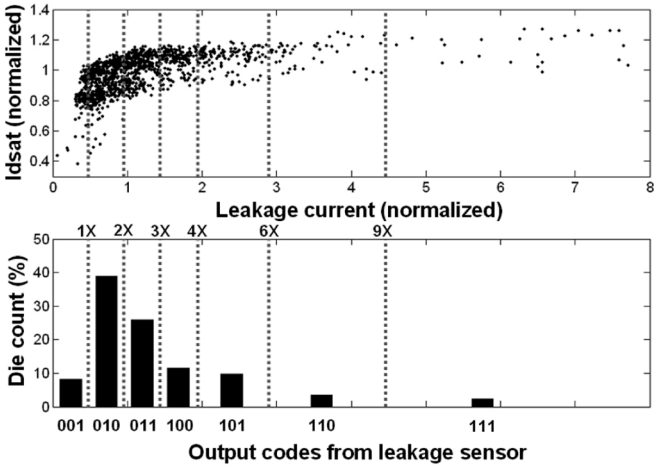
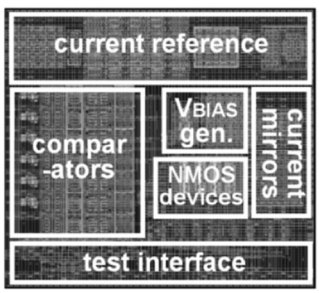


Fig. 9. Leakage binning results of 6-channel leakage sensor.



Resolution	7 levels
Technology	90nm CMOS
V_{DD}	1.2V
Power consumption	0.66 mW @80C°
Dimensions	83 X 73 μm^2

Fig. 10. Layout and key features of the leakage sensor testchip.

curve in Fig. 7 increases. In other words, the sensitivity of drain voltage to leakage variation degrades. Accordingly, the sensing resolution in channels with high bias currents is lower than the others. A nonuniform mirroring ratio of $\{1\times, 2\times, 3\times, 4\times, 6\times, 9\times\}$ is used rather than a uniform mirroring ratio of $\{1\times, 2\times, 3\times, 4\times, 5\times, 6\times\}$ to ensure a target sensing resolution at high bias currents. This also guarantees that the monotonically increasing $V_{\rm SEN}$ across the six channels do not swap levels under worst case WID parameter variations. Since the proposed current mirror based design shares the bias circuits $(I_{\rm REF}, V_{\rm BIAS})$ across each sensing channel, the leakage-sensor circuit becomes simple and dense compared to [8], where separate $V_{\rm BIAS}$ generators are required for each channel. Outputs of the comparators should be a thermometer code with a single transition. Bubbles (lone "1" within a string of "0"s or a "0" within a string of "1"s) due to WID variation or comparator metastability can be eliminated using NAND3 gates which require two "1"s immediately above a "0" in order to determine the transition point [10]. Additional combinational logics remap the one cold encoded bit array V1–V6 into a desired binary code sequence, $\rm OUT[2:0]$. Based on nMOS and pMOS $I_{\rm on}$, $I_{\rm off}$ measurements, the designed 1.2 V, 0.66 mW, 0.006 mm^2 leakage sensor shows excellent leakage binning results in 90-nm CMOS (Figs. 9 and 10).

REFERENCES

[1] C. H. Kim, K. Roy, S. Hsu, A. Alvandpour, R. Krishnamurthy, and S. Borkar, "A process variation compensating technique for sub-90 nm dynamic circuits," in *Proc. Symp. VLSI Circuits*, 2003, pp. 205–206.

[2] A. Alvandpour, R. Krishnamurthy, K. Soumyanath, and S. Borkar, "A conditional keeper technique for sub-0.13 μm wide dynamic gates," in *Proc. Symp. VLSI Circuits*, 2001, pp. 29–30.

[3] C. H. Kim, K. Roy, S. Hsu, R. Krishnamurthy, and S. Borkar, "An on-die CMOS leakage current sensor for measuring process variation in sub-90 nm generations," in *Proc. Symp. VLSI Circuits*, 2004, pp. 250–251.

[4] S. Thompson, "A 90 nm logic technology featuring 50 nm strained silicon channel transistors, 7 layer of Cu interconnects, low *k* ILD, and 1 μm^2 SRAM cell," in *Proc. Int. Electron Devices Meeting*, 2002, pp. 61–64.

[5] A. Dancy and A. Chandrakasan, "Ultra low power control circuits for PWM converters," in *Proc. IEEE Power Electron. Specialists Conf.*, 1997, pp. 21–27.

[6] S. Narendra, D. Klowden, and D. Vivek, "Sub-1 V process-compensated MOS current generation without voltage reference," in *Proc. Symp. VLSI Circuits*, 2001, pp. 143–144.

[7] E. Vittoz and O. Neyroud, "A low-voltage CMOS bandgap reference," *IEEE J. Solid-State Circuits*, vol. SC-14, no. 3, pp. 573–577, Jun. 1979.

