

INDEX

Numbers

2 to the x minus 1, 402
8-bit excess-127 exponent, 93
8-bit registers, 9
16-bit registers, 9
80x86 memory addressing
 modes, 111
96-bit rcl and rcr operations, 508
128-bit by 32-bit division, 494
128-bit comparisons, 488
8087 FPU, 93, 381

Symbols

&& operator, 22–23
@a, 360
@abs compile-time function, 561
@ae, 360
@align procedure option, 657
@b, 360
@be, 361
@c, 18, 360
@cdecl procedure option, 657
@ceil compile-time function, 561
@cos compile-time function, 561
@defined compile-time function,
 562, 568
@delete compile-time function, 562
@e, 361
@elements compile-time
 function, 562
@elementSize compile-time
 function, 562
@exp compile-time function, 561
@floor compile-time function, 561
@g, 361

@ge, 361
@global operator (in
 namespaces), 250
@index compile-time function, 562
@insert compile-time function, 562
@isAlpha compile-time function, 561
@isAlphanum compile-time
 function, 561
@isDigit compile-time function, 561
@isLower compile-time function, 561
@isSpace compile-time function, 561
@isUpper compile-time function, 561
@isxDigit compile-time
 function, 561
@l, 361
@le, 361
@length compile-time function, 562
@lineNumber compile-time
 function, 563
@log compile-time function, 561
@log10 compile-time function, 561
@lowercase compile-time
 function, 562
@max compile-time function, 561
@min compile-time function, 561
@na, 360
@nae, 360
@nb, 360
@nbe, 361
@nc, 18, 360
@ne, 361
@ng, 361
@nge, 361
@nl, 361
@nle, 361
@no, 18, 360

@noalignstack option, 288, 297, 657
 @nodisplay procedure option, 288, 657
 @noframe procedure option, 288, 291, 657
 @nostorage variable option, 124, 186
 @ns, 18, 360
 @nz, 18, 360
 @o, 18, 360
 @odd compile-time function, 563
 @pascal procedure option, 657
 @random compile-time function, 561
 @randomize compile-time function, 561
 @real32 compile-time function, 560
 @returns procedure option, 280, 657
 @rindex compile-time function, 562
 @s, 18, 360
 @sin compile-time function, 561
 @size compile-time function, 148, 245, 562
 @sqrt compile-time function, 561
 @stdcall procedure option, 657
 @strbrk compile-time function, 562
 @string compile-time function, 564
 @strset compile-time function, 562
 @strspan compile-time function, 562
 @substr compile-time function, 562
 @tan compile-time function, 561
 @text compile-time function, 563
 @tokenize compile-time function, 562
 @toString: compile-time function, 564
 @trim compile-time function, 562
 @typename compile-time function, 562, 588
 @uppercase compile-time function, 562
 @use procedure option, 317, 324, 657
 @z, 18, 360
 := (CTL assignment operator), 555
 !register, 429
 # (numeric character constant prefix), 105
 #{ (hybrid parameter passing syntax), 327
 #{ and }# operators, 474
 #else compile-time statement, 566
 #elseif compile-time statement, 566
 #endif compile-time statement, 566
 #endfor compile-time statement, 570, 572, 596
 #endif compile-time statement, 566
 #endwhile compile-time statement, 570
 #error CTL statement, 553–554
 #for..#endif compile-time statement, 570, 572, 596
 #if compile-time statement, 565
 #include, 3
 declarations, 344
 directive, 598, 654
 #includeonce directive, 338
 #macro declaration, 574
 #print CTL statement, 553
 #while..#endwhile compile-time statement, 570
 || operator, 23
 finalize strings in a procedure, 687
 initialize strings in a procedure, 687
 pVMT field in an object, 678
 vars constant, 300
 VMT field in a class, 672

A

aaa instruction, 535
 aad instruction, 535
 aam instruction, 535
 aas instruction, 535
 ABI (application binary interface), 302
 Absolute value (floating point), 399
 Abstract
 base class, 693
 data types, 653
 keyword, 695
 methods, 693
 Accessing
 characters within a string, 194
 data on the stack, 146
 data via a pointer, 177

- an element of a single-dimensional array, 219
 - elements of 3- and 4-dimensional arrays, 226
 - elements of an array, 221
 - elements of multidimensional arrays, 231
 - fields of a union, 244
 - local variables, 299
 - names outside a namespace, 249
 - reference parameters, 318
 - value parameters, 310
- Accessor methods, 653
- Accumulated errors in a floating-point calculation, 90
- Activation record
 - construction at runtime, 294
 - definition, 293
- Actual parameters, 324
- adc instruction, 482, 609
- add instruction, 15
- Adding 1 to a register or memory location, 152
- Adding an integer to a floating-point value, 405
- Addition (extended precision), 480
- Address expressions, 131
- Addresses in a virtual method table, 672
- Addressing modes, 112
- Address-of operator, 115, 152, 414
- Aggregate data types, 185
- ah, 9
 - copying ah to flags register, 88–89, 400
- al, 9
- al/ax/eax register usage in string instructions, 635
- Aliases, 198, 245, 275, 344
- align directive, 130
- Aligning
 - bit strings, 602
 - fields within a record, 241
- Alignment
 - data, 128
 - variable alignment, 131
 - within a record, 241
- Alloc memory function, 148
- Allocating storage for arrays, 229.
 - See also* Arrays
- Alphabetic character, 101
- and instruction, 70, 376, 601, 605–606
- and operation, 67
- Anonymous
 - memory objects, 304
 - unions, 246
 - variable type coercion, 149
 - variables, 115
- anyexception (try..endtry), 48
- Application binary interface (ABI), 302
- Arc tangent, 403
- arg (HLA stdlib module)
 - arg.c, 346
 - arg.v, 346
- Arithmetic
 - expressions, 351, 365–369
 - idioms, 377
 - logical systems, 377
 - operators within a constant expression, 169
 - shift right, 82–83
- Arity, 252
- Array variables, 219
- Arrays
 - arrays of arrays, 227
 - arrays of records, 236
 - definition, 218
 - multidimensional arrays, 224, 227
 - structure fields, 237
- ASCII
 - character set, 61, 101
 - codes for numeric digits, 104
 - groups, 104
- Assigning
 - constant to a variable, 366
 - one variable to another, 366
- Assignment by reference, 197
- Assignments, 366
- Associativity, 369–370
 - in compile-time expressions, 557
- Automatic (local) variables, 125, 299

- Automatic storage allocation, 267
- Avoiding branches by using
 - calculations, 441
- ax, 9
- B**
- Backspace, 34
- Base address (of an array), 219
- Base class methods, 698
- Base classes, 673
- BCD
 - arithmetic, 532
 - numbers, 59
 - representation, 532
- Bell character, 34
- Benefits of object-oriented
 - programming, 652
- bh, 9
- Biased (excess) exponents, 94
- Binary
 - data types, 58
 - digits, 55
 - formats, 55
 - fractions, 93
 - numbering system, 53–54
 - point (binary fractions), 93
- Binary-coded decimal
 - arithmetic, 532
 - instructions, 477
 - numbers, 59
- Bit, 55, 58
 - complement, 601
 - counting, 620
 - data, 600
 - extraction, 601
 - fields, 85
 - inversion, 69, 70, 601, 604
 - manipulation, 599–601
 - mask, 600
 - offset, 600
 - packed data, 85
 - pattern search, 627
 - runs, 600
 - sets, 600
 - string
 - alignment, 602
 - arrays, 615
 - coalescing, 612
 - distribution, 612
 - extraction, 626
 - HLA strings, 70, 600–601
 - merging, 625
 - reversal, 623
 - testing for 1 bits, 607
 - testing for set bits, 607
 - testing, 601
- Bit-by-bit operations, 70, 170
- bits (HLA stdlib module), 628
 - bits.cnt, 629
 - bits.coalesce, 629
 - bits.distribute, 629
 - bits.extract, 630
 - bits.merge8, bits.merge16, and bits.merge32, 630
 - bits.nibbles8, bits.nibbles16, and bits.nibbles32, 630
 - bits.reverse8, bits.reverse16, and bits.reverse32, 630
- Bitwise operations, 70, 170
- bl, 9
- Block copy performance, 642
- Boolean
 - evaluation
 - complete, 432
 - short-circuit, 433
 - expressions, 18, 375
 - logical systems, 377
 - values, 59
 - variables in an if statement, 429
- boolean compile-time function, 559
- bound instruction, 156, 157
- bp, 9
- Bracketing characters in macro
 - parameters, 578
- Branch avoidance using
 - calculations, 441
- Branch out of range, 420
- break statement, 27, 461
- breakif statement, 27
- bsf instruction, 618
- bsr instruction, 618
- bt instruction, 608
- btc instruction, 211, 608
- btr instruction, 211, 608
- bts instruction, 211, 608
- Busy bit (FPU), 387

