A New Multiplication Architecture for DSP Applications

Nguyen Minh Son*, Jong-soo Kim*, Jae-ha Choi*

Abstract

The modern digital logic technology does not yet satisfy the speed requirements of real-time DSP circuits due to synchronized operation of multiplication and accumulation. This operation degrades DSP performance. Therefore, the double-base number system (DBNS) has emerged in DSP system as an alternative methodology because of fast multiplication and hardware simplicity. In this paper, authors propose a novel multiplication architecture. One operand is an output of a flash analog-to-digital converter (ADC) in DBNS format, while the other operand is a coefficient in the IEEE standard floating-point number format. The DBNS digital output from ADC is produced through a new double base number encoder (DBNE). The multiplied output is in the format of the IEEE standard floating-point number (FPNS). The proposed circuits process multiplication and conversion together. Compared to a typical multiplier that uses the FPNS, the proposed multiplier also consumes 45% less gates, and 44% faster than the FPNS multiplier on Spartan-3 FPGA board. The design is verified with FIR filter applications.

Keywords : Double-base Number System, flash ADC, double-base number encoder, logarithm number sysstem, DSP, FIR filter

I. Introduction

In recent emerging technologies such as wireless communications and bio-imaging applications, an insatiable need has arisen for higher speed DSPs. However, the multiplication, one of the most frequently performed operations in real-time DSP, is the main bottleneck in speed improvement due to its complicated structure [1] - [3]. Thus, many researchers have tried to design faster adders and multipliers [2] - [12]. Meanwhile, some researchers introduced other number systems instead of using binary number system to overcome the barrier of speed limitation in DSPs. One of such works is using a logarithm number system (LNS). The LNS multiplication can be simply processed by addition. The LNS addition can be implemented by a look-up-table (LUT) [6] - [9], or a shift - and - add - based method [10]. The LUT-based method can perform fast operations, but demand more chip area. On the other hand, the shift - and - add - based method achieved good trade-off between area, time, and accuracy. All these researches focused onconverting binary numbers into theLNS to replace time-consuming multiplication with additions. Another method for an efficient multiplication with less hardware resource is using the double-base number system (DBNS) [3] - [6]. However, the main drawback of the DBNS is compatibility with other system and multidimensional processing requirement. In addition, these studies have data precision problem due to non linear conversion processing [8], [10]. Thus, the DBNS suggested Greedy algorithm to solve the multidimensional problem. The effectiveness of the DBNS has been proven with FIR filter design [2] - [5].

In this paper, authors present a novel multiplier that uses two different types of operands. One operand uses the DBNS code supplied from ADC, while the other operand is represented in IEEE floating point number for system compatibility.

This paper is organized as follows: In Sections II, the DBNS operations and digital filter arithmetic operations are introduced briefly. The DBNE for converting analog signals to a digitized DBNS operand is discussed in Section III. Mathematical derivation for multiplication algorithm is discussed in Section IV. Experimental and simulation results of synthesized DBNE and the logic level circuits of the multiplier are presented in Section V. Section VI concludes this work.

* 울산대학교

투고 일자 : 2011. 1. 13 수정완료일자 : 2011. 4. 30
게재확정일자 : 2011. 4. 30
* 본 연구는 울산대학교 2010년도 교내 학술연구비 지원에 의하여 수행되었음.

II. DBNS Arithmetic and Digital Filter Operations

The goal of DBNS is to reduce hardware complexity, and enhance the processing speed of a real time DSP [4] - [5]. The following equation represents any real number X in the DBNS format, where, s_k is a signed bit, b_k is a binary exponent, t_k is a ternary exponent, and $\varepsilon$is an error tolerance. The binary and ternary exponents are independent of each other.