[8] M. M. Griffin, J. Zerbe, G. Tsang, M. Ching, and C. L. Portmann, "A process-independent, 800-MB/s, DRAM byte-wide interface featuring command interleaving and concurrent memory operation," *IEEE J. Solid-State Circuits*, vol. 33, no. 11, pp. 1741–1751, Nov. 1998.

[9] T. Kuroda, "A 0.9 V 150 MHz 10 mW 4 $\rm mm^2$ 2-D discrete cosine transform core processor with variable-threshold-voltage scheme," in *Proc. Int. Solid-State Circuits Conf.*, 1996, vol. 437, pp. 166–167.

[10] M. Steyaert, R. Roovers, and J. Craninckx, "A 100 MHz 8 bit CMOS interpolating A/D converter," in *Proc. Custom Integr. Circuits Conf.*, May 1993, pp. 28.1.1–28.1.4.

MICRO: A New Hybrid Test Data Compression/ Decompression Scheme

Sunghoon Chun, YongJoon Kim, Jung-Been Im, and Sungho Kang

***Abstract*—To overcome the limitation of the automatic test equipment (ATE), test data compression/decompression schemes become a more important issue of testing for a system-on-chip (SoC). In order to alleviate the limitation of previous works, a new hybrid test data compression/decompression scheme for an SoC is developed. The new scheme is based on analyzing the factors that influence test parameters: compression ratio and hardware overhead. To improve compression ratio, the proposed scheme, called the Modified Input reduction and CompRessing One block (MICRO), uses the modified input reduction, the one block compression, a novel mapping, and reordering algorithms. Unlike previous approaches using the cyclic scan register architecture, the proposed scheme is to compress original test data and to decompress the compressed test data without the cyclic scan register architecture. Therefore, the proposed scheme leads to high-compression ratio with low-hardware overhead. Experimental results on ISCAS '89 and ITC '99 benchmark circuits prove the efficiency of the new method.**

***Index Terms*—Design for testability, system-on-chip (SoC) test, test data compression, test data decompression.**

I. INTRODUCTION

As the complexity of VLSI circuits increases, it is more important to test VLSI circuits completely. Especially, today's large and complex VLSI circuits in system–on-chip (SoC) environments, need an enormous amount of test data. When SoCs are tested, such test data is transferred to the circuit under test (CUT) from an automatic test equipment

Manuscript received September 21, 2004; revised November 14, 2005. This work was supported by the Korea Science and Engineering Foundation Grant R01-2006-000-11038-0 funded by the Korea government (MOST).

The authors are with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul 120-749, Korea (e-mail: shchun@yonsei.ac.kr; yongjoonkim@yonsei.ac.kr; joazoa@soc.yonsei.ac.kr; shkang@yonsei.ac.kr).

Digital Object Identifier 10.1109/TVLSI.2006.878227

(ATE). Since the channel width and the size of memory for the ATE are limited, the traditional ATE must be modified or more expensive ATE must be required in order to test an SoC with enormous test data.

To alleviate these problems, there are two useful approaches. One is a built-in self-test (BIST) scheme [1]–[5] and the other is a test compression/decompression scheme [6]–[16]. However, the BIST can be applied directly to circuits only if circuits are BIST-ready. In recent SoC environments, since most IP cores provided from core vendors are not BIST-ready and incorporating the BIST into circuits can lead to the performance penalty during normal operation of circuits, BIST schemes are not an acceptable solution for all circuits in order to reduce enormous test data.

In a test data compression/decompression scheme, on the other hand, a test input sequence is compressed and the compressed input sequence is decompressed by a decompressor on the circuit or the ATE. These approaches have been researched over the last several years. There are three types of the previous approaches: the methods using the compacted test sets [6], [7], test data compression methods which reduce the amount of data transmitted to the ATE [8], and the test data compression scheme using an on-chip decompression architecture [9]–[17]. The method proposed in [12] is suitable for only small primary-input circuits. In addition, the disadvantage of the approaches proposed in [13] and [17] are such that, as the block size increases, it leads to high area overhead for the decompressor. Especially in SoC environments, the minimal area overhead of the decompressor is more significant since the decompressor is necessary for each core. In [14], the approach based on the fact that effective test sets in a test sequence are very similar except only a small number of bits is proposed first. In order to use this fact, the original test set T_D is transformed into the differential test sequence set $T_{\rm diff}$. The Golomb code [10], frequency-directed runlength (FDR) code [9], and variable-length input Huffman coding (VIHC) [16] also use the differential test sequence set $T_{\rm diff}$. Although all of them provide high-compression ratio for most of benchmark circuits, they still lead to high–area overhead for the decompression architectures. In addition, in order to regenerate to the original test set T_D from the differential test sequence set $T_{\rm diff}$ inside a chip, the cyclical scan register (CSR) has to be used [14].

Therefore, to alleviate these limitations of previous compression schemes, we focus on a test data compression/decompression scheme without the CSR architecture in order to achieve a high–compression ratio and low–area overhead for the decompression architecture. In this paper, a new hybrid test data compression/decompression scheme is proposed. Unlike the previous approaches, the proposed approach uses the modified input-reduction scheme in order to compensate the loss of the compression ratio which is caused by our approach without the CSR architecture and $T_{\rm diff}$.

