Efficient Test Data Compression and Low Power Scan Testing in SoCs

Jun-Mo Jung and Jong-Wha Chong

Testing time and power consumption during the testing of SoCs are becoming increasingly important with an increasing volume of test data in intellectual property cores in SoCs. This paper presents a new algorithm to reduce the scan-in power and test data volume using a modified scan latch reordering algorithm. We apply a scan latch reordering technique to minimize the column hamming distance in scan vectors. During scan latch reordering, the don't-care inputs in the scan vectors are assigned for low power and high compression. Experimental results for ISCAS 89 benchmark circuits show that reduced test data and low power scan testing can be achieved in all cases.

Keywords: SoC test, scan test, test data compression, low power scan test.

I. Introduction

A system-on-a-chip (SoC) typically consists of user-defined functional blocks and intellectual property (IP) cores, such as SRAM memory and CPU cores. This core-based SoC design greatly improves design productivity and speeds up the time-to-market.

As the complexity and speed of digital circuits increases, the importance of testing circuits more efficiently also increases [1]. The testing of SoCs is especially important.

However, there are two types of problems in SoC testing: the testing time and power dissipation during test application. One of the problems in testing SoCs is the large amount of test data that must be transferred between the tester and the chip coupled with the limited input/output channel capacity, speed, and data memory of the automatic test equipment.

Test data stored in the tester are given to the relevant IP core by demultiplexing in an SoC, and the test response must be observed by the SoC's output. Testing time is determined by the data transmission speed and the transmission width between the tester memory and the SoC.

One solution to this problem is to use a built-in self-test (BIST). However, most IP cores that are available from ASIC vendors have to be modified considerably to incorporate the BIST because the BIST is not normally incorporated into the IP core design.

Another solution to the problem of a long testing time is the test data compression method. The test data for IP cores are compressed to much smaller test data, which are stored in the automatic test equipment memory. The decoder in IP cores decodes the compressed data to obtain the original test data during the test application. Some reports have presented methods for test data compression for non-scan circuits [2] and

Manuscript received Jan. 15, 2003; revised June 9, 2003.

This work was supported by Kimpo College's Research Fund.

Jun-Mo Jung (Phone: +82 31 999 4143, email:jmjung@kimpo.ac.kr) is with Kimpo College, Gimpo, Gyeonggi-do, Korea.

Jong-Wha Chong (email: jchong@hanyang.ac.kr) is with Hanyang University, Seoul, Korea.

for full-scan circuits [3].

The successive test patterns in test data are generally highly correlated and differ in only a small number of bits. The method proposed in [4] exploited this characteristic. The difference in test data between successive test patterns was compressed using run-length codes (variable-to-fixed-length codes), but it was less efficient than the more general variable-to-variable-length codes [5], [6].

Chandra and Chakrabarty presented another method that used Golomb codes, which mapped variable-length runs of 0s in the difference vector to the variable-length code words, and this achieved greater compression [7]. In [8], they presented a new compression scheme based on reordering the test vectors in such a way that enables the generation of geometric shapes that can be highly compressed via perfect lossless compression.

Power consumption is another problem in SoC testing since the power of a digital system is considerably higher in the test mode than in the normal mode. The reason is that test patterns cause the switching of as many nodes as possible while the normal mode activates only a few modules at a time. Special care must be taken to ensure that the power consumption is not exceeded during test application.

A new automatic test pattern generation tool [9] was proposed to overcome the low correlation between consecutive test vectors during test application. A mixed solution based on the reseeding scheme and test vector inhibiting techniques was proposed to reduce the energy consumption of BIST sessions [10]. The primitive polynomial was selected to achieve the lowest switching activity in the circuit. The sub-sequences of linear feedback shift register were also inhibited during test application. The techniques for minimizing power dissipation in scan and combinational circuits were proposed to reduce power dissipation for full-scan circuits [11]. The scan-in power was reduced using test vector ordering and scan latch reordering (SLR).

Other studies proposed methods for low power mapping and test compression methods for unspecified scan vectors [7], [12] and a new compression and low power consumption technique for unspecified scan vectors [13]. The latter technique mapped the don't-care input for low power and performed the SLR. Ghosh et al. proposed SLR techniques for minimizing power dissipation, which were also capable of reducing the area overhead of the circuits [14]. The greedy algorithm found the best order for overhead minimization and power minimization.