$$\left| X - \sum_k s_k 2^{b_k} 3^{t_k} \right| < \varepsilon \tag{1}$$

If error tolerance is small enough, than X can be approximated to (2)

$$X \cong \sum_k s_k 2^{b_k} 3^{t_k} \tag{2}$$

In general, k is larger than 1 to represent X within small error ranges. The error ε must be zero in an ideal case. In case of k=1, X can be represented as (3) with a finite precision.

$$X \cong s 2^b 3^t \tag{3}$$

Now, the four fundamental arithmetic operations of (3) can be derived as follows,

$$X = X_1 \times X_2 = (s_1 \oplus s_2) 2^{b_1 + b_2} 3^{t_1 + t_2}$$
$$X = X_1 \div X_2 = (s_1 \oplus s_2) 2^{b_1 - b_2} 3^{t_1 - t_2}$$
$$X = X_1 \pm X_2 = s_1 2^{b_1} 3^{t_1} \left[1 \pm (s_1 \oplus s_2) 2^{b_2 - b_1} 3^{t_2 - t_1} \right]$$
$$= s_1 2^{b_1} 3^{t_1} \cdot 2^{b_\Phi} 3^{t_\Phi} = s_1 2^{b_1 + b_\Phi} 3^{t_1 + t\Phi}$$

where,

$$\Phi(b_2 - b_1, t_2 - t_1) = 2^{b_\Phi} 3^{t_\Phi} = 1 \pm (s_1 \oplus s_2) 2^{b_2 - b_1} 3^{t_2 - t_1}$$

In reference [4] and [5], the multiplication and division are easily performed by adding or subtracting exponents of operands. However, the addition and subtraction are executed by the LUT in which results of Φ function are stored as a single DBNS term. Due to ROM structure, the addition and subtraction are inflexible. In addition, the DBNS must be converted into binary for readability and compatibility later. Therefore, the authors suggest a new multiplier linked with a flash ADC to speed up the DSP operation.

In digital filter, one input is a digitized signal from ADC, while the other input is a coefficient supplied from external. Typical ADC and FIR filter structures are well known [1], [13]-[16]. In order to design faster ADCs, a double-base number encoder (DBNE) was designed [17]. The DBNE converts input analog signal $x(n)$ into DBNS format $2^{bd}3^{td}$. The filter coefficient input $h(0)$ is supplied in the IEEE 32-bit standard floating point format as $m2^{Bc}$. Therefore, the following well known FIR filter equation can be rewritten as (4),

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) = \sum_{k=1}^{N-1} h(k)x(n-k) + h(0)x(n) \tag{4}$$
$$y(n) = y(n-1) + m2^{Bc} \cdot 2^{bl}3^{td}$$

where, m is the mantissa part of the IEEE standard floating point representation. The mantissa m can be represented by m = 1. f = 1+ f, and the fraction f can be expressed by f=0.$X_{22}X_{21}\cdots X_1X_0\epsilon$[0,1), while Bc and bd and td are within [-127, 127).

In the next section, the ADC's encoder logic that provides input x(n) to a FIR filter in the DBNS format will be discussed.

III. DBNE Design

The error ε in (1) can be redefined as the difference between X and X', where X' is the closest value to X when k=1. It is desirable to minimize ε when finding X'. The error tolerance ε depends on the exponential bit size of binary and ternary. For example, if a given real number range is [0, 1] and the digits of binary and ternary exponents are [-2, 2], the maximum value ε (ε_{MAX}) is 0.125 with X=0.875 and X'=0.75($2^{-2}\cdot 3^1$) or 1($2^0\cdot 3^0$). However, if the digits of exponent are increased to [-3, 3], ε_{MAX} decreases to 0.083 with X=0.583 and X'=0.5($2^{-1}\cdot 3^0$) or 0.667($2^1\cdot 3^{-1}$).

