Application of Constraint Algorithm for High Speed A/D Converters

Minh Son Nguyen, Soo A Yeo, Man-Ho Kim, and Jongsoo Kim*

Abstract

In the paper, a new Constraint algorithm is proposed to solve the fan-in problem occurred in the encoding circuitry of an ADC. The Flash ADC architecture uses a Double-Base Number System(DBNS). The DBNS has been known to represent the Multidimensional Logarithmic Number System (MDLNS) used for implementing the multiplier accumulator architecture of FIR filter in Digital Signal Processing (DSP) applications. The authors use the DBNS with the base 2 and 3 in designing ADC encoder circuits, which is called as Double Base Integer Encoder(DBIE). A symmetric map is analyzed first, and then asymmetric map is followed to provide addition ready DBNS for DSP circuitry. The simulation results of the DBIE circuits in 6-bit and 8-bit ADC show the effectiveness of the Constraint algorithm with 0.18μm CMOS technology. The DBIE yields faster processing speed compared to the speed of Fat Tree Encoder (FAT) circuits by 17% at more power consumption by 39%.

Keywords : Double-Base Number System, Flash ADC, Asymmetric DBIE and Constraint Algorithm

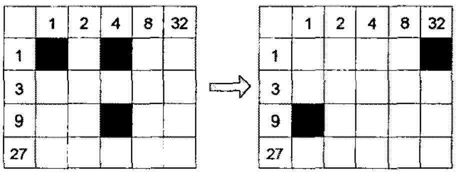
I. Introduction

1. A Double Base Number System (DBNS)

In [1] and [2], a new double-base redundant number system was proposed with the following number representation,

$$X=\sum_{i,j}^{k,m} d_{ij}\, 2^i 3^j \qquad (1)$$

where $d_{ij} \in \{0,1\}$, $i \in (0,k)$ and $j \in (0,m)$ are independent integers. A 2-dimensional table will be used to illustrate the representation scheme. Since *i and j* are independent, we can represent any number as indices in 2 dimensions. The base 2 is on the x-axis, while the base 3 is represented as the y-axis as shown Fig. 1. The summation of non-zero squares is equal to the given number X. In references [1] and [4], the representation of the Greedy algorithm was suggested as the name of Near Canonical Double Base Number Representation (NCDBNR) instead of using Canonic Double Base Number Representation (CDBNR) due to its NP characteristics. Now let's take an example to show the principle briefly. A DBNS representation of X = 41 can be converted to the right side according to the Greedy algorithm as shown in Fig 1. More details about DBNS arithmetic operations can be founded in [1, 2, 4, 5, 6] and [7].



	1	2	4	8	32
1	■		■		
3					
9			■		
27					

⇨

	1	2	4	8	32
1					■
3					
9	■				
27					

Fig. 1. Greedy algorithm and the CDBNR

2. Flash ADC

There were many efforts to design and implement a fast ADC at low power and chip area consumption in [11, 12, 13] and [14]. After converting analog signals into digital signals, a DSP circuit does arithmetic operations heavily. Therefore, the performance of DSP circuits are dependent on the capability of real time processing speed. Thus, we give a new flash ADC with DBNS to enhance the operation speed of DSP circuits.

Fig. 2 describes the architecture of a flash ADC using DBIE to convert analog signal to double base

* 울산대학교
투고 : 2008. 4. 28 수정완료 : 2008. 7. 21
게재확정일자 : 2008. 7. 25

* Corresponding author. All authors are with the School of Electrical Engineering, University of Ulsan, Ulsan, Korea.

integer system. In this architecture, the comparator and 0-1 generator circuits were well discussed in [9]. The voltage comparator and gain booster block uses TIQ architecture, which consists of 2^n CMOS transistors to compare the analog input voltage to 2^n predetermined reference voltages. The voltage comparator is connected with gain booster transistors to recover the smoothed rectangular pulse shape resulted from the voltage comparator characteristics to a sharp pattern. The output of voltage comparator and gain booster block is called as thermometer code. After then, the 0-1 generator circuit is used to convert the thermometer code to a required code. Thus we focus on designing the DBIE circuits with the Constraint algorithm suggested in this paper. The reasons of suggesting the Constrain algorithm are the following. First, the DBNS can do the arithmetic operation fast because the addition and multiplication can be processed by just shifting operation. Second, the DBNS has the fan-in problem in ADC, even though the Greedy algorithm may yield more compact code.