This paper presents efficient low power and higher compression methods for sequential circuits with full-scan circuits. In previous methods, the don't-care inputs in the unspecified scan vectors were assigned to 0 (zero mapping) [8] or to the adjacent input value for low power and higher compression, minimum transition count mapping (MTC mapping) [13], [14]. After mapping, the various techniques for high compression and low power, including SLR, were applied to the mapped scan vectors. However, in our investigation, we were able to effectively reduce power consumption during scan-in and test data volume by assigning the neighboring input value to don't-care inputs during scan latch reordering. The existing studies applied scan latch reordering after the don't-care mapping. However, we applied the don't-care mapping during scan latch reordering which resulted in more efficient power reduction and reduced test data volume. Moreover, the scan latch was ordered to minimize the hamming distance between adjacent scan cells.

This paper is organized as follows. In section II, we describe the power consumption model for digital CMOS circuits. Section III presents a new scan latch reordering method that considers low power mapping. Section IV discusses the results of the ISCAS89 benchmark circuits, and section V presents the conclusion.

II. Power Consumption Model

1. Power Consumption Model of Digital CMOS Circuits

Power dissipation in CMOS circuits can be classified as static, short circuit, leakage, and dynamic power dissipation. The static power dissipation is negligible. Short circuit and leakage power dissipation contribute up to 20% of the total power dissipation. The remaining 80% is attributed to dynamic power dissipation caused by switching of gate outputs. If the gate is part of a synchronous digital circuit controlled by a global clock, it follows that the dynamic power P_d required to charge and discharge the output capacitance load of every gate is:

$$P_d = 0.5 C_{load} V_{dd}^2 F_c N_g, \tag{1}$$

where C_{load} is the load capacitance, V_{dd} is the supply voltage, F_c is the global clock frequency, and N_g is the total number of gate output transitions $(0 \rightarrow 1 \text{ or } 1 \rightarrow 0)$. These transitions are the major factors of power dissipation. The power dissipation during full-scan testing is due to the dynamic power caused by transitions that occurred when the scan vectors were shifted in the scan chain.

2. Scan-in Power Model

Power consumption in testing a sequential circuit with a single scan chain includes the following: scan-in power consumed during the scan-in operations of the scan vectors, scan-out power consumed during the scan-out operations of the

test response, and power consumed in the combinational logic of the sequential circuit. It is difficult to estimate the scan-out power directly from the scan vector set since the test response must be determined from the function of the core being tested. As in [15], therefore, we consider the scan-in power only and measure it in terms of the weighted transition metric (WTM).

The scan-in power depends not only on the number of transitions in it but also on their relative positions. For example, consider the scan vector $S_1S_2S_3S_4S_5$ = 10101 with a scan length of 5. If the leftmost bit (S_1) is first shifted in the scan chain, the transition of (S_1, S_2) causes four (scan length–1) transitions during the scan-in. The transition of (S_2, S_3) causes three (scan length–2) transitions. The transition of (S_j, S_{j+1}), therefore, causes (scan length –j) transitions.

Let each scan vector SV with scan length K be $S_1\ S_2 \ldots\ S_K$. The scan-in power for SV, P_{SV}, is given by:

$$P_{SV} = \mathrm{WTM}(SV) = \sum_{j=1}^{K-1} (S_j \oplus S_{j+1})(K-j). \quad (2)$$

The term $S_j \oplus S_{j+1}$ is 1 if the transition occurs between S_j and S_{j+1}. Therefore, the transition that occurred at the j-th bit position caused as many as (K–j) transitions during the scan-in to the scan chain.

Assuming that the set of the scan vectors used for testing is $SV_{set} = SV_1, SV_2, ..., SV_n$, the power consumed during the scan-in of SV_{set} is

$$\mathrm{WTM}(SV_{set}) = \sum_{i=1}^{n} \mathrm{WTM}(SV_i). \quad (3)$$

The average power consumption and peak power consumption can thus be respectively as

$$P_{avg} = \mathrm{WTM}(SV_{set})/n \quad \text{and}$$
$$P_{peak} = \mathrm{Maximum}(\mathrm{WTM}(SV_i)) \quad \text{for all scan vectors.} \quad (4)$$

III. Scan Latch Reordering Considering Low Power Mapping

Previously proposed methods applied low power and compressing methods that first assigned binary logic values to don't-care inputs within each scan vector. This paper proposes a new method that does not first assign a value to don't-care input but assigns the value to don't-care input while simultaneously applying scan latch reordering.

