Some Explicit Formulae of NAF and its Left-to-Right Analogue Based on Booth Encoding *

Dong-Guk Han[1], Okyeon Yi[1]t, and Tsuyoshi Takagi[2]

fchrista,oyyil@kookmin.ac.kr, takagi@imi.kyushu-u.ac.jp

Kookmin University, Kyushu University

Abstract. Non-Adjacent Form (NAF) is a canonical form of signed binary representation of integers. Joye-Yen proposed a left-to-right analogue of NAF (FAN). It is known that NAF and FAN can be generated by applying a sliding window method with width-2 to the Booth encoding in right-to-left and left-to-right direction, respectively. In this article, we derive some properties of Booth encoding such as pattern, classification, extension, adjacency, and length.

Keywords: *signed binary representation, non-adjacent form, Booth encoding.* 1

Introduction

In some exponentiation-based public-key cryptosystems including RSA and Elliptic Curve Cryptosystems (ECC), a binary representation of a given integer (which may be a secret in most cases) is commonly used as a standard technique. While a non-signed representation of an integer is unique, we have some ways for representing the integer in signed form. For example, an integer 13 can be represented in signed form such as 10101, 10011, or 10111, where 1 denotes -1. Such signed binary representations are especially useful in ECC, since inversions of arbitrary points can be obtained with almost free operations over elliptic curves. Some properties of such signed binary representations are related to the cost of an exponentiation. Especially, the number of non-zero bits (Hamming weight) is important since this value rules the number of multiplications in the exponentiation. Thus analyzing signed representations implies a cost evaluation of exponentiations.

The non-adjacent form (NAF) is a well-known signed binary representation [9]. A NAF of a positive integer a is an expression a = vi2^{i} where vi E -1,0,11, vn-i 0 and no two successive digits are non-zero, i.e. vi • vi$_{+}$i = 0 for i = 0,1, ..., n - 2

* This work was supported by research program 2011 of Kookmin University in Koreat. This work was partly supported by the IT R&D program of MKE/KEIT[10039140, Developement of Crypto Algorithms(ARIA, SEED, KCDSA, etc.) for Smart Devices(ARM7,9,11, UICC)].

[9]. Each integer a has a unique NAF representation denoted by NAF(a). Moreover, NAF(a) can be efficiently computed by right-to-left operations ([5], for example). The average Hamming weight of NAF has been found by Gollmann et al. (Corollary 2.6 in [3]), and more recently Wu and Hasan have presented a closed form expression for the average number of Hamming weight and length in a minimal weight radix-r signed-digit representation where the special case r = 2 is NAF [10].

In [6], Joye-Yen proposed a left-to-right analogue of NAF. We call it "FAN" as the reverse order of NAF. It is known that NAF and FAN can be generated by applying a sliding window method with width-2 to the Booth encoding [1] in right-to-left and left-to-right, respectively [4, 8]. Note that the Booth encoding was also introduced as the reversed binary representation by Knuth [7, Exercise 4.1-27]. The Booth encoding and FAN of an integer a are denoted by Boom(a) and FAN(a), respectively.

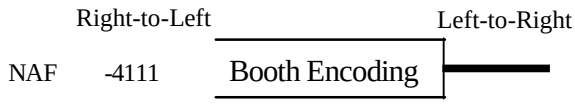


Fig. 1. A relation of the Booth encoding, NAF, and FAN.

In this paper, we provide some explicit formulae of NAF and FAN by using fundamental properties induced from Booth encoding. We propose some results for properties of Booth encoding and relations about Pattern, Classification, Extension, Adjacency, and Length among Booth, NAF, and FAN(Section 3).

2 Preliminaries

2.1 Booth Encoding, NAF and FAN

The n-bit Booth encoding [1] is an n-bit signed binary representation that satisfies the following two conditions:

— Signs of adjacent non-zero bits (without considering zero bits) are opposite.
— The most significant non-zero bit and the least significant non-zero bit are 1 and 1, respectively, unless all bits are zero.

In [8], they proved that for each integer there exists only one representation that satisfies the Booth recoding properties and showed a simple conversion method from an n-bit binary string to (n 1)-bit Booth encoding. Given an integer a, the Booth encoding of a is obtained by

$$2a \ominus a,$$

where $\ominus$ stands for a bitwise subtraction.