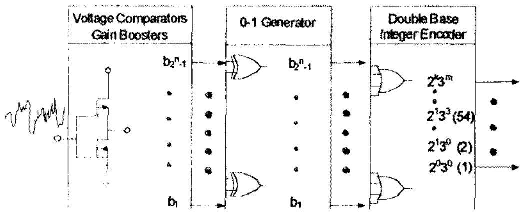


Fig. 2. Flash ADC with DBNS

The remainder of this paper is organized as follows. In Section II, the basic idea of DBNS and the proposed Constraint algorithm are introduced. Section III gives the experimental results of the Constraint algorithm for 6 and 8 bits ADCs. In Section IV, we conclude with a summary and future works.

II. DBNS-Map and Constraint Algorithm

Unlike typical n-bit ADCs, the proposed ADC architecture provides more outputs than the number of n-bit, since any number $X \in (0, 2^n - 1)$ must be represented by equation (1). In [2], the computational complexity of canonical representation was high. Thus, instead of getting CDBNR, the NCDBNR was proposed in [1, 3]. However, our goal is to find the minimum number of additions, while satisfying the fan-in requirements of ADC encoder circuits.

In this section, logical method is presented to calculate the number of bits, and additions based on the following lemma and other definitions referred in [1, 2, 4, 6] and [7].

Definition: DBNS-map is a symmetric if numbers of columns and rows must be equal or greater than by 1.

With every DBNS-map, any number X can be represented by many combinations of double bases (or cell). It means finding minimum number of cells is possible. Assume that DBNS-map has k columns and m rows. $|k - m| = l_0$ is called asymmetric level, where $l_0 =$ positive integer such as $1 < l_0 < k, m$. Let's try to find $|k - m| = l_0$ with $l_0 = 1$ or 0 to get symmetric DBNS-map. Assume $X \in (0, 2^n - 1)$ is binary integer number, thus $X < 2^n$. Besides, we always have $\exists X \in (0, 2^n - 1)$ where

$$X = 2^k 3^m + \sum_{i,j}^{k-1, m-1} d_{ij} 2^i 3^j < 2^n \qquad (2)$$

Thus, $2^k 3^k < 2^n$ and $k \ln 2 + m \ln 3 < n \ln 2$

If DBNS-map is symmetric, then $k = m$. From equation (2), we give:

$$k=m< (n \ln 2) / (\ln 2 + \ln 3) \qquad (3)$$

If a DBNS-map is asymmetric, then we have

$$k = m - l_0, \text{ or } m = k - l_0$$

Assume that $l_0 = k$ or m as maximum asymmetric of DBNS-map, then $m = \frac{k}{2}$ or $m = 2k$. Substitute m into equation (3), then we obtain

$$k < n \ln 2 \; / \; (\ln 2 + 1/2 * (\ln 3)) \text{ or } k > n \ln 2 \; / \; (\ln 2 + 2 \ln 3) \qquad (4)$$

In oder to obtain symmetric DBNS-map, we should refer k in

$$\left(\frac{n \ln 2}{\ln 2 + 2 \ln 3}, \frac{n \ln 2}{\ln 2 + \frac{1}{2} \ln 3} \right) \qquad (5)$$

Algorithm to find m and k that give minimum number of additions under the fan-in limitation in DBNS-map of 2^n-1 numbers

Input: number of bits of ADC - n,
FAN-IN constraint parameter - n_f

Output: number of columns of DBNS-map - k,
number of rows of DBNS-map - m,
number of stages of DBInt-Encoder - $S_{\min}$

PSEUDO algorithm:

```
Procedure DBIE_Architechture(n)
1. Calculate $k_{\min}$ and $k_{\max}$ from (5).
```

2. Find number of stages s responding with every pair (k_0, m_0), where k_0 and $m_0 \in (k_{\min}, k_{\max})$.
3. With each of s, check whether S is minimum.
4. If S is minimum, return $S_{\min}$, m and k; else jump to step 2.