Scan latch reordering for low power reorders the position of the input of the scan vector. The compression ratio and power reduction ratio improved when the number of runs of 0s or runs of 1s increased inside the scan vector. Therefore, the scan latch should be reordered to enable the neighboring scan input value to be the same.

This paper used hamming distance to calculate the similarities of the neighboring scan input value. In addition, the don't-care input was assigned to minimize the hamming distance during scan latch ordering, not prior to scan cell ordering. Therefore, this paper was able to obtain better compression ratio and power reduction ratio than previous methods.

1. The Cost Function for Scan Latch Reordering

The set of the scan vectors used for testing, SV_{set}, can be expressed by a two-dimensional array, $SV[r][c]$, where r is the number of the scan vector and c the input number of the scan vector. For each element of the array, $SV[i][j]$ means the j-th scan input of the i-th scan vector.

$\mathrm{WTM}(SV_{set})$ is described in (5) below.

$$\begin{aligned}\mathrm{WTM}(SV_{set}) &= \sum_{i=1}^{r}\sum_{j=1}^{c-1}(SV[i][j] \oplus SV[i][j+1])(c-j) \\ &= \sum_{j=1}^{c-1}\sum_{i=1}^{r}(SV[i][j] \oplus SV[i][j+1])(c-j) \\ &= \sum_{j=1}^{c-1} HD\_col(j, j+1)(c-j), \end{aligned} \quad (5)$$

where the $HD\_col(j,j+1)$ is the column hamming distance between the j-th column and the (j+1)th column for all rows. For example, assuming the SV_{set} as in Fig. 1, the SV_{set} has four scan vectors with three bits (scan chain length = 3).

$$SV_{set} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Fig. 1. Example of an HD_col.

In this example, the column hamming distances are as follows: $HD\_col(1,2)=3$, $HD\_col(1,3)=1$, and $HD\_col(2,3)=4$.

For $\mathrm{WTM}(SV_{set})$ to be small, the HD_col must be small. The scan latch reordering means the reordering of the column positions in the scan vectors. It is thus advantageous to use the column hamming distance as the cost function for scan latch reordering.

2. Proposed Algorithm

A. Calculation of the Column Hamming Distance

HD_col(j,j+1) is the hamming distance between columns j and j+1: we calculate it considering the don't-care input.

Given *SV*[ ][ ] ∈ {0,1,*X*(don't-care input)} for arbitrary i and j, the hamming distance between *SV*[i][j] and *SV*[i][j+1] is 1 only on the condition that (*SV*[i][j], *SV*[i][j+1]) is (0,1) or (1,0). In the case of (0,*X*) or (1,*X*), the hamming distance becomes 0 because the value of *SV*[i][j] can be assigned to *X*, which exists in *SV*[i][j+1].

Because scan latch reordering is an NP-hard problem, it is very difficult to find a near optimal solution. We propose the heuristic algorithm below.

- Define *SV*[*][j] as the j-th column in the two-dimensional array of the scan vectors.
- Initialize the arbitrary *SV*[*][j] to *SV*[*][1] and assign 0 to all *X*s in *SV*[*][1].
- Calculate the column hamming distance (*HD_col*) between *SV*[*][1] and *SV*[*][j] (j: from 2 to c). The value of j that minimizes the column hamming distance can be obtained.
- Assuming that the column with the minimum *HD_col* is k, *SV*[*][k] can be exchanged with *SV*[*][2].
- Allocate the same values for all the don't-care inputs in *SV*[*][2] as those in the same row in *SV*[*][1] (mapping for lower power and higher compression).
- Repeat the above process until j of *SV*[*][j] becomes c–1.
- Compress *SV*[r][c].

The proposed algorithm is described by pseudo-code in Fig. 2 below.

```
SV_set = SV[r][c];
Initialize SV[*][1];
For (j=1; j<c; j++)
{
for (k=j+1; k<c+1; k++)
HD_col(j,k); /* Calculate HD_col */
Search the index k with the minimum HD_col;
Exchange column j+1 with column k;
Assign the Xs of SV[*][j+1] in the value of SV[*][j];
}
Compress (SV_set);

Function HD_col(j,k)
{HD_sum(k) = 0;
for (i =1; i<r+1; i++)
if ( (SV[i][j] == 0 && SV[i][k] == 1) |,
(SV[i][j] == 1 && SV[i][k]== 0 ) )
HD_sum(k)++
}
```

Fig. 2. Proposed algorithm.