Fig. 1 shows the relationship between the number of exponent bits and the maximum error. It can be seen that the maximum error is close to 0 when the number of exponent bits is bigger than 9. In this calculation, the sizes of exponents of binary and ternary are assumed to be the same. In addition, the authors assumed that the raw input values of X vary from 0.05 to 1.20, since the input voltage of the 6-bit flash ADC changes between 0.55V and 1.05V [18] - [19]. Thus, one LSB resolution of the ADC is 8.065mV. If we use 8-bit exponents for binary and ternary radices, then the maximum error is 0.0049, which is equivalent to 0.6 LSB in the ADC. In case of 9-bit exponents, we get 0.0016 (0.15 LSB) as the maximum error. Thus, the conversion error of an input voltage with single DBNS term is negligible. Therefore, the authors used 9-bit exponents.

The voltage values converted into the double-base numbers with 9-bit exponents are shown in Table I. For simplicity, only 16 voltage levels among 63 discrete

values are shown. The first column in Table I is the input voltage. The second column, DV, denotes decimal values of the ADC input from 1 to 63 corresponding to the input voltages. The third and fourth columns marked with b and t represents the value of binary and ternary exponents respectively. CV stands for calculated values with the given b and t using (3). The difference between "Input" and "CV" is shown in the last column as ε. The DBNE can be easily implemented with ROM encoder architecture.

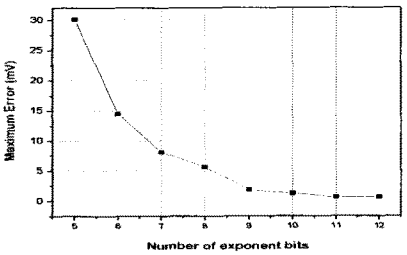


Fig. 1. Relationship between error and exponent

TABLE 1 6-bit DBNS Code and Error

Input (V)	DV	b	t	CV	ε (LSB)
0.55000000	1	-134	84	0.54975094	0.030882
0.58225806	5	191	-121	0.58218009	0.009668
0.61451613	9	115	-73	0.61460757	0.011338
0.64677419	13	207	-131	0.64613718	0.078985
0.67903226	17	-186	117	0.67859024	0.054807
0.71129032	21	-10	6	0.71191406	0.077339
0.74354839	25	-151	95	0.74300178	0.067776
0.77580645	29	193	-122	0.77624012	0.053772
0.80806452	33	136	-86	0.80844691	0.047414
0.84032258	37	163	-103	0.84023362	0.011030
0.87258065	41	190	-120	0.87327014	0.085492
0.90483871	45	-184	116	0.90478699	0.006413
0.93709677	49	11	-7	0.93644262	0.081110
0.96935484	53	206	-130	0.96920578	0.018482
1.00161290	57	-84	53	1.00209031	0.059195
1.03387097	61	195	-123	1.03498683	0.138358

IV. Mathematical Derivation of Multiplication and Conversion

In [3] and [4], the real-time DSP circuit with the DBNS was implemented by LUT based on inner product step processor (IPSP), and both operands are DBNS codes. Instead of using the same types of operands, the authors propose a new LNS multiplier that does not require a LUT. However, the authors use mixed data formats as mentioned before.

The expression $T=m2^{Bc}\cdot 2^{bd}3^{td}$ in (4) is the multiplication of an input signal and a coefficient. This expression should be converted into $M\cdot 2^{B}$, where $M\in[0,1)$ and $B\in[-127,127]$, to yield multiplied result as the IEEE FPNS for iterative addition later. In order to convert and multiply input data simultaneously, an approximation of non-linear function $y=2^{f}$ to a linear function $y=1+f$, where $m=1.f=1+f$ and $f\in[0,1)$, are applied based on the following mathematical derivation.

A. Non-linear and Linear Functions

The polynomial approximation is used to approximate the non-linear function $g(x)$ into the linear function $f(x)$. Fig. 2 shows the graph of functions $f(x)$ and $g(x)$.The maximum distance $Dmax$ between g(x) and $f(x)$ is computed along the x-axis. In Fig. 2, if we use another function $y=f'(x)=(1-d)+x$, which is parallel to the linear function $f(x)$ for every y position from 1 to 2, then we can approximate $g(x+d)=f(x)$, where d is less than $Dmax$. This means that each value y of the linear function $y=1+x$ is approximated to y' of the non-linear function $y=2^{x}$ with a distance d. Since the filter coefficient is usually normalized, the range number between 1 and 2 is valid. In order to obtain more precise conversion between two functions $g(x)$ and $f(x)$,we need to determine the constant d before approximating.