At the 2^{nd} and 3^{rd} step, we must find the minimum number of stages s corresponding to number of rows and columns. We knew that number $X \in (0, 2^n - 1)$ as the following formula

$$X = 2^{i_1}3^{j_1} + 2^{i_2}3^{j_2} + \ldots + 2^{i_s}3^{j_s}$$

where: $i_1 \neq i_2 \neq \ldots \neq i_s$ with $i_\alpha < m, \alpha = 1..s$ and $j_1 \neq j_2 \neq \ldots \neq j_s$ with $j_\beta < k, \beta = 1..s$

In order to find the minimum number of stages, the minimum distance in graph theory is applied. Supposed that graph G is generated by nodes in S_i. Every number $X = 2^0 3^1 + 2^i 3^j + \ldots + 2^k 3^m$ can be represented by sub-graph lines (connected path) in G, and every addition terms are connections between two adjoining nodes.

Let's call $S_1, S_2, \ldots, S_i, \ldots, S_s$ as subset of representing every 2-integer number X. So, every S_i subset will give the set of 2-integers in the following:

$$S_i = \{2^0 3^0, 2^1 3^0, 2^0 3^1, 2^1 3^1, \ldots, 2^i 3^j, \ldots, 2^k 3^m\}$$

with number of elements of set S_i is $k*m$.

Let's assume that the maximum fan-in of every gate is usually 4, and apply this assumption to the graph G. Then maximum degree of every node in G will be 4. However, every X is represented by 2-integer in DBNS will be each of distinguishable sub-graph line in the graph G. Use the handshaking lemma of graph theory in [8]. Therefore, equation (6) is given by

$$n_f * (k*m) * s = 2 * m_e \tag{6}$$

where m_e is the number of edges. Note that the graph G must include all connected paths are represented by number $X \in (0, 2^n - 1)$, thus $m_e > 2^n$. Substitute m_e in (6), we obtain:

$$n_f * (k*m) * s = 2 * 2^n$$

or

$$s > \frac{2^{n+1}}{n_f * k * m}$$

Algorithm to find number of stages and the representation of 2-integer number

Input: number of bits of ADC - n,
number of columns - k, number of rows - m.
Output: number of stages of DBIE - s.

PSEUDO algorithm:

```
Procedure Find_DBInt_Stages(n, k, m)
1. Calculate $S_{temp} = \partial\left(\frac{2^{n+1}}{n_f * k * m}\right)$
2. Naming set $V$ as consisting of integer number $X \in (0, 2^n - 1)$. Find $V' = V - \{X_i \in \text{set } S_i\}$.
3. For every number of stage $i$ $(i < S_{temp})$, find set $U = \{X_j$ has $(i-1)$ additions and every term $\in$ set $S_i\}$. $V'' = V' - U$.
4. If $V''$ is empty, stop and return $S_{\min} = i$. Else increase $i$ by 1 and jump in step 3.
```

From both algorithms, the author can get results, which sufficiently satisfy the proposed requirements with symmetric DBNS-map definition. However, the above algorithms didn't solve the problem of reducing two contiguous cells by rules 3 and 4 in [2], or row and column reduction in [7] to decrease number of additions. With this symmetric DBNS-map, the Constraint algorithm can't yield Addition Ready DBNS(ARDBNS). The ARDBNS means there are not consecutive squares (cells) in the DBNS-map, since the consecutive squares can be reduced into one square by just shifting downward by 1 or right side by 2. If a DSP circuitry needs ARDBNS, the DBNS-map must be asymmetric. Thus, the numbers of encoder output increase more.

To solve this problem, we should append columns or rows from the previously defined symmetric DBNS-map to asymmetric DBNS-map to represent all numbers in $(0, 2^n - 1)$. It is rather simple to know that the maximum number of column in asymmetric DBNS-map will not be greater than n-bit ADC. The following pseudo algorithm will describe to find asymmetric DBNS-map.

PSEUDO algorithm:

```
Procedure Find_Asym DBIE
1. Find number of rows and stages of Asymmetric DBNS-map from symmetric DBNS-map
2. Set $V$ consists of integer number $X \in (0, 2^n - 1)$. Find $V' = V - \{X_i \in \text{set } S_i\}$.
3. For every number of stage $i$ ( $i <$ number of stages of Asymmetric DBNS-map), set $U = \{X_j$ has $(i-1)$ additions, every term $\in$ set $S_i$ and sum of terms is not $\in$ set $S_i\}$. $V'' = V' - U$.
4. If $V''$ is empty or $i$ = number of stages of
```

Asymmetric DBNS-map, stop and exit. Else increase i by 1 and jump in step 3.