We will explain the proposed algorithm using an example (Fig. 3).

Consider the SV_{set} as in Fig. 3(a).

First, search the columns that have a minimum column hamming distance from the first column, $[010]^T$.

Because the 3rd column ($[XXX]^T$) has a minimum hamming distance, the 2nd column can be exchanged with the 3rd column, as in Fig. 3(b). Assign the same value to the don't-care input of the 2nd column as that in the same row in the first column, as in Fig. 3(c). Find the 3rd column with the minimum hamming distance from the 2nd column in Fig. 3(c). The *HD_col*(2,3) and the *HD_col*(2,4) are 3 and 2, respectively. The 3rd column is $[X01]^T$, as in Fig. 3(d). The don't-cares in the 3rd column can be assigned as in Fig. 3(e).

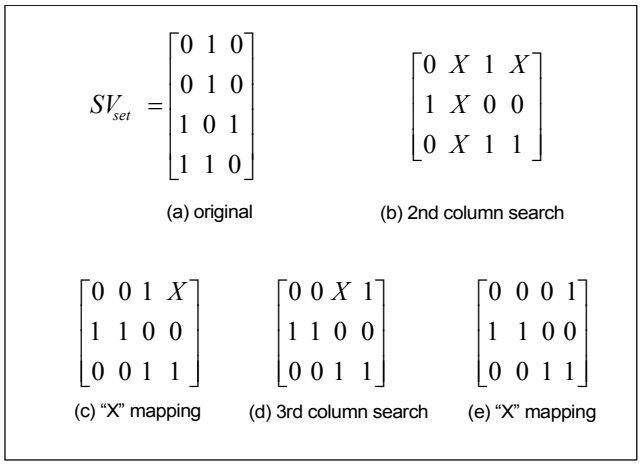

Fig. 3. Example of the proposed method.

B. Compression Method

In Golomb codes with the group size m, the number of encoding bits of the data with runs of 0s of length A is $\left\lfloor \frac{A}{m} \right\rfloor + \log_2 m + 1$. For example, in case of data with A=100 and m=4, the number of encoding bits by Golomb codes is 28 bits. However, if we encode it using binary codes, the number of encoding bits is 8 bits. In this case, we can reduce 20 bits of codeword. When the run length is larger than any threshold value, it can be encoded using binary codes with fixed bits. Therefore, we can compress test data more efficiently using prefix as described below. The prefix is added to codeword according to the type of run length.

- Prefix (00): Golomb encoding of 0's run
- Prefix(01): Golomb encoding of 1's run
- Prefix(10): fixed-length binary encoding of 0's run
- Prefix(11): fixed-length binary encoding of 1's run

The additional overheads of this decoder are the prefix (2bits) decoder and fixed-length bit counter. It is very small.

The pseudo codes for compress( ) function in proposed algorithm are shown in Fig. 4.

```
Function Compress(test data)
{
calculate_runs(run-length);
For(each run-length)
{
If (run-length < k) /* k may be 20 - 25 */
        /* Encodes run-length by golomb; */
        Codeword = ("00" or "01") & (golomb codes);
        /* & means the concatenation */
Else if (run-length > valule) /* value should be 2^p */
        { Do while(run-length > value)
        {/* Encodes it by fixed-length (p bit) binary
        codes */
        Codeword = ("10" or "11") & (fixed-length
        binary codes);
        run-length = run-length - value;
        }
        if (run-length < k)
        {/* Encodes run-length by golomb; */
        Codeword = ("00" or "01") & (golomb codes);}
        Else
        {/* Encodes it by fixed-length binary codes */
        Codeword = ("10" or "11") & (fixed-length
        binary codes);}
        }
}
}
```

Fig. 4. Pseudo codes of compress( ) function.

IV. Experimental Results

In this section, we evaluate the effect of the proposed method on test data volume and power consumption during scan testing for ISCAS89 benchmark circuits. The experiments were conducted on a Sum Ultra 10 workstation.

