Clay S. Turner

A Fast Binary Logarithm Algorithm

"DSP Tips and Tricks" introduces practical design and implementation signal processing algorithms that you may wish to incorporate into your designs. We welcome readers to submit their contributions. Contact Associate Editors Rick Lyons (R.Lyons@ieee.org) or C. Britton Rorabaugh (dspboss@aol.com).

This article presents a computationally fast algorithm for computing logarithms. The algorithm is particularly well suited for implementation using fixed-point processors.

BACKGROUND

Long before the technological age, in 1427, the Persian mathematician and astronomer al Kashi presented an algorithm for efficient integral exponentiation via repeated squaring and multiplying [1]. Al Kashi's method now sees much use in modern cryptographic systems where its efficiency with respect to very large numbers is paramount. Related but not as well known as al Kashi's method is an algorithm for finding binary logarithms via repeated squaring and dividing. Performing these operations in a radix two-number format reduces the divisions to binary shifts, thus making the algorithm amenable to fixed-point implementations on low-complexity microprocessors and field-programmable gate arrays.

THE LOGARITHM ALGORITHM

Now that we have revealed the two main types of computations for finding the binary logarithm, let's go through our algorithm's mathematical development. To start, we simply desire to find

$$y = \log_2(x). \tag{1}$$

Thus we can invert this and find

$$x = 2^y. \tag{2}$$

Since the iterative part of our log algorithm assumes x is in $1 \le x < 2$, we may need to "normalize" x. This normalization is performed by a simple succession of divides/multiplies by two. These divides/multiplies by two may be efficiently effected by binary shifts. The count of the divides/multiplies gives the characteristic of our logarithm result. This will be added to the mantissa (calculated below) to form the complete logarithm. The number of divides is taken as a positive number and the number of multiplies is taken as a negative number.

Now with x properly scaled, then we know from (1) that $0 \le y < 1$. So let's expand y into a binary series, thus

$$y = y_1 \cdot 2^{-1} + y_2 \cdot 2^{-2} + y_3 \cdot 2^{-3} + y_4 \cdot 2^{-4} + \cdots . \tag{3}$$

It will be efficacious to put this into nested parenthetical form

$$y = 2^{-1}(y_1 + 2^{-1}(y_2 + 2^{-1} \times (y_3 + 2^{-1}(y_4 + \cdots)))). \tag{4}$$

So putting (4) into (2) we have

$$x = 2^{2^{-1}(y_1 + 2^{-1}(y_2 + 2^{-1}(y_3 + 2^{-1}(y_4 + \ldots))))}. \tag{5}$$

Now we are ready to see how to sequentially extract the bits comprising y.

Step 1: Square x as

$$x^2 = 2^{y_1} \cdot 2^{2^{-1}(y_2 + 2^{-1}(y_3 + 2^{-1}(y_4 + \cdots)))}. \tag{6}$$

We see the right-hand side of (6) is a product of two factors, where the left factor is equal to either one or two depending on the value of y_1. Thus we have the following two cases:

$$y_1 = 0: \quad x^2 = 2^{2^{-1}(y_2 + 2^{-1}(y_3 + 2^{-1}(y_4 + \cdots)))} \tag{7}$$

or

$$y_1 = 1: \quad x^2 = 2 \cdot 2^{2^{-1}(y_2 + 2^{-1}(y_3 + 2^{-1}(y_4 + \cdots)))}. \tag{8}$$

Since both (7) and (8) contain the common factor $2^{2^{-1}(y_2 + 2^{-1}(y_3 + 2^{-1}(y_4 + \ldots)))}$, and this factor will have a value that is greater than or equal to one and less than two, we see the square of x will the greater than or equal to two if and only if $y_1 = 1$, otherwise $y_1 = 0$.

Step 2: Compare the result of squaring and if x^2 is greater than or equal to two, then set mantissa bit to "1" and divide x^2 by two. Otherwise set mantissa bit to "0" and leave x^2 unchanged.

So now at the end of step 2, we have

$$x^2 = 2^{2^{-1}(y_2 + 2^{-1}(y_3 + 2^{-1}(y_4 + \ldots)))}. \tag{9}$$

Realizing that x^2 is just a number, we see that (9) has the exact same form as (5). Thus we can repeat Steps 1 and 2 to extract the remaining bits in y.

After the prerequisite number of bits in the mantissa is obtained, then simply add the characteristic to the mantissa. If we require a logarithm having a radix other than two, then we may employ the following property of logarithms

$$\log_a(x) = \frac{\log_2(x)}{\log_2(a)}, \tag{10}$$

where the division by $\log_2(a)$ is implemented as a multiplication by the fixed value $1/\log_2(a)$.

