International Journal of Research Studies in Computer Science and Engineering (IJRSCSE)
Volume 2, Issue 2, February 2015, PP 28-31
ISSN 2349-4840 (Print) & ISSN 2349-4859 (Online)
www.arcjournals.org

Estimation of Number of Bits in Binary Representation of an Integer

WANG Xingbo

Department of Mechatronics
Foshan University, Foshan City,
Guangdong Province, PRC
wxbmail@msn.com

Abstract: *Two bounds to estimate the number of bits in binary representation of an integer of N decimal bits are proposed. One is more approximate to real value and the other is easier to compute. The former one is suitable for classic programming and the later is more practical in embedded computing, especially in development of system on chip (SoC), when used to be a range of computation or to be a condition of searching.*

Keywords: *Embedded System, System on chip, Binary Representation, Number of bits, Estimation*

1. Introduction

Real-time performance is a characteristic and a fundamental demand of embedded system. To ensure the real-time demands, algorithm and programs for embedded system must be efficient and compact, as commented in bibliography [1]. Hence, both hardware and software of embedded system are required to strictly keep a minimal cost, as bibliographies [2]-[5] investigated and proposed. Consequently, algorithm for embedded system programming must be very refinery even for a simple or a trivial problem.

In programming practice, it often requires estimating number of bits of an integer in its binary representation. To a programmer of classical computer, this problem is trivial to cost his or her spends, however it dose have to spend time to design a proper algorithm for a programmer of embedded system.

Looking into many bibliographies, I cannot find one that has an answer to the question. Therefore, this paper makes an investigation on the problem and presents the results.

2. Preliminaries

We first need the following lemmas.

Lemma 1 ([6][7]). The floor function $\lfloor x \rfloor$, which is defined by $x-1<\lfloor x \rfloor \le x$, has the following properties:

(1). for any real x and integers n: $\lfloor n+x \rfloor = n+\lfloor x \rfloor$ and $n > x \Rightarrow n > \lfloor x \rfloor$;

(2) for any real x and y: $x \le y \Rightarrow \lfloor x \rfloor \le \lfloor y \rfloor$ and $\lfloor x \rfloor > \lfloor y \rfloor \Rightarrow x > y$.

Where the symbol $a \Rightarrow b$ means that conclusion b is derived from condition a.

Lemma 2 ([7]). Total valid bits of positive integer α's binary representation is $\lfloor \log_2 \alpha \rfloor + 1$.

Lemma 3. $\frac{301}{1000} < \lg 2 < \frac{302}{1000}$.

3. Main Results

Theorem 1. A positive integer m that has n (n>1) decimal bits requires no less that $\lfloor (n-1)\log_2 10 \rfloor + 1$ and no more than $1+\lfloor n\log_2 10 \rfloor$ binary bits to represent it in binary representation.

Proof. The largest positive integer in decimal expression is $\underbrace{99...9}_{n}=10^n-1$ and the smallest is $\underbrace{10...0}_{n-1}=10^{n-1}$. Hence it yields

$$10^{n-1}\le m\le 10^n-1<10^n$$

which leads to

$$\frac{n-1}{\lg 2}\le \log_2 m<\frac{n}{\lg 2} \tag{1}$$

Namely,

$$(n-1)\log_2 10\le \log_2 m<n\log_2 10 \tag{2}$$

Therefore, it yields by Lemma 1

$$\lfloor (n-1)\log_2 10\rfloor+1\le 1+\lfloor \log_2 m\rfloor\le 1+\lfloor n\log_2 10\rfloor \tag{3}$$

By Lemma 2, this validates the theorem.

Theorem 2. A positive integer m that has n (n>1) decimal bits requires no less that $3n$-2 and no more than $4n$-1 binary bits to represent it in binary representation.

Proof. Let us begin with the inequality (1).

By Lemma 3, it yields

$$\frac{1000(n-1)}{302}<\log_2 m<\frac{1000n}{301}$$

and thus

$$1+\frac{1000(n-1)}{302}<1+\log_2 m<\frac{1000n}{301}+1 \tag{4}$$

Substituting $\frac{1000n}{301}+1=4n-1-\frac{204n}{301}$ and $1+\frac{1000(n-1)}{302}=\frac{1000(n-1)}{302}=3n-2+\frac{94(n-1)}{302}$ in (4) yields

$$3n-2+\frac{94(n-1)}{302}\le 1+\log_2 m<4n-1-\frac{204n}{301} \tag{5}$$

This immediately leads to

$$3n-2<1+\log_2 m<4n-1 \tag{6}$$

By Lemma 1 and Lemma 2, we obtain

$$3n-2\le 1+\lfloor \log_2 m\rfloor\le 4n-1 \tag{7}$$

which is what Theorem 2 claims.

4. Numerical Experiments and Conclusions

Table 1 lists integers of 1 to 10 decimal bits and their binary representations and table 2 lists datum computed by both Theorem 1 and Theorem 2 together with the binary bits from table 1.