II. Modified Input Reduction Scheme

In order to reduce test sets for a BIST, the input reduction was proposed first in [18]. A new input-reduction scheme is used in this paper, which is the modified version of the input-reduction scheme in [18], in order to achieve high–compression ratio. The input-reduction scheme is to identify circuit inputs that can be combined into other test inputs without the loss of fault coverage. This method is based on the concepts of the compatibility and the inverse compatibility.

When a test set T_D whose input size is N and length is L is given, let $v(i,k)$ be a value of input i $(0 \le i \le N-1)$ at sequence k $(0 \le k \le L-1)$. The input compatibility is defined as follows.

Definition 1— Input Compatibility: For a given test input set T_D, two inputs i and j are compatible, if and only if $v(i,k) = v(j,k)$. If $v(i,k) = X$ (don't care) or $v(j,k) = X$, it should not conflict with the value of any other compatible or inverse compatible inputs.

```
input_reduction()
T_D: test set
N : the number of inputs
L : the length of test sequence
C : the set to check whether target input is reduced previously
{
 int i ; // input i(0 ≤ i ≤ N-1)
 int j; // input j (i+1≤ j ≤ N-1)
 Int k; // sequence k(0 ≤ k ≤ L-1)
 int check;
 initialize_input_check();
 for(i=0; i<N; i++) {
   if(C_i == UNIQUE) {
          v(i, k); // target input
          for(j=i+1; j<N; j++) {
            if(C_j == UNIQUE) {
                  v(j, k); // comparison input
                  compatible_check = is_compatible(v(i,k), v(j,k));
                // is_compatible function includes the conflict_check function
                  switch(compatible_check) {
                   case COMPATIBLE:
                    C_j = COMPATIBLE; break;
                   case INV_COMPATIBLE:
                    C_j = INV_COMPATIBLE; break;
                   case DONT_CARE:
                    C_j = DON'T_CARE; break;
                   case NONE:
                    C_j = UNIQUE; break; } }
            else  break; } } }
}
```

Fig. 1. Modified input-reduction algorithm.

Let $\overline{v(i,k)}$ be an inverse value of $v(i,k)$. The inverse compatibility is defined as follows.

Definition 2—Input Inverse Compatibility: For a given test input set T_D, two inputs i and j are compatible, if and only if $v(i,k) = \overline{v(j,k)}$. If $v(i,k) = X$ (don't care) or $v(j,k) = X$, it should not conflict with the value of any other compatible or inverse compatible inputs.

Unlike the input reduction in [18], we consider a case where test sets are given by the deterministic automatic test pattern generation (ATPG) or the combination of random and deterministic ATPGs. Therefore, we propose a new input-reduction algorithm for this situation. The proposed algorithm is shown in Fig. 1. Note that the process of don't care identification is required for a given test set obtained from the combination of random and deterministic ATPG in order to reduce test inputs efficiently. At the first step of the don't care identification process, we perform fault simulation for a given test set T_S to mark the essential faults of each test vector. Next, we drop faults which are detectable by a currently used test vector in the entire fault list and the combination of input values which excited and propagated these faults is identified by the processes of backtrace and the forward implication as similar methods in the deterministic ATPG. Input values which are not fixed are reassigned to X's.

In this way, the scheme to compress test set by identifying compatible inputs and inverse compatible inputs is called the Input Reduction (IR) scheme. Note that the reduced test set by the input-reduction method is called $T_{\rm IR}$.

III. Micro Scheme

This section introduces a new compression/decompression scheme for the reduced test data $T_{\rm IR}$. We propose an efficient test data compression/decompression scheme, the Modified Input reduction and CompRessing One block (MICRO), without using $T_{\rm diff}$, which satisfies two requirements: a lossless compression algorithm and a simple decompression architecture. In this scheme, only one block which occurs most

frequently in $T_{\rm IR}$ is compressed since it is important as to make the decoder for the compression code simple and small as well as to increase the compression ratio.

A. Proposed Compression Code

In this paper, a new compression code, called the MICRO code, is proposed in order to compress test data efficiently. The main key idea is that the compression ratio is enhanced by increasing the occurrence frequency of one block. It can be achieved by appropriately filling specific values to numerous unspecified value ("X" values) in test sets. Compressing one block leads to lower hardware overhead for the decoder while, in the methods like the Huffman code, as the number of the compressed blocks is increased, the hardware overhead for the decompression architecture is increased enormously.

To simplify the presentation, the following definitions are used.

Definition 3—CB: The compression block (CB) is defined as the 4-bit block which occurs most frequently in the test data.

Definition 4—UB: The test data which exclude the compression block are divided into 2-bit blocks. The uncompression blocks (UB) are defined as theses divided 2-bit blocks.

The idea of the MICRO code is to make the 4-bit block *CB*, which occurs most frequently, have a 1-bit code, and the rest of test sets have 3-bit codes which consist of a 2-bit original block and a prefix bit to identify uncompressed 2-bit blocks. In general, it is easy to increase the occurrence frequency of one 4-bit block by filling appropriate specific values to unspecified test cubes since deterministic test patterns contain many X values. Therefore, increasing the occurrence frequency of the specific 4-bit block is implemented easily and it is an efficient method to improve the compression ratio for test data.

