

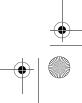




Numbers	propagating through
0-bits	add and subtract instructions, 54-57
counting. See counting bits.	logical operations, 58–63
isolating, 11	range analysis, 54
trailing 0's	searching for values in, 95–96
counting, 74, 84–87	arithmetic tables, 285–288
identifying, 11	arrays
turning on, 12	checking bounds. See arithmetic
0-bytes, finding, 91–95	bounds.
1-bits	counting 1-bits, 72–73
counting. See counting bits.	indexes, checking. <i>See</i> arithmetic bounds.
identifying, 11	
isolating, 11	indexing a sparse array, 73–74 rearranging, 127
right-propagating, 12	of short integers, 36–37
rightmost, turning off, 11–12	shuffling/unshuffling, 127
	shuffing/unshuffing, 127
A	В
absolute value	_
computing, 17–18	base $-1 + i$ number system, 230–232
multibyte, 36–37	base $-1 - i$ number system, 232–233
negative of, 21	base –2 number system, 223–230, 239
add instruction	big-endian format, converting to little-
condition codes, 33–34	endian, 101
propagating arithmetic bounds, 54-57	binary decomposition, integer exponentia-
addition	tion, 212–214
arithmetic tables, 285	binary search
double-length, 34–35	counting leading 0's, 77–80
and logical operations, 15–16	integer logarithm, 215–221
multibyte, 36–37	integer square root, 203–210
of negabinary numbers, 225–226	bit matrices, multiplying, 77
overflow detection, 26–28	bit numbering, 1
in various number encodings, 228–229	bit operations
Advanced Encryption Standard, 126	compress operation, 93, 104, 108, 116–123
alternating among values, 41–44	computing parity. See parity.
Alverson's method, 188–189	counting bits. See counting bits.
and, in three instructions, 16	cycling through bit combinations. See
arithmetic, computer vs. ordinary, 1	Gray code.
arithmetic bounds	extracting alternate bits, 104
checking, 51–53	finding strings of 1-bits, 96–99
of expressions, 54–55	initing strings of 1-bits, 90–99













INDEX

flipping bits, 102 number of leading zeros (nlz) function, 21-22, 83 general permutations, 122-126 generalized extract, 116-122 signed comparisons, from unsigned, half shuffle, 108 23 - 24inner shuffle, 106-108 true/false results, 21 outer shuffle, 106-108 using negative absolute values, 21-22 perfect shuffle, 106-108 comparisons reversing bits computer evaluation of, 25 6-, 7-, 8-, and 9-bit quantities, floating-point comparisons using inte-102-104 ger operations, 263-264 definition, 101 three-valued compare function, 19-20. generalized, 102 See also sign function. on rightmost bits. See rightmost bits. compress operation, 93, 104, 108, 116-123 searching words for bit strings, 83, computer algebra, 2-4 95-99 computer arithmetic, 1 sheep and goats operation, 122-126 condition codes, 33-34 shuffling bits, 106-108, 116, 127 constants transposing a bit matrix, 108-116 dividing by. See division by constants. unshuffling bits, 107-108, 116, multiplying by, 133-136 123, 127 counting bits. See also number of leading bit vectors, 1 zeros (nlz) function; number of bitgather instruction, 124-126 trailing zeros (ntz) function; popubitsize function, 83 lation count function. boundary crossings, powers of 2, 49-50 1-bits in 7- and 8-bit quantities, 71 bounds, arithmetic. See arithmetic bounds. 1-bits in a word, 65-72 branch on carry and register result nonzero 1-bits in an array, 72-73 instruction, 49 bitsize function, 83 bytes divide and conquer strategy, 65-66 definition, 1 leading 0's, with finding first 0-byte, 91-95 binary search method, 77 reversing, 92, 101-102 floating-point methods, 81-82 population count instruction, 79-80 \mathbf{C} rotate and sum method, 69-70 search tree method, 84-85 C language with table lookup, 71 arithmetic on pointers, 81, 190 trailing 0's, 74, 84-87 GNU extensions, 71-72, 82 by turning off 1-bits, 69 referring to same location with different cryptography types, 81 Advanced Encryption Standard, 126 representation of character strings, 91 bitgather instruction, 124-126 summary of elements, 2-4 DES (Data Encryption Standard), ceiling function, identities, 139-140 125-126 Chang, Albert, 96 Rijndael algorithm, 126 character strings, 91 SAG method, 122-126 checking arithmetic bounds, 51-53 shuffling bits, 106-108, 125-126 comparison predicates Triple DES, 126 from the carry bit, 24-25 cube root, integer, 211-212 definition, 21 cycling among values, 41-44









