MGSDNAF - A Modified Signed Digit Generalized Non-Adjacent Form for Integers Representation

Arash Eghdamian, Azman Samsudin
School of Computer Sciences, Universiti Sains Malaysia, 11800, Penang, Malaysia.
ae11_com109@student.usm.my

***Abstract*—This paper describes a new radix r integer representation, which is an improved version over of an existing integer representation, namely "Modified Generalized Non-Adjacent Form" (MGNAF). Similar to MGNAF, MGSDNAF reduces the integer's Hamming Weight better than the GNAF, a well-known integer representation method. With a reduced Hamming Weight, exponentiations can be quickly calculated. Moreover, contrary to MGNAF, the digit set size in the proposed method is smaller; and therefore improved the memory usage in point multiplication -an operation that is highly used in the calculation of pairing-based cryptosystems.**

***Index Terms*—Cryptography; Generalized NAF; Hamming Weight; Radix-r Representation.**

I. Introduction

Modular exponentiation is among the most time-consuming operations in most cryptosystems. For this reason, efficient modular exponentiation algorithm is very important for the performance of the corresponding cryptosystems. The fundamental requirement in enhancing the modular exponentiation is to have an efficient multiplication technique. To have faster scalar multiplication, an efficient class of radix-r representation should be used. An example of efficient class representation is a representation that has lower number of non-zero digits (lower Hamming Weight).

Reitwiesner showed that his canonical recoding (NAF) is unique [1]. However, in general, the signed radix-r representation need not to be unique. Ebeid and Hasan [2], show that representations of integer M can be obtained by replacing 01 with $1\bar{1}$, $0\bar{1}$ with $\bar{1}1$ and vice versa [1]. Subsequently, they proposed an algorithm that is able to create all possible Binary Signed Digit representations of an integer M in 2007. As an illustration, Figure 1 demonstrates all binary signed representations of 13 with the length of 5 bits.

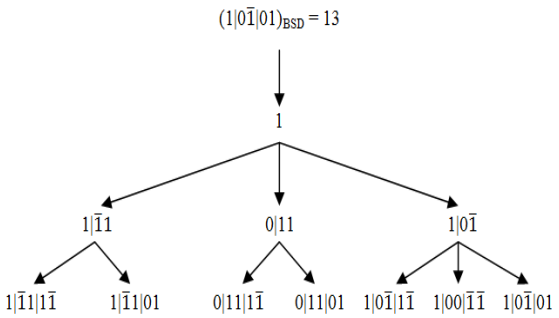

Figure 1: All 5-bit signed digit representations of 13 [3]

Based on Ebied and Hasan work, Clark and Liang [4] introduced the Generalized Non-Adjacent Form (GNAF) - a minimal representation for any signed-radix r integer. By analyzing GNAF, Amo and Wheeler [5] proved that the average percentage of non-zero digits in that representation for an integer in radix r is equal to (r - 1)/(r + 1). This should be in comparison with the average percentage of non-zero digits in the standard radix-r representation, which is (r - 1)/r. In 2014 Eghdamian and Samsudin proposed a representation, Modified Generalized Non-Adjacent Form (MGNAF) [6], which was an improvement over the GNAF representation.

II. MGNAF

In this section some of the main properties of Modified Generalized Non-Adjacent Form (AKA. MGNAF) [6] are explained.

An integer M is presented by using the radix-r representation as follows:

$$M = \sum_{j=0}^{n-1} M_j r^j, \quad M_j \in \{0,1,\dots,r-1\} \tag{1}$$

In this paper the radix-*r* representation of an integer *M* is written as $M = (M_{n-1}, \dots, M_1, M_0)$; and in this representation, the *j*-th digit of *M* is shown as M_j, while *n* shows the digit length of *M*. The number of non-zero digits of this radix-*r* representation is known as the Hamming Weight of *M*. The average Hamming Weight of an integer representation in radix *r* is $(r-1)/r$.