If partition technique is applied [15], the graph of function $f'(x)$ will be shifted to the right by (5),

$$d=\frac{D_{MAX}}{N} \tag{5}$$

where N is the number of shifting steps. The error E is defined as the maximum deviation between two values of x_1 and x_2. The x_1 corresponds to x of the linear function $y=f'(x)=1-d+x$, while x_2 corresponds to x of the non-linear function. The error E must be less than d, while x_2 approaches x_1. In order to reduce the error E, they range [1,2) is divided by large N steps. If E needs to be less than $10^{-3}\approx 2^{-10}$, then N must be greater than $(2^{10}*\mathrm{Dmax})\approx 88$. If N is greater than 127, the maximum error E_{MAX} is less than 0.0006. Table II shows the relationship between the maximum error and the number of shifting steps.

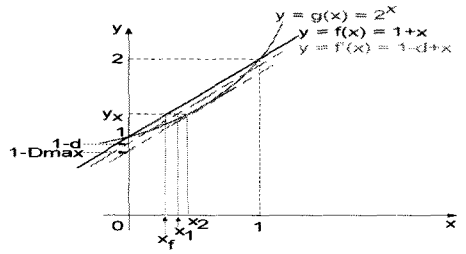


Fig. 2. Approximation between nonlinear and linear functions

TABLE 2 The Relationship Between Maximum Error and Number of Shifting Steps

Shifting steps (N)	Max. error (E_{MAX})	Shifting steps (N)	Max. error (E_{MAX})
0	0.0861	31	0.0027
1	0.0430	63	0.0013
3	0.0215	127	$0.0006 < 2^{-10}$
7	0.0108	255	$0.0003 < 2^{-11}$
15	0.0054	511	$0.00015 < 2^{-12}$

B. Approximation

Since $m=1.f=1+f$ and $f=\in[0,1)$, the $T=m2^{Bc}\cdot 2^{bd}3^{td}$ in (4) can be approximated as follows,

$$T=m2^{Bc}\cdot 2^{bd}3^{td}\cong 2^{f+d}\cdot 2^{Bc}\cdot 2^{bd}3^{td} \tag{6}$$

where d is a constant obtained from (5). Moreover, $3^{td}=2^{td*\log_2(3)}\approx 2^{I+F}$, where I is integer of the product $[td*\log_2(3)]$ and F is fraction of the product $[td*\log_2(3)]$. Since Bc, bd, and td are in the range of [-127,127) in (4), I and F can be found as follows

$$I=\sum_{i=0,1,4,6} int[td\cdot(2^{-i})] \tag{7}$$

$$F=\sum_{i=1,4,6,8,9,20,21} frac[td\cdot(2^{-i})] \tag{8}$$

The fraction and integer of $[td\cdot(2^{-i})]$ are obtained by shifting operation. If we substitute I and F into(6), then T is

$$T\cong 2^{f+d+F}\cdot 2^{Bc+bd+I} \tag{9}$$

There are two cases when approximating (9) into IEEE floating-point format

$$(a)\ IF\,(f+d+F)>1\ then \quad T\cong(f+d+F)\cdot 2^{Bc+bd+I+1} \tag{10}$$

$$(b)\ IF\,(f+d+F)\le 1\ then \quad T\cong(1+f+d+F)\cdot 2^{Bc+bd+I} \tag{11}$$

Here, the T is in the format of the floating-point after multiplication. The mantissa part is a binary addition of $(1+f+d+F)$ or $(f+d+F)$, and the exponent part is a binary addition of $(Bc+bd+I)$ or $(Bc+bd+I+1)$. Thus, the expression I and F can be processed by binary adders and shifters.