Digital Object Identifier 10.1109/MSP.2010.937503

(continued on page 140)

1053-5888/10/$26.00©2010IEEE

Please send calendar submissions to:
Dates Ahead, c/o Jessica Barragué,
IEEE Signal Processing Magazine
445 Hoes Lane
Piscataway, NJ 08855 USA,
e-mail: j.barrague@ieee.org
(Colored conference title indicates SP-sponsored conference.)

2010

SEPTEMBER

2010 International Conference on Image Processing (ICIP 2010)
26–29 September, Hong Kong.
General Chair: Wan-Chi Siu
URL: http://www.icip2010.org

OCTOBER

The 6th IEEE Sensor Array and Multichannel Signal Processing Workshop (SAM '10)
4–7 October, Israel.
General Cochairs: Hagit Messer and Jeffrey L. Krolik
URL: http://www.sam-2010.org/

The 1st IEEE International Conference on Smart Grid Communications
4–6 October, Gaithersburg, Maryland.
General Cochairs: George Arnold and Stefano Galli
URL: http://www.ieee-smartgridcomm.org

Digital Object Identifier 10.1109/MSP.2010.937314

2010 IEEE Workshop on Signal Processing Systems (SiPS 2010)
6–8 October, San Francisco, California.
General Cochairs: Shuvra Battacharyya and Jorn Janneck
URL: http://www.sips2010.org/

2010 IEEE International Symposium on Phased Array Systems and Technology (ARRAY'10)
12–15 October, Waltham, Massachusetts.
Conference Chair: Mark Russell
URL: http://www.array2010.org/

2010 The 10th IEEE International Conference on Signal Processing (ICSP'10)
24–28 October, Beijing, China.
Conference Chair: Yuan Baozong
URL: http://icsp10.bjtu.edu.cn

NOVEMBER

2010 2nd International Conference on Audio, Language, and Image Processing
23–25 November 2010, Shanghai, China.
General Chairs: Fa-Long Luo, Wanggen Wan, and Thomas Sikora
URL: http://www.icalip2010.cn/

DECEMBER

28th Picture Coding Symposium (PCS'10)
7–10 December, Nagoya, Japan.
General Chair: Masayuki Tanimoto
URL: http://www.pcs2010.org/

2010 IEEE Spoken Language Technology Workshop (SLT'10)
12–15 December, Berkeley, California.
General Chairs: Dilek Hakkani-Tür and Mari Ostendorf
URL: http://www.slt2010.org/

The IEEE International Workshop on Information Forensics and Security (WIFS)
12–15 December, Seattle, Washington.
General Cochairs: Darko Kirovski and Paul Van Oorschot
URL: http://www.wifs10.org

2011

JANUARY

2011 IEEE Digital Signal Processing and Signal Processing Education Workshop (DSP/SPE'10)
4–7 January, Sedona, Arizona.
General Chairs: Lina Karam and Ronald Schafer
URL: http://www.dspe2011.org

MARCH

Data Compression Conference
29–31 March, Snowbird, Utah.
General Chair: James A. Storer
URL: http://www.cs.brandeis.edu/~dcc/

---

dsp TIPS&TRICKS

continued from page 124

We list the binary logarithm algorithm's steps as follows:

1) Initialize result to 0: $y = 0$.
2) Initialize mantissa-bit decimal value to 0.5: $b = 1/2$.
3) While $x < 1, x = 2x, y = y - 1$.
4) While $x \geq 2, x = x/2, y = y + 1$.
5) Go to Step 3 and repeat until $1 \leq x < 2$.
6) Square: $x = x \cdot x$.
7) If $x \geq 2, x = x/2, y = y + b$.
8) Scale for next bit: $b = b/2$.
9) Go to Step 6 and repeat until desired number of mantissa bits are found.
10) Final $\log(x)$ value: y.

A "C" program, and MATLAB code, demonstrating the algorithm are available for download at http://www.signalprocessingsociety.org/publications/periodicals/spm/columns-resources-archive/2010-columns-resources/#tips.

AUTHOR

Clay S. Turner (Clay.Turner@PaceOMatic.com) is chief scientist for Pace-O-Matic, Inc. He has over 30 years of experience in digital signal processing and mathematical and embedded programming.

REFERENCE

[1] D. E. Knuth, *The Art of Computer Programming: Seminumerical Algorithms*, 2nd ed. Reading, MA: Addison-Wesley, vol. 2, 1981, pp. 441–466.

SP