The non-adjacent form (NAF) also represents an integer in signed form [9]. Since there is no successive non-zero bits in the representation, NAF is a standard technique for computing exponentiations [5]. According to [4, 8], NAF can be interpreted as a combination of the Booth encoding and a right-to-left sliding window method with width-2 ($SW_{2^{-t-1}}$), i.e. width-2 window, moving right-to-left, skipping consecutive zero entries after a nonzero digit is processed. 01, 01, 11, and 11 are converted to 01, 01, 01, and 01, respectively.

Integer Booth Recoding $^{sw7_2'-t-1}$ NAF representation

Integer Booth Recoding FAN representation

2.2 Notations

For a given binary n-bit integer a (between 0 and 2n-1) we use the following notations:

- BINARY(a), BooTH(a), NAF(a), and FAN(a) denote the binary, Booth encoding, NAF, and FAN representation of the integer a respectively.
  - BINARY(a) := (a_n_i, al, ao)2 with ai E {0,1},
  - BOOTH(a) := (/3n, •••,131, 00)2 with 13i E {-1, 0,1},
  - NAF(a) := *(Vn, •••,V1,vo)2* with v_i E {-1,O,1},
  - FAN(a) := *(O_n*, .•., 01, (6, 0)2 with Oi E { -1, 0, 1}.

  Note that throughout this paper each of the symbols 0i, vi, and Oi are only utilized to denote a digit of the representation of Booth, NAF, and FAN respectively.
- */3(n) := {Bouni(a)* 0 < a < 2^n — 1}, that is a set of all Booth encodings of integers between 0 and 2n - 1.
  - *Case_I _8(n) :=* {BooTH(a) with *f3$_n$* = 0 0 < a < 2^n —1}.
  - *Case_II_B(n):=* {BooTH(a) with *(/3$_n$, /3$_{n_1}$)* = (1,1) I 0 < a < 2^n —
  - *Case_III_B(n):=* {BooTH(a) with (/3$_n$, /3$_{n_1}$) = (1,0) 0 < a < 2n -1}.
- Ai(n) := {NAF(a) 0 < a < 2^n — 1}, that is a set of all NAF representations of integers between 0 and 2n - 1.
- .F(n) := {FAN(a) 0 < a < 2n - 1}, that is a set of all FAN representations of integers between 0 and *2n -1.*

– E_n is the negligible function in n, namely for every constant $c > 0$ there exists an integer m such that 1E,1 < 1/nc for all n > *m.*
— If *t* is a real number, then [t] is the largest integer < *t* and it] is the smallest integer > *t.*

2.3 Some cases of NAF and FAN

In this section, we show that how to compute NAF and FAN representations from the Booth encoding. For example, for an integer 13 = (1, 1, 0,1)2_i we have BooTH(13) = (1,0,1,1,1)2 from Algorithm ??. Then we divide BooTH(13) (as a string) into width-2 windows from *right to left:* 01, 01, 11 (the leftmost 0 was padded), and convert 11 to 01 and 11 to 01, if any. Thus we have NAF(13) = (1,0, 1,0,1)2.

In order to generate FAN representation of 13, we divide the string of BooTH(13) into width-2 windows from *left to right:* 10, 11, 10 (the rightmost 0 was padded). Then, similarly to NAF, convert 11 to 01 and 11 to 01, if any. Thus we have FAN(13) = (1,0, 0,1,1)2. Note that FAN can have successive non-zero bits unlike NAF.

3 Several results of Booth encodings

In this section, we prove several results for the Booth encodings such as Pattern, Classifications, Extension, Adjacency, and Length.

Table 1. NAF, FAN, Booth encoding representations of some integers

Integer a	Signed Binary			Non-signed Binary
	NAF (a)	**FAN** (a)	**BOOTH** (a)	
0	0000000	0000000	0000000	0000000
1	0000001	0000001	0000011	0000001
2	0000010	0000010	0000110	0000010
3	0000101	0000101	0000101	0000011
4	0000100	0000100	0001100	0000100
5	0000101	0000101	0001111	0000101
6	0001010	0001010	0001010	0000110
7	0001001	0001001	0001001	0000111
8	0001000	0001000	0011000	0001000
9	0001001	0001001	0011011	0001001
10	0001010	0001010	0011110	0001010
11	0010101	0001101	0011101	0001011
12	0010100	0010100	0010100	0001100
13	0010101	0010011	0010111	0001101
14	0010010	0010010	0010010	0001110
15	0010001	0010001	0010001	0001111

3.1 Some Properties of Booth Encoding

Property 1. Due to the definition of Booth encoding (refer to Section 2.1), the Hamming weight of BooTH(a) is always even, if the original integer a is positive.