Furthermore, test data, except the compression block, are divided into the 2-bit uncompression blocks, in order to make the most frequent 4-bit block in test sets occur more frequently. This is because dividing the rest of test data which exclude the CB into smaller size, has a high probability to increase the occurrence frequency of the CB in the original test data. For this reason, the compression block occurs more frequently as the length of the UB is 2 bits. The goal of the MICRO code is to improve the compression ratio by increasing the occurrence frequency of the most frequent 4-bit block in $T_{\rm IR}$.

To evaluate the relative efficiency of the Huffman and the MICRO code, we compute the compression efficiency. In the MICRO code, the compression efficiency is computed as the following equation, where OF_i is the occurrence frequency. The lower compression efficiency value means that the compression ratio is much higher. Note that the OF_1 is the occurrence frequency of the most frequent block and the occurrence frequency of the block which decreases as i becomes larger

$$CE_{\rm MICRO} = 0.25OF_1 + 1.5\sum_{j=2}^{5} OF_j. \tag{1}$$

As mentioned before, the OF_1 for the CB increases by 0.25 times because the CB is encoded from a 4–bit original block to a 1-bit code. Similarly, the sum of the occurrence frequencies of the remaining uncompression blocks product is 1.5. Note that when $CE = 1$, the length of the encoded codes is the same length of the original data. In the Huffman code, the compression efficiency depends on the Huffman encoding tree and occurrence frequencies of blocks for a given piece of test data. Empirically, assuming the size of the block is 4 bit in the Huffman code, the compression efficiency equation for the test data of most circuits is one of the following equations

$$\begin{cases} CE_{\rm Huffman} = \sum_{j=1}^{16} 0.25 \cdot j \cdot OF_j \\ CE_{\rm Huffman} = 0.5(OF_1 + OF_2 + OF_3) + \sum_{j=4}^{16} 0.25(j-1) \cdot OF_j. \end{cases} \tag{2}$$

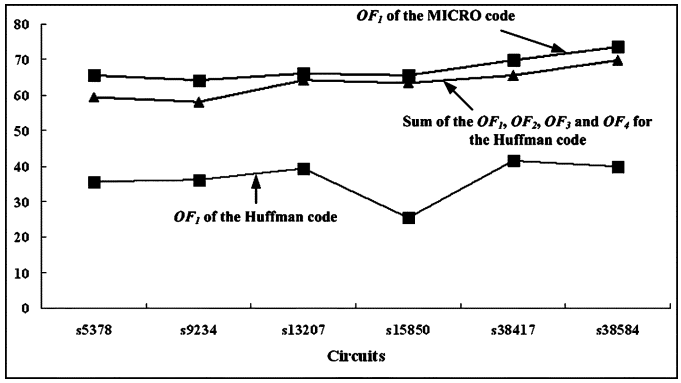


Fig. 2. Example of occurrence frequencies of the MICRO code and the Huffman code.

In these cases, the compression efficiency of the MICRO code is less than that of the Huffman code, if the following condition is satisfied:

$$OF_{1\_\rm MICRO} \geq \sum_{j=1}^{4} OF_{j\_\rm Huffman}. \tag{3}$$

For this case, using the MICRO code guarantees a better compression ratio than using the Huffman code. Although the sum of OF_1, OF_2, OF_3, and OF_4 for the Huffman code is larger than the OF_1 of the MICRO code, the compression ratio of the MICRO code can be less than that of the Huffman code; however, it cannot always guarantee that the compression ratio of the MICRO code is better since the compression result also heavily depends on the occurrence frequencies of the uncompression blocks.

Fig. 2 shows the OF_1 value of the MICRO code and the sum of $OF_1 \sim OF_4$ values in ISCAS '89 benchmark circuits. Since dividing the rest of the test data, which excludes the CB into 2-bit leads to increase the occurrence frequency of the CB in the original test data, the average difference between the OF_1 of the MICRO code and OF_1 of the Huffman code is almost 30%. In addition, although the compression efficiency of the Huffman code is marginally better than that of the MICRO code, it is inevitable that the decoder-area overhead of the Huffman code will be much higher that that of the MICRO code. The size of the compression block of the MICRO code can be extended to 5 bits more if a larger size of the CB can guarantee the higher compression ratio.

B. Proposed Compression Algorithm

The algorithm used to implement the proposed scheme (MICRO) has three procedures: 1) preparing initial test set; 2) reordering the test set; and 3) compressing the test set using the MICRO code. The detail of each step is as follows.

