

Index

(numeric character constant prefix) 100
 # character literal constants 402
 #(and)# (macro parameter brackets) 976
 #{ (Hybrid parameter passing syntax) 838
 #asm 1151
 #emit 1151
 #endasm 1151
 #ERROR statement 951
 #include 20
 #includeonce directive 571
 #KEYWORD section in a macro 989
 #PRINT statement 951
 #TERMINATOR section in a macro 988
 ? Compile-time operator 407
 @c 30
 @ELEMENTS compile-time function 974
 @Elements compile-time function 479
 @EVAL function 980
 @global operator (in name spaces) 497
 @nc 30
 @no 30
 @NOALIGNSTACK option 805
 @NODISPLAY option 805
 @NOFRAME option 805
 @NOSTORAGE 169
 @ns 30
 @nz 30
 @o 30
 @offset compile-time function 1153
 @s 30
 @size compile time function 487
 @Size compile-time function 479
 @size function 187
 @size function (applied to UNIONS) 493
 @StaticName function 1152
 @STRING
 name operator 975
 @TYPENAME function 992
 @USE procedure option 830, 836
 @z 30
 finalize strings 1087
 initialize strings 1087
 pVMT 1080, 1084
 vars constant 816
 VMT 1075

16-bit registers 24
 32016 microprocessor 235
 32-bit registers 24

4004 microprocessor 234
 4040 microprocessor 234
 64-bit constant expressions 1119
 6502 microprocessor 234
 6800 microprocessor 234
 68000 microprocessor 235
 8008 microprocessor 234
 8080 microprocessor 234
 8085 microprocessor 234
 8088 microprocessor 235
 8-bit registers 24

A

AAA instruction 900
 AAD instruction 900
 AAM instruction 900
 AAS instruction 900
 Absolute value (floating point) 628
 Abstract base classes 1091
 Abstract data types 1060
 ABSTRACT keyword 1093
 Abstract method declarations 1091
 Accessing a word in byte addressable memory 141
 Accessing an element of a single dimension array 464
 Accessing data on the stack. 186
 Accessing data via a pointer 410
 Accessing data with a 16-bit bus 144
 Accessing double words in memory 145
 Accessing elements of 3 & 4 dimensional arrays 471
 Accessing elements of a two-dimensional array 470
 Accessing elements of an array 465
 Accessing elements of multidimensional arrays 475
 Accessing fields of a structure 484
 Accessing local variables 815
 Accessing names outside a namespace 497
 Accessing reference parameters 831
 Accessing the characters within a string 426
 Accessing the fields of a class 1069
 Accessing the fields of a UNION 492
 Accessing value parameters 824
 Accessing words at odd addresses 145
 Accessor methods 1060
 Activation records 810
 Active high logic 225
 Active low logic 225
 Actual parameters 836
 ADC instruction 855
 ADD 28
 Adders 223
 Addition (extended precision) 853
 Addition table 848
 Address binding 1375, 1376
 Address bus 139
 Address expressions 171
 Address of operator 160
 Address spaces 140

- Addressable memory 139
- Addressing modes 157
- Addressing modes (Y86) 278
- Address-of operator 191
- AGP Bus 336
- AH 24
- AL 24
- Algorithm 541
- Aliases 430, 494, 557
- Aligning fields within a record 490
- Allocating objects dynamically 1081
- Allocating storage for arrays 474
- AND 605
- AND instruction 68, 910
- AND operation 65, 204
- Anonymous unions 494
- Anonymous variables 160
- Anyexception (try..endtry) 735
- Arbitrary text as macro parameters 976
- Arc cosecant 639
- Arc cosine 639
- Arc cotangent 639
- Arc secant 639
- Arc sin 639
- Architecture 137
- Arctangent 632
- arg.c function 640
- arg.v function 640
- Arithmetic expressions 597, 600
- Arithmetic idioms 606
- Arithmetic logical systems 605
- Arithmetic operators within a constant expression 404
- Arithmetic shift right 78
- Arity 478, 479
- Array access 464
- Array variables 464
- array.cpy function 481
- array.daAlloc function 480
- array.daFree function 480
- array.dArray declaration 479
- array.index function 480
- Arrays 463
- Arrays as structure fields 487
- Arrays of arrays 471
- Arrays of records 486
- Arrays of two or more dimensions 468
- ASCII character set 58, 97, 104
- Assembly language newsgroups 8
- Assert Macro 1252
- Assigning a constant to a variable 597
- Assigning one variable to another 597
- Assignment by reference 428
- Assignments 597
- Associativity 203, 600, 601
- Audio and video data 1117
- Automatic storage allocation 551
- Automatic variables 169
- AX 24

B

- Background colors on the text display 195
- backspace 40
- Base address (of an array) 463
- Base classes 1075
- Based indexed addressing mode 163
- Basic System Components 137
- BCD 87
- BCD arithmetic 897
- BCD numbers 56
- BCD values 397
- BEGIN..END statement 740
- bell character 40
- BH 24
- Biased (excess) exponents 91
- Big-endian data format 928
- binary 53
- Binary Coded Decimal 87
- Binary coded decimal arithmetic 897
- Binary coded decimal numbers 56
- binary data types 56
- Binary Formats 55
- Binary Numbering System 54
- Binary operator 203
- Binding an address to a variable 1376
- Bit data 909
- Bit fields and packed data 81
- Bit masks 910
- Bit offset 909
- Bit runs 909
- Bit scanning 923
- Bit sets 909
- Bit strings 909
- Bits 56
- bits.cnt function 932
- bits.coalesce function 932
- bits.distribute function 932
- bits.extract function 932
- bits.merge8, bits.merge16, and bits.merge32 functions 933
- bits.nibbles8, bits.nibbles16, and bits.nibbles32 functions 933
- bits.reverse8, bits.reverse16, and bits.reverse32 functions 933
- Bitwise logical operators within a constant expression 404
- Bitwise operations 68
- BL 24
- Boolean Algebra 203
- Boolean algebra theorems 204
- Boolean expression canonical form 209
- Boolean expressions 30, 604
- Boolean function equivalence to electronic circuits 221
- Boolean function names 207
- Boolean function numbers 208
- Boolean function simplification 214
- Boolean functions 205
- Boolean functions of n variables 207