Let $(11)^k$ be a pattern of non-zero bits in Booth encoding such that
k 2 1

1, 1 , , 1, 1, 1, 1 (exactly k-times) after omitting all zero bits between 1 and 1. Let #[$(11)^k$] be the total number of strings having (11)k pattern. For example, in 8(4), #[$(11)^1$] = 10 (i.e. integers 1,2,3,4,6,7,8,12,14,15) and #[$(11)^2$] = 5 (i.e. integers 5,9,10,11,13). Refer to the fourth column of Table 1.

Theorem 1 (Pattern). *B(n) consists of exactly all possible representations with (11)k pattern for* $0 < k < [n/21$.

Proof. For $0 < k < [n/21$,

$$\#[(11)^{\circ}] = {}^{n+}{}_{0\,1),}\#[(11)^1] = n\,1) \qquad \#[(1i)k] = n^{+}{}_{2k^1} \qquad \#[(1\bar{1})^{i}{}_{r_{\#/21}} = {}^{n+1}{}_{21n/21)}$$

where the binomial coefficient M denotes the number of b-combinations from a set *S* with a elements.

Thus $_E r_k n/_i p$ $\#[(i)k_l = {}_E r_{ic} n/1$ (92+1) $= 2^n$ from $i_{E\,i}(n$ $n = 2^{n+1}$ and Ek=0 (^{n1}) = Ek-0 (a++11)* This implies that there are 2^n different representations with $(11)^k$ pattern. As the Booth recoding is unique, the assertion is proved. 0

Theorem 2 (Classification). *8(n) can be divided into the following three cases;*

i) Case_l_B(n) with #[Case_I_8(n)]=2'[1],
ii) Case_ILB(n) with #[Case_IL13(n)] = 2^{n-2},
iii) CaseJILB(n) with #[Case_III_13(n)] = 2^{n-2}.

Proof. From Property 1 and Lemma 1,

$$\#[Case\_I13(n)] = 1 \quad {}^{n} \quad ,.\text{-}1 \qquad 2\,k)\ `$$
$$k = 0$$

$$\#[Case\_II\_13(n)] = \sum_{k=0}^{J} n\ 2k\ 1) = 2^{n-2}{}_5$$

$$\#[Case\_III\_B(n)] = 2^{n-2} = {}^{Ld\,1'}{}_{71}\,1 \begin{cases} \backslash\_, \text{-}1 \ {}^{n-1}_{2k+1} & \text{(if n is even)} \\ {}^{n-1}_{2k+1} \cdot & \text{(otherwise)} \end{cases}$$

Theorem 3 (Extension). *8(n) can be constructed from 13(n — 1) according to the following rules;*

i) Case_I_B(n) = {(/3 = 0)11(1³n-i, • • •, 00) (0.-1, • • •, 130) E B(n - 1)},
ii) Case_H_B(n)= (3n, i³,-,.-1)=(1,1)11(N-2, • • •,00) (/n-2, • • 03o) E B(n - 2) ,
iii) Case_III_B(n) = On-1) = (1, 0)11(i³n-2, • • •, 00) (1, On-2, • •
Case_I_13(n-1)} .

Proof. From Property 1 and Lemma 1, 2, we can see that the assertion is true. ❑

Theorem 4 (Case 11-Classification).

i) # *[a* E *Case_II_13(n) the number of most significant consecutive non-zero bits of a is t, where t is even]* = $\frac{\quad}{3}$

ii) # *[a* E *Case_II_13(n) the number of most significant consecutive non-zero bits of a is t, where t is odd]* = $\frac{\text{on-2—nn}}{3}$

where in = 2 if n is odd and ic$_n$ = 1 if n is even. Especially,

iii) # *[a* E *Case_II_13(n) the number of most significant consecutive non-zero bits of a is 2]* = *2n-3*

Proof. Let n be even and s_t := # [a E *Case_IIJ3(n)* the number of most significant consecutive non-zero bits of a is even *t].* Then, based on Property 1

$$82 \quad :(2k) = 2", \; 84 = \sum_{k=0}^{2^{n-2}-1} 2k = 2", \; \cdots S2t = \sum_{k=0}^{2^{n-t}-1} 2k \quad 2n\text{-}2t\text{-}1,$$