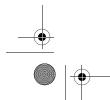




INDEX

D Data Encryption Standard (DES), 125-126 DEC PDP-10 computer, xi, 68 decryption. See cryptography. denormalized numbers, 262 denorms, 262 DES (Data Encryption Standard), 125–126 difference or zero (doz) function, 37-38 distribution of leading digits, 264-267 divide and conquer strategy, 65-66 division arithmetic tables, 287 doubleword by single word, 148-153 floor, 137-138, 188 modulus, 137-138, 188 multiword, 140-145 of negabinary numbers, 226-228 nonrestoring algorithm, 148, 150-151 notation, 137 overflow detection, 32-33 restoring algorithm, 148-150 shift-and-subtract algorithms (hardware), 148-151 short, 145-148, 151-152 signed computer, 137 long, 145-146, 153 multiword, 144-145 short, 146-148 unsigned computer, 137 long, 148-153 short from signed, 145-148 division by constants by 3, 5, and 7, 158–160 exact division definition, 190 multiplicative inverse, Euclidean algorithm, 192-195 multiplicative inverse, Newton's method, 195-197 multiplicative inverse, samples, 197-198 floor division, 188 incorporating into a compiler, signed, 171-173

incorporating into a compiler, unsigned, 183-184 magic algorithm, 171-174 magic numbers Alverson's method, 188 calculating, signed, 162-163, 171-174 calculating, unsigned, 182-187 definition, 161 sample numbers, 189-190 table lookup, 188-189 uniqueness, 173-175 magicu algorithm, 183-185 magicu2 algorithm, 186-187 modulus division, 188 signed best programs for, 175-178 by divisors ≤ -2 , 168–171 by divisors ≥ 2 , 160–168 incorporating into a compiler, 171-173 by powers of 2, 155-156 remainders from powers of 2, 156-157 test for zero remainder, 200-201 uniqueness, 173-175 unsigned by 3 and 7, 178-180 best programs for, 184-186 by divisors ≥ 1 , 180–183 incorporating into a compiler, 183-184 incremental division and remainder technique, 172, 183-184 by powers of 2, 178 remainders from powers of 2, 178 test for zero remainder, 198-199 double buffering, 39 double-length addition/subtraction, 34-35 double-length shifts, 35-36 doublewords definition, 1 dividing by single words, 148–153 doz (difference or zero) function, 37-38









•



300 INDEX

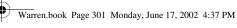
E	first uppercase letter, 96
encryption. See cryptography.	length of character strings, 91
end-around-carry technique, 228	next higher number, same number of 1
estimating multiplication overflow, 31	bits, 13–14
Euclidean algorithm, 192–195	the <i>n</i> th prime, 271–278, 282
Euler, Leonhard, 271–272	strings of 1-bits
	application, 96
even parity, 74 exact division	first string of a given length, 96–99
	values within arithmetic bounds, 95–9
definition, 190	flipping bits, 102
multiplicative inverse, Euclidean algo-	floating-point numbers
rithm, 192–195	comparing using integer operations,
multiplicative inverse, Newton's	263–264
method, 195–197	denormalized numbers, 262
multiplicative inverse, samples, 197	distribution of leading digits, 264–267
exchanging	gradual underflow, 262
conditionally, 41	IEEE format, 261–263
corresponding register fields, 39-40	normalized numbers, 262
two fields in same register, 40	table of values, 267–269
two registers, 38–39	
exclusive or	floating-point operations
propagating arithmetic bounds	counting leading 0's with, 81–82
through, 62	simulating, 83
scan operation on an array of bits, 75	floor division, 137–138, 188
in three instructions, 16	floor function, identities, 139–140
execution time model, 9	Floyd, R. W., 87
exponentiation	formula functions, 278–284
by binary decomposition, 212–214	formulas for primes, 271–284
in Fortran, 214–215	Fortran, integer exponentiation, 214–215
powers of 2, testing for, 11–12	
expressions	G
arithmetic bounds, 54–55	Gardner, Martin, 239
range analysis, 54–55	generalized extract operation, 116–122
extract instruction, 122	Gosper, R. W.
extracting bits	iterating through subsets, 14
alternate bits, 104	loop-detection, 87–89
generalized extract, 116–122	gradual underflow, 262
	graphics-rendering, Hilbert's curve,
F	258–259
factoring, 136	Gray, Frank, 239
	Gray code
Fast Fourier Transform (FFT), 104–105	applications, 239–240
Fermat numbers, 271 FFT (Fast Fourier Transform), 104–105	converting integers to, 75, 236
	definition, 235
find leftmost 0-byte, 91–94	incrementing Gray-coded integers,
find rightmost 0-byte, 91–92, 94–95	237–239
finding	negabinary Gray code, 239
decimal digits, 95	reflected, 235–237
first 0-byte, 91–95	GRP instruction, 126
	GIG Instruction, 120