- Boolean logical systems 605
 - Boolean map simplification 214
 - Boolean term 209
 - Boolean values 56
 - BOUND instruction 393
 - BP 24
 - Branch out of range 758
 - BREAK 791
 - BREAK statement 36
 - BREAKIF statement 36
 - bs 40
 - BSF and BSR instructions 924
 - BSWAP instruction 928
 - BT, BTC, BTR, and BTS instructions 915
 - Buffering data to improve I/O performance 336
 - Bus contention 261
 - Bus interface unit 256
 - BX 24
 - Byte 57
 - Byte addressable memory array 143
 - Byte enable lines 140, 145
 - Byte strings 935
 - Bytes 56
- C**
- Cache and I/O devices 352
 - Cache Architecture 308
 - Cache associativity 308
 - Cache coherency 269
 - Cache hit 154
 - Cache hit ratio 154
 - Cache line replacement policy 310
 - Cache memory 153
 - Cache miss 154
 - Cache write policies 311
 - Cache, two level 155
 - CALL Instruction 805
 - CALL instruction 541, 805
 - Callee register preservation 544
 - Caller register preservation 544
 - Calling Base Class Methods 1095
 - Canonical form of boolean expressions 208
 - Canonical forms 209
 - Carriage return 40
 - carry 26
 - Carry flag 26, 916
 - carry flag 592
 - Case insensitive string comparisons 436
 - Case labels (non-contiguous) 782
 - Case neutral 19
 - CASE statement 747
 - Case Statement 776
 - Case statement 761
 - cbw instruction 74
 - cdq instruction 74
 - CD-Quality recording 113
 - Celeron microprocessor 239
 - Central Processing Unit 137
 - Central processing unit 24
 - CH 24
 - Change sign (floating point) 629
 - Changing the value of a VAL constant at different points in your program 406
 - Char data type 101
 - Character classification compile-time functions 958
 - Character constants 401
 - Character literal constants 100
 - Character strings 419
 - Characters 96
 - Circular queues 343
 - Circular queues and output transfers 349
 - CL 24
 - CL register in rotate operations 80
 - CL register in SHL instruction 77
 - Class Methods, Iterators, and Procedures 1067
 - Classes 1061
 - clc instruction 84
 - cld instruction 84
 - Clearing the FPU exception bits 633
 - Clearing the Screen 192
 - CLI instruction 348
 - cli instruction 84
 - Clipping (saturation) 76
 - Clock 149
 - Clock frequency 150
 - Clock period 150
 - Clocked logic 228
 - Closure 203
 - Closure of an operator 203
 - cls (clear screen) routine 192
 - CMC instruction 917
 - cmc instruction 84
 - CMPS 935, 943
 - Coalescing bit sets 920
 - Coarse-grained parallelism 269
 - Code reuse 580
 - Code stream parameters 1341
 - Coercion 173
 - Color depth 109
 - Colors on a video display 109
 - Column major ordering 469, 473
 - Combinatorial circuits 223
 - Command line arguments 640
 - Command line parameters 641
 - Comments 22
 - Commutative operators 603
 - Commutativity 203
 - comp.lang.asm.x86 newsgroup 8
 - Compare strings 935
 - Comparing dates 84
 - Comparing floating point numbers 89
 - Comparing registers with signed integer values 175
 - Comparing two strings 436
 - Comparison Instructions (MMX) 1134

- Comparison operators in a constant expression 404
- Compile-Time Constants and Variables 952
- Compile-Time Expressions and Operators 953
- Compile-Time Functions 956
- Compile-Time Language 949
- Compile-time loops 966
- Compile-Time Pattern Matching Functions 958
- Compile-time procedures 969, 985
- Compile-time programs 995
- Compile-time string functions 958
- Compile-time symbol information 959
- Compile-time variables 961
- Complete boolean evaluation 768
- Complex arithmetic expressions 600
- Complex string functions 947
- Composite data types 419
- Computer Architecture 137
- Computing 2**x 631
- Computing MOD using an AND instruction 345
- Concatenating two string literals 401
- Concatenation 433
- Condition codes 26
- Condition jump instructions (opposite conditions). 758
- Conditional compilation 962
- Conditional jump aliases 758
- Conditional Jump Instructions 755
- Conditional jumps (x86) 282
- Conditional statements 761
- Console application 20
- CONSOLE Module (Standard Library) 192
- console.cls routine 192
- console.fillRect routine (standard library) 197
- console.getX 194
- console.getY 194
- console.gotoxy routine 193
- console.puts routine (standard library) 199
- console.putsx routine (standard library) 199
- console.setOutputAttr routine (standard library) 196
- CONST declarations 397
- Constant expressions 172, 403
- Constructing a truth map 215
- Constructing data tables at compile time 996
- Constructing logic functions using only NAND operations 222
- Constructing truth tables from the canonical form 210
- Constructors 1079, 1081
- Constructors and Inheritance 1082
- Contention (for the bus) 261
- Context-free macros 985
- CONTINUE 791
- CONTINUE and CONTINUEIF statements 745
- Control bus 139
- Control characters 98
- Control characters within string constants 402
- Controlling field offsets within a record 489
- conv.strToFt function 640
- Converting Arithmetic Expressions to Postfix Notation 635
- Converting BCD to floating point 624
- Converting between canonical forms 213
- Converting between HLA time format and seconds 515
- Converting binary to hex 61
- Converting Floating Point Expressions to Assembly Language 634
- Converting hex to binary 61
- Converting IF statements to assembly language 761
- Converting Postfix Notation to Assembly Language 637
- Converting UNICODE to ASCII 1124
- Copy by reference 430
- cosecant 639
- Cosine 631
- cot function 639
- Cotangent 639
- Count (string elements) 936
- Counters 231
- Counting bits 925
- CPI (clocks per instruction) 265
- CPU 24, 137
- CPUID instruction 1113
- cr 40
- Create procedure for an object 1081
- Creating libraries 581
- Creating lookup tables 651
- cs.difference function 1134, 1141
- CTL (compile time language) 949
- Current string length 420
- Cursor location on the screen (standard library) 194
- Cursor positioning 193
- cwd instruction 74
- cwde instruction 74
- CX 24
- CYMK color space 110