V. Logical Circuit Implementation and Simulation Results

Fig. 3 shows the logic level circuits of inner product processor for (4). The proposed multiplier consists of 23-bit binary comparators, shifters, and adders. The 23-bit comparators compare the value of the mantissa $(1+f)$ with known constant Y of the linear and non-linear functions. The detail of the "Fraction and Integer Conversion" block in Fig 3 performs the above equations (7) and (8), which consists of simple shifters and adders. The maximum number of stages of cascaded adders is only 3 in 23-bit expression. These operations also can be done in parallel. The proposed multiplier needs smaller number of shifters and adders compared to a 24-bit fixed-point multiplier. The worst delay time of fixed-point multiplier depends on the number of bits and round-off methodology [11], [13]. The delay time of the proposed multiplier is one stage of parallel shifters and 4 stages of 23-bit addition only. As the number of comparator in Fig. 3 increases, the multiplier output becomes more accurate. However, the multiplier's speed is not reduced due to the parallel processing scheme of comparison. Fig. 3 does not have any relation with partitioning numbers, and its delay time is fixed according to the word size. The error rate is 0.06% with 127 comparators (N=127) and 0.13% with 63 comparator.

In this work, the Spartan-3 FPGA board was used to verify the functionality, and to compare performance of two multipliers: the proposed multiplier and the floating-point multiplier. Also, the board is used to model the FIR filter with DBNS and floating-point representation. Table IV shows the results of the two multipliers. The proposed multiplier requires 45% less gates and it is 44% faster because of the simple structure.

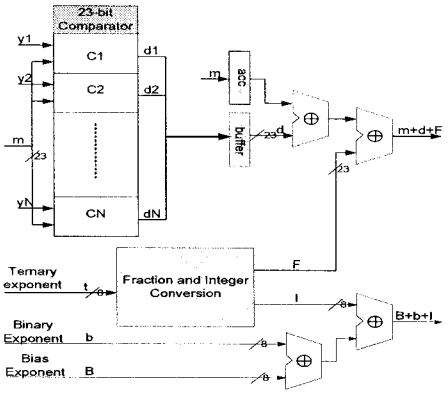


Fig. 3. Multiplication and conversion to floating point

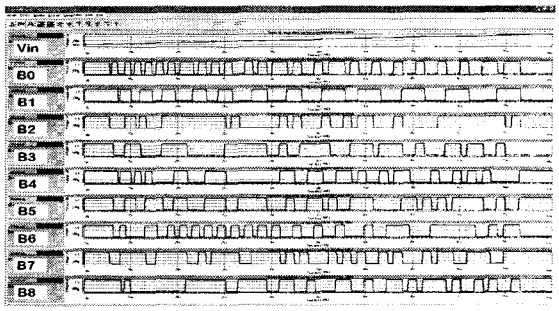
TABLE 3 Comparison of DBNS and FPNS Multiplication Designs in VERILOG

	FPNS Multiplication	DBNS Multiplication
No. of gates	3965	2134
Delay time	34.93 ns	15.27 ns

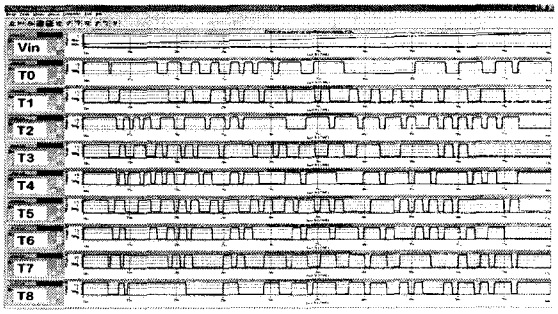
HSPICE simulation result of the 6-bit flash ADC encoded with 9-bit DBNE is shown in Fig. 4. These

waveform correspond to the binary and ternary exponent in Table I. The first waveform in each plot shows the analog input signal V_{in}. The second row waveform is the LSB, while the last waveform is the MSB.