The goal of algorithm gets the double integer representation of X, and number of additions. The order of algorithm can take $O(2^n)$ after step 2, and O $(n^*\log(n))$ for 2 steps 3^{rd} and 4^{th}. The order of algorithm is $O(2^n)$ in general, where 2^n is number of DBNS represented integers. The Constrain algorithm gives better results for representing number X in CDBNR, although it is slow as comparing with the Greedy algorithm referred in [1, 2] and [4][16].

To check accuracy of above algorithms, we implemented these algorithms with $n=$ 6 and 8, and show the result in next Section.

DBIE Results

Appling above procedures with 63 numbers(X) from 1 to (2^6-1), we get 4x4 DNBS-map, and the maximum number of additions is 2 in symmetric case, while getting 4x6 DBNS-map of asymmetric case with the same number of addition. However, the Greed algorithm in [1] and [4] shows 3 times addition when the input is 53_{10}. Due to the increasing the input terms, the asymmetric map provided less fan-in problem, even though it needs more chip area and output pins. For example, twelve cases among 63 inputs are required 2 additions in symmetric map, which mean 3 operands (or 3 cells are used in DBNS-map). But, the asymmetric map can process 5 inputs as 2 additions. As shown in table 1, the number of fan-in of DBIE circuits must satisfy electrical design requirements. However, the Greedy algorithm approach can't meet the requirement of fan-in limitation..

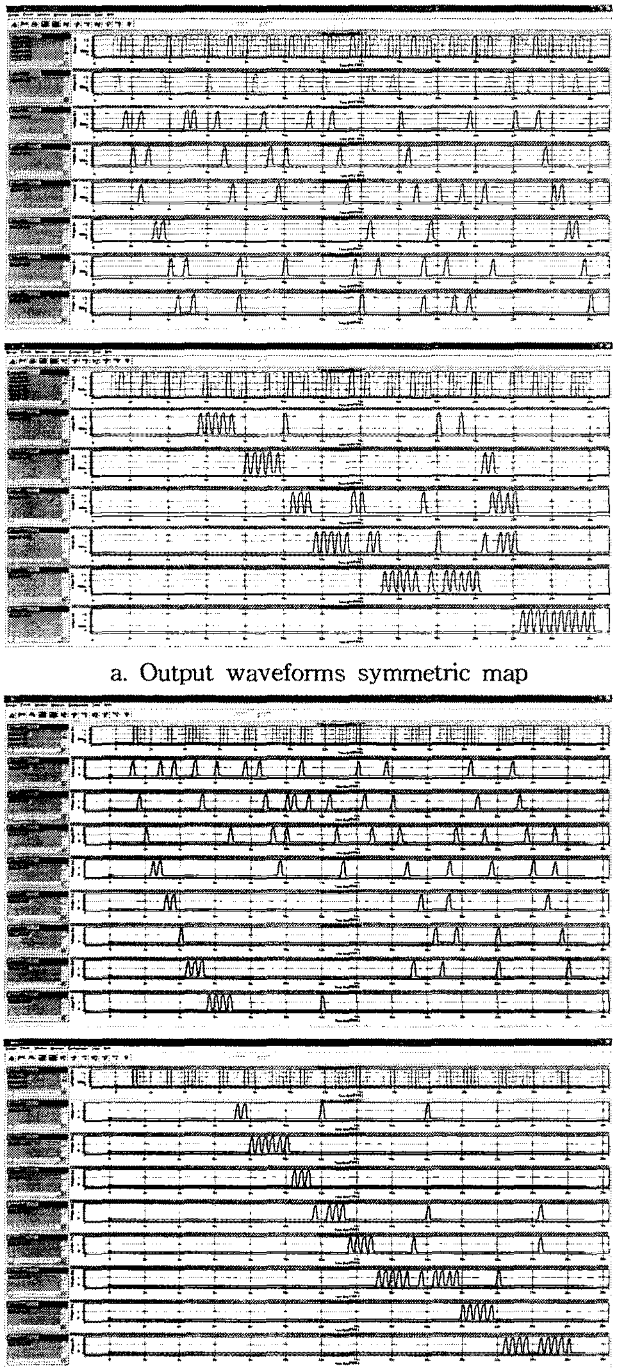
To prove the effectiveness of the Constraint algorithm, the DBIE circuits were designed with the minimum transistor size. The simulation was done by H-spice software. From the simulation results, it is possible that the designed circuit worked properly. In designing DBIE circuit, the 0.18μm CMOS technology is used. The output wave forms of the 6-bit DBIE are shown in Figure 3a and 3b. The figure 3a is the case of symmetric map, while the figure 3b is the case of asymmetric map. As mentioned already, the symmetric map requires only 13 output pins, while the asymmetric map needs 16 output pins. The first rows in each figures indicates the input number form 1 to 63. The left side indicates number 1, and right side means number 63. The correctness of the Constraint algorithm can be verified from the figure 3. For example, the number 32_{10} can be converted to $2^3 3^0 + 2^3 3^1$ (symmetric) or $2^5 3^0$ (asymmetric) respectively by the algorithm. Table 2 is the summary of timing information between the input and output waveform for the Constrain algorithm in ns unit. From the table, the asymmetric algorithm shows better performance than the symmetric case in speed by 23,9%.