- bx, 9
- Byte, 60
- byte compile-time function, 559
- Byte strings, 633
- Bytes, 58

C

- C integer types, 478
- C programming language, 478
- C/C++ switch statement, 451
- Cache, 12
- Call indirect, 329
- call instruction, 255–256, 288
- Callee/caller register
 - preservation, 259
- Calling base class methods, 698
- Carriage return character, 34
- Carry flag, 10, 358, 418
 - and, or, and xor instruction effect, 605
 - as a bit accumulator, 609
- Case
 - insensitive comparison, 207
 - labels (noncontiguous), 450
 - neutral identifiers, 2
- case statement, 423, 442
- cbw instruction, 77
- cdq instruction, 77
- Central processing unit (CPU), 8
- ch, 9
- Change sign (floating point), 399
- Changing the value of a val
 - object, 173
- char
 - compile-time function, 559
 - data type, 106
- Character
 - classification compile-time functions, 561
 - constants, 165
 - data type, 101
 - literal constants, 105, 165
 - strings, 185
- Character sets, 209
 - expressions, 212
 - implementation, 210
 - operators, 213
- Choosing an alignment value for
 - variables, 131
- c1 (register), 9
 - in rotate operations, 84
 - in shl instruction, 81
- Classes
 - class implementation in HLA, 654
 - classes and objects, 651
 - information hiding, 683
 - procedures vs. methods, 663
- c1c instruction, 88, 609
- c1d instruction, 88
- c1i instruction, 88
- Clipping (saturation), 80
- cmc instruction, 88, 609
- cmp instruction, 357
- cmps string instruction, 644
- cmpsb instruction, 634
- cmpsd instruction, 634
- cmpsw instruction, 634
- Coalescing bit strings, 612
- Code sections, 120
- Coercion, 111, 133
- Column-major ordering, 225, 228
- Command-line compiler, 5
- Comments, 7
- Commutative operators, 374–375
- Comparing
 - bits, 601
 - dates, 88
 - floating-point numbers, 92
 - a register to zero, 365
 - registers with signed integer values, 136
 - strings, 206–207, 633, 645
- Comparison operators in a constant expression, 169
- Comparisons
 - dates, 88
 - floating point, 92, 386
 - unsigned, 363
- Compile-time
 - conversion of text objects, 564
 - decisions, 565
 - expressions and operators, 555
 - functions, 148, 558. *See also* Compile-time functions

- Compile-time, *continued*
 - language, 551
 - debugging, 554
 - loops, 570
 - operators, 556
 - procedures, 573, 585
 - programs, 592
 - string functions, 561
 - symbol information, 562
- Compile-time functions
 - @abs, 561
 - boolean, 559
 - byte, 559
 - @ceil, 561
 - char, 559
 - character classification, 561
 - @cos, 561
 - cset, 559
 - @defined, 562, 568
 - @delete, 562
 - dword, 559
 - @elements, 562
 - @elementSize, 562
 - @exp, 561
 - @floor, 561
 - @index, 562
 - @insert, 562
 - int8, 559
 - int16, 559
 - int32, 559
 - int64, 559
 - int128, 559
 - @isAlpha, 561
 - @isAlphanum, 561
 - @isDigit, 561
 - @isLower, 561
 - @isSpace, 561
 - @isUpper, 561
 - @isxDigit, 561
 - @length, 562
 - @lineNumber, 563
 - @log, 561
 - @log10, 561
 - @lowercase, 562
 - lword, 559
 - @max, 561
 - @min, 561
 - numeric CTL functions, 561
 - @odd, 563
 - qword, 559
 - @random, 561
 - @randomize, 561
 - @real32, 560
 - real32, 559
 - real64, 559
 - real80, 559
 - @sin, 561
 - @size, 148, 245, 562
 - @sqrt, 561
 - @strbrk, 562
 - @string, 564
 - string, 559–560
 - @strset, 562
 - @strspan, 562
 - @substr, 562
 - @tan, 561
 - @text, 563
 - text, 559
 - @tokenize, 562
 - @toString:, 564
 - @trim, 562
 - type conversion, 559
 - @typename, 562, 588
 - uns8, 559
 - uns16, 559
 - uns32, 559
 - uns64, 559
 - uns128, 559
 - @uppercase, 562
 - word, 559
- Compile-time statements
 - #else, 566
 - #elseif, 566
 - #endif, 566
 - #endif, 566
 - #endwhile, 570
 - #error, 553, 554
 - #for, 570, 572, 596
 - #if, 565
 - #print, 553
 - #while, 570
- Complete boolean evaluation, 432
- Complex arithmetic
 - expressions, 369
- Complex string functions, 649
- Composite data types, 185

- Concatenating two string literals, 165
- Concatenation, 203
- Condition codes
 - @a, 360
 - @ae, 360
 - @b, 360
 - @be, 361
 - @c, 18, 360
 - @e, 361
 - @g, 361
 - @ge, 361
 - @l, 361
 - @le, 361
 - @na, 360
 - @nae, 360
 - @nb, 360
 - @nbe, 361
 - @nc, 18, 360
 - @ne, 361
 - @ng, 361
 - @nge, 361
 - @nl, 361
 - @nle, 361
 - @no, 18, 360
 - @ns, 18, 360
 - @nz, 18, 360
 - @o, 18, 360
 - @s, 18, 360
 - @z, 18, 360
 - flags, 10
 - FPU condition codes, 385
 - settings after `cmp` instruction, 359
- Conditional
 - compilation, 565
 - as a debugging tool, 569
 - `jmp` aliases, 420
 - `jmp` instructions, 418–420
 - statements, 423
- Conditional jumps
 - ja, 419
 - jae, 419
 - jb, 419
 - jbe, 419
 - jc, 419, 609
 - je, 419
 - jf, 421
 - lg, 419
 - jge, 419
 - jl, 419
 - jle, 419
 - jna, 419
 - jnae, 419
 - jnb, 419
 - jnbe, 419
 - jnc, 419, 609
 - jne, 419
 - jng, 419
 - jnge, 419
 - jnl, 419
 - jnle, 419
 - jno, 419
 - jnp, 419
 - jns, 419
 - jnz, 419
 - jo, 419
 - jp, 419
 - jpe, 419
 - jpo, 419
 - js, 419
 - jt, 421
 - jz, 419
- Configuring software for several different environments, 567
- Console application, 3
- `const`
 - declarations, 160
 - fields in a class, 656
 - sections in an HLA program, 171
- Constant
 - 0.0 (FPU load instruction), 402
 - expressions, 132, 169
 - expressions in CTL statements, 555
 - log base 2 of e , 402
 - log base 2 of 10, 402
 - log base e of 2, 402
 - log base 10 of 2, 402
 - `nl` (newline), 33, 168
 - `pi`, 402
- Constructing data tables at compile time, 592
- Construction of an activation record, 294