INDEX

H J-K hacker, definition, xiv Jensen, Eric, 87 half shuffle, 108 halfwords, 1 Hamming distance, 73 high-order half of product, 132-133 Hilbert's curve. See also space-filling curves. applications, 258-259 coordinates from distance 253-254 description, 246 Landry, F., 271 Lam and Shapiro method, 248-250 parallel prefix operation, 251–252 state transition table, 246-248 description, 241–242 distance from coordinates, 252–254 endian, 101 examples, 241, 243 generating, 242-245 load instruction, 9 incrementing coordinates, 254–257 non-recursive generation, 257 tion, 92 ray tracing, 258-259 logarithms definition, 215–216 IBM Stretch computer, 65 IBM System/360 computer, 49, 264-265 IEEE format, floating-point numbers, 261-263 logical operations image processing, Hilbert's curve, 258-259 incremental division and remainder techbinary, table of, 17 nique, 172, 183-184 58-63 inequalities, logical and arithmetic expressions, 16-17 inner shuffle, 106-107 loop detection, 87-89 insert instruction, 122 instruction-level parallelism, 9 tion, 92 instruction set for this book, 4-9 integer log base 2 function, 83, 215-216 M integers. See also specific operations on integers. magic numbers complex, 230-233 converting to Gray code, 75, 236 reversed, incrementing, 104-106 reversing, 101-104 definition, 161 ISIGN (transfer of sign) function, 20

isolating rightmost bits, 11

iterating through subsets, 14

Knuth's Algorithm D, 140-144 Knuth's Algorithm M, 129-130 Kronecker, Leopold, 261

Lam and Shapiro method, 248–250, leading 0's, counting, 77-82. See also number of leading zeros (nlz) function. leading digits, distribution, 264–267 little-endian format, converting to bigload immediate instruction, 8-9 load word byte-reverse (lwbrx) instrucbinary search method, 216-218 log base 2, 83, 215-216

log base 10, 215–221 table lookup, 216, 218–221 with addition and subtraction, 15-16 propagating arithmetic bounds through, tight bounds, 58-63

lwbrx (load word byte-reverse) instruc-

magic algorithm, 171-174 Alverson's method, 188-189 calculating, signed, 162-163, 171-174 calculating, unsigned, 182-187 samples, 189-190 table lookup, 188 uniqueness, 173-175







negabinary number system description, 223-230 Gray code, 239

302





magicu algorithm, 183-185 negative absolute value, 21 magicu2 algorithm, 186-187 negative overflow, 28 max function, 37-38 Newton's method Mills, W. H., 284 description, 289-290 min function, 37-38 integer cube root, 211 modu (unsigned modulus) function, 68 integer square root, 203-206 modulus division, 137-138, 188 multiplicative inverse, 195-197 Moore, Eliakim Hastings, 257-258 next higher number, same number of 1-bits, mulhs (multiply high signed) instruction 13 - 14division with, 158-160, 186 nibbles, 1 implementing in software, 132 nlz (number of leading zeros) function. See mulhu (multiply high unsigned) number of leading zeros (nlz) instruction function. division with, 178-180, 186 nonrestoring algorithm, 148, 150-151 implementing in software, 132 normalized numbers, 262 multibyte absolute value, 37 notation used in this book, 1-4 multibyte addition/subtraction, 36-37 ntz (number of trailing zeros) function. See multiplication number of trailing zeros (ntz) arithmetic tables, 286 function. by constants, 133-136 number of leading zeros (nlz) function factoring, 136 applications, 83, 96 high-order half of 64-bit product, 132 bitsize function, 83 high-order product signed from/to comparison predicates, 21-22, 83 unsigned, 132-133 counting trailing 0's, 84 multiword, 129-132 finding 0-bytes, 91-92 of negabinary numbers, 226 finding strings of 1-bits, 96-99 overflow detection, 29-32 incrementing reversed integers, 106 multiplicative inverse and integer log base 2 function, 83 Euclidean algorithm, 192-195 overflow detection, multiplication, 31 Newton's method, 195-197 random number generation, 83 samples, 197-198 rounding to powers of 2, 47 multiply high signed (mulhs) instruction number of trailing zeros (ntz) function division with, 158-160, 186 applications, 87-89 implementing in software, 132 counting trailing 0's, 84-87 multiply high unsigned (mulhu) finding strings of 1-bits, 83 instruction loop detection, 87-89 division with, 178-180, 186 random number generation, 83 ruler function, 87 implementing in software, 132 multiply instruction, condition codes, number systems 33-34 base -1 + i, 230–232 multiword division, 140-145 base -1 - i, 232–233 multiword multiplication, 129-132 base -2, 223-230, 239 base efficiency, 233-234 negabinary, 223-230, 239

