Power of two

In mathematics, a **power of two** means a number of the form 2^n where n is an integer, i.e. the result of exponentiation with as base the number two and as exponent the integer n.

In a context where only integers are considered, n is restricted to non-negative values,[1] so we have 1, 2, and 2 multiplied by itself a certain number of times.[2]

Because two is the base of the binary numeral system, powers of two are common in computer science. Written in binary, a power of two always has the form 100…0 or 0.00…01, just like a power of ten in the decimal system.

Expressions and notations

Verbal expressions, mathematical notations, and computer programming expressions using a power operator or function include:

2 to the power of n

2 power n

power(2, n)

pow(2, n)

2^n

2 ^ n

2 ** n

Computer science

Two to the power of *n*, written as 2^n, is the number of ways the bits in a binary integer of length *n* can be arranged; depending on the integer type these may represent partly positive and partly negative numbers and zero, or just non-negative ones. Either way, one less than a power of two is often the upper bound of an integer in binary computers. As a consequence, numbers of this form show up frequently in computer software. As an example, a video game running on an 8-bit system might limit the score or the number of items the player can hold to 255—the result of using a byte, which is 8 bits long, to store the number, giving a maximum value of $2^8 - 1 = 255$. For example, in the original *Legend of Zelda* the main character was limited to carrying 255 rupees (the currency of the game) at any given time.

Powers of two are often used to measure computer memory. A byte is now considered to be eight bits (an octet, resulting in the possibility of 256 values (2^8). (The term *byte* has been, and in some case continues to be, used to be a collection of bits, typically of 5 to 32 bits, rather than only an 8-bit unit.) The prefix *kilo*, in conjunction with *byte*, may be, and has traditionally been, used, to mean 1,024 (2^{10}). However, in general, the term *kilo* has been used in the System International to mean 1,000 (10^3). Binary prefixes have been standardised, such as *kibi*' meaning 1,024. Nearly all processor registers have sizes that are powers of two, 32 or 64 being most common.

Powers of two occur in a range of other places as well. For many disk drives, at least one of the sector size, number of sectors per track, and number of tracks per surface is a power of two. The logical block size is almost always a power of two.

Numbers which are not powers of two occur in a number of situations such as video resolutions, but they are often the sum or product of only two or three powers of two, or powers of two minus one. For example, 640 = 512 + 128, and 480 = 32 × 15. Put another way, they have fairly regular bit patterns.

Mersenne primes

A prime number that is one less than a power of two is called a Mersenne prime. For example, the prime number 31 is a Mersenne prime because it is 1 less than 32 (2^5). Similarly, a prime number (like 257) that is one more than a power of two is called a Fermat prime; the exponent will itself be a power of two. A fraction that has a power of two as its denominator is called a dyadic rational. The numbers that can be represented as sums of consecutive positive integers are called polite numbers; they are exactly the numbers that are not powers of two.