- Constructors, 678
 - definition, 677
 - inheritance with, 681
 - object initialization, 677
 - parameters, 685
 - Container object (for bits), 600
 - continue statement, 461
 - Control characters, 103
 - within string constants, 167
 - Control word, 384, 404
 - conv (HLA stdlib module)
 - conv.getDelimiters function, 517
 - conv.setDelimiters function, 517
 - Conversions (floating-point instructions), 391
 - Converting
 - arithmetic expressions to postfix notation, 407
 - BCD to floating point, 393
 - binary to hex, 57
 - break statements to pure assembly language, 461
 - complex expressions to assembly language, 369
 - continue statements to pure assembly language, 462
 - floating-point expressions to assembly language, 406
 - for loops to pure assembly, 460–461
 - forever loops to pure assembly, 456, 460
 - hex to binary, 57
 - if statement to pure assembly language, 422–424
 - integer to floating point, 392
 - noncommutative arithmetic operators to assembly language, 372
 - postfix notation to assembly language, 409
 - repeat..until loops to pure assembly, 458
 - simple expressions to assembly language, 366
 - while loops to pure assembly, 457
 - Copying
 - arbitrary number of bytes using the movsd instruction, 642
 - overlapping arrays using the movs string instructions, 641
 - by reference, 198
 - strings, 201, 346
 - Cosine, 403
 - Counting bits, 620
 - in a 16-bit operand, 629
 - CPU registers, 9
 - Creating lookup tables, 545
 - cs (HLA stdlib module)
 - cs.charToCset function, 213–214
 - cs.cpy function, 213–214
 - cs.empty function, 213
 - cs.eq function, 216
 - cs.extract function, 217
 - cs.IsEmpty function, 215
 - cs.member function, 216, 276
 - cs.ne function, 216
 - cs.psubset function, 216
 - cs.psuperset function, 216
 - cs.rangeChar, 213–214
 - cs.removeChar function, 213–214
 - cs.strToCset function, 213, 215
 - cs.subset function, 216
 - cs.superset function, 216
 - cs.unionChar function, 213–214
 - cs.unionStr function, 213, 215
 - cset compile-time function, 559
 - CTL (compile-time language), 551
 - CTL assignment statement, 555
 - Current string length, 187
 - cwd instruction, 77
 - cwde instruction, 77
 - cx, 9
- D**
- daa instruction, 534–535
 - Dangling pointers, 182
 - das instruction, 534–535
 - Data alignment, 128
 - Data representation, 155
 - Data tables, constructing at compile time, 592

- Data type coercion, 111
- Date comparison, 88
- Debugging CTL programs, 554
- Debugging with conditional compilation, 569
- dec instruction, 152
- Decimal arithmetic, 100, 477, 532
- Decimal input, unsigned (extended precision), 525
- Decimal output, unsigned (extended precision), 510
- Decimal numbering system, 54
- Decisions in HLA, 17, 422–424
- Declarations
 - readonly, 123
 - static, 122
 - storage, 123
 - type, 173
 - val, 172
 - var, 125
 - VMT, 672
- Declaring local symbols in a macro, 584
- default clause in an HLA switch statement, 448
- delete memory deallocation operator (C++), 147
- Delimiters character set, 516
- Delimiting macro parameters, 578
- Denormalized
 - exception (FPU), 383
 - floating-point values, 388
 - values, 96
- Destructors, 686
- Destructured code, 439
- dh, 9
- di, 9
- Different-sized operands, 530
- Direct addressing mode, 113
- Direct jump instructions, 416
- Direction flag, 635–637
- Directives
 - align, 130
 - external, 338
 - forward, 286, 692
 - #include, 336
- Disadvantages of macros (vs. procedures), 575
- Displacement-only addressing mode, 112–113
- Display (in an activation record), 289
- dispose memory deallocation operator (Pascal), 147
- Distributing bit strings, 612
- div (within a constant expression), 169
- div and idiv instructions, 355, 492
- Divide-by-zero exception (FPU), 384
- Division, unsigned, 355–356
- Division without div or idiv, 379
- d1, 9
- Domain conditioning, 544
- Domain of a function, 541
- Dope vector, 252
- Dot operator, 234
- Double-precision floating-point format, 95
- Double words, 58, 62
- Double-word strings, 633
- dup operator, 221
- Duplicate symbol errors in macro expansions, 583
- dword compile-time function, 559
- dwords, 58
- dx, 9
- Dyadic operations, 67
- Dynamic
 - arrays, 251
 - memory allocation, 112, 147, 180
 - nesting of control structures, 44
 - string length, 187
 - type systems, 247
- Dynamically nesting try..endtry statements, 43

E

- eax, 9
- ebp, 9
- ebx, 9
- ecx, 9

- ecx register usage in string instructions, 635
- edi, 9
- edi register usage in string instructions, 635
- edx, 9
- Effective address, 116, 153, 322
- eflags register, 10, 89, 144
- else statement, 17, 424
- elseif statement, 17
- Embedding control characters in string constants, 167
- End of line, 108
- endfor statement, 25
- endif statement, 17
- endwhile statement, 17, 24
- enum declaration, 175
- Enumerated data types, 174
- Errors in a floating-point calculation, 90
- Errors when using pointers, 150
- Escape character sequences, 166
- esi, 9
- esi register usage in string instructions, 635
- esp, 9
- Exception
 - flags (FPU), 385
 - FPU exception bits, 405
 - handling, 28
 - masks (FPU), 383
 - overflow (FPU), 384
- Exception codes
 - ex.AccessDenied, 30
 - ex.AccessViolation, 31
 - ex.ArrayBounds, 31
 - ex.ArrayShapeViolation, 30
 - ex.AssertionFailed, 31
 - ex.AttemptToDerefNULL, 30
 - ex.AttemptToFreeNULL, 30
 - ex.BadFileHandle, 30
 - ex.BadObjPtr, 29
 - ex.BlockAlreadyFree, 30
 - ex.BoundInstr, 31, 157
 - ex.Breakpoint, 31
 - ex.BufferOverflow, 30
 - ex.BufferUnderflow, 30
 - ex.CannotCreateDir, 30
 - ex.CannotFreeMemory, 30
 - ex.CannotRemoveDir, 30
 - ex.CannotRemoveFile, 30
 - ex.CannotRenameFile, 30
 - ex.CDFailed, 30
 - ex.ControlC, 31
 - ex.ConversionError, 30, 46, 516–517
 - ex.DiskFullError, 30
 - ex.DivideError, 31, 493
 - ex.DivisionError, 356, 501
 - ex.EndOfFile, 30
 - ex.ExecutedAbstract, 31
 - ex.fDenormal, 31, 383
 - ex.fDivByZero, 31
 - ex.FileCloseError, 30
 - ex.FileNotFoundException, 30
 - ex.FileOpenFailure, 30
 - ex.FileReadError, 30
 - ex.FileSeekError, 30
 - ex.FileWriteError, 30
 - ex.fInexactResult, 31
 - ex.fInvalidOperation, 31, 383
 - ex.fOverflow, 31
 - ex.FractionTooBig, 30
 - ex.fStackCheck, 31
 - ex.fUnderflow, 31
 - ex.IllegalChar, 29, 517
 - ex.IllegalInstr, 31
 - ex.IllegalSize, 30
 - ex.IllegalStringOperation, 29
 - ex.InPageError, 31
 - ex.IntoInstr, 31, 159, 356
 - ex.InvalidAlignment, 30
 - ex.InvalidArgument, 30
 - ex.InvalidDate, 31
 - ex.InvalidDateFormat, 31
 - ex.InvalidHandle, 31
 - ex.InvalidTime, 31
 - ex.InvalidTimeFormat, 31
 - ex.MemoryAllocationCorruption, 29
 - ex.MemoryAllocationFailure, 30, 149
 - ex.MemoryFreeFailure, 30
 - ex.NoMemory, 31
 - ex.PointerNotInHeap, 30
 - ex.PrivInstr, 31

- ex.SingleStep, 31
- ex.SocketError, 31
- ex.StringAlignment, 29
- ex.StringIndexError, 29, 196
- ex.StringMetaData, 29
- ex.StringOverflow, 29, 191, 203, 205
- ex.StringOverlap, 29
- ex.StringUnderflow, 29
- ex.ThreadError, 31
- ex.TimeOverflow, 31
- ex.TooManyCmdLnParms, 29
- ex.ValueOutOfRange, 29, 46, 529
- ex.WidthTooBig, 30
- exception statement, 28
- Exceptions, divide by zero (FPU), 384
- Excess (biased) exponents, 94
- Exclusive-or operation, 67, 69
- Executing a loop backwards, 469
- exit statement, 262
- exitif statement, 262
- Exponent, 90
- Expressions, 369
 - and temporary values, 374
- Extended Base Pointer register, 295
- Extended-precision
 - addition, 480
 - and, 503
 - arithmetic, 477
 - comparisons, 485
 - division, 492
 - floating-point format, 95
 - formatted I/O, 515
 - I/O, 509
 - input routines, 516
 - multiplication, 488
 - neg, 501
 - not, 504
 - operations, 478
 - or, 503
 - rotates, 508
 - shifts, 504
 - shifts and their effect on the flags, 506
 - xor, 504
- External declaration
 - limitations, 343