D

- D (data) flip-flop 229
- DAA instruction 898
- Dangling pointers 414
- DAS instruction 898
- Data bus 138
- Data Transfer Rates 334
- data.pack function 511
- Date arithmetic 512
- Date comparison 84
- Date to string conversions 510
- date.a_toString function 510
- date.datePlusDays function 512
- date.datePlusMonths function 512
- date.dayNumber function 513
- date.dayOfWeek function 513
- date.daysBetween function 512
- date.daysLeft function 513
- date.fromJulian function 512
- date.IsValid function 505
- date.Julian function 512

- date.OutputFormat values 510
 - date.Print function 510
 - date.today function 509
 - date.toString function 510
 - date.unpack function 511
 - date.validate function 505
 - Deadlock 349
 - DEC instruction 190
 - decimal 53
 - Decimal arithmetic 96, 897
 - Decisions 760
 - Declarations
 - static 21
 - Declaring arrays 464
 - Decoder circuits 224
 - Decoding instruction opcodes 225
 - DEFAULT section of a SWITCH statement 748
 - Deferring macro parameter text expansion 977
 - delete memory deallocation operator (C++) 187
 - DeMorgan's Theorems 205
 - Denormalized exception (FPU) 614
 - Denormalized floating point values 621
 - Denormalized values 92
 - Destination index 936
 - Destroy procedure in a class 1086
 - Destructors 1086
 - Destructuring 774
 - Device Drivers 353
 - DH 24
 - DI 24
 - Digital video 115
 - Direct addressing mode 158
 - Direct jump instructions 753
 - Direct mapped caches 308
 - Direct Memory Access 333
 - Direct memory access 331
 - Direction flag 936, 937
 - Dirty bits 312
 - Disassembly 379
 - Displacement only addressing mode 158
 - Display (in an activation record) 806
 - Display (lexical nesting data structure) 1375
 - dispose memory deallocation operator (Pascal) 187
 - Distributed Shared Memory 304
 - Distributing bit strings 920
 - Distributive law 204
 - div (within a constant expression) 404
 - DIV simulation 607
 - DL 24
 - DMA 331
 - Domain conditioning 650
 - Domain of a function 649
 - Dope vector 478
 - Dot operator 484
 - Double precision floating point format 91
 - Double word storage in byte addressable memory 141
 - Double word strings 935
 - Double words 59
 - double words 56
 - Downloading MASM 8
 - Dual I/O ports 329
 - Duality 205
 - DUP operator 464
 - dwords 56
 - DX 24
 - Dynamic Arrays 477
 - Dynamic link 1308, 1378
 - Dynamic memory allocation 187, 412
 - Dynamic nesting of control structures 731
 - Dynamic Object Allocation 1081
 - Dynamic type systems 495
- ## E
- Eager macro parameter text expansion 977
 - Eager vs. deferred macro parameter evaluation 977
 - EAX 24
 - EBP 24
 - EBX 24
 - ECX 24
 - EDI 24
 - EDX 24
 - Effective addresses 160
 - EFLAGS 25
 - EFLAGS register 85
 - EFLAGS register 184
 - EIP register 26
 - EISA bus 334
 - Electronic circuit equivalence to boolean functions 221
 - ELSE 30, 32, 761
 - ELSEIF 30, 32
 - Embedding control characters in string constants 402
 - EMMS Instruction 1139
 - EMMS instruction 1115
 - Encoding instructions 271
 - ENDFOR 34, 36
 - ENDIF 30, 32
 - ENDTRY 37
 - ENDWHILE 30, 33
 - ENUM 408
 - Enumerated data types 408
 - eoln 40
 - Errors when using pointers 189
 - Escape character sequences 401
 - ESI 24
 - ESP 24
 - ex.DivisionError exception 590
 - ex.FileOpenFailure exception 522
 - ex.IntoInstr exception 590
 - ex.InvalidDate exception 505
 - ex.MemoryAllocationFailure exception 188
 - ex.StringIndexError exception 428
 - ex.StringOverflow exception 423, 434
 - EXCEPTION 37
 - Exception flags (FPU) 616

- Exception handling 37
- Exception masks (FPU) 614
- Exception values 735
- Exclusive-or 65
- Exclusive-OR operation 207
- Exclusive-or operation 67
- Executing a loop backwards 798
- Execution units 265
- EXIT 740
- EXIT and EXITF statements 546
- EXITIF 740
- exp function 640
- Exponent 88
- Expression classification functions 960
- Expressions 600
- Expressions and temporary values 603
- Extended precision (80 bit) floating point values 397
- Extended precision addition 853
- Extended precision AND 873
- Extended precision comparisons 857
- Extended precision division 864
- Extended precision floating point format 91
- Extended precision formatted I/O 883
- Extended precision I/O 878
- Extended precision input routines 884
- Extended precision multiplication 860
- Extended precision NEG 872
- Extended precision NOT 874
- Extended precision OR 874
- Extended Precision Rotates 878
- Extended precision shifts 875
- Extended precision XOR 874
- EXTERNAL directive 572, 575
- Extracting bit sets 920
- Extracting bit strings 919, 930

F

- F2XM1 instruction 631
- FABS instruction 628
- FADD/FADDP instructions 625
- Falling edge of a clock 150
- False (representation) 604
- FBLD instruction 624, 901
- FBLD/FBSTP instructions 624
- FBSTP Instruction 624
- FBSTP instruction 901
- FCHS instruction 629
- FCLEX/FNCLEX instructions 633
- FCOM, FCOMP, and FCOMPP instructions 629
- FCOM/FCOMP/FCOMPP instructions 629
- FCOMI and FCOMIP instructions 629
- FCOS instruction 631
- FDIV/FDIVP/FDIVR/FDIVRP instructions 626
- FIADD instruction 634
- Fibonacci sequence 846
- FICOM instruction 634