The first 84 powers of two

2^{0}	**=**	**1**	2^{12}	**=**	**4,096**	2^{24}	**=**	**16,777,216**	2^{36}	**=**	**68,719,476,736**	2^{48}	**=**	**281,474,976,710,656**	2^{60}	**=**	**1,152,921,504,606,846,976**	2^{72}	**=**	**4,722,366,482,869,645,213,696**
2^{1}	=	2	2^{13}	=	8,192	2^{25}	=	33,554,432	2^{37}	=	137,438,953,472	2^{49}	=	562,949,953,421,312	2^{61}	=	2,305,843,009,213,693,952	2^{73}	=	9,444,732,965,739,290,427,392
2^{2}	=	4	2^{14}	=	16,384	2^{26}	=	67,108,864	2^{38}	=	274,877,906,944	2^{50}	=	1,125,899,906,842,624	2^{62}	=	4,611,686,018,427,387,904	2^{74}	=	18,889,465,931,478,580,854,784
2^{3}	=	8	2^{15}	=	32,768	2^{27}	=	134,217,728	2^{39}	=	549,755,813,888	2^{51}	=	2,251,799,813,685,248	2^{63}	=	9,223,372,036,854,775,808	2^{75}	=	37,778,931,862,957,161,709,568
2^{4}	**=**	16	2^{16}	**=**	**65,536**	2^{28}	**=**	**268,435,456**	2^{40}	**=**	**1,099,511,627,776**	2^{52}	**=**	**4,503,599,627,370,496**	2^{64}	**=**	**18,446,744,073,709,551,616**	2^{76}	**=**	**75,557,863,725,914,323,419,136**
2^{5}	=	32	2^{17}	=	131,072	2^{29}	=	536,870,912	2^{41}	=	2,199,023,255,552	2^{53}	=	9,007,199,254,740,992	2^{65}	=	36,893,488,147,419,103,232	2^{77}	=	151,115,727,451,828,646,838,272
2^{6}	=	64	2^{18}	=	262,144	2^{30}	=	1,073,741,824	2^{42}	=	4,398,046,511,104	2^{54}	=	18,014,398,509,481,984	2^{66}	=	73,786,976,294,838,206,464	2^{78}	=	302,231,454,903,657,293,676,544
2^{7}	=	128	2^{19}	=	524,288	2^{31}	=	2,147,483,648	2^{43}	=	8,796,093,022,208	2^{55}	=	36,028,797,018,963,968	2^{67}	=	147,573,952,589,676,412,928	2^{79}	=	604,462,909,807,314,587,353,088
2^{8}	**=**	256	2^{20}	**=**	**1,048,576**	2^{32}	**=**	**4,294,967,296**	2^{44}	**=**	**17,592,186,044,416**	2^{56}	**=**	**72,057,594,037,927,936**	2^{68}	**=**	**295,147,905,179,352,825,856**	2^{80}	**=**	**1,208,925,819,614,629,174,706,176**
2^{9}	=	512	2^{21}	=	2,097,152	2^{33}	=	8,589,934,592	2^{45}	=	35,184,372,088,832	2^{57}	=	144,115,188,075,855,872	2^{69}	=	590,295,810,358,705,651,712	2^{81}	=	2,417,851,639,229,258,349,412,352
2^{10}	=	1,024	2^{22}	=	4,194,304	2^{34}	=	17,179,869,184	2^{46}	=	70,368,744,177,664	2^{58}	=	288,230,376,151,711,744	2^{70}	=	1,180,591,620,717,411,303,424	2^{82}	=	4,835,703,278,458,516,698,824,704
2^{11}	=	2,048	2^{23}	=	8,388,608	2^{35}	=	34,359,738,368	2^{47}	=	140,737,488,355,328	2^{59}	=	576,460,752,303,423,488	2^{71}	=	2,361,183,241,434,822,606,848	2^{83}	=	9,671,406,556,917,033,397,649,408

One can see that starting with 2 the last digit is periodic with period 4, with the cycle 2 4 8 6, and starting with 4 the last two digits are periodic with period 20. These patterns are generally true of any power, with respect to any base. The pattern continues, of course, where each pattern has starting point 2^k length the multiplicative order of 2 modulo 5^k.

Powers of 1024

The first few powers of 2^{10} are a little more than those of 1000:

2^{10}	=	1 024	$\approx 10^{3}$	(2.3% deviation)
2^{20}	=	1 048 576	$\approx 10^{6}$	(4.6% deviation)
2^{30}	=	1 073 741 824	$\approx 10^{9}$	(6.9% deviation)
2^{40}	=	1 099 511 627 776	$\approx 10^{12}$	(9.1% deviation)
2^{50}	=	1 125 899 906 842 624	$\approx 10^{15}$	(11.2% deviation)
2^{60}	=	1 152 921 504 606 846 976	$\approx 10^{18}$	(13.3% deviation)
2^{70}	=	1 180 591 620 717 411 303 424	$\approx 10^{21}$	(15.3% deviation)

See also IEEE 1541-2002.

Powers of two whose exponents are powers of two

Because data (specifically integers) and the addresses of data are stored using the same hardware, and the data is stored in one or more octets (2^3), double exponentials of two are common. For example,

$2^1 = 2$

$2^2 = 4$

$2^4 = 16$

$2^8 = 256$

$2^{16} = 65,536$

$2^{32} = 4,294,967,296$

$2^{64} = 18,446,744,073,709,551,616$

$2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456$

$2^{256} = 115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936$.

