction, 139,140,171,354-356,359,368,371,372,377,381,391,393,
404,430,432,433,440,450,454,457,458,462,464,466,482,581,610,613
repeated, <see <B>repz scas<b> and <B>repnz scas<b>>
<B>scasb<b>, <see <B>scas<b>>
<B>scasw<b>, <see <B>scas<b>>
screen filling, 735-741
<B>seg<b> operator, 190,191,198,358 <B>segment<b> directive, 168,192
segment:offset addressing, 156-159,162,167,198,200,202,514
4096 pairs point to each address, 159
don't count on wrapping back to 0, 160
not particularly fast, 158
offset portion very flexible, 198
sums greater than 20 bits normally wrap around to 0, 160,162
segment:offset pointers
loading, 176,198,736
organization in memory, 187,722,723
passing in stack frames, 176
usually loaded with <B>les<b>, 176,737
segment override prefixes, 142,143,171,179,192-196,211,246,347,351,359,361,
363,381,391,394-398,401

and the <B>assume<b> directive, 196,197
don't use with <B>rep<b>, 401
segment registers, 66,67,141,142,155,158,168,343,380,383,397,506,716
and protected mode, 142,169,170,700
avoid loading whenever possible, 177,182
avoiding loading by sharing segments across modules, 192,623,624
bug in disabling of interrupts while loading, 144
can be loaded directly from any addressable memory location, 175,176,736,742
can each only point to a 64-Kb chunk, 160-162,167,380,381
common, 192
copying, 173,174
copying and storing, 142,169
directives, 168
disabling of interrupts while loading, 143,149
grouping, 192,199
loading with a segment that can vary during program execution, 175
manipulating, 169,170
must be loaded when a high level language passes a far pointer, 177
organize so that override prefixes aren't needed inside loops, 194-196,397
setting, 173-175,198,358
speed of manipulation, 169,170
use for temporary storage, 142,170-173,700
segment selectors, 169,170,700
segmented memory architecture, 6,133,155,167,269,505
segments, 155-159,167,168,198,380-383,396,400,401,420,700,723
default memory-addressing segments, 142,143,159,192,208,239,391,393,397,401
joining or separating via the <B>segment<b> directive, 191,192
sharing among multiple modules, 191,192,623,624
sharing between high level language and assembler, 199
shifted when used to address memory, 156-158
working with multiple segments, 193,194
working with multiple segments in multi-module programs, 191,623,624
self-modifying code, 65,615-617
self-reliance, 7
serial communications support by DOS and BIOS, 61
serial ports/adapters, 8,28,29,129,401,672,712,728
SF (sign flag>, 144,148,291,292,329,330,354,561,655,657 sharing code, 646-650
Sheppard, Byron, 52
shifting and rotating memory, 326
shifts and rotates by 1 bit, relative undesirability of, 324-327
shifts and rotates by CL bits, 324-327,538
relative desirability of, 324-327

shifts and rotates by CL bits don't affect CL, 326,373
shifts, multi-bit, <see multi-bit shifts and rotates>
<B>shl<b> instruction, 139,329
short jumps, <see <B>jmp short<b>>
shorter is better (smaller/faster), 83,92,93,110,281,299,305,333,589,698,699,
705,717,720
<B>shr<b> instruction, 68,116,325,329-331,538
effect on prefetch queue, 86-89
suffers from DRAM refresh, 97,98
SI register, 137,140,149,159,347,349,350,353,358-361,380,643
index addressing component, 207
specializes as memory-addressing register for string instructions, 140,193,
316,350,351,359
used as memory-addressing register, 140,200,208,223,246,309,347,391
sign flag, <see SF>
size versus speed, <see bytes versus cycle>
smaller is better, <see shorter is better>
smaller is slower, 589
source code control, 9
SP register (stack pointer), 135,137,144,159,170,232,233,237-239,291,306,630,
711,740
always addresses the stack segment, 141
is a general-purpose register when not being used to maintain a stack, 136,
141
never push directly, 141
not usually available as a general-purpose register, 136,141,172
points to top of stack, 136,141,232,233,237
should always be even, 306,711,712
used as memory-addressing register, 232,233,239
speaker, 24,25,60,64,650
special forms of common instructions, 224,226,230,232
assembler automatically selects whenever possible, 226,232,274,287,306,342
look the same as more general forms, 273
two legitimate machine language forms of, 274,306,342,344
special compressed forms of instructions, 643
assembler automatically uses only when it knows enough to do so, 643
forward references to, 643-645
speed of development, 9
speed versus size, <see bytes versus cycles>
SS register, 137,142-144,158,159,190,192,193,396
default access to, 143,159,208,239
don't use for temporary storage, 170