- FICOMP instruction 634
- FIDIV instruction 634
- FIDIVR instruction 634
- Field alignment within a record 490
- Field Offsets Within a Record 489
- Field width 41
- FILD Instruction 623
- File handles 521
- File Storage (in the memory hierarchy) 304
- fileio.getf function 639
- Filling a Rectangular Section of the Screen 197
- fillRect routine (standard library) 197
- FIMUL instruction 634
- Fine-grained parallelism 268
- FINIT/FNINIT instructions 633
- First-in, First-out (FIFO) cache replacement policy 311
- FIST instruction 623
- FISTP Instruction 623
- FISUB instruction 634
- FISUBR instruction 634
- FLAG register 85
- Flags 25
- Flags (and CMP) 592
- FLD Instruction 621
- FLD1 instruction (load 1.0) 631
- FLDCW instruction 633
- FLDL2E instruction (load lg(e)) 631
- FLDL2T instruction (load lg(10)) 631
- FLDLG2 instruction (load log(2)) 631
- FLDLN2 instruction (load ln(2)) 631
- FLDPI instruction (load pi) 631
- FLDZ instruction (load 0.0) 631
- Flip-flops 229
- Floating point arithmetic 611
- Floating point comparisons 89, 629
- Floating point data types 619
- Floating point registers as procedure parameters 1341
- Floating point unit 237
- Floating point values 60
- Flushing the pipeline 261
- FMUL/FMULP instructions 626
- for 923
- For loops 790
- FOR statement 34
- Forcing bits to one 68
- Forcing bits to zero 68
- Forcing bits to zero (MMX) 1134
- Forcing selected bits to one. 911
- FOREACH..ENDFOR 843
- Foreground colors on the text display 195
- FOREVER loops 787
- FOREVER statement 36
- Formal parameters 836
- FORWARD (variable and type declarations) 1089
- Forward procedure declarations 567
- Four-way set associative caches 310
- FPATAN instruction 632
- FPREM/FPREM1 instructions 628

FPTAN instruction 632
 FPU busy bit 618
 FPU condition code bits 616
 FPU Control Register 612
 FPU control word 633
 FPU exception bits 633
 FPU exception flags 616
 FPU exception masks 614
 FPU interrupt enable mask 615
 FPU precision control 614
 FPU Registers 611
 FPU rounding control 613
 FPU stack fault flag 616
 FPU Status Register 615
 FPU Status register 633
 FPU top of stack pointer 618
 free 187
 Free function 413
 FRNDINT instruction 628
 FSIN instruction 631
 FSINCOS instruction 631
 FSQRT instruction 627
 FST instruction 622
 FSTCW instruction 633
 FSTP Instruction 622
 FSTSW instruction 615, 629
 FSTSW/FNSTSW instructions 633
 FSUB/FSUBP/FSUBR/FSUBRP instructions 625
 FTST instruction 630
 Full adders 223
 Function Computation via Table Look-up 647
 Function instance 1376
 Function numbers 208
 Function overloading 990
 Function results 557, 1370
 Functional units 255
 FXCH Instruction 622
 FYL2X instruction 632
 FYL2XP1 instruction 632

G

General protection fault 165
 General purpose registers 24
 Generating a unique label in an HLA program 984
 Get routine 46
 Getc routine 43
 Getting an integer value 44
 getY routine (standard library) 194
 Global memory locations as parameters 1341
 gotoxy routine (standard library) 193
 Guard digits/bits 88

H

H.O. 55
 Half adder 223

Handshaking 337
 Hard Copy storage (in the memory hierarchy) 305
 Harvard architecture 262
 Header files 576
 heap 187
 Hello World 20
 Hertz (Hz) 150
 Hexadecimal 56
 hexadecimal 53
 Hexadecimal Calculators 62
 Hexadecimal calculators 62
 Hexadecimal input (extended precision) 887
 Hexadecimal numbering system 60
 Hexadecimal output (extended precision) 879
 High order bit 55, 57
 High order byte 58
 High order nibble 57
 High order word 60
 High-speed devices 333
 History of the 80x86 CPU 234
 HLA 4
 Identifiers 19
 HLA pointers 410
 HLA Standard Library 12, 15, 38
 HLA stdlib
 stdin.get 22
 stdout.put 20
 HLA strings 421
 Hybrid control structures 802
 Hybrid parameter passing facilities 838

I

I/O 24, 331
 I/O address bus 140
 I/O and the cache 352
 I/O mapped input/output 331
 I/O port 327
 I/O Speed Hierarchy 333
 I/O subsystem 146
 I/O-mapped input/output 332
 iAPX432 microprocessor 235
 Icon programming language 428
 Identifiers 19
 Identity element for boolean operations 204
 Identity elements 204
 IEEE floating point standard (754 & 854) 90
 IF 30
 IF statement 32
 IF..THEN..ELSE 760, 761
 Implementation section of an abstract data type 1060
 IN instruction 332
 IN operator 31
 INC instruction 190
 INCLUDE directive 570
 Include files 20
 Indexed addressing mode 160

- Indexed addressing mode (x86) 279
- Indirect addressing mode 279
- Indirect calls 839
- Indirect jump 787
- Indirect jump instructions 753
- Indirect Jumps 784
- Indirect jumps 761
- Induction variables 801
- Industry Standard Architecture (ISA) 334
- Infinite loops 787
- Infinite precision arithmetic 87
- Infix notation 634
- Information hiding 1060
- Inheritance 1064, 1075
- INHERITS keyword (classes) 1065
- Inhibition function 1134
- Inhibition operation 207
- Initializing a string 935
- Initializing strings and arrays 946
- Input conditioning 651
- Input/output 24
- Inserting a bit field into some other value 911
- Instance 1376
- Instances (of a class) 1063
- Instruction composition 558
- Instruction pointer register 247
- Instruction set architecture 270
- int16 21
- int32 21
- int8 21
- Integer input 44
- Integer output 41
- Interface section of an abstract data type 1060
- Interrupt enable mask (FPU) 615
- Interrupt service routine 342
- Interrupt service routine (x86) 282
- Interrupt vector 343
- Interrupts 342
- INTMUL instruction 393
- INTO instruction 393
- Invalid operation exception (FPU) 614
- Invariant computations 799
- Inverse element 204
- Inverse element for boolean operations 204
- Inverting bits 68
- Inverting selected bits 913
- IRET instruction 343
- IS operator (object IS someType) 1094
- ISA bus 334
- ISR 342
- Iterators 843