Several of these numbers represent the number of values representable using common computer data types. For example, a 32-bit word consisting of 4 bytes can represent 2^{32} distinct values, which can either be regarded as mere bit-patterns, or are more commonly interpreted as the unsigned numbers from 0 to $2^{32} - 1$, or as the range of signed numbers between -2^{31} and $2^{31} - 1$. Also see tetration and lower hyperoperations. For more about representing signed numbers see two's complement.

Some selected powers of two

$2^8 = 256$

The number of values represented by the 8 bits in a byte, also known as an octet. (The term byte is often defined as a collection of bits rather than the strict definition of an 8-bit quantity, as demonstrated by the term kilobyte.)

$2^{10} = 1,024$

The binary approximation of the kilo-, or 1,000 multiplier, which causes a change of prefix. For example: 1,024 bytes = 1 kilobyte (or kibibyte).

This number has no special significance to computers, but is important to humans because we make use of powers of ten.

$2^{12} = 4,096$

The hardware page size of Intel x86 processor.

$2^{16} = 65,536$

The number of distinct values representable in a single word on a 16-bit processor, such as the original x86 processors.[3]

The maximum range of a short integer variable in the C, C++, C#, and Java programming languages. The maximum range of a **Word** or **Smallint** variable in the Pascal programming language.

$2^{20} = 1,048,576$

The binary approximation of the mega-, or 1,000,000 multiplier, which causes a change of prefix. For example: 1,048,576 bytes = 1 megabyte (or mibibyte).

This number has no special significance to computers, but is important to humans because we make use of powers of ten.

$2^{24} = 16,777,216$

The number of unique colors that can be displayed in truecolor, which is used by common computer monitors.

This number is the result of using the three-channel RGB system, with 8 bits for each channel, or 24 bits in total.

2^{30} = 1,073,741,824

The binary approximation of the giga-, or 1,000,000,000 multiplier, which causes a change of prefix. For example, 1,073,741,824 bytes = 1 gigabyte (or gibibyte).

This number has no special significance to computers, but is important to humans because we make use of powers of ten.

2^{32} = 4,294,967,296

The number of distinct values representable in a single word on a 32-bit processor. Or, the number of values representable in a doubleword on a 16-bit processor, such as the original x86 processors.[3]

The range of an `int` variable in the Java and C# programming languages.

The range of an `Cardinal` or `Integer` variable in the Pascal programming language.

The minimum range of a long integer variable in the C and C++ programming languages.

The total number of IP addresses under IPv4. Although this is a seemingly large number, IPv4 address exhaustion is imminent.

2^{40} = 1,099,511,627,776

The binary approximation of the tera-, or 1,000,000,000,000 multiplier, which causes a change of prefix. For example, 1,099,511,627,776 bytes = 1 terabyte (or tebibyte).

This number has no special significance to computers, but is important to humans because we make use of powers of ten.

2^{64} = 18,446,744,073,709,551,616

The number of distinct values representable in a single word on a 64-bit processor. Or, the number of values representable in a doubleword on a 32-bit processor. Or, the number of values representable in a quadword on a 16-bit processor, such as the original x86 processors.[3]

The range of a long variable in the Java and C# programming languages.

The range of a **Int64** or **QWord** variable in the Pascal programming language.

The total number of IPv6 addresses generally given to a single LAN or subnet.

One more than the number of grains of rice on a chessboard, according to the old story, where the first square contains one grain of rice and each succeeding square twice as many as the previous square.

2^{96} = 79,228,162,514,264,337,593,543,950,336

The total number of IPv6 addresses generally given to a local Internet registry. In CIDR notation, ISPs are given a /32, which means that 128-32=96 bits are available for addresses (as opposed to network designation). Thus, 2^{96} addresses.

2^{128} = 340,282,366,920,938,463,463,374,607,431,768,211,456

The total number of IP addresses available under IPv6.

$2^{43,112,609} - 1$ = 316,470,269, …, 697,152,511

The largest known prime number as of 2009. It has 12,978,189 digits.