 together with SP must point to a valid stack, 143,190
<B>SS:<b> segment override prefix, 193
stack, 136,232-234,237-240,291,294,517,568,582,626,627,630,632,633,635,650,
651,657,711,721,722,725,740,741
 allocating space on in a subroutine, 635
 clearing by reloading SP, 630
 don't access popped stack data, 237,238
 not inactive even when not accessed directly because of interrupts, 136,143
stack addressing, 224,232,237,244
stack addressing registers, 137
stack frames, 141,175,179,193,208,233,239-245,286,544,715,740
 all <I>mod-reg-rm<i> addressing modes can often be used to point to, 244
 negative displacements, 242,243
 try to use 1-byte displacements, 243,244
stack-oriented instructions, 224
stack pointer, <see SP register>
stack segment, <see SS register>
standard input, 638,639
 redirected, 639,640
standard output, 638,639
static-column RAM, 698,702,703
static RAM (SRAM), 94,712
static variables, 193
statuses, carrying along, <see carrying results along in a flag>
statuses, saving, 291
status flags, 144-147
<B>stc<b> instruction, 146
<B>std<b> instruction, 149,362,363,420
<B>sti<b> instruction, 149
<B>stos<b> instruction, 139,140,288,351,352,359,365,377,381,391,393,404-407,
412,413,613
 doesn't affect any flags, 375
 repeated, <see <B>rep stos<b>>
<B>stosb<b>, <see <B>stos<b>>
<B>stosw<b>, <see <B>stos<b>>
string instructions, 79,109,139,140,143,149,158,159,182,192,209,211,218,219,
221,224-226,264,288,297,319,344,346,347,351,357-359,362-364,367,
381-383,393,395-397,400,401,404,407,413,416,417,424,466,479,482,488,
497,504,506,534,538,601,613,698,720
 advance their pointer registers, 350,359,377,391,407,444
 byte- and word-sized operations, hazards of mixing, 420,422-425,427
 data size, 361,375,377,393,394

 definition of advancing pointer registers, 359,361,362
 operands to, 393-396
 pointing back to last element processed, 377-379,444
 repeated, <see repeated string instructions>
strings, comparing, 359,404,457,462,464,610,611,613
strings, copying, 351,405,576,582,610,611,613
strings, searching, 354,404,433-436,439,442-445,454,457,464,471-473,479,482,
565,610,611,613,679,685
 case-insensitive, 457,466,658,661,662,693
 double-search approach, 433-435,445,462,688
 substrings, <see substrings, searching for>
 word-sized operations, advantages of, 440,449,450,610,611 <B>struc<b> directive, 179,242
 negative displacements, 242,243
structure elements, forward references to, 644,645
<B>sub<b> instruction, 147,281-283,298,300,305,306,309,311,313,314,319-321,
323,341,343,448,449
subroutines, 508
 returning status from, 147,487,488
substrings, searching for, 457,471-473,479
Symdeb
 run from a remote terminal, 29
 screen flipping, 28
system clock, 25
system memory performance relative to display memory, 63,703
system memory wait state cycle eater, 702-704,720
system software, 59,60,63,64
TASM (Turbo Assembler), 32,168,244,372,614,645
<B>test<b> instruction, 285,321,343
TESTCODE Zen timer file containing code to be timed, 29-31,50
TF (trap flag), 144,145,149
think functionally, 533,535
Time command, 25,35
time of day count, 25,36,37,185
timer channel 0 of the 8253, 24-28,36,37
 interrupts, 25,26
timer channel 1 of the 8253, 24,25
 DRAM refresh, 25,95
timer channel 2 of the 8253, 24-26
 interrupts, 25,26
 system clock, 25,26
timer interrupt, 26,35,129,136,149,186
 BIOS code to handle, 35,51

timer, <see 8253 timer chip>
TIMER_INT BIOS routine, 25
timing diagrams, 118
tlink, 32
traditional assembler programming model, 59,60
 compatibility benefits, 59,60
 inadequacy of hardware representation, 61
 knowing well, 59,61
 knowing when to break the rules, 61
 performance shortcomings, 60,61
 portability benefits, 59,60
 replacing with own code, 61
trap flag, <see TF>
"Tricks of the Trade," 344
Turbo C, 241,623
Turbo Pascal, 735
two's complement arithmetic, 322,323,546
unconditional jumps, <see jumps, unconditional>
Unix, 697,700
unpacked BCD, 334,335,339,340 upper case, converting to, 351,405-407,466,468,470,471,576,577,579,614,637,
638,661,693
user interaction, 9
V20, 267
vertical retrace, 106
VGA, <see Video Graphics Array>
Video Graphics Array (VGA), and display adapter cycle eater, 105,262,728
 and word-sized string instructions, 390-392
video registers, indexed, setting, 262,263
VisiCalc, 269
<B>wait<b> instruction, 730,731
wait state cycle eater, 83,702-704
wait states, 14,62,75-77,100,106,698,703-705,713,718
 affect everything, even other cycle eaters, 100
 allow slower devices to complete bus accesses, 100
 are of no particular duration, 100
 as they relate to display memory, 100,101
 can occur only during a memory or I/O read or write, 100
 code can't directly control, 94
 compared to DRAM refresh, 100
 controlled by device being accessed, 100
 do not occur on a regularly scheduled basis, 100