J

- JA instruction 757
- JAE instruction 757
- JB instruction 757

- JBE instruction 757
- JC instruction 756
- JE instruction 757, 758
- JF Instruction 759
- JG instruction 758
- JGE instruction 758
- JL instruction 758
- JLE instruction 758
- JMP instruction 753
- JNA instruction 757
- JNAE instruction 757
- JNB instruction 757
- JNBE instruction 757
- JNC instruction 756
- JNE instruction 757, 758
- JNG instruction 758
- JNGE instruction 758
- JNL instruction 758
- JNLE instruction 758
- JNO instruction 756
- JNP instruction 756
- JNS instruction 756
- JNZ instruction 251, 756
- JO instruction 756
- JP instruction 756
- JPE instruction 756
- JPO instruction 756
- JS instruction 756
- JT instruction 759
- Julian day numbers 512
- JZ instruction 756

K

- Karnaugh Maps 203
- Kost significant bit 57

L

- L.O. 55
- Labels 751
- LAHF instruction 85
- lahf instruction 84
- Large parameters 832
- Large programs 569
- Last-in, first-out data structures 180
- Latency (of a cache access) 307
- Lazy evaluation 1354
- LEA instruction 191
- Leap years 507
- Least recently used (LRU) cache replacement 311
- Least significant bit 57
- Left associative 204
- Left associative operators 601
- Left shift operation 76
- Length (field of an HLA string) 422
- Length-prefixed strings 420

- Level One Cache 304
 - Level Two Cache 304
 - Lexical Nesting 1375
 - Lexical scope 547
 - Lexicographical ordering 437, 944
 - lf 40
 - LIB (library) files 581
 - LIB.EXE program 582
 - Libraries 581
 - Lifetime 170
 - Lifetime (of a variable) 547, 551
 - Lifetime of a variable 1376
 - LIFO 180
 - Linefeed 40
 - LINK.EXE program 582
 - Linker 569
 - Literal record constants 485
 - Literals (boolean) 209
 - Little endian data format 928
 - ln function 640
 - Local symbols in a macro 981
 - Local variables 547, 815
 - Locality of reference 153, 306
 - Locating the Cursor (standard library) 194
 - LOCK prefix 1459
 - LODS 935, 947
 - log function 640
 - Logic Instructions (MMX) 1133
 - Logical AND 204
 - Logical AND operation 65
 - Logical complement 204
 - Logical exclusive-OR 207
 - Logical exclusive-or operation 65, 67
 - Logical inhibition 207
 - Logical NAND 207
 - Logical NOR 207
 - Logical NOT 207
 - Logical NOT operation 65, 67
 - Logical Operations on Binary Numbers 68
 - Logical Operations on Bits 65
 - Logical operators within a constant expression 404
 - Logical OR 204
 - Logical OR operation 65, 66
 - Logical shift right 78
 - Logical XOR operation 65
 - Loop control variables 788
 - LOOP instruction 251
 - Loop invariant computations 799
 - Loop register usage 795
 - Loop termination 796
 - Loop termination test 787
 - Loop unraveling 800
 - Loops 787
 - LOOPZ and LOOPNZ instructions 341
 - Low Level Control Structures 751
 - Low order bit 55, 57
 - Low order byte 58
 - Low order nibble 57
 - Low order word 60
 - Low-speed devices 333
- ## M
- Machine idioms 606
 - Machine state, saving the 543
 - Macro parameter brackets 976
 - Macro parameter expansion 971
 - Macro parameters 971
 - Macros 969
 - Make files 578
 - malloc 187
 - Malloc function 412
 - Managing large programs 569
 - Managing libraries 581
 - Manifest constants 398
 - Mantissa 88
 - Map method for boolean function simplification 214
 - mask 910
 - Masking 68
 - Masking in bits 68
 - Masking out 57
 - Masking out bits 68
 - Masking out bits (setting them to zero) 910
 - MASM 8
 - MASM32 12
 - Maximum addressable memory 139
 - Maximum string length 421
 - MaxStrLen 422
 - Medium-level control structures 759
 - Medium-speed devices 333
 - Megahertz (Mhz) 150
 - Memory 24
 - Memory access 150
 - Memory access time 150
 - Memory access violation exception 414
 - Memory banks 143
 - Memory cells 229
 - Memory Hierarchy 303, 305
 - Memory mapped files 314
 - Memory protection 312
 - Memory subsystem 140
 - MemoryAllocationFailure exception 188
 - Memory-mapped I/O 331
 - Merging bit strings 929
 - Merging source files during assembly 570
 - Metaware Professional Pascal 1307
 - Methods 1061
 - Microprocessor clock 149
 - MIDI 114
 - MIMD (Multiple Instruction, Multiple Data) 268
 - Minimum field width 41
 - Mixed Integer and Floating Point Arithmetic 638
 - MM0, MM1, MM2, MM3, MM4, MM5, MM6, and MM7 (MMX Registers) 1114
 - MMU (memory management unit) 314

- MMX (multimedia extensions) 238
- MMX arithmetic instructions 1131
- MMX Comparison Instructions 1134
- MMX Data Types 1116
- MMX Instruction Operands 1118
- MMX Logic Instructions 1133
- MMX Programming Paradigm 1140
- MMX Registers 1114
- MMX Shift Instructions 1138
- mod (within a constant expression) 404
- MOD calculation using AND 345
- Modulo (floating point remainder) 628
- Monochrome displays 110
- MOV instruction 157
- Mov instruction 27
- MOVD instruction 1123
- Move strings 935
- MOVQ instruction 1123
- MOVS 935, 938
- Movsx instruction 74
- movzx instruction 74
- MP3 files 111
- Multi-precision Division 864
- MUL simulation 606
- Multidimensional arrays 468
- Multi-level page tables 313
- Multi-part macros 985
- Multiplication table 848
- Multiprecision addition 853
- Multi-precision comparisons 857
- Multiprecision operations 853
- Multiprecision subtraction 856
- Multiprocessing 268

