

Low-Weight Left-to-Right Binary Signed-Digit Representation of N Integers

Xiaoyu Ruan and Rajendra S. Katti
Department of Electrical and Computer Engineering
North Dakota State University
Fargo, North Dakota 58105, USA
e-mail: {xiaoyu.ruan, rajendra.katti}@ndsu.nodak.edu

Abstract — An algorithm for computing the binary signed-digit representation of N integers is introduced. The algorithm operates from left to right (i.e., from the most significant bit to the least significant bit) and the resulting representation has minimum joint weight. Such an algorithm is useful in elliptic curve cryptosystems (ECC) [1].

I. INTRODUCTION

The binary signed-digit (BSD) representation uses ± 1 and 0 to recode integers. i.e., the BSD representation of integer n is denoted as $n = (\dots u_3 u_2 u_1 u_0)$. Here, $n = \dots + u_3 2^3 + u_2 2^2 + u_1 2^1 + u_0$, where $u_i \in \{1, -1, 0\}$ for all i . The BSD representation for a given integer is not unique. We define the *weight* to be the number of nonzero bits in a BSD representation. If BSD representations of N integers are written one below another to form a BSD table, the number of nonzero columns in the table is defined as the *joint weight*. A BSD representation with low joint weight can speed up the scalar point multiplication operation in elliptic curve cryptosystems (ECC) [1]. Computing the BSD representation from left to right can significantly reduce the amount of memory required.

II. COMPUTING THE BSD REPRESENTATION

Solinas and Proos introduce one kind of BSD representation with minimum joint weight, named the Joint Sparse Form (JSF) [2,3]. However the algorithm for computing the JSF is right-to-left, which suggests large amount of memory is required.

Our left-to-right method for computing another kind of BSD representation for N integers is given in Algorithm 1. The basic principle is that in a BSD representation, the consecutive bits of the form $x0\dots 0\bar{x}$, where $x \in \{1, -1\}$, can be replaced by $0x\dots x$.

Algorithm 1.

Input: L -bit binary expansions of N ($N > 0$) integers k_i ($0 \leq i \leq N-1$). Each k_i is denoted as $(k_{i,L-1}, k_{i,L-2}, \dots, k_{i,1}, k_{i,0})$.

Output: The BSD representation of the N given integers with minimum joint weight.

Step 1: Convert the binary expansion of each k_i ($0 \leq i \leq N-1$) into an $(L+1)$ -bits BSD representation using the following rule:

$$k_i = ((k_{i,L-1} - 0), (k_{i,L-2} - k_{i,L-1}), \dots, (k_{i,0} - k_{i,1}), (0 - k_{i,0})).$$

Step 2: Scan all the $L+1$ columns in the array from the leftmost column (L) to the rightmost column (0), one column at a time. The column being scanned is called the “scanning column”.

Step 3a: If all the N entries in the scanning column are zero, then skip the scanning column and scan the column to its right.

Step 3b: If at least one entry in the scanning column is nonzero, then perform Step 4.

Step 4: Mark the rows, which have a nonzero bit in the scanning column. The nonzero bit is called a “reducible bit”.

Step 5: Scan the marked rows from the nonzero bit and go rightwards. Look at at most N bits (excluding the reducible bit).

Step 6a: If for at least one marked row, the next rightward nonzero bit is not within the next N bits, i.e., the next N bits to the right of the reducible bit are all zero, then skip the scanning column and scan the column to its right.

Step 6b: If for all marked rows, the next rightward nonzero bit is within the next N bits, then among all marked rows let the maximum distance between the reducible bit and the next rightward nonzero bit be $C-1$, i.e., there are $C-1$ zeros between the two nonzero bits, and then perform Step 7 ($C \geq 1$).

Step 7: Scan the columns from the column with the farthest rightward nonzero bit found in Step 6b to the column with reducible bits. Note that this is a right-to-left sweep of $C+1$ bits.

Step 8: See if there exists at least one nonzero entry in each of the $C+1$ columns being scanned in Step 7. Note that, except for the leftmost column within the $N \times (C+1)$ table, for every nonzero column, at least one of the nonzero values must be the rightmost in that row.

Step 9a: If at least one column of the $C+1$ columns does not satisfy the condition of Step 8, then skip the scanning column and scan the column to its right.

Step 9b: If all the $C+1$ columns are nonzero and satisfy the condition of Step 8, then perform Step 10.

Step 10: Suppose the reducible bit in one marked row is x ($x \in \{1, -1\}$). First replace x by 0. Then replace the bits to its right by x . This second replacement is performed until the next nonzero bit, \bar{x} . i.e., replace $x0\dots 0\bar{x}$ by $0x\dots x$. This replacement is performed in all rows with reducible bits.

Step 11: Skip C columns and continue to scan rightwards. Note that the C columns have already been replaced. \square

The optimality of Algorithm 1 is stated in Theorem 1.

Theorem 1. The output of Algorithm 1 has minimum joint weight among any BSD representation of the N given integers.

III. COMPARISON AND CONCLUSION

Simulation results show that Algorithm 1 is more efficient than the JSF method when implemented in software. Another major advantage of Algorithm 1 is that it scans the integers from left to right, which is compatible with Shamir’s method [4] for scalar multiplication in ECC, thus requiring far less memory than a right-to-left method does.

REFERENCES

- [1] J. Lopez and R. Dahab, “An Overview of Elliptic Curve Cryptography,” Technical report, Institute of Computing, State University of Campinas, Brazil, May 2000.
- [2] J. A. Solinas, “Low-Weight Binary Representations for Pairs of Integers,” Technical Report CORR 2001-41, Center for Applied Cryptographic Research, University of Waterloo, Canada, 2001.
- [3] J. Proos, “Joint Sparse Forms and Generating Zero Columns when Combing,” Technical Report CORR 2003-23, Center for Applied Cryptographic Research, University of Waterloo, Canada, 2003.
- [4] T. ElGamal, “A Public-Key Cryptosystem and Signature Scheme Based on Discrete Logarithms,” The IEEE Transactions on Information Theory, Vol. 31, pp 469-472, 1985.