Fast Modular Exponentiation

Ömer Eğecioğlu
Department of Computer Science
University of California
Santa Barbara, CA 93106

Çetin K. Koç
Department of Electrical Engineering
University of Houston
Houston, TX 77204

Abstract

The well-known binary method computes $C = M^E \pmod N$ using an average number of $1.5(n-1)$ multiplications, where n is the number of bits in the binary expansion of E. When the exponent is recoded using the canonical bit recoding technique then the average number of multiplications can be reduced to $1.33(n-1)$. We show that a further reduction is achieved if the bits of the exponent are scanned at $d > 1$ bits at a time: for $n = 2^{10}$, for example, the average number of multiplications becomes $1.212(n-1)$ with $d = 5$. Furthermore, given any $\epsilon > 0$, the computation can be done using an average of $(1+\epsilon)(n-1)$ multiplications for large n by taking $d = \lceil \frac{1}{\epsilon} \rceil$.

1 Binary and Recoded Binary Methods

The *binary method* (the square and multiply method) computes

$$C = M^E \pmod N \tag{1}$$

using $n-1$ squarings and as many multiplications as the number of nonzero bits in the binary expansion of the exponent, where $n = 1 + \lfloor \log_2 E \rfloor$ [3]. It is clear that $n-1$ is a lower bound for the number of squaring operations required. However, it is possible to reduce the number of consequent multiplications using a *recoding* of the the exponent [10]. Recoding techniques (Booth recoding, bit-pair recoding, etc.) for sparse representations of binary numbers have been effectively used in multiplication algorithms [2, 9]. For example, the original Booth recoding technique [1] scans the bits of the multiplier one bit at a time, and adds or subtracts the multiplicand to or from the partial product, depending on the value of the current bit and the previous bit. The modified versions of the Booth algorithm scans the bits of the multiplier two bits at a time [6] or three bits at a time [9]. These techniques are equivalent in the sense that a signed-digit representation which is based on the identity $2^k - 1 = 2^{k-1} + \cdots + 2^1 + 2^0$ is used to collapse blocks of 1's appearing in a binary representation. Thus in a signed-digit number with radix 2, three symbols $\{\bar{a}, 0, a\}$ are allowed for the digit set, in which a represents 1 and $\bar{a}$ represents -1.

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

A minimal signed-digit vector $D = (D_{n-1}D_{n-2}\cdots D_1D_0)_2$ that contains no adjacent nonzero digits (i.e. $D_iD_{i-1} = 0;\ 0 < i < n$) is called a *canonical signed-digit vector* [7, 2]. If the binary expansion of E is viewed as padded with an initial zero, then it can be proved that there exists a unique canonical signed-digit vector for E. An algorithm for this recoding is described in [7, 2].

The bit recoding techniques applied to E can be used for the modular exponentiation problem provided that $M^{-1} \pmod N$ is supplied along with M [10]. Throughout this paper, we will ignore the preprocessing time required for the computation of $M^{-1} \pmod N$ by the well known extended Euclidean algorithm [3, 5].

2 The m-ary Method

The binary method can be generalized to the *m-ary method* which scans the digits of E expressed in radix m [3]. We restrict our attention to the case when $m = 2^d$. Let $E = (E_{n-1}E_{n-2}\cdots E_1E_0)_2$ be the binary expansion of the exponent. This representation of E is partitioned into k blocks of length d each, for $kd = n$ (if d does not divide n, the exponent is padded with at most $d-1$ zeros). Now, define

$$F^{(i)} = (E_{id+d-1}E_{id+d-2}\cdots E_{id})_2 = \sum_{r=0}^{d-1} E_{id+r}2^r \quad . \tag{2}$$

Note that $0 \le F^{(i)} \le 2^d - 1$ and $E = \sum_{i=0}^{k-1} F^{(i)}2^{id}$. In the m-ary method, first the values of $M^j \pmod N$ for $j = 2, 3, \ldots, 2^d - 1$ are computed. Then the bits of E are scanned d bits at a time from the most significant to the least significant. At each step the partial result is raised to the 2^d power and multiplied with $M^{F^{(i)}}$ where $F^{(i)}$ is the value of the current bit section.

---

m-ary method

Input: M, N, E, n, d where $n = 1 + \lfloor \log_2 E \rfloor$ and $n = kd$ for $k \ge 1$.

Output: $C = M^E \pmod N$.

Step 1. Compute $M^j \pmod N$ for $j = 2, 3, \ldots, 2^d - 1$.