INDEX

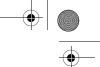












INDEX

303

0	planar curves, 241. See also Hilbert's
odd parity, 74	curve.
1-bits	poems, 202, 211
counting. See counting bits.	population count function
identifying, 11	applications, 73–74
isolating, 11	computing Hamming distance, 73
right-propagating, 12	counting 1-bits, 65
rightmost, turning off, 11–12	counting leading 0's, 79–80
or, in three instructions, 16	counting trailing 0's, 84
ordinary arithmetic, 1	position sensor, 239–240
ordinary rational division, 137	powers of 2
outer shuffle, 106–108	boundary crossings, detecting, 49–50
overflow detection	identifying, 11–12
definition, 26	rounding to, 45–48
division, 32–33	signed division, 155–156
	unsigned division, 178
estimating multiplication overflow, 31	PPERM instruction, 126
multiplication, 29–32	prime numbers
negative overflow, 28	Fermat numbers, 271
with number of leading zeros (nlz)	finding the <i>n</i> th prime
instruction, 31	formula functions, 278–284
signed add/subtract, 26–28	Willans's formulas, 273–277
subtraction, 27–29	Wormell's formulas, 277–278
unsigned add/subtract, 29	formulas for, 271–284
_	from polynomials, 272
P	propagating
parallel prefix operation	arithmetic bounds
compress operation, 119-120	add and subtract instructions, 54–57
definition, 75	logical operations, 58–63
generalized extract, 117	through exclusive or, 62
Hilbert's curve, 251–252	rightmost 1-bit, 12
parity, 75	rightmost 1-ort, 12
swap and complement, 251-252	O D
vs. insert/extract instructions, 122	Q–R
parity	quicksort, 65
adding to 7-bit quantities, 76	
application, 77	range analysis, 54
computing, 74–76	ranges. See arithmetic bounds.
definition, 74	ray tracing, Hilbert's curve, 258–259
parallel prefix operation, 75	rearranging arrays, 127
scan operation, 75	reflected binary Gray code, 235–236, 239
Peano, Giuseppe, 241. See also Hilbert's	registers
curve.	exchanging, 38–39
Peano-Hilbert curve. See Hilbert's curve.	exchanging conditionally, 41
perfect shuffle, 106–108	exchanging fields of, 39–40
permutations on bits, 122–126. See also bit	reversing contents of, 101-104
operations.	RISC computers, 4–6