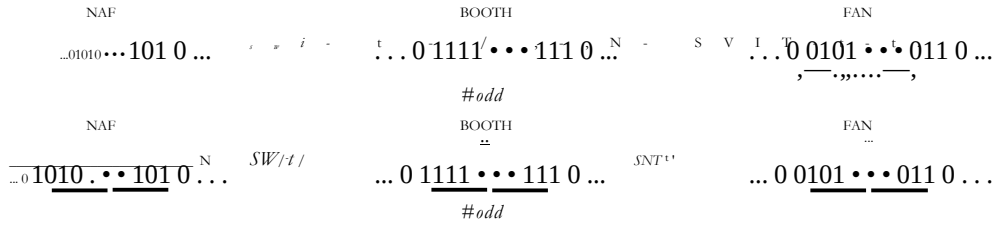
$$sn\text{-}2 = \sum_{k=0}^{1} (2k) = s_n = 1.$$

Thus, the result of Lemma 4.i) when n is even is $S2k = \frac{2^n - 3}{3} + 1$-1 From Lemma 2.ii), we can derive the result of Lemma 4.ii) when n is even from the above result that is 2n-2 $\frac{2^{n-1}+1}{3} - \frac{2^{n-2}-1}{3}$ •

Similarly, the result of Lemma 4.i) when n is odd is $\frac{2^n - 1}{3} + 2$ and the result of Lemma 4.ii) when n is odd is 2n-2 $\frac{2^n - 2}{3} + 2 =$ The third one is clearly s2. ❑

3.2 Relations among Booth, NAF, and FAN

Theorem 5 (Adjacent). *A substring with an odd number (> 1) of consecutive nonzero bits in Booth representations is converted into a substring with* 11 *or* 11 *at the least significant bits of the substring when transforming to FAN representation. Moreover, the resulting string in FAN is shorter than the original string in Booth, i.e. the most significant 1 of the FAN string will be one position lower than that of the Booth string. In the case of NAF string, the most significant 1 or 1 stays at the same position.*



Define *L [a]* as the bit-length of a representation of a, counting the bits from the least significant to the most significant 1. For example, if a = (10110)2 then L[BINARY(a)] = 5. The results of Lemma 5 directly serves the proof of Lemma 6.

Theorem 6 (Length). *For an arbitrary integer a*

i. L[Bouni(a) with *#(the most significant consecutive nonzero bits) = even] = L[NAF(a)] + 1, = L[FAN(a)] +1,*

ii. L[BooTH(a) with *#(the most significant consecutive nonzero bits) = odd (> 1)] = L[NAF(a)] = L[FAN(a)]* + 1,

iii. L[Boom(a) with *#(the most significant consecutive nonzero bits) = 1] = L[NAF(a)] = L[FAN(a)].*

4 Conclusion

In this article, we derived several interesting contributions from the relation between NAF and FAN based on the Booth encoding. The results would be applied to analyze the probability of the NAF representation of an n-bit integer, the average Hamming weight of NAF and FAN, and the average length of zero runs in both the NAF and FAN.

References

1. A. Booth, "A signed binary multiplication technique", Journ. Mech. and Applied Math., 4(2), pp.236-240, 1951.
2. D. Gollmann, Y. Han, and C.J. Mitchell, "Redundant Integer Representations and Fast Exponentation", Designs, Codes, and Cryptography, vol. 7, pp.135-151, 1996.
3. IEEE 1363-2000, IEEE Standard Specifications for Public-Key Cryptography, 2000.

4. M. Joye, and S.-M. Yen, "Optimal Left-to-Right Binary Signed-digit Exponent Recoding", IEEE Transactions on Computers 49(7), pp.740-748, 2000.
5. D.E. Knuth, "*The Art of Computer Programming, vol. 2, Seminumerical Algorithms*, 2nd ed., Addison-Wesley, Reading, Mass, 1981.
6. G.W. Reitwiesner, *Binary arithmetic*, Advances in Computers, vol.1, pp.231-308, 1960.
7. H. Wu, and M.A. Hasan, "Closed-Form Expression for the Average Weight of Signed-Digit Representations", IEEE Transactions on Computers 48(8), pp.848-851, 1999.