- external directive, 338
- external option in a class declaration, 657

- Extracting
 - bit strings, 626
 - bits, 601

F

- f2xm1 instruction, 402
- fabs instruction, 399
- fadd instruction, 394
- faddp instruction, 394
- false
 - boolean constant, 7, 375
 - label, 475
- fbld instruction, 393, 537–538
- fbstp instruction, 393, 537–538
- fchs instruction, 399
- fclex instruction, 405
- fcom instruction, 386, 399–400
- fcomi instruction, 400–401
- fcomip instruction, 400–401, 567
- fcomp instruction, 386, 399–400
- fcompp instruction, 386, 399–400
- fcos instruction, 403
- fdiv instruction, 396
- fdivp instruction, 396
- fdivr instruction, 396
- fdivrp instruction, 396
- fiadd instruction, 405
- ficom instruction, 386, 405
- ficomp instruction, 386, 405
- fidiv instruction, 405
- fidivr instruction, 405
- Field alignment within a record, 241
- Field width, 35
- fild instruction, 392
- fimul instruction, 405
- finit instruction, 404
- First clear bit, 600, 618
- First set bit, 600, 618
- fist instruction, 392
- fistp instruction, 392
- fisub instruction, 405
- fisubr instruction, 405

- Flags, 10
 - and instruction, 605
 - carry, 10, 358, 418
 - cmp instruction affect on
 - flags, 358
 - copying ah register to flags,
 - 88–89, 400
 - direction, 635–637
 - lahf instruction, 89
 - or instruction, 605
 - overflow, 156–159, 358, 418
 - register, 89
 - sign, 358, 418
 - xor instruction, 605
 - zero, 358, 418
- fld instruction, 389
- fld1 instruction, 402
- fldcw instruction, 384, 404
- fldl2e instruction, 402
- fldl2t instruction, 402
- fldlg2 instruction, 402
- fldln2 instruction, 402
- fldpi instruction, 402
- fldz instruction, 402
- Floating-point
 - adding an integer to a floating-point value, 405
 - arithmetic, 380
 - calculations, 380
 - comparisons, 92, 386, 399–400, 405
 - control word, 384, 404
 - data registers, 380
 - data types, 387
 - division, 396, 405
 - exchange registers, 391
 - FPU, 380
 - integer conversion, 392
 - integer division, 405
 - integer multiplication, 405
 - integer subtraction, 405
 - multiplication, 396
 - negation, 399
 - overflow, 92
 - overflow exception, 384
 - partial remainder, 398
 - pushing a value onto the FPU stack, 389
 - pushing the constant 1.0 onto the FPU stack, 402
 - remainder, 398
 - reverse division, 405
 - subtraction, 395
 - test for zero, 386, 402
 - underflow, 92
 - values, 62
- Floating-point unit, 380
- fmul instruction, 396
- fmulp instruction, 396
- fnclx instruction, 405
- fninit instruction, 404
- fnstsw instruction, 405
- for loops, 25, 460
- Forcing
 - a 0 result, 68
 - bits to one, 70
 - bits to zero, 70
- forever loop, 27, 456
- Formal parameters, 270, 324
- FORTRAN programming
 - language, 452
- forward directive, 286, 692
- Forward procedure
 - declarations, 286
- fpatan instruction, 403
- fprem instruction, 398
- fprem1 instruction, 398
- fptan instruction, 403
- FPU
 - busy bit, 387
 - condition code bits, 385
 - control register, 381
 - control word, 384, 404
 - data movement instructions, 389
 - data registers, 381
 - data types, 387
 - exception bits, 405
 - exception flags, 385
 - exception masks, 383
 - floating-point unit, 380
 - initialization, 404
 - popping the FPU stack, 390
 - registers, 380
 - stack fault flag, 385
 - status register, 385, 405

- status word, 385, 399–400, 405
 - top of stack pointer, 387
- Free memory function, 148
- frndint instruction, 399
- fsin instruction, 403
- fsincos instruction, 403
- fsqrt instruction, 391, 397
- fst instruction, 390
- fstcw instruction, 384, 404
- fstp instruction, 390
- fstsw instruction, 385, 399–400, 405
- fsub instruction, 395
- fsubp instruction, 395
- fsubr instruction, 395
- fsubrp instruction, 395
- ftst instruction, 386, 402
- fucom instruction, 386
- fucomp instruction, 386
- fucompp instruction, 386
- Function
 - computation via table
 - lookup, 539
 - overloading using macros, 586
 - results, 275
- fxam instruction, 386
- fxch instruction, 391
- fyl2x instruction, 404
- fyl2xp1 instruction, 404

G

- General protection fault, 120
- General purpose registers, 9
- GenerateSines program, 546
- Generating sine data at compile time, 592
- Generic object pointers, 662
- Global variables, 263, 267, 652
- Guard digits/bits, 90

H

- Hash mark (#) operator, 105
- Header files, 344
- Heap, 148
- Hello, world!
 - compile-time program, 553
 - HLA program, 2

- Hexadecimal, 53
 - calculators, 64
 - hex to decimal conversion, 66
 - input (extended precision), 520
 - numbering system, 56
 - numbers, 59
 - output (extended precision), 509
 - string-to-numeric conversion, 520
- High order
 - bit, 56, 60
 - byte, 61
 - nibble, 60
 - word, 62
- High-level assembly language vs. low-level assembly language, 50
- High-level control structures, 17
- HLA
 - pointers, 177
 - Standard Library, 32
 - strings, 188
 - support for ASCII characters, 105
- HLA v2.0 procedure declarations, 287
- H.O., 56. *See also* High order
- Hybrid
 - boolean expressions, 474
 - control structures, 413, 473
 - parameter passing facilities, 327

I

- i128Size routine, 515
- Icon programming language, 539
- Identifiers, 2
- Idioms (machine idiosyncrasies), 377
- idiv instruction, 355, 492
- IEEE (754 and 853) floating-point format, 93, 381, 383, 537
- if statement, 17, 20, 422–424
- imod instruction, 439
- imul instruction, 352, 488
- in (clause in a #for..#endfor compile-time loop), 573

- in operator, 19
- inc instruction, 152
- #include directive, 336
- Include files, 3
- Inclusive-or operation, 69
- Indexed addressing mode, 116
- Indexed and scaled-indexed addressing modes, 111
- Indirect
 - calls, 329
 - jumps, 416, 423, 452–456
- Induction variables, 472–473
- Infinite loops, 27, 456
- Infinite precision arithmetic, 89
- Infix notation, 406
- Information hiding, 653
- Inheritance
 - in HLA classes, 659
 - implementation, 674
- inherits keyword in classes, 660
- Initializing the FPU, 404
- Initializing the virtual method table
 - pointer in an object, 678
- Input conditioning, 544
- Input/output, 8
- Inputting values in an HLA program, 7
- Instance of a class, 657
- Instruction composition
 - definition, 277
 - effect on program readability, 279
- Instructions
 - aaa, 535
 - aad, 535
 - aam, 535
 - aas, 535
 - adc, 482, 609
 - add, 15
 - and, 70, 376, 601, 605
 - binary-coded decimal, 477
 - bound, 156
 - bsf, 618
 - bsr, 618
 - bt, 608
 - btc, 211, 608
 - btr, 211, 608
 - bts, 608
 - call, 255–256, 288
 - cbw, 77
 - cdq, 77
 - clic, 88, 609
 - cld, 88
 - cli, 88
 - cmc, 88, 609
 - cmp, 357
 - cmps, 644
 - cmpsb, 634
 - cmpsd, 634
 - cmpsw, 634
 - cwd, 77
 - cwde, 77
 - daa, 534–535
 - das, 534–535
 - dec, 152
 - div, 355, 492
 - f2xm1, 402
 - fabs, 399
 - fadd, 394
 - faddp, 394
 - fbld, 393, 537–538
 - fbstp, 393, 537–538
 - fchs, 399
 - fclex, 405
 - fcom, 386, 399–400
 - fcomi, 400–401
 - fcomip, 400–401, 567
 - fcomp, 386, 399–400
 - fcompp, 386, 399–400
 - fcos, 403
 - fdiv, 396
 - fdivp, 396
 - fdivr, 396
 - fdivrp, 396
 - fiadd, 405
 - ficom, 386, 405
 - ficomp, 386, 405
 - fidiv, 405
 - fidivr, 405
 - fild, 392
 - fimul, 405
 - finit, 404
 - fist, 392
 - fistp, 392
 - fisub, 405
 - fisubr, 405