don't stop the 8088 completely, 101
external to 8088, 94
lowest level, 94,100
only display adapter wait states seriously affect PC performance, 101
system memory, <see system memory wait state cycle eater>
transparent to code, 100
usually do stop the 8088, 101
warm boot and system clock, 26,27
white space, 488
word alignment, <see data alignment>
word-to-doubleword conversion, 312-314
word-sized memory access
advantage over two byte-sized accesses by two instructions, 81,83
penalty for using, 79,80,84
split into two byte-sized accesses, 67,78,79
WordStar, 269
<B>xchg<b> instruction, 139,172,226,267,285,287,288,316-319,342
AX-specific form of, 285-288,306,318,342
used to get and set a memory variable, 318
<B>xlat<b> instruction, 139,158,193,209,211,224,246-249,256,257,264,312,333,
468,471
<B>xor<b> instruction, 146,148,281-284,298,311,320-322,343,489-492,496-499,
504,505,533-537,594
Z80, 266,267,509
Zen (coined verb form; also Zenned, Zenning), 735,741
Zen of assembly language, 3-11,13,14,53,73,113,114,117,164,167,182,237,289,
315,332,396,489,504,508,509,533-535,537,545,559,589,615,617,635,651,
701,722,727,728,733-735,742
inquisitive, skeptical mind, 81 learning, 10
mastering, 10
Zen timer, 14,15,22,23,26,30,53,87,93,113,115,164,251,275-279,281,291,293,
294,456,492,514,517,518,557,622,694,696,703,708,719,721,733
accuracy, 29,30,34,53,86
accurate only for the particular code sequence you've timed, 116,129
avoiding interrupts while long-period Zen timer runs, 49
alternative output methods, 28,29
assembling the long-period timer for PS/2 computers, 36,37
calling interface, 15,23,27-29,34,48
choosing the long-period timing mode, 36,37
compatibility with PC-compatible computers, 30
DS set equal to CS to allow data in TESTCODE, 30-32
effect of DRAM refresh on accuracy, 30,34

effect of preceding code on accuracy, 86
effect of prefetch queue on accuracy, 30,34,53,86
effect on calling code, 29
erratic operation when undocumented timer stop is used, 37
excluding execution time of start-up code, 32
far calls to, 29
fractional counts can result from DRAM refresh, 99,128,712
fractional counts can result from display adapter cycle eater, 99,128
ideally used to measure the performance of an entire subroutine, 86
imperfections, 53
inability to measure across midnight, 35
inaccuracy introduced into system clock, 23,26,27,35,36
interrupts, 23,25,26,28,35
least variable over longer periods, 129
long-period inaccuracy in PS/2 computers, 36
long-period timer 26,34-37,48-51,164,186,492
maximum timing intervals, 34,35
minimum timing interval, 86
overflow detection, 26-28
overhead adjustment, 27,28
precision timer, 35,37,48,50,51,492
rebooting at the end of timing sessions, 26,27,35-37
rebooting immediately in case of erratic results, 37
repeating tests multiple times, 37
rounding results, 34
sample use of long-period timer, 48,49
sample use of precision timer, 30
sequence of calls to subroutines, 28
short code sequences may vary greatly from one measurement to next, 129
use with assembly language, 29
use with high level languages, 29
using, 30-32,34,48-52
variation of results on computers other than PCs, 30
variations can result from DRAM refresh, 99
zero, handling efficiently, 229,230,277-279,297-299,301,302,303,311,657,738
zero flag, <see ZF>
ZF (zero flag), 144-148,277,301,307,329,330,354,371,372,375,432,433,448,449,482,488,586,658
Zilog, 266
ZTimerOff, 15,23,26-28,30-32,34
ZTimerOn, 15,23,26-28,30-32,34,49
interrupts are turned off and must remain off for precision timer, 23

INDEX/ABRASH/

ZTimerReport, 15,28-31,34