The main difference between MGNAF and GNAF is the consecutive of same-nonzero elements, which might appear in big integers. GNAF is only able to decrease the non-zero density of consecutive (*r-1*)s of a big number, while MGNAF is able to reduce the Hamming Weight of any sequence of same-nonzero elements. In GNAF recoding, when a sequence of *n* (*r-1*)s appears in a number of radix *r* number:

$$\underbrace{r-1, r-1, \dots, r-1}_{n\ times} \tag{2}$$

GNAF will change it to:

$$1, \underbrace{0,0,\dots,0}_{n-1\ times}, \bar{1} \tag{3}$$

In the example given above, the bit '*1*' is added to the next digit to the left of the sequence (note that, '$\bar{1}$' represents '*-1*' in this number). In GNAF, other sequences

with any other same-nonzero element would remain untouched. For example, a sequence of six *3*s in a radix *4* representation, is changed to $(100000\bar{1})_4$ but a sequence of six *2*s in a radix *4* representation remains untouched. The reason behind such transformation is that, if '*1*' is added to $(333333)_4$ the result would become $(1000000)_4$. Subsequently, the value carried by the added '*1*' should be reduced so that the new representation will still hold the same value as its original representation. If such reduction is carried out, then the new representation will be recorded as $(100000\bar{1})_4$ which has the same value as $(333333)_4$ but with different formation and less non-zero density.

On the other hand, MGNAF has the capability to change the formation of any sequence that has same-nonzero elements. For example:

$$\underbrace{x, x, \dots, x}_{n} \tag{4}$$

will be changed to:

$$(\frac{x}{r-1}, \underbrace{0,0,\dots,0}_{n-1\ times}, \frac{\bar{x}}{r-1}) \tag{5}$$

Here the $\frac{x}{r-1}$ which is at the most significant bit of the new formation is added to the next digit, left to the sequence. The explanation for this recoding is almost similar to GNAF but with some improvement, which is clarified with the following example. MGNAF recoding for *6* x-es in radix *r*, is as shown below ($x \neq 0$ and $r \neq 1$):

$$\begin{aligned}(x,x,x,x,x,x)_r &= \tfrac{x}{r-1}(r\text{-}1,r\text{-}1,r\text{-}1,r\text{-}1,r\text{-}1,r\text{-}1)_r \\ &= \tfrac{x}{r-1}(1,0,0,0,0,0,\bar{1})_r \\ &= (\tfrac{x}{r-1},0,0,0,0,0,\tfrac{x}{r-1})_r\end{aligned} \tag{6}$$

Numerical details of the comparison between GNAF and MGNAF is provided in Table 1. The result shows that the proposed MGNAF performs better in scalar multiplication with the cost of additional digits in its digit set.

Table 1
Comparison of GNAF and MGNAF [6]

Radix	Normal (%)	GNAF HW[1] (%)	MGNAF HW[1] (%)	Improvement[2] (%)
2	50	33.33	33.33	0
3	66.67	50	44.87	30.78
4	75	60	53.25	45
5	80	66.67	59.55	53.37
6	83.33	71.43	64.7	56.52
7	85.71	75	68.5	60.67
8	87.5	77.78	71.7	62.51
9	88.89	80	74.15	65.81

[1] Hamming Weight
[2] MGNAF improvement compare to GNAF

III. PROPOSED WORK

The main weakness of MGNAF, is the size of its digit set. MGNAF can reduce the Hamming Weight of radix-r integer more than GNAF can, but its weakness might affect efficiency. Therefore, reducing the size of MGNAF's digit set can help to improve its efficiency dramatically. The work reported in this paper tries to address this drawback of MGNAF. The new proposed algorithm has similar Hamming Weight reduction as MGNAF does, but the new algorithm, MGSDNAF, cuts the digit set size of MGNAF to almost half.

Algorithm: Given integer $M = \sum_{i=0}^{n} a_i r^i$, $|a_i| < r$, i = 0,1,...,n , this algorithm computes an integer representation with Hamming Weight similar to MGNAF for M with three main steps.