N

- Name space pollution 496, 583
- Names of boolean functions 207
- NAMESPACE declarations 584
- Namespaces 496
- NAND gates 221
- NAND operation 207
- Near-Line Storage subsystems 305
- NEG instruction 71
- Negation (floating point) 629
- Negative numbers 70
- Nesting record definitions 488
- Nesting TRY..ENDTRY statements 730
- Network Storage (in the memory hierarchy) 304
- New line 41
- NEW memory allocation operator (C++ or Pascal) 187
- newln 41
- Newsgroups 8
- Nibble 56
- Nibbles 56
- nl 20, 40
- nl (newline) constant 403

- NOALIGNSTK option 813
- Nonuniform Memory Access (NUMA) 304
- NOR operation 207
- Normalized floating point numbers 620
- Normalized values 92
- NOT 605
- NOT IN operator 31
- NOT instruction 68
- NOT operation 65, 67, 204, 207
- NuBus bus 334
- NULL pointer references 165
- NUMA 304, 315
- Number of boolean functions 207
- Numeric Input 44
- Numeric output 41
- Numeric representation 63
- N-way set associative caches 309

O

- Object Initialization 1079
- Objects 1063
- Off-Line storage subsystems 305
- One-way set associative cache 308
- On-line and memory subsystems 304
- Opcodes 247
- Operation codes 247
- Operator precedence 600
- Operator Precedence and Associativity (compile-time operators) 955
- Opposite condition jump conditions 758
- Opposite jumps 758
- Optional macro parameters 975
- OR 65, 605
- OR instruction 68, 911
- OR Operation 66
- OR operation 204
- OUT instruction 332
- Out of Order Execution 266
- Outer product 848
- Outputting register values 176
- Overflow exception (FPU) 614
- Overflow flag 26
- overflow flag 592
- Overlapping blocks (string operations) 940
- Overloading 990
- Overriding a method 1065

P

- Packed arithmetic instructions 1131
- Packed arrays of bit strings 922
- Packed data 81
- Packed decimal arithmetic 901
- Packing and unpacking bit strings 917
- PACKSSDW instruction 1123
- PACKSSWB instruction 1123

- PACKUSDW instruction 1123
- PACKUSWB instruction 1124
- PADDB, PADDW, and PADD instructions 1131
- Padding a record to some number of bytes 491
- Padding parameter data 827
- PADDSB and PADDSW instructions 1131
- PADDUSB and PADDUSW instructions 1132
- Paging 312
- Palette (video card) 109
- PAND instruction 1133
- PANDN instruction 1133
- Parallel computation with MMX instructions 1117
- Parallel execution of instructions 253
- Parallel printer port 337
- Parameter expansion in macros 971
- Parameters 552, 816, 1341
- Parameters (macros) 971
- Parameters, variable length 821
- Parity flag 914
- Parse 286
- Pass by lazy evaluation 1354, 1395
- Pass by name 1395
- Pass by name parameters 1354
- Pass by reference 1395
- Pass by reference parameters 555, 817, 1354
- Pass by result 1395
- Pass by value 1394
- Pass by value parameters 552, 817, 1354
- Pass by value/returned 1354
- Pass by value/returned parameters 1354
- Pass by value-result 1395
- Passing large objects as parameters 832
- Passing parameters as parameters 836
- Passing parameters by name 1360
- Passing parameters by result 1359
- Passing parameters by value 1394
- Passing parameters from one procedure as parameters to another 1363
- Passing parameters in a parameter block 1341, 1353
- Passing parameters in global memory locations 1341
- Passing parameters in global variables 1346
- Passing parameters in registers 818, 1341, 1342
- Passing parameters in the code stream 820, 1341, 1351
- Passing parameters on the stack 822, 1341, 1347
- Passing reference parameters 834
- Passing value parameters 825
- Passing variables from different lex levels as parameters 1394
- Patch panel programming 246
- Pattern matching functions (compile-time) 958
- PCI bus 334
- PCMPEQB, PCMPEQW, and PCMPEQD instructions 1134
- PCMPGTB, PCMPGTW, and PCMPGTD instructions 1134
- PCMPLTx instructions 1136
- Pentium™ Processor 237
- Performance improvements for loops 796
- Performance of Memory Subsystems 306
- Peripheral Connection Interface (PCI) 334
- Pipeline flush 261
- Pipeline stalls 261
- Pipelined instruction execution 237
- Pipelining 259
- PMADDWD instruction 1132
- PMULHUW instruction 1132
- PMULHW instruction 1132
- PMULLW instruction 1132
- Pointer constants and pointer constant expressions 411
- Pointer errors 189
- Pointer problems 413
- POINTER TO type declaration 411
- Pointers 409
- Polled I/O 342
- polymorphism 1066
- POP instruction 177
- POPA and POPAD instructions 183
- POPF and POPFD instructions 184
- POR instruction 1133
- Port 327
- Positioning the Cursor 193
- Postfix notation 635
- Pound sign operator ("#") 100
- Precedence 204, 600
- Precision exception (FPU) 614
- Prefetch Queue 255
- Prefetch queue 256
- Preserving registers 179, 544
- Priming the pump (for output devices) 350
- Principle of duality 205
- Private fields in a class 1062
- Procedural parameters (passing procedures as parameters) 842
- Procedure call syntax 542
- Procedure instance 1376
- Procedure invocation 541, 805
- Procedure Overloading in classes 1085
- Procedure pointers 839
- Procedures and the Stack 807
- Processor size 139
- Product of maxterms representation 209
- Professional Pascal 1307
- Program unit 1380
- Programming in the large 569
- PSARW and PSARD instructions 1138
- Pseudo-opcode 166
- PSLLW, PSLLD, and PSLLQ instructions 1138
- PSLRW, PSLRD, and PSLRQ instructions 1138
- PSUBB, PSUBW, and PSUBD instructions 1132
- PSUBSB and PSUBSW instructions 1132
- PSUBUSB and PSUBUSW instructions 1132
- PUNPCKHBW instruction 1124
- PUNPCKHDQ instruction 1124
- PUNPCKHWD instruction 1124
- PUNPCKLBW instruction 1124
- PUNPCKLDQ instruction 1124