1) *Preparing initial test set:* The procedure consists of two steps. In the first step, using the modified input-reduction scheme, the original test set T_D is transformed to the reduced test set $T_{\rm IR}$. In the next step, by computing the occurrence frequency of 4-bit blocks, the CB is determined and the unspecified values in the reduced test set $T_{\rm IR}$ are mapped to appropriate specific values to make the frequency of the occurrence for the CB higher.
2) *Reordering the test set $T_{\rm IR}$:* The sequence of the reduced test set $T_{\rm IR}$, which is mapped to specific values, is reordered so that the compression block occurs more frequently. Initially, in this procedure, first test sequence $T_{\rm IR1}$ is prepared as the initial test sequence and then $T_{\rm IR1}$ is divided into 4-bit blocks and 2-bit blocks according to whether the divided 4-bit block is the CB or not. If there is the remaining block rb_1 in $T_{\rm IR1}$ ($rb_1 \leq 3$) bits and a filling block fb_2, which the compression block is

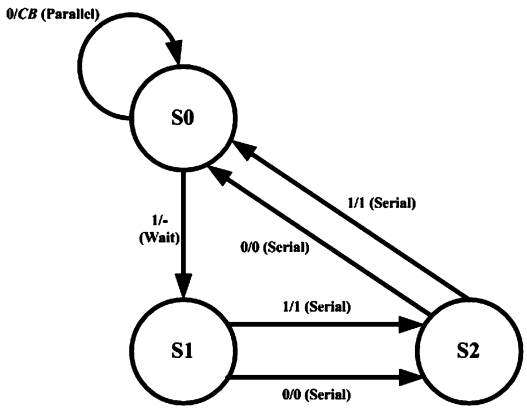


Fig. 3. State diagram of the FSM decoder for the MICRO.

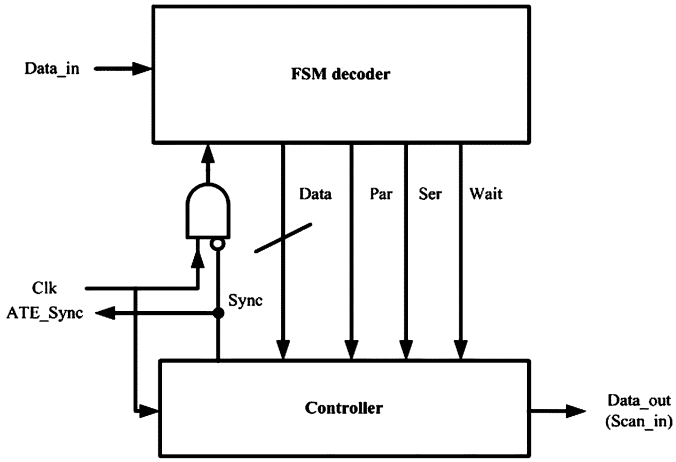


Fig. 4. MICRO decompression architecture.

generated by adding rb_1 to, is found in the other sequences, this sequence which includes the filling block fb_2 becomes the second test sequence $T_{\mathrm{IR}2}$. Otherwise, the original second test sequence $T_{\mathrm{IR}2}$ is prepared for the reordering procedure. For the entire sequence $k(1 \leq k \leq L-1)$, repeating the same steps mentioned above, the occurrence frequency of the compression block in T_{IR} can be much higher.

3) *Compressing the test set using the MICRO code:* The reduced and reordered test set T_{IR} is compressed by the MICRO code proposed in Section III-A. This procedure is to make the compression block $\{CB\}$ which occurs most frequently include a 1-bit code, and the uncompression block $\{UB_1, UB_2, UB_3, UB_4\}$ have 3-bit codes which consist of a 2-bit original block and a prefix bit to identify uncompressed 2-bit blocks. Let $N = \{n_{\mathrm{CB}}, n_{\mathrm{UB1}}, n_{\mathrm{UB2}}, n_{\mathrm{UB3}}, n_{\mathrm{UB4}}\}$ be the number of occurrences of the test set T_{IR}. The total number of the compressed test data is $n_{\mathrm{CB}} + \sum_{i=1}^{4} 3 \cdot n_{\mathrm{UBi}}$.

C. Decompression Architecture

A general on-chip decompression architecture consists of the FSM decoder to decode the compression code and the control logic which is responsible for controlling the data transfer between the ATE and the FSM decoder. We assume that the ATE can be external clock synchronization as shown in [19].

Once the MICRO code has been chosen, then an FSM decoder for the MICRO code is synthesized. The three control signals are parallel, serial, and wait. These signals control the buffering and the loading of data into the controller when the data has been decoded and the synchronization of the ATE. An example of the state diagram for the FSM decoder of the MICRO code is shown in Fig. 3. Each encoded block has a prefix bit which indicates whether remaining bits which exclude the prefix bit are compressed bits or not.

The controller to transfer the decoded test data into scan chains in the CUT and to control the signals between the ATE and the FSM decoder is illustrated in Fig. 4. The FSM does not perform the decoding process but the controller transfers data into scan chains since the FSM clock is stable, when the sync output signal in the controller is a "1." If the sync signal is a "0," then the FSM decoder decodes the compressed test data with the shifting process in the controller.