Fig. 5 shows two FIR filters' characteristics simulated with MATLAB. Both filters have 52 coefficients, and their cut-off frequencies are in the same range as 0.45~0.55 rad/s. The pass band's ripples of both filters are the same, but the attenuation in stop band of DBNS FIR filter and FPNS FIR filter are -55dB and -75dB respectively. The stop band's attenuation difference results from the conversion error of DBNS.



(a) Output waveform of binary exponent



(b) Output waveform of ternary exponent

Fig. 4. Output waveform

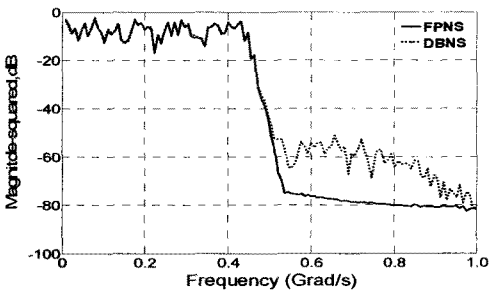


Fig. 5. Filter responses

VI. Conclusions

In this work, an algorithm to design a double-base number encoder (DBNE) in flash ADC has been developed, and verified by SPICE. The proposed DBNE has been designed in 0.18um CMOS technology, and its performance is superior to other encoders. In order to utilize the DBNE circuits efficiently, the authors developed faster multiplier that supports DBNE operand and IEEE floating-point numbered operand for system compatibility. The suggested multiplier was simulated by Verilog HDL on Spartan-3 FPGA board. The multiplier circuit consists of comparators, adders, and shifters. The design could reduce the hardware resources by 45%, and improve speed by 44%. Although the proposed DBNS filter has accuracy problem, the DBNS filter can be used within allowable tolerance.

As future work, the authors will compare the proposed multiplier's performance with other types, since there is close correlation between filter coefficient's value and multiplier structure. With the same filter design specification, delay time and chip area will be analyzed, since the proposed circuits require more hardware resources. Finally, the percentage error of the proposed algorithm will be further analyzed, and compared with other conversion algorithms that convert input into logarithm number to replace multiplication with addition.

References

[1] Walt Kester, Mixed-Signal and DSP Design Techniques, Analog Devices, 2003.

[2] Amitabha, Kolkata, Pavel Sinha, Kenneth Alan Newton, Krishanu Mulherjee, Triple-Base Number Digital Signal and Numerical Processing System, US Patent, Jan 24, 2008.

[3] Kacem, R.; Khouja, N.; Grati, K.; Ghazel, A., Low Power Implementation of Digital Filters using DBNS Representation and Sub-expression Sharing, The 2nd International Conference on Signals, Circuits and Systems, Nov 2008, pp. 1-6.

[4] Vassil S. Dimitrov, Graham A. Jullien, and W. C. Miller, Theory and Applications of the Double-Base Number System, IEEE Tran. on Computers, vol. 48, 1999, pp. 1098 - 1106.

[5] Vassil S. Dimitrov and Graham A. Jullien, A New Number Representation with Applications, IEEE Circuits and Systems Magazine, 2003.

[6] Roberto Muscedere , Vassil Dimitrov , Graham A. Jullien , William C. Miller, Efficient Techniques for Binary to Multidigit Multidimensional Logarithmic Number System Conversion Using Range Addressable Look-Up Tables, IEEE Tran. on Computers, vol. 54-3, March 2005, pp. 257 - 271.

[7] Sheng-Chien Huang and Liang-Gee Chen, A 32-bit Logarithmic Number System Processor, Journal of VLSI Signal Processing, vol.14, 1996, pp. 311 - 319.