Table 1. The fan-in numbers

Greedy		Constrain Symmetric		Constrain Asymmetric	
Cell	Fan IN	Cell	Fan IN	Cell	Fan IN
54	10	54	10	54	9
				48	5
36	18	36	11	36	11
				32	6
27	9	27	12	27	6
24	3	24	10	24	3
18	6	18	7	18	6
				16	4
12	11	12	8	12	5
9	7	9	8	9	7
8	4	8	10	8	5
6	8	6	7	6	5
4	14	4	10	4	9
3	7	3	8	3	11
2	10	2	12	2	11
1	21	1	12	1	12

Table 2. Timing information (unit: ns)

		delay	rising	falling	high	low	period
Symm	Input		0.05	0.05	0.175	0.125	0.4
	Output	0.171	0.09	0.082	0.09	0.114	0.376
Asym	Input		0.05	0.05	0.175	0.125	0.4
	Output	0.13	0.087	0.063	0.127	0.096	0.373



a. Output waveforms symmetric map

b. Output waveforms asymmetric map

Fig. 3 H-spice output waveforms of constraint algorithm

In addition, the DBIE circuits improve the encoding speed by 16.64% in case of 6 bits, and by 15.89% in case of 8 bits respectively compared with the Fat Tree Encoders. The results of the FAT Encoder are simulated according to the reference [10]. Besides, it will be easy to layout the DBIE circuits because the numbers of transistors were less than the Fat Tree encoders about 17.45% for ADC 6 bits and 8.86% for ADC 8 bits. All comparative results of Double Base Integer and FAT encoder are shown in table 3. The DBIE circuits are more efficient in ADC when embedding DSP circuitry into SoC architecture.

Table 3. Hspice results with 0.18um CMOS.

HSpice results is simulated by technology of CMOS 0.18um				
ADC	8bits		6bits	
encoder	Asymmetric DBIE	FAT	Asymmetric DBIE	FAT
max speed (Ghz)	5.26	4.54	6.66	5.71
no. of transistor	2234	2432	424	498
max power (mW)	26.05	19.11	17.26	12.44
output bits	27	8	16	6

III. Conclusions

In this paper, a new algorithm called Constraint algorithm was presented to design Double Base Integer Encoder for TIQ based ADC. With the Constraint algorithm, it was possible to design fast ADC, because the algorithm reduced addition stages in DSP. Of course, the direct comparison between two algorithms of Greedy and Constraint may be unreasonable, since the Greedy algorithm was not developed for encoder circuits. In this paper, the comparison is just to show that the Greedy's approach of minimum representation is not appropriate in ADC design. More important design factors are power consumption and speed now. The new circuits took more power consumption by 26.64% in 8 bits, and 29.26% in 6 bits. This drawback results from the increasing of input number of each logic gate, and the gate's switching speed[15]. This problem will be researched next by considering more parameters that reduce switching frequency. To improve this algorithm later, we can apply the depth first search with some parameters to find more effective double base number system. Furthermore, applying asynchronous digital circuit design for fast processing will be next goal to detect and correct when errors occur in the front stages. Finally, new DSP arithmetic circuits, which support the DBNS arithmetic operation, will be merged into ADC circuits as future research.