IV. Experimental Results

To demonstrate the efficiency of the proposed method, the proposed compression/decompression scheme is used to compress test sets for ISCAS '89 and ITC '99 benchmark circuits. The proposed MICRO scheme is implemented in C and experiments are performed on a Pentium 3 667-MHz system with the Linux. The test set for each circuit is generated by MinTest [20] in order to compare with SC [13] and the Huffman code. Note that, in order to compare fairly, we use the reduced test data set T_{IR} for all cases. Note that the compression ratio is computed as follows:

$$\text{Compression ratio} = \frac{|T_{\mathrm{IR}}| - T_{\mathrm{comp}}}{|T_{\mathrm{IR}}|} \times 100 \tag{4}$$

where T_{comp} is the size of the compressed test set.

The results of the compression are presented in Table I. As shown in Table I, the MICRO code provides an almost similar compression as a Huffman code, but the number of states for the MICRO is fixed, while the number of states for the Huffman decoder grows exponentially and the number of states for SC [13] grows linearly.

We cannot compare the compression ratio of the MICRO with that of other compression schemes fairly, Golomb [10], FDR [9], and VIHC [16], since they show different compression ratios according to the test data set. Therefore, we just compare the area of the decompression architecture.

TABLE I
Compression Ratios for ISCAS '89 and ITC '99 Benchmark Circuits

Circuits	Group size	IR + SC [13]		IR + Huffman		MICRO	
		FSM State	Comp (%)	FSM State	Comp (%)	FSM State	Comp (%)
s5378	4	5	66.95	15	79.53	3	**79.64**
s9234	4	5	61.82	15	75.96	3	**76.14**
s13207	8	20	73.81	255	**86.33**	3	86.20
s15850	4	5	72.90	15	**85.49**	3	85.44
s38417	4	8	81.19	15	89.56	3	**89.74**
s38584	4	8	77.78	15	**93.75**	3	93.53
b10	4	8	24.3	15	55.64	3	**61.97**
b11	4	8	53.57	15	61.11	3	**68.84**
b12	4	8	68.61	255	79.22	3	**79.51**
b13	4	8	26.83	15	33.78	3	**80.29**
b14	4	20	53.37	255	65.17	3	**66.75**
b15	4	20	63.54	255	77.46	3	**78.54**

TABLE II
Decoder Area Comparison

Group size	SC [13]	Golomb [10]	FDR [9]	VIHC [16]	SHC [17]	MICRO
4	349	125	320	136	349	120
8	587	227		201	587	
16	900	307		296	900	

TABLE III
COMPARISON OF THE AREA OVERHEAD FOR THE DECODER

Circuits	Area of the circuit with a single scan chain	SC [13]	Golomb [10]	FDR [9]	VIHC [16]	SHC [17]	MICRO
s5378	2704	11.43	4.41	10.58	4.78	21.55	3.87
s9234	3361	9.4	3.58	8.69	3.88	18.10	3.14
s13207	7965	10.15	3.71	3.86	3.58	8.53	1.35
s15850	7672	4.35	1.6	4	1.74	8.82	1.40
s38417	22841	1.5	0.54	1.38	0.59	3.15	0.47
s38584	23236	1.47	0.53	1.35	0.58	3.09	0.46
b10	4491	7.77	2.78	7.12	3.03	13.07	2.67
b11	10228	3.41	1.22	3.13	1.33	5.74	1.17
b12	28479	1.22	1.04	1.12	0.71	2.06	0.42
b13	10709	3.25	1.17	2.99	1.27	5.48	1.12
b14	131549	0.26	0.22	0.24	0.15	0.45	0.09
b15	175369	0.20	0.07	0.18	0.07	0.33	0.06

Table II shows the area comparison of the decompression architecture for previous works and the proposed approach. The area is computed by the Synopsys Design compiler with the lsi 10-k library. Note that the block size of the proposed MICRO code is fixed as 4 bits and, for SC, Golomb, VIHC and SHC, the area overhead is computed for the group size of 4, 8, and 16. The area for previous approaches is obtained in [16]. To compare exactly with previous works, the decoder of the MICRO for s5378 benchmark circuit, whose decoders are synthesized in [16] for the comparison, is also synthesized.

In addition, the area overhead of the decoders for the different compression methods is compared in Table III. The ISCAS '89 and ITC '99 benchmark circuits are synthesized with a single scan chain, and a single scan chain in the benchmark circuit is inserted by the DFT advisor of the Mentor Graphics with the class library. The area overhead of the decoder is computed as follows. See equation (5) at the bottom of the page

$$\text{decoder area overhead} = \frac{\text{the area of decoder}}{(\text{the area of benchmark circuit} + \text{the area of decoder})} \times 100. \quad (5)$$

The area for decoders and benchmark circuits is computed by the Synopsys Design compiler with the class library. Note that the decoder for each compression scheme is configured by the parameter, the group size, shown in Table I. Column 2 in Table III is the area of benchmark circuits without the decoder.