We consider sequential circuits with full-scan. For each full-scan circuit, we assumed a single scan chain. We used partially-specified scan vector sets generated by the MINTEST automatic test pattern generation program with dynamic compaction [16]. Table 1 presents the experimental results of the scan vector compression. We use the 9 bits fixed-length binary codes in compress sub function. In the Compression Ratios column, sub column Golomb coding shows the compression ratios by Golomb coding after mapping only 0 to the don't-care inputs [7]. The sub column MTC map & SLR method is the method in [13]. The sub column Proposed is our method. The compression ratios obtained were computed as follows:

$$Comp.Ratio = \frac{\#Original\ Bits - \#Compressed\ Bits}{\#Original\ Bits} \times 100$$

Table 1. Experimental results on test data compression.

Circuits	Compression ratios (%)		
	Golomb coding (zero mapping)	MTC map & SLR	Proposed
S5378	37.13	46.10	59.20
S9234	45.27	47.20	65.74
S13207	79.75	81.07	90.15
S15850	62.83	64.59	86.10
S38417	28.38	58.56	85.17
S38584	57.17	63.41	79.15
Average	51.76	60.16	77.58

As Table 1 shows, the compression ratio in Golomb coding for S5378 was about 37.13%, about 46% for the MTC & SLR method and about 59% for the proposed method. On average, the compression ratio was about 51.76% of the compression ratio in Golomb, about 60.16% in the MTC & SLR method, and 77.58% in the proposed method. The proposed method showed about a 26% better compression ratio than Golomb and about 17% better than the MTC map & SLR method.

Table 2 shows the reduction ratios for peak power. The reduction ratio in the proposed method was reduced to 47% less than that in zero mapping and 13% less than that in the MTC map & SLR method for S5378. The average reduction ratio was about 21% in zero mapping, about 66% in the MTC & SLR method, and about 81.68% in the proposed method. The experimental results show that the proposed method has a better reduction ratio than the previous methods.

In Table 3, we report the results of the reduction ratios for average power. While the reduction ratio was about 89.8% in zero mapping and 96% in MTC & SLR for S13207, the proposed method gives a good result of 99.3%. The power reduction rates on average were about 74% for zero mapping, 88% for MTC & SLR, and about 96% for the proposed method.

As the experimental results confirm, the proposed method has higher compression ratios and power reduction ratios.

Table 2. Reduction ratios for peak power.

Circuit	zero mapping		MTC map & SLR		Proposed	
	Peak	%	Peak	%	Peak	%
S5378	10,127	24.55	5,556	58.61	3,849	71.3
S9234	12,994	25.72	7,400	57.70	5,331	69.5
S13207	101,127	25.42	35,486	73.83	12,585	90.7
S15850	81,832	18.35	33,207	66.87	13,479	86.6
S38417	505,295	26.10	181,436	73.46	54,034	92.1
S38584	531,321	7.21	187,379	67.28	111,466	80.1
Average		21.23		66.29		81.68

Table 3. Reduction ratios for average power.

Circuit	zero mapping		MTC map & SLR		Proposed	
	Avg	%	Avg	%	Avg	%
S5378	3,336	68.89	1,896	82.89	860	92.2
S9234	5,692	61.09	2,515	82.81	1,021	93.0
S13207	12,416	89.82	5,188	95.75	766	99.3
S15850	20,742	77.18	8,945	90.16	2,162	97.7
S38417	172,665	71.31	85,913	85.72	15,960	97.3
S38584	136,634	74.50	49,473	90.77	18,235	96.6
Average		73.79		88.02		96.02

V. Conclusion

As the number of IP cores increases in an SoC, the test data volume and testing time also increase. As a result, the testing and chip costs go up and productivity goes down. Power consumption in the test mode is much larger than that in the normal mode and causes damage to the chip due to excessive power consumption.

This paper proposes a new algorithm that has efficient low power and a high compression ratio for unspecified scan vectors. It applies mapping don't-care input and scan latch reordering at the same time using the column hamming distance as the cost function. The proposed method shows extremely high compression and power reduction compared to previous methods.

Routing complexity should also be considered for much larger scan cells and more complex systems. Further studies will be conducted along these lines.

References

[1] S. Kang, B. Underwood, W. Law, and H. Konuk, "Efficient Path Delay Test Generation for Custom Designs," *ETRI J.*, vol. 23, no. 3, Sept. 2001, pp. 138-149.