Fast algorithm to check if a positive number is a power of two

The binary representation of integers makes it possible to apply a very fast test to determine whether a given positive integer *x* is a power of two:

positive *x* is a power of two ⇔ $(x \ \& \ (x - 1))$ equals zero.

where *&* is a bitwise logical *AND* operator. Note that if *x* is 0, this incorrectly indicates that 0 is a power of two, so this check only works if $x > 0$.

Examples:

−1	=	1…111…1	−1	=	1…111…111…1
x	=	0…010…0	y	=	0…010…010…0
x − 1	=	0…001…1	y−1	=	0…010…001…1
x & (x − 1)	=	0…000…0	y & (y − 1)	=	0…010…000…0

Proof of Concept:

Proof uses the technique of contrapositive.

Statement, S: If x&(x-1) = 0 and x is an integer greater than zero then $x = 2^k$ (where k is an integer such that k>=0).

Concept of Contrapositive:

S1: P -> Q is same as S2: ~Q -> ~P

In above statement S1 and S2 both are contrapositive of each other.

So statement S can be re-stated as below

S': If x is a positive integer and $x \neq 2^k$ (k is some non negative integer)then x&(x-1) ≠ 0

Proof:

If $x \neq 2^k$ then at least two bits of x are set.(Let's assume m bits are set.)

Now, bit pattern of x - 1 can be obtained by inverting all the bits of x up to first set bit of x(starting from LSB and moving towards MSB, this set bit inculsive).

Now, we observe that expression x & (x-1) has all the bits zero up to the first set bit of x and since x & (x-1) has remaining bits same as x and x has at least two set bits hence predicate x & (x-1) ≠ 0 is true.

Fast algorithm to find a number modulo a power of two

As a generalization of the above, the binary representation of integers makes it possible to calculate the modulos of a non-negative integer (*x*) with a power of two (*y*) very quickly:

x modulo y ⇔ $(x \ \& \ (y - 1))$.

where *&* is a bitwise logical *AND* operator. This bypasses the need to perform an expensive division. This is useful if the modulo operation is a significant part of the performance critical path as this can be much faster than the regular modulo operator.

Algorithm to convert any number into nearest power of two number

The following formula finds the nearest power of two, on a logarithmic scale, of a given value $x > 0$:

$$2^{\text{round}[\log_2(x)]}$$

Computer pseudocode:

```
pot = 2^roundup(log2(npot));
```

This should be distinguished from the nearest power of two on a linear scale. For example, 23 is nearer to 16 than it is to 32, but the previous formula rounds it to 32, corresponding to the fact that 23/16=1.4375, larger than 32/23=1.3913.

If x is an integer value, following steps can be taken to find the nearest value (with respect to actual value rather than the binary logarithm) in a computer program:

1. Find the most significant bit k, that is set (*1*) from the binary representation of x, when $k = 0$ means the least significant bit
2. Assume that all bits $k < 0$ are zero. Then, if bit $k-1$ is zero, the result is 2^k. Otherwise the result is 2^{k+1}.

A C++ version of this code for the unsigned integer type T would be:

```
template <class T>
T nearestpower2(T v)
{
        int k;
        if (v == 0)
                return 1;
        for (k = sizeof(T) * 8 - 1; ((static_cast<T>(1U) << k) & v) == 0; k--);
        if (((static_cast<T>(1U) << (k - 1)) & v) == 0)
                return static_cast<T>(1U) << k;
        return static_cast<T>(1U) << (k + 1);
}
```

Algorithm to round up to power of two

Sometimes it is desired to find the least power of two that is not less than a particular integer, *n*. The pseudocode for an algorithm to compute the next-higher power of two is as follows. If the input is a power of two it is returned unchanged.[4]

```
n = n - 1;
n = n | (n >> 1);
n = n | (n >> 2);
n = n | (n >> 4);
n = n | (n >> 8);
n = n | (n >> 16);
...
n = n | (n >> (bitspace / 2));
n = n + 1;
```