PUNPCKLWD instruction 1124
PUSH instruction 176
PUSHA instruction 183
PUSHAD instruction 183
PUSHD instruction 176
PUSHF and PUSHFD instructions 184
PUSHW instruction 176
Put routine 42
putiXsize 41
PXOR instruction 1133

Q

Quicksort 564
Quicktime 115
QWORD data type 397
qwords 56

R

radix 61
RAISE statement 427, 735
Range of a function 649
RCL instruction 80
RCR instruction 80
Read control line 140
Read/write input/output ports 327
Read/write ports 329
Reading from memory 141
Reading integer values 44
Read-only (input) ports 327
READONLY declaration section 167
Read-only ports 329
READONLY variables as constants 398
Realloc function 413
Rearranging expressions to make them more efficient 773
Record constants 485
Record field alignment 490
Record offsets 489
Records 483
Records as record fields 487
Recursion 563
Reference parameters 831, 834
Register addressing modes 157
Register indirect addressing mode 159
Register indirect jump instruction 753
Register preservation 544
Register preservation in a TRY..ENDTRY statement 739
Register Renaming 266
Register type coercion 175
Register usage in loops 795
Registers 24
Registers (electronic implementation) 230
Registers (in the memory hierarchy) 303
Registers as procedure parameters 818, 1341, 1342
Registers as signed integer values 175
Relational operators 31

Remainder (floating point) 628
Removing unwanted data from the stack 184
REPEAT 30
Repeat Until loop 788
REPEAT..UNTIL loops 787
REPEAT..UNTIL statement 35
Replacement policy (for caches) 310
Representing audio information 111
Required macro parameters 975
Resume frame (for iterators) 1308
RET instruction 541, 805
RETURNS Option 560
Reverse polish notation 634
Reversing a bit string 927
RGB color space 109
Right associative operators 204, 601
Right shift operation 77
Rising edge of a clock 150
ROL instruction 79
ROR instruction 79
Rotate left 79
Rotate right 79
Rounding a floating point value to an integer 628
Rounding control 613
Rounding control (FPU) 613
Row major ordering 469
RPN 634
Run of ones 909
Run of zeros 909
Run-time language 949
Run-time Type Information 1094

S

SAHF instruction 85, 629
sahf instruction 84
SAR instruction 79
Saturation 73
Saturation arithmetic 1118
Saving the machine state 543
SBB instruction 856
Scanning for bits 923
SCAS 935, 946
Schematic Symbols 221
Scope 1375
Scope (of a name) 547
Searching for a bit 923
Searching for a bit pattern 931
Searching for data within a string 935
secant 639
Self-modifying code 386
Separate compilation 569
Sequential logic 228
Set/reset flip-flop (SR flip-flop) 229
SETcc Instructions 593
setOutputAttr routine (standard library) 196
Setting selected bits 911

- Seven segment decoder 223
- Shift arithmetic right operation 79
- Shift Instructions (MMX) 1138
- Shift registers 230
- SHL instruction 76
- SHLD and SHRD instructions 876
- Short circuit boolean evaluation 769
- SHR instruction 77
- SI 24
- Side effects 562
- Sign bit 70
- Sign extension 73, 590
- Sign flag 26
- sign flag 592
- Signed 69
- Signed and unsigned numbers 69
- Signed comparisons 594
- Signed decimal input (extended precision) 895
- Signed decimal output (extended precision) 882
- Signed division 590
- Signed integer output 41
- Significant digits 88
- SIMD 1117
- SIMD (Single Instruction, Multiple Data) 268
- Simplification of boolean functions 214
- Simulating DIV 607
- Simulating MUL 606
- Sine 631
- Single Instruction Multiple Data model 1117
- Single Instruction, Single Data execution model 268
- Single precision floating point format 90
- SISD (single instruction, single data) 268
- Sixteen-bit bus data access 144
- Size of a processor 139
- SNOBOL4 programming language 428
- Source index 936
- SP 24
- Spaghetti code 786
- Spatial locality of reference 153
- Square root 627
- SR (set/reset) flip flop 229
- ST0..ST7 (FPU registers) aliasing with MMX registers 1114
- Stack fault flag (FPU) 616
- Stack frame 810, 1308
- Stack manipulation by procedure calls 807
- Stack Segment 176
- Stack-based parameters for procedures 1341
- Stalls 261
- Standard entry sequence (to a procedure) 813
- Standard exit sequence (from a procedure) 814
- Standard input 40
- Standard Library 38
- Standard Macros 969
- Standard output 40
- State machine 784
- State machines 232
- State variable 784
- Statement Labels 751
- Static data objects in a class 1063
- STATIC declaration section 167
- Static declaration section 21
- Static link 1378
- Static Procedures (in a class) 1066
- std instruction 84
- Stdin.a_gets function 425
- stdin.eoln 103
- Stdin.FlushInput 46
- stdin.FlushInput 103
- stdin.get 22, 65, 102
- Stdin.Get routine 46
- Stdin.getc 43
- stdin.getdw 65
- stdin.getf function 638
- stdin.geth 65
- Stdin.gets function 425
- stdin.getu16 72
- stdin.getu32 72
- stdin.getu8 72
- stdin.getw 65
- Stdin.ReadLn 46
- stdio.bell 40
- stdio.bs 40
- stdio.cr 40
- stdio.lf 40
- stdio.tab 40
- stdlib.hhf 20
- stdout.newln 41, 541
- stdout.newln function 805
- stdout.put 20, 42, 65, 101
- stdout.putc 101
- stdout.putcsize 101
- stdout.putdw 65
- stdout.puth 65
- stdout.puti16 41
- stdout.puti32 41
- stdout.puti8 41
- stdout.putiXsize 41
- stdout.putr32 94
- stdout.putr64 94
- stdout.putr80 94
- stdout.putu16 72
- stdout.putu16size 72
- stdout.putu32 72
- stdout.putu32size 72
- stdout.putu8 72
- stdout.putu8size 72
- stdout.putw 65
- STI instruction 348
- sti instruction 84
- STORAGE declaration section 168
- Stored program computer systems 246
- Storing double words in byte addressable memory 141
- Storing words in byte addressable memory 141
- STOS 935, 946, 947
- str.a_cat function 433