fld, 389
 fld1, 402
 fld2e, 402
 fldcw, 384, 404
 fldl2t, 402
 fldlg2, 402
 fldln2, 402
 fldpi, 402
 fldz, 402
 floating-point comparisons, 399
 floating-point conversions, 391
 fmul, 396
 fmulp, 396
 fnclex, 405
 fninit, 404
 fnstsw, 405
 fpatan, 403
 fprem, 398
 fprem1, 398
 fptan, 403
 FPU data movement, 389
 frndint, 399
 fsin, 403
 fsincos, 403
 fsqrt, 391, 397
 fst, 390
 fstcw, 384, 404
 fstp, 390
 fstsw, 385, 399–400, 405
 fsub, 395
 fsubp, 395
 fsubr, 395
 fsubrp, 395
 ftst, 386, 402
 fucom, 386
 fucomp, 386
 fxam, 386
 fxch, 391
 fyl2x, 404
 fyl2xp1, 404
 idiv, 355, 492
 imod, 439
 imul, 352, 488
 inc, 152
 indirect jumps, 416
 intmul, 156, 354
 into, 156, 159
 ja, 419
 jae, 419
 jb, 419
 jbe, 419
 jc, 419, 609
 je, 419
 jf, 421
 jg, 419
 jge, 419
 jl, 419
 jle, 419
 jmp, 416
 jna, 419
 jnae, 419
 jnb, 419
 jnbe, 419
 jnc, 419, 609
 jne, 419
 jng, 419
 jnge, 419
 jnl, 419
 jnle, 419
 jno, 419
 jnp, 419
 jns, 419
 jnz, 419
 jo, 419
 jp, 419
 jpe, 419
 jpo, 419
 js, 419
 jt, 421
 jz, 419
 lahf, 89
 lea, 153, 322, 414
 lodbs, 634
 lods, 648
 lodsd, 634
 lodsw, 634
 mov, 14, 112
 movs, 638
 movsb, 634, 638
 movsd, 634, 638
 movsw, 634, 638
 mul, 352, 488
 neg, 74, 501
 not, 70, 376, 601
 or, 70, 376, 601, 605
 pop, 138, 259

Instructions, *continued*

popa, 144
popad, 144
popf, 144
popfd, 144
push, 137, 259
pusha, 143
pushad, 143
pushd, 138
pushf, 144
pushfd, 144
pushw, 138
rcl, 84, 609
rcr, 84, 609
rep.movsb, 635
rep.movsd, 635
rep.movsw, 635
rep.stosb, 636
rep.stosd, 636
rep.stosw, 636
repe.cmpsb, 635
repe.cmpsd, 635
repe.cmpsw, 635
repe.scasb, 635
repe.scasd, 635
repe.scasw, 635
repne.cmpsb, 635
repne.cmpsd, 635
repne.cmpsw, 635
repne.scasb, 635
repne.scasd, 635
repne.scasw, 635
rol, 83
ror, 83
sahf, 88–89, 400
sar, 83, 379
sbb, 484, 501, 609
scas, 647
scasb, 634
scasd, 634
scasw, 634
seta, 363
setae, 363
setb, 363
setbe, 363
setc, 362, 609
sete, 363
setg, 363
setge, 363
setl, 363
setna, 363
setnae, 363
setnb, 363
setnbe, 363
setnc, 362, 609
setne, 363
setng, 363
setnge, 363
setnl, 363
setnle, 363
setno, 362
setnp, 362
setns, 362
setnz, 362, 364
seto, 362
setp, 362
setpe, 362
setpo, 362
sets, 362
setz, 362, 364
shl, 80, 378
shld, 506
shr, 81, 379
shrd, 506
stc, 609
std, 88
sti, 88
stos, 648
stosb, 634
stosd, 634
stosw, 634
sub, 15
test, 364, 601, 606
xlat, 540
xor, 70, 376, 601, 604–605
int8, 5
int8, int16, int32, int64, and
int128 compile-time
functions, 559
int16, 5
int32, 5
Integer
integer portion of a floating-
point number, 399
integer to floating point
conversion, 392

- integer to floating point
 - comparisons, 405
 - signed remainder/modulo, 439
 - subtraction from floating point, 405
- Integer types in C, 478
- Integer/floating-point division, 405
- Intel ABI, 302
- intmul instruction, 156, 354
- into instruction, 156, 159
- Invalid operation exception (FPU), 383
- Invariant computations, 470
- I/O, 8
- Iterator entries in a virtual method table, 671

J

- ja instruction, 419
- jae instruction, 419
- jb instruction, 419
- jbe instruction, 419
- jc instruction, 419, 609
- je instruction, 419
- jf medium-level instruction, 421
- jg instruction, 419
- jge instruction, 419
- jl instruction, 419
- jle instruction, 419
- jmp instruction, 416
- jna instruction, 419
- jnae instruction, 419
- jnb instruction, 419
- jnbe instruction, 419
- jnc instruction, 419, 609
- jne instruction, 419
- jng instruction, 419
- jnge instruction, 419
- jnl instruction, 419
- jnle instruction, 419
- jno instruction, 419
- jnp instruction, 419
- jns instruction, 419
- jnz instruction, 419
- jo instruction, 419
- jp instruction, 419

- jpe instruction, 419
- jpo instruction, 419
- js instruction, 419
- jt medium-level instruction, 421
- Jump indirect, 456
- Jump instructions, 416
- jz instruction, 419

K

- KCS Floating-Point Standard, 93

L

- Labels, 414
- lahf instruction, 88, 89
- Large parameter objects, 319
- Large programs, 335
- Last clear bit, 600, 618
- Last set bit, 618
- Last-in, first-out data structures, 140
- lea instruction, 153, 322, 414
- Least significant bit, 56, 60
- Left-associative operators, 370
- Left rotates, 80
- Left shifts, 80
- Legal boolean expressions in HLA
 - high-level language statements, 19
- Length (field of an HLA string), 190
- Length-prefixed strings, 187
- Lexical scope, 263
- Lexicographical ordering, 207, 645
- Lifetime (of a variable), 125, 263, 267
- LIFO, 140
- Linefeed character, 34
- Linker, 335
- Literal quotes in string
 - constants, 166
- Literal record constants, 235
- L.O., 56. *See also* Low order
- Load effective address (lea)
 - instruction, 153, 322, 414
- Loading the flags register
 - from ah, 89

Loading the FPU control word, 404

Local

- macro symbols, 582–583
- variables, accessing, 262, 299

Location counter, 127

lods instruction, 648

lods instruction, 634

lods instruction, 634

lodsw instruction, 634

Log base 2 of e , 402

Log base 2 of x , 404

Log base 2 of x plus 1, 404

Logical

- and operation, 67, 376
- exclusive-or (xor) operation, 67, 69
- not operation, 67, 69
- operations on binary numbers, 70
- operations on bits, 67
- operators within a constant expression, 170
- or operation, 67, 68, 376
- shift left, 80
- shift right, 82
- xor operation, 67, 376

Lookup table creation, 545

Loops, 24–25, 27, 456–457, 460

- control variables, 457
- invariant computations, 470
- performance improvement, 466, 469
- register usage, 465
- termination test, 457, 466
- unraveling/unrolling, 471

Low order

- bit, 56, 60
- byte, 61
- nibble, 60
- word, 62

Low-level control structures, 414

Low-level parameter

- implementation, 301

lword, 58

- compile-time function, 559
- data declarations, 63

M

Machine idioms, 377

Machine state, saving, 258

Macros, 573

- parameters, 576, 578, 579
- vs. procedures, 574
- in several different source files, 598