[8] Suganth Paul, Nikhil Jayakumar, and Sunil P. Khatri, A Fast Hardware Approach for Approximate, Efficient Logarithm and Antilogarithm Computations, IEEE Tran. on VLSI, vol 17, Feb. 2009, pp. 269 - 277.

[9] K. Johansson, O. Gustafson, and L. Wanhammar, Implementation of elementary functions for logarithmic number systems, IET Comput. Digit. . Tech., vol.2, Jul. 2008, pp. 295-304.

[10] Tso-Bing Juang, Sheng-Hung Chen, and Huang-Jia Cheng, A Lower Error and ROM-Free Logarithmic Converter for Digital Signal Processing Applications, IEEE Tran. on Circuit and Systems, vol 56, Dec. 2009, pp. 931 - 935.

[11] Huey Ling, High-Speed Binary Adder, IBM Journal of Research and Development, vol.5-3, 1981.

[12] Mustafa GOK, A Novel IEEE Rounding Algorithm for High-speed Floating-point Multipliers, The VLSI journal, 2007, pp. 549 - 560.

[13] Bhaskar D. Rao, Floating-point Arithmetic and Digital filter, IEEE Tran. on Signal Processing, vol 40, 1992, pp. 85 - 95.

[14] Christian Piguet, Low-Power CMOS Circuit: Technology, Logic Design and CAD tools, Taylor & Fracis, 2006.

[15] Minh Son Nguyen and Jongsoo Kim, The Conversion Algorithm between Double-Base Number System and Floating-Point Number System applied for FIR filter, The 24th ITC-CSCC 2009, May 2009, pp 1458-1461.

[16] Minh Son Nguyen, Insoo Kim, Kyusun Choi and Jongsoo Kim, Design and Implementation of Flash ADC and DBNS FIR filter integrated on DSP System, ISOCC'09, vol.1, Nov 2009, pp. 430-435.

[17] Minh Son Nguyen, Insoo Kim, Jae ha Choi and Jongsoo Kim, Algorithm and Design of Double-base Log Encoder for Flash A/D Converters, KISPS, vol. 10, Nov 2009, pp. 289-293.

[18] Daegyu Lee, Jincheol Yoo, Kyusun Choi, Design Method and Automation of Comparator Generation for Flash A/D Converter, ISQED, 2002, pp. 138 - 142.

[19] Jaehyun Lim, Insoo Kim, Nguyen Minh Son, Jincheol Yoo, Jongsoo Kim and Kyusun Choi, Low Power Flash A/D Converter with TIQ Comparators for Multi-Standard Mobile Applications, IREE, vol.4, Dec 2009, pp. 1447-1452.

[20] Daegyu Lee, Jincheol Yoo, Kyusun Choi and Jahan Ghaznavi, Fat Tree Encoder Design for Ultra-High Speed Flash A/D Converters, The 45th Midwest Symposium on Circuits and Systems, vol.2, Aug 2002, pp. 87 - 90.

[21] Jincheol Yoo, A TIQ Based CMOS Flash A/D Converter for SoC Applications, Ph.D. Dissertation, Dept. of Comp. Sci. and Eng. The Pennsylvania State University, 2003.

Nguyen Minh Son received B.E and M.E degree in Computer Engineering from The HCMcity University of Technology, Vietnam, in 2001 and 2005 respectively, and received Ph.D degree in Electrical Engineering at the University of Ulsan, Korea in 2010. He is a lecturer in the Schoolp of Computer Science and Engineering, Vietnam National University, Vietnam.

Jongsoo Kim received his Ph.D. degree from the University of Alabama in Hunstville in 1994. Currently, he is a professor in the School of Electrical Engineering, University of Ulsan, Ulsan, Republic of Korea. His current research interests include low-power circuits, mixed signal circuits, A/D converter circuits design, and high-level synthesis.

Jaeha Choi is a professor in the School of Electrical Engineering, University of Ulsan, Ulsan, Republic of Korea. His current research interests is designing RF analog circuits and MMICs.