Where | is a binary or operator, >> is the binary right-shift operator, and bitspace is the size (in bits) of the integer space represented by n. For most computer architectures, this value is either 8, 16, 32, or 64. This operator works by setting all bits on the right-hand side of the most significant flagged bit to *1*, and then incrementing the entire value at the end so it "rolls over" to the nearest power of two. An example of each step of this algorithm for the number

2689 is as follows:

Binary representation	Decimal representation
0101010000001	2,689
0101010000000	2,688
0111111000000	4,032
0111111110000	4,080
0111111111111	4,095
1000000000000	4,096

As demonstrated above, the algorithm yields the correct value of 4,096. The nearest power to 2,689 happens to be 2,048; however, this algorithm is designed only to give the *next highest* power of two to a given number, not the nearest.

A C++ version of this code for the signed integer type T would be:

```
template <class T>
T nexthigher(T k) {
        k--;
        for (int i=1; i<sizeof(T)*CHAR_BIT; i<<=1)
                k = k | k >> i;
        return k+1;
}
```

For unsigned integers, the code would be:

```
template <class T>
T nexthigher(T k) {
        if (k == 0)
                return 1;
        k--;
        for (int i=1; i<sizeof(T)*CHAR_BIT; i<<=1)
                k = k | k >> i;
        return k+1;
}
```

Note: CHAR_BIT is defined in <climits>

Other properties

The number of vertices of an n-dimensional hypercube is 2^n. Similarly, the number of (n − 1)-faces of an n-dimensional cross-polytope is also 2^n and the formula for the number of x-faces an n-dimensional cross-polytope has is $2^x\binom{n}{x}$.

The sum of the reciprocals of the powers of two is 2. The sum of the reciprocals of the squared powers of two is 1⅓.

References

[1] Lipschutz, Seymour (1982). *Schaum's Outline of Theory and Problems of Essential Computer Mathematics*. New York: McGraw-Hill. pp. 3. ISBN 0070379904.

[2] Sewell, Michael J. (1997). *Mathematics Masterclasses*. Oxford: Oxford University Press. pp. 78. ISBN 0198514948.

[3] Though they vary in word size, all x86 processors use the term "word" to mean 16 bits; thus, a 32-bit x86 processor refers to its native wordsize as a dword

[4] Warren Jr., Henry S. (2002). *Hacker's Delight*. Addison Wesley. pp. 48. ISBN 978-0201914658

Article Sources and Contributors

Power of two *Source*: http://en.wikipedia.org/w/index.php?oldid=482716979 *Contributors*: 121a0012, Aaron Brenneman, AndrewWTaylor, Avalon2009, Bevo, BioTube, Bobo192, Bpatton, Calabe1992, Catapult, Chandlerklebs, CodeMercenary, Colfulus, Cyan, David Eppstein, Davidhorman, Dcoetzee, DerHexer, Direvus, Dori, Długosz, Esprit15d, Fred Gandt, Fredrik, Gaius Cornelius, Georgia guy, Giftlite, Glome83, GorillaWarfare, Gus Polly, Hemanshu, HenningThielemann, Herbee, Hooperbloob, Hu, Ixfd64, J.delanoy, Jernejl, Jobe6, Julienlecomte, Justin W Smith, Karl Palmen, Katalaveno, Khazar, Laurentius, Logan, MathMartin, MegaHasher, Melchoir, Michael Hardy, Midnightcomm, Mini-Geek, Neelix, Neo139, Numbo3, Omegatron, Patrick, PeterNLewis, Petersk, Piet Delport, Pithree, PrimeFan, PrimeHunter, Proofreader77, Qiq, R. S. Shaw, Ramsey2006, RedWolf, RekishiEJ, Renesis, Rich Farmbrough, Robo37, Rollett3785, Ryan Cable, Safalra, Saxonjf, Sbrools, School3, ScottMHoward, Sekelsenmat, Shd, Shlomital, Sindikat, Smithj, Softlavender, Stdazi, Stevenj, Stw, SunnySideOfStreet, Superunintelligible, TakuyaMurata, Tamfang, Team4Technologies, The Anome, Tobias Bergemann, Twri, Vadmium, Wegge, Wiwaxia, Wl219, Yoichi123, 172 anonymous edits

License