Managing large programs, 335

Manifest constants, 161, 164

Manipulating bits in memory, 599

Mantissa, 90, 93

Mask (bits), 600

Masking, 70

math (HLA stdlib module)

- math.acos, 411
- math.acot, 411
- math.acsc, 411
- math.addl, 479, 482
- math.addq, 479, 482
- math.andl, 479
- math.andq, 479
- math.asec, 411
- math.asin, 411
- math.cot, 411
- math.csc, 411
- math.divl, 479
- math.divq, 479
- math.exp, 411
- math.idivl, 479
- math.imodl, 479
- math.imulq, 479
- math.ln, 411
- math.log, 411
- math.modl, 479
- math.modq, 479
- math.mull, 479
- math.mulq, 479
- math.negl, 480
- math.negq, 480
- math.notl, 480
- math.notq, 480
- math.orl, 479
- math.orq, 479
- math.sec, 411

- math.shll, 480
- math.shlq, 480
- math.shrl, 480
- math.shrq, 480
- math.subl, 479
- math.subq, 479
- math.tentox, 411
- math.twotox, 411
- math.xorl, 479
- math.xorq, 479
- math.ytox, 411
- Maximum string length, 188
- maxlen, 190
- Medium-level control
 - structures, 421
- mem (HLA stdlib module), 148
 - mem.alloc, 148, 180, 191, 251, 686
 - mem.free, 148, 182
 - mem.isInHeap, 686
- Memory, 8
 - access violation exception, 181
 - addressing modes, 111, 113
 - allocation, 111, 148
 - anonymous memory objects, 304
 - indirect jump through
 - memory, 418
 - leaks, 183
 - organization, 111
 - read operation, 12
 - subsystem, 11
 - write operation, 12
- Merging source files during
 - assembly, 336
- Methods, 653, 655
 - accessor methods, 653
 - calling mechanism, 671
 - definition, 653
 - entries in a virtual method
 - table, 671
 - methods vs. class
 - procedures, 663
- Minimal procedures, 289
- Minimum field width, 35
- Misaligned data and the system
 - cache, 131
- Mixed-size operands, 530
- mod (within a constant
 - expression), 169
- Modulo
 - floating-point remainder, 398
 - integer remainder, 439
- Modulo-n counters, 380
- Monadic operations, 69
- Most significant bit, 56, 60
- mov instruction, 14, 112
 - operands, 15
- Moving string data, 633
- movs instruction, 638
 - performance, 642
- movsb instruction, 634, 638
- movsd instruction, 634, 638
- movsw instruction, 634, 638
- movsx instruction, 78
- movzx instruction, 78
- mul instruction, 352, 488
- Multiplication, 156, 352, 354, 488
 - floating-point, 396
 - integer and floating-point
 - numbers, 405
 - by a reciprocal to simulate
 - division, 379
 - register value by 10, 378
 - unsigned, 352–353, 488
 - without mul, imul, or intmul, 378
- Multiprecision
 - addition, 480
 - arithmetic, 477
 - comparisons, 485
 - operations, 478
 - subtraction, 483

N

- N/No N rule, 420
- Namespace
 - declarations, 346
 - definition, 248
 - pollution, 248, 345
- neg instruction, 74, 501
- neg64 (macro), 574

- Negation
 - of boolean variables in an if statement, 429
 - floating-point, 399
 - of large values, 502
 - Negative numbers, 72
 - Nested array constants, 231
 - Nesting record definitions, 239
 - Nesting try...endtry statements, 43
 - new memory allocation operator (C++ or Pascal), 147
 - New style procedure
 - declarations, 287
 - Newline constant, 33
 - newln, 35
 - Nibbles, 58, 59
 - n1 (newline) constant, 3, 33, 168
 - Noncommutative binary
 - operators, 375
 - Normalized floating-point numbers, 95, 388
 - not in operator, 19
 - not instruction, 70, 376, 601
 - not operation, 67, 69
 - NUL character, 187, 306, 518
 - NULL, 120
 - Numbering systems, 54
 - Numbers, unsigned, 72
 - Numeric
 - compile-time functions, 561
 - output field width, 35
 - representation, 65
- O**
- Object
 - constructors, 677
 - in HLA, 657
 - implementation, 668
 - initialization, 677
 - memory allocation, 679
 - pointers (generic objects), 662
 - representation with
 - inheritance, 673
 - Object-oriented programming, 651
 - benefits, 652
 - general principles, 652
 - One's complement numbering
 - system, 73
 - Operands, mixed size, 530
 - Operations
 - and, 605
 - on binary numbers, 70
 - not, 67, 69
 - or, 67, 68, 376, 605
 - rotation, 80
 - shift arithmetic right, 83
 - shifts, 80
 - xor, 67, 69, 376, 605
 - Operator precedence, 23, 370
 - in compile-time expressions, 557
 - Operators
 - in, 19
 - logical, 170
 - not in, 19
 - type, 134
 - Opposite condition jump, 420
 - Options
 - @align, 657
 - @cdecl, 657
 - @noalignstack, 288, 297, 657
 - @nodisplay, 288, 657
 - @noframe, 288, 291, 657
 - @nostorage, 124, 186
 - @pascal, 657
 - @returns, 280, 657
 - @stdcall, 657
 - @use, 317, 324, 657
 - Optional parameters in a macro
 - expansion, 581
 - or instruction, 70, 376, 601, 605
 - or operation, 67, 68
 - Output field width, 35
 - Outputting values in HLA, 3, 137
 - Overflow exception (FPU), 384
 - Overflow flag, 10, 156, 159, 358, 418
 - and the and, or, and xor instructions, 605
 - testing, 156, 159
- P**
- Packed
 - arrays of bit strings, 615
 - data, 85
 - decimal arithmetic, 537
 - Packing and unpacking bit
 - strings, 609

- Padding
 - parameter data, 313
 - records, 242
- Parameters. *See* Procedures,
 - parameters
- Partial remainder, 398
- Pascal programming language, 539
- Passing. *See also* Procedures
 - ah, bh, ch, or dh on the stack as a
 - procedure parameter, 314
 - a byte parameter to a
 - procedure, 314
 - large objects as parameters, 319
 - large objects passed by value on
 - the stack, 316
 - parameters as parameters, 324
 - parameters by reference, 323
 - parameters in registers, 301
 - parameters in the code
 - stream, 304
 - parameters on the stack, 307
 - reference parameters, 273, 275,
 - 321, 323
 - value parameters, 269, 311
 - word parameters on the
 - stack, 316
- Performance improvements for
 - loops, 466
- Performance of the string
 - instructions, 650
- pi (FPU load instruction), 402
- Placement of the VMT, 672
- point class, 659
- pointer to type declaration, 178
- Pointers, 175
 - and code readability, 456
 - constants and pointer constant
 - expressions, 179
 - dangling, 182
 - problems, 180
 - to records, 242
- pop instruction, 138, 259
- popa instruction, 144
- popad instruction, 144
- popf instruction, 144
- popfd instruction, 144
- Popping the FPU stack, 390
- Postfix notation, 406
- Precedence
 - of arithmetic operators, 370
 - rules, 370
- Precision
 - control bits (FPU), 383
 - exception (FPU), 384
- Preserving
 - the direction flag, 637
 - machine state, 258
 - registers, 140, 258
 - registers in loops, 465
- Printing
 - boolean values, 8
 - character values, 8
 - values with HLA, 3
- Private declarations, 340
- Problems with macro parameter
 - expansion, 579
- proc keyword in HLA, 287
- Procedures, 255
 - call syntax, 257
 - compared to macros, 574
 - effect on the stack, 290
 - invocation, 255
 - overloading, 685
 - options
 - @align, 657
 - @cdecl, 657
 - @noalignstack, 288, 297, 657
 - @nodisplay, 288, 657
 - @noframe, 288, 291, 657
 - @pascal, 657
 - @returns, 280, 657
 - @stdcall, 657
 - @use, 317, 324, 657
 - parameters
 - ah, bh, ch, or dh as
 - parameters, 314
 - byte parameters on the
 - stack, 314
 - large objects as
 - parameters, 319
 - large objects passed by
 - value, 316
 - parameters as
 - parameters, 324
 - parameters in registers, 301