[2] V. Iyengar, K. Charabarty, and B.T. Murray, "Built-in Self Testing of Sequential Circuits Using Precomputed Test Sets," *Proc. of IEEE VLSI Test Symposium*, May 1998, pp. 418-423.

[3] V. Iyengar, K. Charabarty, and B.T. Murray, "Deterministic Built-in Pattern Generation for Sequential Circuits," *J. Electron. Tet. Theory Applicat.*, vol. 15, Aug/Oct. 1999, pp. 97-115.

[4] A. Jas and N.A. Touba, "Test Vector Decompression via Cyclical Scan Chains and its Application to Testing Core-Based Design," *Proc. of Int'l Test Conf.*, Nov. 1998, pp. 458-464.

[5] S.W. Golomb, "Run-Length Encoding," *IEEE Trans. Inform. Theory*, vol. IT-12, 1966, pp. 399-401.

[6] H. Kobayashi and L.R. Bahl, "Image Data Compression by Predictive Coding, Part I: Prediction Algorithm," *IBM J. of Research & Development*, vol. 18, 1974, p. 164.

[7] A. Chandra and K. Chakrabarty, "System-on-a-Chip Test Data Compression and Decompression Architectures Based on Golomb Codes," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Syst.*, vol. 20, no. 3, Mar. 2001, pp. 355-368.

[8] S. al Zahir, A.El-Maleh, and E. Khan, "An Efficient Test Vector Compression Technique Based on Geometric Shapes," *Proc. of IEEE Int'l Conf. on Electronics, Circuits and Systems* (ICECS 2001), 2001, pp. 1561-1564.

[9] S. Wang and S.K. Gupta, "ATPG for Heat Dissipation Minimization during Test Application," *IEEE Trans. Comput.*, 1998, pp. 256-262.

[10] P. Girad, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A Test Vector Inhibiting Technique for Low Energy BIST Design," *Proc. of VLSI Test Symposium*, 1999, pp. 407-412.

[11] V. Dabhokar, S. Chakravarty, I. Pomeranz, and S.M. Reddy, "Techniques for Minimizing Power Dissipation in Scan and Combinational Circuits during Test Application," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Syst.*, vol. 17, no. 12, 1998, pp. 1325-1333.

[12] A. Chandra and K. Chakrabarty, "Combining Low-Power Scan Testing and Test Data Compression for System-on-a-Chip," *Proc. of IEEE/ACM Design Automation Conf.* (DAC), June 2001, pp. 166-169.

[13] P. Rosinger, P.T. Gonciari, B.M. Al-Hashimi, and N.Nicolici, "Simultaneously Reduction in Volume of Test Data and Power Dissipation for Systems-on-a-Chip," *Electronics Lett.*, vol. 37, no. 24, Nov. 2001, pp. 1434-1436.

[14] S. Ghosh, S.Basu, and N.A. Touba, "Joint Minimization of Power and Area in Scan Testing by Scan Cell Reordering," *Proc. of IEEE Symposium on VLSI* (ISVLSI), 2003, pp 246-249.

[15] R. Sankaralingam, R.R. Oruganti, and N.A. Touba, "Static Compaction Techniques to Control Scan Vector Power Dissipation," *Proc. of IEEE VLSI Test Symposium*, 2000, pp. 35-40.

[16] I. Hamzaoglu and J.H. Patel, "Test Set Compaction Algorithms

for Combinational Circuits," *Proc. of Int'l Conf. Computer-Aided Design*, Nov. 1998, pp. 283-289.

Jun-Mo Jung received the BS and the MS degrees in electronics engineering from Hanyang University in Seoul, Korea, in 1987 and 1989. From March 1989 to June 1996, he worked at Samsung Electronics. Since July 1996, he has been an Associate Professor at Kimpo College. His current research interests include low power design, ASIC design, VLSI testing, and DFT.

Jong-Wha Chong received the BS and MS degrees in electronic engineering from Hanyang University in Seoul, Korea, in 1975 and 1977 and the PhD degree from Waseda University, Japan in 1981 in electronic communication engineering. From 1979 to 1980, he was with NEC Central Laboratory. Since 1981, he has been a Professor at Hanyang University. His current research interests are in VLSI design for the digital signal and image processing, video compression, high-speed wireless LAN, and digital communication systems.