Step 1: Set $a_{n+1} = 0$. Set i = 0.
Step 2: Do step 3 while i <= n; the algorithm terminates with $M = \sum_{i=0}^{n+1} a_i r^i$ as the new presentation for M.
Step 3: If $a_i \neq 0$, then based on a_i and a_{i+1} values, one of the following cases would be considered:

Case:

1. $a_i = a_{i+1}$
   set $a_i = \frac{-a_i}{r\text{-}1}$
   set Digit $= a_{i+1}$
   while a_{i+1} = Digit
   set $a_{i+1} = 0$
   set $i = i + 1$
   end while
   set $a_{i+1} = a_{i+1} + \frac{a_i}{r\text{-}1}$,
   if ($a_{i+1} > \frac{r}{2}$ and Digit $\neq r$-1)
   set $a_{i+1} = a_{i+1}\text{-}r$
   set $a_{i+2} = a_{i+2} + 1$
   end if

2. $a_i > 0$ and $a_{i+1} \geq 0$
   if ($a_i + a_{i+1} \geq r$ and $a_{i+1} \neq a_{i+2}$)
   set $a_i = a_i\text{-}r$
   if ($a_{i+1} < r$-1)
   set $a_{i+1} = a_{i+1} + 1$
   end if
   if ($a_{i+1} = r$-1)
   set $a_{i+1} = 0$
   set $a_{i+2} = a_{i+2} + 1$
   end if
   end if

3. $a_i < 0$ and $a_{i+1} \leq 0$
   if ($a_i + a_{i+1} \leq -r$)
   set $a_i = +r$
   if ($a_{i+1} > -(r\text{-}1)$)
   set $a_{i+1} = a_{i+1}\text{-}1$
   end if
   if ($a_{i+1} = -(r\text{-}1)$)
   set $a_{i+1} = 0$
   set $a_{i+2} = a_{i+2}\text{-}1$
   end if
   end if

4. $a_i > 0$ and $a_{i+1} < 0$
   if ($a_i \geq -a_{i+1}$)
   set $a_i = a_i\text{-}r$
   set $a_{i+1} = a_{i+1} + 1$
   end if

5. $a_i < 0$ and $a_{i+1} > 0$
   if ($-a_i \geq a_{i+1}$)
   set $a_i = a_i + r$
   set $a_{i+1} = a_{i+1}\text{-}1$
   end if

IV. ANALYSIS

The proposed algorithm in this paper is designed to reduce the size of MGNAF's digit set while keeping it as efficient as the original MGNAF. The digit set size for MGNAF is (r-2)(r-1). During MGNAF recoding process of an integer in radix r, a sequence of one non-zero digit, 'x', with length of n, will change to:

$$(\frac{x}{r-1}, \underbrace{0,0,\dots,0}_{n-1\ times}, \frac{\bar{x}}{r-1}) \quad (7)$$

The '$(\frac{x}{r-1})$', would be added to the first digit, next to this sequence. Here is when the new digits are created.

For radix r:

$$x \quad \{0, 1, ..., r\text{-}1\} \quad (8)$$

In general, '*x*' can be written in r different ways. If the added digit is '$(\frac{x}{r-1})$', then '*x*' has (*r-2*) possibilities (note that, '*x*' cannot take the value of zero or '*r-1*' for the obvious reasons). Moreover, the digit next to '$(\frac{x}{r-1})$' has *r-1* different possibilities, since '$(\frac{x}{r-1})$' cannot be neighboring to digit '*x*'.

The number of digits in MGNAF representation and also in its left to right version [7], ignoring the sign, would be (*r-2*)(*r-1*). This is relatively a big digit-set. To reduce the digit-set size, step *3* in the proposed algorithm, MGSDNAF, will first checks if the number is greater than half of the radix. If so, the number is deducted by *r* (i.e. radix) and the next digit will be increased by *1*. With this approach, the size of the digit-set can be cut to half.