Procedures, *continued*

- parameters, *continued*
 - parameters in the code stream, 304
 - parameters on the stack, 307
 - pass by reference, 273, 321, 323
 - pass by value, 269, 311
 - procedural parameters, 333
 - required vs. optional in a macro, 581
 - variable length, 306
 - word parameters on the stack, 316
- pointers, 329

Programming in the large, 335

Programming languages

- C, 478
- FORTRAN, 452
- Icon, 539
- Pascal, 539
- SNOBOL4, 539

Protected block in a try statement, 28

Protected statements in a try..endtry statement, 43

Prototypes, 286

Prototypes for class methods, 655

Pseudo-opcode, 121

Public declarations, 340

push instruction, 137, 259

pusha instruction, 143

pushad instruction, 143

pushd instruction, 138

pushf instruction, 144

pushfd instruction, 144

Pushing a value onto the floating-point stack, 389

Pushing the constant 1.0 onto the FPU stack, 402

pushw instruction, 138

puth128, 509

puti (overloaded function), 586

puti128, 514

puti128Size, 515

putu128, 514

Q

Quicksort, 283

qword compile-time function, 559

qwords, 58

- data declarations, 63

R

Radix, 56

raise statement, 42, 196

Range of a function, 542

rcl instruction, 84, 609

rcr instruction, 84, 609

Reading

- integer values from the standard input device, 39
- from memory, 12
- values into an HLA program, 7

readonly

- declaration section, 123
- variables as constants, 161

real32, real64, and real180 compile-time functions, 559

Rearranging expressions

- in if statements to improve performance, 438
- to make them more efficient, 438

Reassigning text objects, 564

Records, 233

- constants, 235
- declarations, 233
- field alignment, 241
- pointers, 242
- as record fields, 237

Recursion, 282

Reference parameters, 273, 318, 321

Register

- addressing modes, 112
- comparison to zero, 365
- indirect addressing mode, 114
- indirect jump instruction, 416
- output, 137
- preservation, 258
- callee/caller, 259
- in a try..endtry statement, 48

- as procedure parameters, 301
 - as a signed integer value, 136
 - type coercion, 136
 - usage in loops, 465
 - usage in string instructions, 635
- Registers, 9
 - 8-bit, 16-bit, 32-bit, 9
 - al, ah, bl, bh, cl, ch, dl, dh, 9
 - ax, bx, cx, dx, bp, sp, si, di, 9
 - eax, ebx, ecx, edx, ebp, esp, esi, edi, 9
 - FPU, 380, 381
 - preservation, 140, 258, 465
- Relational operators, 19
- Remainder
 - floating point, 398
 - signed integer, 439
- Removing unwanted data from the stack, 144
- rep.movsb instruction, 635
- rep.movsd instruction, 635
- rep.movsw instruction, 635
- rep.stosb instruction, 636
- rep.stosd instruction, 636
- rep.stosw instruction, 636
- rep/repe/repz and repnz/repne string instruction prefixes, 635
- repe.cmpsb instruction, 635
- repe.cmpsd instruction, 635
- repe.cmpsw instruction, 635
- repe.scasb instruction, 635
- repe.scasd instruction, 635
- repe.scasw instruction, 635
- repeat..until loops, 17, 26, 456, 458
- Repetitive compilation, 570
- repne.cmpsb instruction, 635
- repne.cmpsd instruction, 635
- repne.cmpsw instruction, 635
- repne.scasb instruction, 635
- repne.scasd instruction, 635
- repne.scasw instruction, 635
- Required vs. optional macro parameters, 581
- Restrictions in simple switch statement implementations, 446
- ret instruction, 255, 288
- Return address, 288
- Reverse division (floating point), 396
- Reverse Polish notation (RPN), 406
- Reverse subtraction (floating point), 395
- Reversing bits in a bit string, 623
- Right-associative operators, 371
- Right shifts, 80–82
- rol instruction, 83
- ror instruction, 83
- Rotate
 - left, 83
 - operations, 80
 - right, 83
- Rounding
 - control (FPU), 382
 - floating-point numbers, 399
 - floating-point value to an integer, 399
- Row-major ordering, 225
- RPN (Reverse Polish notation), 406
- RTTI (runtime type information), 696
- Run of zero bits, 600
- Runtime
 - language, 551
 - memory organization, 119
 - type information (RTTI), 696

S

- sahf instruction, 88–89, 400
- sar instruction, 83, 379
- Saturation, 76, 80
- Saving the machine state, 258
- sbb instruction, 484, 501, 609
- Scaled-indexed addressing mode, 118
- scas instruction, 647
- scasb instruction, 634
- scasd instruction, 634
- scasw instruction, 634
- Scope (of a name), 263
- Searching
 - for a bit, 617
 - for a bit pattern, 627
 - for the first (or last) set bit, 618
- Self-modifying programs, 121

- Separate compilation, 335
- Set on condition instructions, 362
- seta instruction, 363
- setae instruction, 363
- setb instruction, 363
- setbe instruction, 363
- setc instruction, 362, 609
- setcc instructions, 362
- sete instruction, 363
- setg instruction, 363
- setge instruction, 363
- setl instruction, 363
- setna instruction, 363
- setnae instruction, 363
- setnb instruction, 363
- setnbe instruction, 363
- setnc instruction, 362, 609
- setne instruction, 363
- setng instruction, 363
- setnge instruction, 363
- setnl instruction, 363
- setnle instruction, 363
- setno instruction, 362
- setnp instruction, 362
- setns instruction, 362
- setnz instruction, 362, 364
- seto instruction, 362
- setp instruction, 362
- setpe instruction, 362
- setpo instruction, 362
- sets instruction, 362
- Setting bits, 601
- setz instruction, 362, 364
- Shift
 - arithmetic right operation, 83
 - left operation, 80
 - operations, 80
 - right operation, 82
 - rotate instructions, 601, 608
- shl instruction, 80, 378
- shld instruction, 506
- Short-circuit boolean
 - evaluation, 433
 - vs. complete boolean evaluation, 435
- shr instruction, 81, 379
- shrd instruction, 506
- si, 9
- Side effects, 435
- Sign
 - bit, 72
 - contraction, 76, 79
 - extension, 76, 356
 - extension prior to division, 372
 - zero flag settings after mul and imul instructions, 353
- Sign flag, 10, 358, 418
 - and the and, or, and xor instructions, 605
- Signed
 - comparison flag settings, 359
 - comparisons, 363
 - decimal input (extended precision), 529
 - decimal output (extended precision), 514
 - division, 356
 - integer output, 35
 - integer remainder/modulo, 439
 - multiplication, 156, 352, 354, 488
 - numbers, 72
- Significant digits, 90
- Simple assignments (conversion to assembly language), 366
- Simulating div, 379
- Sine, 403
- Sine data table generation, 592
- Single precision floating point format, 93
- Size of a data type (compile-time function), 148
- SNOBOL4 programming language, 539
- Software configuration
 - via conditional compilation, 568
 - for different environments, 567
- Source file merging during assembly, 336
- sp, 9
- Spaghetti code, 455
- Square root, 391, 397
- st0, 381
- st1, 381
- Stack fault flag (FPU), 385
- Stack frame, 293