As shown in Tables II and III, the decoder of the MICRO has the lowest area overhead. Furthermore, for Golomb, FDR, and VIHC, the area overheads in Tables II and III exclude the area overhead for the CSR architecture. As mentioned before, it is inevitable that the high area overhead for the CSR architecture is required. Of course, to resolve this problem, the method using the user defined logic (UDL) or the boundary scan around another core was proposed in [14]. However, the usage of the boundary scan around another core is not possible in all cases and the control logic to reconfigure the boundary scan to the CSR architecture is also required. Although the problem is alleviated, the area overhead to use the UDL is still high, especially in an SoC. Therefore, the proposed compression/decompression scheme is an effective solution of the test data compression/decompression for SoCs.

Finally, it is necessary to compare the total test application time (TAT) reduction that can be achieved using the different compression techniques. Table IV shows the results of comparing lower bounds on $f_{\rm sys}/f_T$ to obtain maximum TAT reduction. Note that the value of the parameters for each scheme is obtained in Table II. The same concept in [17] is used for the experiments in order to compare the total TAT. Using this concept, the total test application time is easily compared

TABLE IV
LOWER BOUNDS ON $f_{\rm sys}/f_T$

Circuit	Lower bounds on f_{sys}/f_T				
	Golomb [10]	FDR [9]	VIHC [13]	SHC [17]	MICRO
s5378	4	128	4	6	4
s9234	4	64	4	6	4
s13207	16	512	16	6	4
s15850	4	512	4	6	4
s38417	4	1024	4	6	4
s38584	4	512	4	6	4

by using lower bounds on $f_{\rm sy}/f_T$. Note that $f_{\rm sys}$ is the ratio of the system clock frequency and f_T is the tester clock frequency. For the MICRO code, the block size b determines the lower bound on $f_{\rm sys}/f_T$. When the decoder receives a complete codeword, it needs to output the corresponding block of b bits into the serializer in the controller of the MICRO decompression architecture. Therefore, the lower bound is given by $f_{\rm sys}/f_T \geq (b/L_{\min})$, where $L_{\min}$ is the size of the smallest codeword. Since it is more difficult to achieve the maximum reduction of the total TAT when the value of the lower bound on $f_{\rm sys}/f_T$ is higher, having lower value of the lower bound on $f_{\rm sys}/f_T$ has the advantage.

V. CONCLUSION

In this paper, we propose a new hybrid test data compression/decompression scheme using the input-reduction scheme for SoCs. Instead of $T_{\rm diff}$, the input-reduction scheme is used in order to identify redundant inputs for testing, and by the input-reduction, we obtain effectively compressed test data without any loss of test data information. This compressed test data $T_{\rm IR}$ is compressed again by the proposed compression method which consists of three procedures. The decompression architecture for the proposed MICRO code is simple and small since the MICRO code has a simple decoding process. Therefore, the area overhead for our approach is much lower than that of previous approaches.

As shown in experimental results, the on-chip decoder is smaller than that of previous works and easy to implement. In addition, higher compression ratio is achieved by the MICRO scheme. Therefore, the proposed approach is an attractive and effective solution of test data compression/decompression for SoCs.

REFERENCES

[1] I. Hamzaoglu and J. H. Patel, "Reducing test application time for built-in-self-test test pattern generators," in *Proc. VLSI Test Symp.*, 2000, pp. 369–375.

[2] C. A. Chen and S. K. Gupta, "Efficient BIST TPG design and test set compaction via input reduction," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 8, pp. 692–705, Aug. 1998.
[3] K. Chakrabarty, B. T. Murray, J. Liu, and M. Zhu, "Test width compression for built-in-self-testing," in *Proc. Int. Test Conf.*, 1997, pp. 328–337.
[4] C. V. Krishna, A. Jas, and N. A. Touba, "Test vector encoding using partial LFSR reseeding," in *Proc. Int. Test Conf.*, 2001, pp. 885–893.
[5] H. G. Liang, S. Hellebrand, and H. J. Wunderlich, "Two dimensional test data compression for scan based deterministic BIST," in *Proc. Int. Test Conf.*, 2001, pp. 894–902.
[6] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithms for combinational circuits," in *Proc. Int. Conf. Comput.-Aided Des.*, 1998, pp. 283–289.
[7] I. Pormeranz, L. Reddy, and S. Reddy, "Compactest: A method to generate compact test set for combinational circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 12, no. 7, pp. 1040–1049, Jul. 1993.
[8] M. Ishida, D. S. Ha, and T. Yamaguchi, "Compact: A hybrid method for compressing test data," in *Proc. IEEE VLSI Test Symp.*, 1998, pp. 62–69.
[9] A. Chandra and K. Chakrabarty, "Frequency-directed run-length (FDR) codes with application to system on a chip test data compression," in *Proc. IEEE VLSI Test Symp.*, 2001, pp. 114–121.
[10] ——, "System-on-a-chip test data compression and decompression architectures based on Golomb codes," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 20, no. 3, pp. 113–120, Mar. 2001.
[11] A. El-Maleh, S. al Zahir, and E. Khan, "A geometric primitives based compression scheme for tesing system-on-chip," in *Proc. IEEE VLSI Test Symp.*, 2001, pp. 114–121.
[12] V. Iyengar, K. Chakrabarty, and B. Murray, "Deterministic built in pattern generation for sequential circuits," *J. Electron. Testing: Theory Appl.*, vol. 15, no. 1, pp. 97–114, Aug. 1999.
[13] A. Jas, J. Ghosh-Dastidar, and N. A. Touba, "Scan vector compression/decompression using statical coding," in *Proc. IEEE VLSI Test Symp.*, 1999, pp. 114–121.
[14] A. Jas and N. Touba, "Test vector decompression via cyclical scan chains and its application to testing core based designs," in *Proc. IEEE Int. Test Conf.*, 1998, pp. 458–464.
[15] ——, "Using embedded processor for efficient deterministic testing of system-on-chip," in *Proc. Int. Conf. Comput. Des.*, 1999, pp. 418–423.
[16] P. Y. Gonciari, B. M. Al-Hashimi, and N. Nicolici, "Improving compression ratio, area overhead, and test application time for system-on-a-chip test data compression/decompression," in *Proc. Des. Automat. Test Eur. Conf. Exhibition*, 2002, pp. 604–611.
[17] A. Jas, J. Ghosh-Dastidar, M. Ng, and N. A. Touba, "An efficient test vector compression scheme using selective Huffman coding," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 6, pp. 797–806, Jun. 2003.
[18] C. A. Chen and S. K. Gupta, "Efficient BIST TPG design and test set compaction via input reduction," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 17, no. 8, pp. 692–705, Aug. 1998.
[19] D. Heidel, S. Dhong, P. Hofstee, M. Immediato, K. Nowka, J. Silaberman, and K. Stawiasz, "High-speed serializing/deserializing design-for-test methods for evaluating a 1 GHz microprocessor," in *Proc. IEEE VLSI Test Symp.*, 1998, pp. 234–238.
[20] I. Hamzaoglu and J. H. Patel, "Test set compaction algorithm for combinational circuits," in *Proc. IEEE Int. Conf. Comput.-Aided Des. Integr. Circuits Syst.*, 1998, pp. 283–289.