In terms of Hamming Weight, the probability that the neighboring digit is zero is *1/r*. If the neighboring digit is zero, the Hamming Weight will increase by 1. This is because a zero digit will change to digit '*1*'. On the other hand, if the neighboring digit is *r-1*, then adding '*1*' to the number will change it to zero and the Hamming Weight is reduced by *1*. However another digit '*1*' will be added to the next digit as carry, and again that digit might be '*0*' or '*r-1*'. In summary, this step increases the Hamming Weight slightly, but another improvement which was applied to the second part of the *3rd* step of MGSDNAF, reduced the impact of this increase. The improvement that was mentioned will give the priority to record the sequence of non-zero digits. Therefore, the algorithm checks the next two digits; if they are the same, then to prevent passing any carry to that sequence of the same-nonzero digits, the current digit will not be recoded. To implement such checking, the condition "If $a_{i+1} \neq a_{i+2}$" is added to the algorithm.

As it is shown in Table 2, the experimental results show that the Hamming Weight of the proposed method is slightly higher than MGNAF for radix *3*. This difference decreases as the radix goes higher till the Hamming Weight of MGSDNAF is lower than MGNAF's in radix 6 and above.

Table 2
Hamming Weight Comparison of MGNAF and the Proposed Method

Radix	MGNAF (%)	Proposed Method (%)
2	33.33	33.33
3	44.87	45.09
4	53.25	53.38
5	59.55	59.63
6	64.7	64.61
7	68.5	68.39
8	71.7	71.47
9	74.15	73.78

V. CONCLUSION

Reducing the Hamming Weight of exponent *m* can improve the calculation performance of g^m. To achieve this, exponent m can be recorded by using a signed-digit representation. Recently "Modified Generalized Non-Adjacent Form" (MGNAF) was proposed as a new recoding method. In this paper, a new signed-digit representation, MGSDNAF, is proposed as an improved version of MGNAF. Although the proposed improvement reduces the size of MGNAF's digit set to half, the performance of the proposed method is as good as the original MGNAF. Using this modified method instead of MGNAF enhances the memory usage of computing modular exponentiations when the points should be pre-computed and stored in memory. Moreover, experimental results show that the Hamming Weight of integers recoded in MGSDNAF are lower than MGNAF for radixes higher than 5 and this difference increases as the radix of the numbers increases.

ACKNOWLEDGEMENT

This work has been supported by Fundamental Research Grant Scheme (FRGS-203/PKOMP/6711427) funded by the Ministry of Higher Education of Malaysia (MOHE).

REFERENCES

[1] Reitwiesner, G. W. 1960. Binary Arithmetic. *Advances in computers,* 1, 231-308.

[2] Ebeid, N., Hasan, and M. A. 2007. On Binary Signed Digit Representations Of Integers. *Designs, Codes and Cryptography*, 42(1), 43-65.

[3] Wu, T., Zhang, M., Du, H. Q., and Wang, R. B. 2010. On Optimal Binary Signed Digit Representations Of Integers. *Applied Mathematics-A Journal of Chinese Universities*, 25(3), 331-340.

[4] Clark, W. E., and Liang, J. J. 1973. On Arithmetic Weight For A General Radix Representation Of Integers. *Information Theory, IEEE Transactions on*, 19(6), 823-826.

[5] Arno, S., and Wheeler, F. S. 1993. Signed Digit Representations Of Minimal Hamming Weight. *Computers, IEEE Transactions on,* 42(8), 1007-1010.

[6] Eghdamian, A., and Samsudin, A. 2014. An Improved Signed Digit Representation of Integers. *3rd International Conference on Computer Engineering & Mathematical Sciences*, 287-290.

[7] Eghdamian, A., and Samsudin, A. 2015. A Modified Left-to-Right Radix-r Representation. *Second International Symposium on Technology Management and Emerging Technologies (ISTMET), IEEE,* 278-281.