- Stack manipulation by procedure calls, 290
- Stack operations
 - pop, 138, 259
 - popa, 144
 - popad, 144
 - popf, 144
 - popfd, 144
 - push, 137, 259
 - pusha, 143
 - pushad, 143
 - pushd, 138
 - pushf, 144
 - pushfd, 144
 - pushw, 138
- Stack segment, 137
- Standard entry sequence (to a procedure), 296
- Standard exit sequence (from a procedure), 298
- Standard input, 34
- Standard library (HLA), 32
- Standard macro parameter expansion, 576
- Standard macros, 574
- Standard Output, 34
- State machine, 416, 452
- State variable, 452
- Statement labels, 414
- Statements
 - break, 27, 461
 - breakif, 27
 - case, 423, 442
 - conditional, 423
 - continue, 461
 - else, 17, 424
 - elseif, 17
 - endfor, 25
 - endif, 17
 - endwhile, 17, 24
 - exception, 28
 - exit, 262
 - exitif, 262
 - for, 25, 460
 - forever, 27, 456
 - if, 17, 20, 422, 424
 - jf, 421
 - raise, 42, 196
 - repeat..until, 17, 26, 456, 458
 - then, 17
 - try..endtry, 28, 42
 - until, 17
 - while, 17, 24, 456, 457
- static
 - declaration section, 6, 122
 - procedures in a class, 661
 - variable section, 5
- Status register (FPU), 385, 405
- Status word, 399, 400, 405
- stc instruction, 609
- std instruction, 88
- stdin (HLA stdlib module)
 - stdin.a_gets, 193
 - stdin.eoln, 108
 - stdin.flushInput, 40, 108
 - stdin.get, 7, 41, 67, 99, 107, 516
 - stdin.getc, 38, 519
 - stdin.getf, 99, 411
 - stdin.geth8, 67
 - stdin.geth16, 67
 - stdin.geth32, 67
 - stdin.geth64, 67
 - stdin.geth128, 67
 - stdin.geti32, 75
 - stdin.gets, 191, 193
 - stdin.getu8, 76
 - stdin.getu16, 76
 - stdin.getu32, 76
 - stdin.getu64, 76
 - stdin.getu128, 76
 - stdin.peekc, 519
 - stdin.readLn, 40
- stdio (HLA stdlib module)
 - stdio.bell, 34
 - stdio.bs, 34
 - stdio.cr, 34
 - stdio.lf, 34
 - stdio.tab, 34
- stdlib.hhf, 3
- stdout (HLA stdlib module)
 - stdout.newLine, 35, 256
 - stdout.put, 3, 7, 37, 67, 107, 307
 - stdout.putc, 106
 - stdout.putcSize, 106
 - stdout.puth8, 67
 - stdout.puth16, 67

- stdout (HLA stdlib module),
 - continued*
 - stdout.puth32, 67, 509
 - stdout.puth64, 67
 - stdout.puth128, 67
 - stdout.puti8, 35, 75
 - stdout.puti8Size, 35
 - stdout.puti16, 35
 - stdout.puti16Size, 35
 - stdout.puti32, 35, 276, 590
 - stdout.puti32Size, 35, 590
 - stdout.puti128, 515
 - stdout.putiXSize, 35
 - stdout.putr32, 97
 - stdout.putr64, 97
 - stdout.putr80, 97
 - stdout.putu8, 75
 - stdout.putu8Size, 76
 - stdout.putu16, 75
 - stdout.putu16Size, 76
 - stdout.putu32, 75
 - stdout.putu32Size, 76
 - stdout.putu64, 75
 - stdout.putu64Size, 76
 - stdout.putu128, 75, 515
 - stdout.putu128Size, 76
- sti instruction, 88
- storage declaration section, 123
- Storing ah register into flags,
 - 88–89, 400
- Storing the FPU control word, 384
- Storing the FPU status word, 385,
 - 399–400, 405
- stos instruction, 648
- stosb instruction, 634
- stosd instruction, 634
- stosw instruction, 634
- str (HLA stdlib module)
 - str.a_cat, 203
 - str.a_cpy, 201
 - str.a_delete, 205
 - str.a_insert, 204
 - str.a_substr, 205
 - str.alloc, 191
 - str.cat, 203
 - str.cpy, 199, 346
 - str.delete, 205
 - str.eq, 206
 - str.free, 192
 - str.ge, 206
 - str.gt, 206
 - str.ieq, 207
 - str.ige, 207
 - str.igt, 207
 - str.ile, 207
 - str.ilt, 207
 - str.index, 208
 - str.ine, 207
 - str.insert, 204
 - str.le, 206
 - str.length, 203
 - str.lt, 206
 - str.ne, 206
 - str.put, 208
 - str.strRec data type, 190
 - str.substr, 205
- strfill procedure, 302, 307
- string compile-time function,
 - 559–560
- Strings, 185
 - assignment by reference, 197
 - comparisons, 206, 633, 645
 - concatenation, 165, 203
 - constant initializers in the const section, 167
 - constants, 165
 - constants containing control characters, 167
 - instruction operation, 634
 - instruction performance, 650
 - instructions, 633–634, 648
 - operators within a constant expression, 170
 - pointers, 188
- Structures as structure fields, 237
- sub instruction, 15
- Substring operation, 205
- Subtract with borrow, 484, 501, 609
- Subtraction, floating point, 395
- Swapping registers on the FPU stack, 391
- switch statement, 442

- Synthesizing in assembly language
 - break statements, 461
 - continue statements, 462
 - for statements, 461
 - forever..endfor loops, 460
 - repeat..until loops, 458
 - while loops, 457

T

- Tab character, 34
- Tables, 539
- Tag field, 247
- Taking the address of a statement
 - label, 414
- Tangent, 403
- tbyte values (BCD), 538
- Temporary values in an
 - expression, 374
- Termination test (while loops), 457
- Test for zero (floating point), 402
- test instruction, 364, 601, 606
- Testing a floating-point operand for
 - zero, 386, 402
- Testing bits, 601
- Testing for set bits in a bit
 - string, 607
- Testing the overflow flag, 156, 159
- Text
 - constants, 167, 243
 - object assignment, 564
 - type, 564
- text compile-time function, 559
- then statement, 17
- Top of stack pointer (FPU), 387
- Transcendental function
 - instructions, 402
- Translating arithmetic
 - expressions into
 - assembly language, 351
- Treating registers as signed integer
 - values, 136
- Tricky programming, 377
- true
 - boolean constant, 7, 375
 - label, 475
- Truth table, 68

- try..endtry
 - effect on the stack, 46
 - protected statements, 28, 43
 - statement, 28, 42

- Two's complement
 - numbering system, 62
 - numeric representation, 72
 - operation, 73

- Type checking
 - coercion, 133–134, 243
 - procedure pointer
 - invocations, 332

- type declaration section, 173

- Type operator, 134

- Type-conversion compile-time
 - functions, 559

U

- u128Size, 515

- Unary operator (conversion to
 - assembly language), 368

- Underflow exception (FPU), 384

- Unicode, 62, 109

- Uninitialized pointers, 180

- Unions, 243

- accessing fields of a union, 244

- anonymous, 246

- definition, 243

- syntax (declaration), 243

- Units, 339

- Unpacking bit strings, 609

- Unprotected (try..endtry), 45

- Unraveling/unrolling loops,
 - 471, 596

- uns8, 75

- uns8 compile-time function, 559

- uns16, 75

- uns16 compile-time function, 559

- uns32, 75

- uns32 compile-time function, 559

- uns128 compile-time function, 559

- Unsigned comparisons, 363

- Unsigned decimal input (extended
 - precision), 525

- Unsigned decimal output
 - (extended precision), 510

Unsigned division, 355–356
Unsigned multiplication,
 352–353, 488
Unsigned numbers, 72
Unsigned variable declarations, 75
Unstructured code, 441
until statement, 17
Untyped reference parameters, 334

V

val
 declarations, 160
 fields in a class, 656
 section, 172
 val object modification, 173
 value parameter
 specification, 272
Value parameters, 269, 310
Values, inputting in an HLA
 program, 7
var
 declarations, 125
 pass-by-reference
 parameters, 273
Variable alignment, 131
Variable declarations, 75
Variable number of parameters in a
 macro, 579
Variable option, @nostorage, 124, 186
Variable-length parameters, 306
Variant types, 247
Vars (`_vars_`) constant in a
 procedure, 300
Virtual method table pointer
 initialization, 678
Virtual method tables, 671. *See*
 also VMT
Virtual methods in a class, 661

VMT

 declaration, 672
 initialization, 678
 record structure, 671
 virtual method tables, 671
Von Neumann Architecture, 8

W

while statement, 17, 24, 456, 457
word compile-time function, 559
Word strings, 633
Words, 58, 61
Writing compile-time
 programs, 592

X

xlat instruction, 540
xor instruction, 70, 376, 601,
 604, 605
xor operation, 67, 69

Y

Y2K, 87

Z

Zero divide exception (FPU), 384
Zero extension, 356
Zero flag, 10, 358, 418, 606
 setting after a multiprecision or,
 503
 settings after `mul` and `imul`
 instructions, 353
Zero-terminating byte (in HLA
 strings), 188
Zero-terminated strings, 186
zstring data type, 186