- Str.a_cpy function 432
- str.a_delete function 435
- str.a_insert function 435
- str.a_substr function 435
- str.cat function 433
- str.cpy function 430
- str.delete function 435
- str.eq function 436
- str.ge function 436
- str.gt function 436
- str.ieq function 436
- str.ige function 437
- str.igt function 437
- str.ile function 437
- str.ilt function 437
- str.index function 437
- str.in_ function 437
- str.insert function 435
- str.le function 436
- str.length function 433
- str.lt function 436
- str.ne function 436
- str.strRec data type 422
- str.strRec definition 489
- str.substr function 435
- str.uppercase function 1142
- Stralloc function 423
- Strfree function 424
- String assignment by reference 428
- String comparisons 436
- String concatenation 401, 433
- String constant initializers in the CONST section 402
- String constants 401
- String constants containing control characters 402
- String Functions (compile-time functions) 958
- String instructions 935
- String Operators within a constant expression 404
- String pointers 421
- String primitives 935
- String representation 489
- STRUCT assembler directive 483
- Structure, accessing fields of... 484
- Structured gotos 740
- Structures 483
- Structures as structure fields 487
- SUB 28
- Subroutine instance 1376
- Substring operation 435
- Subtraction table 848
- Sum of minterms representation 209
- Superscalar CPUs 237, 265
- SWITCH Statement 776
- SWITCH statement 747
- Symbol tables 287
- Symbols reserved by HLA 982
- Symbols that begin and end with a single underscore 982
- Synthesizing a While loop 787
- System bus 24, 138

- System Buses 334
- System clock 149
- System clock frequency 150
- System clock period 150
- System date function 509
- System time 514
- System timing 149

T

- tab 40
- Tables 647
- Tag field 495
- Taking the address of a statement label 751
- Tangent 632
- TBYTE data type 397
- Tbyte values (BCD) 902
- Temporal locality of reference 153
- Temporary values in an expression 603
- TenToX function 640
- Term (boolean) 209
- Termination test (for loops) 787
- Termination test for loops 796
- Test for zero (floating point) 630
- TEST Instruction 596
- TEST instruction 338, 914
- Text Attributes (on the display) 195
- Text constants 402, 492
- THEN 30
- Theorems of boolean algebra 204
- THIS 1069
- Thrashing 314
- Thunk 1361
- Time 514
- Time Input/Output 515
- time.curTime function 514
- time.hmsToSecs function 515
- time.secstoHMS function 515
- time.timerec definition 514
- Time-outs on peripheral devices 340
- Translation Lookaside Buffer (TLB) 313
- Treating registers as signed integer values 175
- True (representation) 604
- Truth maps 214, 215
- truth table 66
- Truth tables 205
- TRY..ENDTRY statement 37, 729
- TTL logic levels 138
- Two level caching system 155
- Two's complement 59
- Two's complement representation 70
- TwoToX function 640
- Two-way set associative caches 309
- Type coercion 173, 491
- Type conversion 957
- TYPE declaration section 407
- Type operator 174

U

UCR Standard Library for 80x86 Assembly Language
 Programmers 3
 Underflow exception (FPU) 614
 UNICODE 59, 108, 1124
 Uninitialized pointers 413
 Unions 492
 Unique boolean functions 207
 Unit activation 1376
 UNITS 572
 Universal boolean function (NAND) 221
 Universal boolean functions (NOR) 223
 Unpacking bit strings 917
 Unprotected (try..endtry) 732
 Unraveling loops 800
 Unravelling loops 999
 Unrolling loops 999
 Uns16 72
 Uns32 72
 Uns8 72
 Unsigned comparisons 594
 Unsigned decimal input (extended precision) 891
 Unsigned Decimal Output (extended precision) 879
 Unsigned division 590
 unsigned multiplication 588
 Unsigned numbers 69
 Unsigned variable declarations 72
 UNTIL 30, 35
 Untyped Reference Parameters 843
 Upper case conversion (MMX) 1144
 User-defined exceptions 735

V

VAL (value parameter specification) 554
 VAL declaration section 406
 VAL declarations 397
 Value parameters 824
 VAR (pass by reference parameters) 555
 VAR declarations 169
 Variable length parameters 821
 Variable lifetime 170, 1375, 1376
 Variable number of macro parameters 974
 Variable-length instructions 274
 Variant types 495
 Vars (_vars_) constant in a procedure 816
 Veitch Diagrams 203
 Very Long Instruction Word 267
 Video and audio data 1117
 Video display 109
 Virtual Memory 304, 312
 Virtual method calls 1066
 Virtual method table 1072
 Virtual Method Tables 1073
 Virtual Methods 1066
 VMT 1072, 1075

Von Neuman Architecture 24
 Von Neumann, John 137

W

Wait states 151
 WAV files 111
 WHILE 30
 While loop 787
 WHILE loops 787
 WHILE statement 33
 Word access in byte addressable memory 141
 Word strings 935
 Words 56, 58
 Words stored at odd addresses 145
 Working sets 314
 Wraparound arithmetic 1118
 Write control line 140
 Write-back cache write policy 311
 Write-only ports 329
 Write-through cache write policy 311
 Writing to memory 140

X

x86 conditional jumps 282
 XLAT instruction 648
 XOR 605
 XOR instruction 68, 913
 XOR operation 65, 67

Y

Y2K 83
 Y86 Addressing modes 278
 Y86 Hypothetical Processor 276
 Y86 opcodes 279
 Yield 844
 YtoX function 640

Z

Z80 microprocessor 234
 Z8000 microprocessor 235
 Zero divide exception (FPU) 614
 Zero extension 590
 Zero flag 26
 zero flag 592
 Zero terminating byte (in HLA strings) 421
 Zeroing selected bits 910
 Zero-terminated strings 419