Table 1. *Integers and their binary representations and number of bits*

Decimal bits n	Integer of n decimal bits and binary representations		Integer of n decimal bits and binary representations	
	Minimal value in decimal and binary representation	Binary bits	Maximal value in decimal and binary representation	Binary bits
1	$1=(1)_2$	*1*	$9=(1001)_2$	*4*
2	$10=(1010)_2$	*4*	$99=(1100011)_2$	*7*
3	$100=(1100100)_2$	*7*	$999=(1111100111)_2$	*10*

4	$1000=(1111101000)_2$	*10*	$9999=(10011100001111)_2$	*14*
5	$10000=(10011100010000)_2$	*14*	$99999=(11000011010011111)_2$	*17*
6	$100000=(11000011010100000)_2$	*17*	$999999=(11110100001000111111)_2$	*20*
7	$1000000=(11110100001001000000)_2$	*20*	$9999999=(100110001001011001111111)_2$	*24*
8	$10000000=(100110001001011010000000)_2$	*24*	$99999999=(101111101011110000011111111)_2$	*27*
9	$100000000=(101111101011110000100000000)_2$	*27*	$999999999=(1110111001101011001001111111111)_2$	*30*
10	$1000000000=(111011100110101100101000000000)_2$	*30*	$9999999999=(1001010100000010111110001111111111)_2$	*34*

Table 2. *Comparison of datum from Theorem 1, 2 and table 1*

n	3*n*-2	$\lfloor (n-1)\log_2 10 \rfloor +1$	Actual num.	4*n*-1	$\lfloor n\log_2 10 \rfloor +1$	Actual num.
1	1	1	1	3	4	4
2	4	4	4	7	7	7
3	7	7	7	11	10	10
4	10	10	10	15	14	14
5	13	14	14	19	17	17
6	16	17	17	23	20	20
7	19	20	20	27	24	24
8	22	24	24	31	27	27
9	25	27	27	35	30	30
10	28	30	30	39	34	34

From the 2 tables, one can see that, Theorem 1 is more accurate than Theorem 2. However, it is obvious that the number 3n-2 and 4n-1 proposed by Theorem 2 are more efficient to compute than the number $\lfloor (n-1)\log_2 10 \rfloor +1$ and $\lfloor n\log_2 10 \rfloor +1$proposed by Theorem 1 because, for example, the number $\lfloor n\log_2 10 \rfloor +1$ is calculate by two logarithm operations: $\ln 2$ and $\ln 10$, one division: $\ln 10 / \ln 2$, one multiplication: $n(\ln 10 / \ln 2)$, one $\lfloor \ \rfloor$ operation and one addition while 4*n*-1 needs only 3 simple operations. Seeing from bibliographies [8], [9], we know that computations of logarithm and floor function is never a simple process. Therefore, the estimates proposed by Theorem 1 are suitable for classical programming, and the estimates proposed by Theorem 2 are more preferred to engineering applications such as embed system or SoC.

ACKNOWLEDGEMENTS

The research work is supported by the national Ministry of science and technology under project 2013GA780052, Department of Guangdong Science and Technology under projects 2012B011300068, Foshan Bureau of Science and Technology under projects 2013AG10007, Special Innovative Projects from Guangdong Education Department, and Chancheng government under projects 2013B1018 and 2013A1021. The authors sincerely present thanks to them all.

REFERENCES

[1]. Embree Paul M., C Algorithms For Real-time DSP, Prentice Hall, Englewood Cliff, 92(1995)

[2]. Ali Dasdan, Rajesh K., Gupta, Time Driven Hardware-Software Codesign, Proc. 3rd. Conf. ASIC, 1999

[3]. Janka, Randall S., Specification and Design Methodology for Real-Time Embedded Systems, Springer, (2002)

[4]. Wayne Wolf, High-Performance Embedded Computing: Architectures, Applications, and Methodologies, Elsevier Inc, (2007)

[5]. Marc Poulhies, Jacques Pulou, Christophe Rippert, Joseph Sifakis, A Methodology and Supporting Tools for the Development of Component-Based Embedded Systems, Lecture Notes in Computer Science. 4888, 75(2008)

[6]. Graham R. L., Knuth D. E. and Patashnik O. Concrete Mathematics: A Foundation for Computer Science, Addison-Wesley, 67 (1994)

[7]. WANG Xingbo, Some Supplemental Properties with Appendix Application of Floor Function, Journal of Science of Teacher's College and University (In Chinese), 34(3), 7(2014)

[8]. Suganth P., Nikhil J., Sunil P. K., A Fast Hardware Approach for Approximate, Efficient Logarithm and Antilogarithm Computations, IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, 17(2), 269(2009)

[9]. Al-Mohy A. H., A more accurate Briggs method for the logarithm, Numerical Algorithm. (59), 393(2012)

AUTHOR'S BIOGRAPHY

WANG Xingbo was born in Hubei, China. He got his Master and Doctor's degree at National University of Defense Technology of China and had been a staff in charge of researching and developing CAD/CAM/NC technologies in the university. Since 2010, he has been a professor in Foshan University, still in charge of researching and developing CAD/CAM/NC technologies. Wang has published 8 books, over 70 papers and obtained more than 20 patents in mechanical engineering.