Step 2. Set $C = M^{F^{(k-1)}} \pmod N$ and for $i = k-2, k-3, \ldots, 1, 0$ compute

$$\begin{aligned} C &= C^{2^d} \pmod N \ , \\ C &= C * M^{F^{(i)}} \pmod N \quad \text{if} \quad F^{(i)} \neq 0 \ . \end{aligned}$$

Theorem 1 *The m-ary method requires*

$$T(n,d) = n + \left(\frac{n}{d} - 1\right)\left(1 - \frac{1}{2^d}\right) + 2^d - d - 2 \tag{3}$$

multiplications on the average.

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

Proof Step 1 of the m-ary method requires $2^d - 2$ multiplications regardless of the value of the exponent. The number of squaring operations in Step 2 is equal to $(k-1)d$. Multiplications in Step 2 are performed for nonzero values of $F^{(i)}$. Since $m-1$ out of m values of $F^{(i)}$ are nonzero, the average number of multiplications required is $(k-1)\left(\frac{m-1}{m}\right)$. Substituting $m = 2^d$ and $n = kd$, we obtain the stated result. □

The average number of multiplications for the binary method can be found simply by substituting $d = 1$ in (3). Thus

$$T(n,1) = n + (n-1)\left(1 - \frac{1}{2}\right) + 2 - 1 - 2 \quad = \quad \frac{3}{2}(n-1) \ .$$

The optimal value d^* of d which minimizes the average number of multiplications required by the m-ary method can be shown to be $d^* = O(\log n)$. Exact values of d^* can be obtained by enumeration [4].

3 The Recoded m-ary Method

Next, we consider the recoded version of the m-ary method, in which the partitioning that determines $F^{(i)}$ in (2) is applied to the canonical signed-digit vector of E instead of the its binary expansion.

Recoded m-ary method

Input: M, M^{-1}, N, E, n, d where $n = 1 + \lfloor \log_2 E \rfloor$ and $n = kd$ for $k \geq 1$.

Output: $C = M^E \pmod N$.

Step 1. Compute the canonical signed-digit recoding of the exponent E using Reitwiesner's algorithm [7, 2]. The resulting exponent D is the minimal signed-digit vector.

Step 2. Compute $M^{F^{(i)}} \pmod N$ for all possible $F^{(i)}$. Note that here $F^{(i)}$ is a bit-section of the minimally recoded exponent D. The length of $F^{(i)}$ is equal to d.

Step 3. Set $C = M^{F^{(k-1)}} \pmod N$ and for $i = k-2, k-3, \ldots, 1, 0$ compute

$$\begin{aligned} C &= C^{2^d} \pmod N \ , \\ C &= C * M^{F^{(i)}} \pmod N \quad \text{if} \quad F^{(i)} \neq 0 \ . \end{aligned}$$

Our point of departure for the analysis of the recoded m-ary method is the study of the collection of all canonical signed-digit vectors. Equivalently, we denote by $\mathcal{L}$ the formal language of all words w over the alphabet $\{\bar{a}, 0, a\}$ in which none of the patterns

$$aa, \ a\bar{a}, \ \bar{a}a, \ \bar{a}\bar{a}$$

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

appears. Thus the words w of length d in $\mathcal{L}$ correspond to possible bit-sections $F^{(i)}$ of the recoded binary expansion of E. For $w \in \mathcal{L}$, let $|w|$ and $|w|_0$ denote the length of w and the number of occurrences of the letter 0 in w, respectively. Suppose τ_n is the total number of words of length n in $\mathcal{L}$, and ζ_n is the total number of occurrences of the letter 0 over all words of length n in $\mathcal{L}$. In other words

$$\zeta_n = \sum_{\substack{w \in \mathcal{L} \\ |w| = n}} |w|_0 \quad .$$

Theorem 2 *We have*

$$\tau_n = \frac{1}{3}\left[2^{n+2} + (-1)^{n+1}\right] \quad \textit{and} \quad \zeta_n = \frac{(24n+56)2^n + (8-3n)(-1)^n}{27} \quad .$$

In particular,

$$\lim_{n\to\infty} \frac{\zeta_n}{n\tau_n} = \frac{2}{3} \quad . \tag{4}$$

Proof By considering the words in $\mathcal{L}$ according to their first letter, it is easy to see that $\mathcal{L}$ satisfies the relation

$$\mathcal{L} = 1 + a + \bar{a} + a0\mathcal{L} + \bar{a}0\mathcal{L} + 0\mathcal{L} \quad , \tag{5}$$

where 1 denotes the empty word and + denotes disjoint union. Consider the generating function

$$f_{\mathcal{L}}(t,x) = \sum_{w\in\mathcal{L}} t^{|w|}x^{|w|_0} \quad .$$

It follows from (5) that $f_{\mathcal{L}}$ satisfies

$$f_{\mathcal{L}}(t,x) = 1 + 2t + 2t^2 x f_{\mathcal{L}}(t,x) + tx f_{\mathcal{L}}(t,x) \quad ,$$

and therefore

$$f_{\mathcal{L}}(t,x) = \frac{1+2t}{1-tx-2t^2x} \quad . \tag{6}$$

We have

$$\sum_{n=0}^{\infty} \tau_n t^n = f_{\mathcal{L}}(t,1) = \frac{1+2t}{1-t-2t^2} \quad . \tag{7}$$