REFERENCES

[1] Vassil S. Dimitrov and Graham A. Jullien, "A New Number Representation with applications", IEEE

Circuits and Systems Magazine, 2003.

[2] Vassil S. Dimitrov and Graham A. Jullien, and W. C. Miller, "Theory and applications of the double-base number system", IEEE Transactions on Computers, vol. 48, pp. 1098 - 1106, 1999.

[3] G. Gilbert and J. M. P. Langlois, "Multipath greedy algorithm for canonical representation of numbers in the double base number system", IEEE-NEWCAS Conference, pp. 39-42, 19 - 22, June 2005.

[4] V. Berthe, Laurent Imbert, and Graham A. Jullien, "More on Converting Numbers to the Double-Base Number System", Research Report LIRMM - 0403 1, Montpellier France, October 2004.

[5] Y. Ibrahim, W. C. Miller, Graham A. Jullien, and Vassil S. Dimitrov, "DBNS addition using cellular neural networks", IEEE International Symposium on Circuits and Systems, vol. 4, pp. 3914 - 3917, 23 - 26 May 2005.

[6] M. Pankaala, A. Paasio, and M. Laiho, "Implementation alternatives of a DBNS adder", 9th International Workshop on Cellular Neural Networks and Their Applications, pp. 138 - 141, 28-30 May 2005.

[7] Kriangyut Wangjitman and Athasit Surarerks, "Addition transducer for double base number system", ISCIT'06, pp. 994-999, Sep 2006.

[8] Reinhard Diestel, Graph Theory, Springer-Verlag New York 2000.

[9] Jincheol Yoo, ''A TIQ Based CMOS Flash A/D Converter for SoC Applications", PhD Thesis Department of Computer Science and Engineering of The Pennsylvania State University, 2003.

[10] 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, pp. 87-90, Aug 2002.

[11] Jincheol Yoo, Daegyu Lee, Kyusun Choi and Jongsoo Kim, "A Power and Resolution Adaptive Flash Analog-to-Digital Converter", ISLPED'02, pp. 233-236, August 2002.

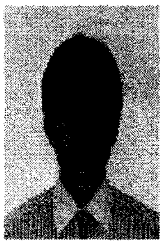
[12] Daegyu Lee, Jincheol Yoo, Kyusun Choi, "Design Method and Automation of Comparator Generation for Flash A/D Converter", IEEE International Symposium on Quality Electronic Design, pp 138-142, March 2002.

[13] D.P. Dimitrov and T.K. Vasileva, "8-bit semi-Flash A/D Converter", Proceedings of the International Conference Mixed Design of Integrated Circuits and System, pp. 171-174, June 2006.

[14] Chia-Chun Tsai, Kai-Wei Hing, Yuh-Shyan Hwang, Wen-Ta Lee and Trong-Yen Lee, "New Power Saving Design Method for CMOS Flash ADC", The 47th IEEE International Midwest Symposium on Circuit and System, pp. Ⅲ-371 - Ⅲ-374, 2004.

[15] Neil H.E. Weste and David Harris, "CMOS VLSI design-A circuits and systems perspective", 3rd edition, Addison-Wesley, 2004.

[16] Jongsoo Kim, Man-Ho Kim, Eun-Hwa Jang, "A New Flash A/D Converter Adopting Double Base Number System", KISPS Journal of Signal Processing and Systems, vol. 9, no. 1, pp. 54-61, 2008.



Minh Son Nguyen

He received B.E. and M.E. degree in computer engineering from the HCMcity University of Technology, Vietnam, in 2001 and 2005 respectively, and is working toward the Ph. D. degree in Electrical Engineering at the University of Ulsan. Ulsan, Korea. He is studying on the fields of VLSI, and DSP system design.

Soo A Yeo

He is working toward a Bachelor degree in the School of Electral Engineering, University of Ulsam, Ulsan, Korea. He received a scholarship from the General Electric Foundation Scholarship Program.

Man-Ho Kim

He received his Ir degree from Delft University of Technology in Holland (1986-1989) and his Ph.D. from Leicester University in the UK (1992-1995). His research interests are the CCD/CMOS image sensor and optical laser transmission system

Jongsoo Kim

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