Wafer-Level Package Interconnect Options

Jayaprakash Balachandran, Steven Brebels, Geert Carchon, Maarten Kuijk, Walter De Raedt, Bart K. J. C. Nauwelaers, and Eric Beyne

***Abstract*—As integrated circuit technology enters the nanometer era, global interconnects are becoming a bottleneck for overall chip performance. In this paper, we show that wafer-level package interconnects are an effective alternative to conventional on-chip global wires. These interconnects behave as *LC* transmission lines and can be exploited for their near speed of light transmission and low attenuation characteristics. We compare performance measures such as bandwidth, bandwidth density, latency, and power consumption of the package-level transmission lines with conventional on-chip global interconnects for different International Technology Roadmap for Semiconductors (ITRS) technology nodes. Based on these results, we show that package-level interconnects are well suited for power demanding low-latency applications. We also analyze different interconnect options such as memory buses, long inter tile interconnects, clock, and power distribution.**

***Index Terms*—Global interconnects, package, performance metrics, transmission lines.**

I. Introduction

On-Chip global interconnects are perceived as overall performance limiters of integrated circuits in nano-CMOS technologies [1]. The delay of global interconnects exceeds transistor logic delay and their performance deteriorates with length, negating the general performance improvement expected from scaling. They also consume significant amounts of power, as much as 53% of the total interconnect power as reported in [2]. Several solutions such as Cu/low-k, three–dimensional (3-D) integration, X-routing and optical interconnects are proposed to address the global interconnect conundrum. By far, the optical solution is the best in terms of delay and power but is useful only for long lines with critical length exceeding 10 mm [3]. Unfortunately, the optical interconnects are also very disruptive and offer many technical challenges such as dense integration of optical components on silicon die [4], [5]. An effective alternative would be to exploit *LC* transmission line properties of metal interconnect. Considering the range of possible lengths for on-chip wiring, performance of well-designed *LC* transmission lines would be comparable to that of the optical solution. The optical interconnects outperform transmission lines mainly by their ability to magnify bandwidth through wavelength-division multiplexing (WDM) [4].

The benefits of *LC* transmission lines for large on-die L2 cache memory application were first discussed in [6]. However, implementing the *LC* transmission lines on-chip requires additional wiring layers, significantly impacting on the product costs [6]. Instead, transmission lines implemented on package wiring layers can be effectively used for on-chip global interconnect applications. In this paper, we propose implementation of *LC* transmission lines in wafer-level packaging (WLP) layers and explore their interconnect options for on-chip global wiring.

Manuscript received August 5, 2005; revised January 25, 2006.

J. Balachandran, S. Brebels, G. Carchon, W. De Raedt, and E. Beyne are with the IMEC VZW, Heverlee, Leuven 3001, Belgium (e-mail: bchandra@imec.be).

M. Kuijk is with the Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Brussels 1050, Belgium (e-mail: mkuijk@vub.ac.be).

B. Nauwelaers is with the Katholieke Universiteit Leuven, Heverlee, Leuven 3001, Belgium (e-mail: bart.nauwelaers@esat.kuleuven.ac.be).

Digital Object Identifier 10.1109/TVLSI.2006.878229