By partial fractions expansion of the right hand side of (7), we obtain

$$\tau_n = \frac{1}{3}\left[2^{n+2} + (-1)^{n+1}\right] \quad . \tag{8}$$

The generating function of the sequence ζ_n can easily be found from $f_{\mathcal{L}}(t,x)$ as

$$\sum_{n=0}^{\infty} \zeta_n t^n = \frac{\partial}{\partial x} f_{\mathcal{L}}(t,1) \quad ,$$

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

where the substitution $x = 1$ is carried out after the differentiation with respect to x. From the expression (6) we obtain the formula

$$\frac{\partial}{\partial x} f_{\mathcal{L}}(t,1) = \frac{t(1+2t)^2}{(1-t-2t^2)^2} = \frac{8}{9}\frac{1}{(1-2t)^2} - \frac{32}{27}\frac{1}{1-2t} + \frac{11}{27}\frac{1}{1+t} - \frac{1}{9}\frac{1}{(1+t)^2} \quad .$$

Therefore,

$$\zeta_n = \frac{(24n+56)2^n + (8-3n)(-1)^n}{27} \tag{9}$$

as claimed. The limiting ratio (4) now easily follows from the expressions (8) and (9). □

The average number of multiplications required by the recoded binary method after $n-1$ squaring operations is equal to the average number of nonzero digits of the recoded form of E. By Theorem 2, this number is

$$1 - \frac{\zeta_n}{n\tau_n} \;\to\; \frac{1}{3} \quad . \tag{10}$$

In particular, (10) yields the average number of multiplications required by the recoded binary method which was stated in [10]:

Corollary 1 *For n large, the average number of multiplications required by the recoded binary method is $\frac{4}{3}(n-1)$.*

Now we compute the number of multiplications necessary in the preprocessing stage when the powers of M corresponding to all the recoded bit-sections of length d are evaluated.

Theorem 3 *The number of multiplications required to compute M^w for all length d recoded bit-sections w is*

$$\tau_d - 3 = \frac{1}{3}\left[2^{d+2} + (-1)^{d+1}\right] - 3 \quad .$$

Proof First we compute M^w where w contains only one nonzero letter. Since $1, M$, and M^{-1} are already available, this step requires $2(d-1)$ multiplications. After this, each value M^w where $|w|_a + |w|_{\bar{a}} = k$ can be computed recursively from the already computed values M^w, $|w|_a + |w|_{\bar{a}} < k$ by a single multiplication. It follows that the total number of multiplications required is

$$\tau_d - (1+2d) + 2(d-1) = \tau_d - 3$$

as claimed. □

In the following theorem we give the average number of multiplications $T'(n,d)$ required by the recoded m-ary method.

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

Theorem 4 *Recoded m-ary method requires*

$$\begin{aligned} T'(n,d) &= n-1+\tau_d-3+\left(\frac{n}{d}-1\right)\left[1-\left(\frac{2}{3}\right)^d\right] \\ &= \left[1+\frac{1}{d}-\frac{1}{d}(\frac{2}{3})^d\right]n \;+\; \frac{2^{d+2}}{3}+\frac{(-1)^{d-1}}{3}+(\frac{2}{3})^d-5 \end{aligned}$$

multiplications on the average.

Proof The number of squaring operations in Step 3 of the recoded m-ary method is equal to $n-1$. The preprocessing time required to compute all necessary powers M^w for all bit sections w with $|w|=d$ is τ_d-3 by Theorem 3. Finally, by Theorem 2 a recoded bit section w of length d is equal to zero with probability $(\frac{2}{3})^d$. Since we require a multiplication for each bit-section after the most significant one (for which the corresponding power of M is already available as the initial value), we need to perform