•



INDEX

 \mathbf{S} remainders arithmetic tables, 288 SAG (sheep and goats) operation, 122-126 signed division scan operation, 75 from non-powers of 2, 157-160 Schroeppel's formula, 229-230 from powers of 2, 156-157 search tree method, 84-85 test for zero, 200-201 searching. See finding. unsigned division sheep and goats (SAG) operation, 122-126 and immediate instruction, 178 shift-and-subtract algorithm incremental division and remainder hardware, 148-151 technique, 172, 183-184 integer square root, 209–210 test for zero, 198-199 shift left double operation, 35 remu function, 102-103 shift right arithmetic instruction, 8 restoring algorithm, 148-150 shift right double signed operation, 36 reversing shift right double unsigned operation, 35 6-, 7-, 8-, and 9-bit quantities, 102-104 shift right extended immediate (shrxi) bits, 101-104 instruction, 179-180 bytes, 92, 101-102 shift right signed instruction generalized, 102 alternative to, for sign extension, 18 integers, 101-104 division by power of 2, 155–156 load word byte-reverse (lwbrx) from unsigned, 18-19 instruction, 92 register contents, 101-104 double-length, 35-36 right-propagating 1-bits, 12 rotate, 34 rightmost bits short division, 145-148, 151-152 0-bits shrxi (shift right extended immediate) isolating, 11 instruction, 179-180 trailing 0's, counting, 74, 84-87 shuffling trailing 0's, identifying, 11 arrays, 127 turning on, 12 bits, 106-108, 116, 127 1-bits sign extension, 18 identifying, 11 sign function, 19. See also three-valued isolating, 11 compare function. right-propagating, 12 signed bounds, 62-63 turning off, 11-12 signed comparisons, from unsigned, 23-24 Rijndael algorithm, 126 signed computer division, 137-138 RISC signed division basic instruction set, 5-8 arithmetic tables, 287 execution time model, 9 best programs for, 175–178 extended mnemonics, 7 by divisors ≤ -2 , 168–171 full instruction set, 6-8 by divisors ≥ 2 , 160–168 rotate and sum method, 69-70 incorporating into a compiler, 171-173 rotate shifts, 34 by powers of 2, 155–156 rounding to powers of 2, 45-48 remainder from non-powers of 2, ruler function, 87 157-160 remainder from powers of 2, 156–157 test for zero remainder, 200-201 uniqueness of magic number, 173-175











INDEX

signed long division, 145-146, 153 transposing a bit matrix signed short division, 146-148 signum function. See sign function; threevalued compare function. Triple DES, 126 space-filling curves, 257-258. See also Hilbert's curve. sparse array indexing, 73-74 square root, integer binary search, 207-209 hardware algorithm, 209-211 Newton's method, 203-207 unshuffling shift-and-subtract algorithm, 209-211 arrays, 127 Stibitz, George, 232 Stretch computer, 65 string length (strlen) function, 91 strings. See bit operations; character strings. strlen (string length) function, 91 subtract instruction condition codes, 33-34 propagating arithmetic bounds, 54-57 subtraction arithmetic tables, 285 difference or zero (doz) function, 37-38 double-length, 34-35 and logical operations, 15-16 multibyte, 36-37 of negabinary numbers, 225-226 overflow detection, 26-28 uppercase letters, finding, 96 in various number encodings, 228-229 swap and complement method, 249-250 V-Wswapping pointers, 39 System/360 computer, 49, 264–265 Willans, C. P., 273–277 table lookup, counting bits, 71

three-valued compare function, 19-20. See also sign function. tight bounds add and subtract instructions, 54-57 logical operations, 58-63 toggling among values, 41-44 Tower of Hanoi puzzle, 89, 239 trailing 0's. See also number of trailing zeros (ntz) function. counting, 84-87 identifying, 11 transfer of sign (ISIGN) function, 20

8 x 8, 108–111, 116 32 x 32, 111–116 true/false comparison results, 21 turning off 1-bits, 11-12, 69 turning on 0-bits, 12

uniqueness, of magic numbers, 173-175 bits, 107-108, 116, 127 unsigned division by 3 and 7, 178-180 arithmetic tables, 287 best programs for, 184-186 computer division, 137 by divisors ≥ 1 , 180–183 incorporating into a compiler, 183-184 incremental division and remainder technique, 172, 183-184 long division, 148-153 by powers of 2, 178 remainders, from powers of 2, 178 short division, from signed, 145-148 test for zero remainder, 198-199 unsigned modulus (modu) function, 68

Voorhies, Douglas, 259

Wilson's theorem, 273 word parity. See parity. words counting bits, 65–72 definition, 1 division doubleword by single word, 148-153 Knuth's Algorithm D, 140-144 multiword, 140-145 signed, multiword, 144-145 single word by single word, 145-148





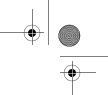












INDEX

multiplication, multiword, 129-132 reversing, 101-104 searching for first 0-byte, 91-95 first uppercase letter, 96 strings of 1-bits, 96-99 a value within a range, 95–96 word parallel operations, 12-13 Wormell, C. P., 277–278

\mathbf{Z} zbytel function, 91-95 zbyter function, 91-95 zero means 2ⁿ, 20–21 0-bits counting. See counting bits. isolating, 11 trailing 0's counting, 74, 84-87 identifying, 11 turning on, 12 0-bytes, finding, 91-95