$$\left(\frac{n}{d}-1\right)\left[1-\left(\frac{2}{3}\right)^d\right]$$

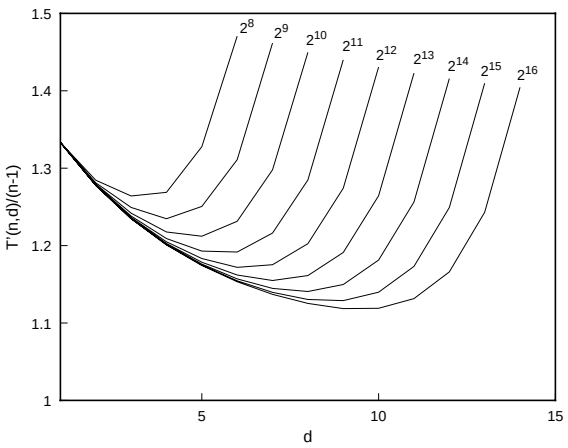
additional multiplications. □

In Figure 1, the values of $\frac{T'(n,d)}{n-1}$ are shown for $n=2^8,2^9,\ldots 2^{16}$ and for d ranging from 1 to 14. For a given value of n there exists an optimal value of d^* of d which minimizes the average number of multiplications $T'(n,d)$. It can be seen that the recoded m-ary method allows for the computation of (1) with fewer than $1.33(n-1)$ multiplications. For example, when $n=2^{10}$ with the optimal choice of $d^*=5$, the average number of multiplications is found to be $1.212(n-1)$. The optimal values d^* and $T'(n,d^*)$ for several values of n are tabulated in Table 1.

Table 1.

n	d^*	$T'(n,d^*)$
2^8	3	$1.264(n-1)$
2^9	5	$1.235(n-1)$
2^{10}	5	$1.212(n-1)$
2^{11}	6	$1.192(n-1)$
2^{12}	6	$1.172(n-1)$
2^{13}	7	$1.155(n-1)$
2^{14}	8	$1.141(n-1)$
2^{15}	8	$1.130(n-1)$
2^{16}	9	$1.118(n-1)$

Figure 1.



Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

Next we consider the magnitude of the optimal value of d which minimizes $T'(n,d)$ for large n. More precisely, given a fixed $\epsilon > 0$, we try to find $d = d(\epsilon)$ such that

$$T'(n,d) \leq (1+\epsilon)(n-1) \quad .$$

This means that for large n, we should have

$$\frac{1}{d} - \frac{1}{d}\left(\frac{2}{3}\right)^d \leq \epsilon \tag{11}$$

by Theorem 4. To satify (11) it suffices to take $d = \lceil \frac{1}{\epsilon} \rceil$. Thus we have proved

Theorem 5 *Given any $\epsilon > 0$, the recoded m-ary method with $d = \lceil \frac{1}{\epsilon} \rceil$ requires only*

$$(1+\epsilon)(n-1)$$

average number of multiplications to compute $C = M^E \pmod N$ for large n.

Note that the values of d^* given in Table 1 are in agreement with Theorem 5: for $\epsilon = 0.155$ for example, $d^* = 7$ and $\frac{1}{\epsilon} = 6.451$; and for $\epsilon = 0.118$, $d^* = 9$ and $\frac{1}{\epsilon} = 8.474$.

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.

References

[1] A. D. Booth, "A Signed Binary Multiplication Technique," *Q. J. Mech. Appl. Math.*, Vol. 4, No. 2, pp. 236–240, 1951. (Also reprinted in [8], pp. 100-104).

[2] K. Hwang, *Computer Arithmetic, Principles, Architecture, and Design*, John Wiley & Sons, Inc., 1979.

[3] D. E. Knuth, *The Art of Computer Programming*, Volume 2, *Seminumerical Algorithms*, 2nd Edition, Addison-Wesley Publishing Company, 1981.

[4] Ç. K. Koç "High Radix and Bit Recoding Techniques for Modular Exponentiation," Unpublished Manuscript, November 1989.

[5] J. D. Lipson, *Elements of Algebra and Algebraic Computing*, Addison-Wesley Publishing Company, 1981.

[6] O. L. MacSorley, "High-Speed Arithmetic in Binary Computers," *Proceedings of the IRE*, Vol. 49, pp. 67–91, January 1961. (Also reprinted in [8], pp. 14-38).

[7] G. W. Reitwiesner, "Binary Arithmetic," *Advances in Computers*, Vol. 1, pp. 231–308, 1960.

[8] E. E. Swartzlander (Editor), *Computer Arithmetic*, Benchmark Papers in Electrical Engineering and Computer Science, Vol. 21, Dowden, Hutchinson, & Ross, Inc., 1980.

[9] S. Waser and M. J. Flynn, *Introduction to Arithmetic for Digital System Designers*, CBS College Publishing, 1982.

[10] C. N. Zhang, H. L. Martin, and D. Y. Y. Yun, "Parallel Algorithms and Systolic Arrays Designs for RSA Cryptosystem," *Proceedings of the International Conference on Systolic Arrays*, pp. 341–350, K. Bromley, S. Y. Kung, and E. Swartzlander (Eds.), Computer Society Press, San Diego, California, May 25–27, 1988.

Communication, Control, and Signal Processing: Proceedings of 1990 Bilkent International Conference on New Trends in Communication, Control, and Signal Processing, E. Arikan, editor, Vol. 1, pages 188-194, Ankara, Turkey, July 2-5, 1990, Amsterdam, Netherlands: Elsevier.