Efficient Built-In Self-Test Scheme for Inter-Die Interconnects of Chiplet-Based Chips

Yi-Chun Huang, Pei-Yun Lin, Jin-Fu Li, and Hong-Siang Fu
Department of Electrical Engineering
National Central University
Taoyuan, Taiwan 320

Yung-Ping Lee
Electronic and Optoelectronic System Research Lab.
Industrial Technology Research Institute
Hsinchu, Taiwan 310

2024 IEEE International Test Conference (ITC) | 979-8-3315-2013-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITC51657.2024.00034

***Abstract*—Various advanced packaging technologies have been proposed to integrate multiple dies into a package. In the multi-die package, a significant amount of inter-die interconnects are used to assemble the dies. The bonding interfaces, e.g., microbumps (ubumps), of the inter-die interconnects are prone to defects during the assembly process. Therefore, it is crucial to test the inter-die interconnects to ensure the quality of the assembly. In this paper, we propose an efficient built-in self-test (BIST) scheme for the testing of inter-die interconnects of chiplet-based ICs. The transmitters/receivers under test are logically arranged as a 2D array. A test algorithm that utilizes alternating row stripe and column stripe test patterns is proposed. These 2D test patterns are then compressed into 1D test patterns such that the test application time can be reduced. Simulation and analysis results show that the proposed BIST scheme can achieve 99.9% test time reduction and save about 53.5% area cost for a 32×64 ubump array in comparison with a typical test scheme, IEEE 1500 test scheme.**

***Index Terms*—Chiplet, inter-die interconnect, interconnection testing, advanced packaging, built-in self-test.**

I. Introduction

Advanced packaging technologies enable the system-level scaling by integrating multiple dies in 2.5D or 3D structures [1]–[5]. In the multi-die package, a huge amount of inter-die interconnects are used for the communication among dies. An inter-die interconnect is usually constituted of in-die metal lines and bonding interface, such as a microbump. During the assembly process, the bonding interfaces of the inter-die interconnects are prone to defects such as shorts and opens [1]. Over the lifetime of the system, the bonding interfaces are prone to the defects caused by thermal-mechanical stress [6]. Thus, the testing of inter-die interconnects is very important for ensuring the quality of chiplet-based ICs.

Testing algorithms of stuck-at and bridge faults of interconnection were studied in [7]–[11]. In [7], a test algorithm using binary counting sequence was proposed to detect the bridge faults of between nets. For n nets, the number of required test patterns is $\lceil log_2 n \rceil$. However, the binary counting sequence consists of all-0 and all-1 test patterns which cannot cover stuck-at-1 and stuck-at-0 faults, respectively. To cope with this issue, a test algorithm using a modified binary counting sequence which is the binary counting sequence excluding all-0 and all-1 test patterns was proposed in [8]. This algorithm requires $\lceil log_2(n+2) \rceil$ test patterns for n nets. However, binary counting sequence and modified binary counting sequence do not have diagnosis capability. A true/complement modified binary counting sequence which requires $2\lceil log_2(n+2) \rceil$ test patterns for n nets was proposed to identify the faulty nets [9]. In the true/complement modified binary counting sequence, each net receives a unique bit stream and the bit streams for all the nets under test have an equal number of “0” and “1” bits. Therefore, any nets having short faults will cause the corresponding receivers to receive an unexpected number of “0” and “1” bits and the faulty nets can be identified [11]. Since true/complement bitstreams are included, we do not need to exclude the all-0 and all-1 bitstreams for the detection of stuck-at faults. That is, true/complement binary counting sequence can fully cover stuck-at and short faults and identify faulty nets, requiring only $2\lceil log_2 n \rceil$ test patterns for n nets [10]. In [10], test and diagnosis methods using a built-in self-test (BIST) technique based on boundary scan architecture were reported.

Despite the test algorithms mentioned above having a small number of test patterns, these test patterns are typically applied sequentially through scan-based design-for-testability (DFT) circuits, e.g., IEEE 1149.1 (boundary scan) and IEEE 1500, which consumes a significant amount of test time when the number of interconnects is large. To reduce the test application time, a BIST scheme with word-oriented test application method was proposed to test the through-silicon vias (TSVs) of 3D ICs [12], [13]. Recently, interconnect automatic test and diagnosis pattern generation algorithms method exploiting the knowledge of layout positions to reduce the required number of test patterns were proposed [14]–[17].

In advanced packaging technology, the 3D interconnect density (3DID) is the technology scaling index which is defined as the product of the number of horizontal interconnects per mm and the number of vertical interconnects per mm^2. The 3DID of current packaging technology can reach $10^6/mm^3$ and is expected to double every two years [3]. Therefore, efficient test techniques for the testing of inter-die interconnects of chiplet-based ICs are imperative for the production testing of advantage packages. In this paper, we propose an efficient BIST scheme for the testing of inter-die interconnects in chiplet-based ICs. An alternating row and column stripe (ARCS) test algorithm is proposed to detect and diagnose stuck-at and bridging faults of inter-die interconnects. Due to the homogeneous property of test patterns generated by

2378-2250/24/$31.00 ©2024 IEEE
DOI 10.1109/ITC51657.2024.00034

ARCS, a simple test compression and decompression method is proposed to reduce the test application time. Finally, a cost-effective BIST scheme is proposed to generate the test patterns and evaluate the test responses.

The rest of this paper is organized as follows. Section II briefly introduces the assembly process of chiplet-based ICs and common defects induced during the assembly process. Section III describes the proposed ARCS test algorithm and the test compression and decompression method. Section IV introduces a cost-effective BIST scheme realizing the ARCS test algorithm. Section V shows the simulation and analysis results. Finally, Section VI gives a brief summary.

II. Preliminary

A. Assembly Process and Common Defects

Chip-on-wafer-on-substrate (CoWoS) assembly process is one prominent method for the assembly of a 3D IC package. CoWoS assembly process aggregates dies using a passive interposer. A simplified CoWoS packaging assembly process [1] is as follows: 1) dies are sorted for known good die (KGD) and microbump (ubump) process is performed. Subsequently, the dies are diced and prepared for assembly; 2) the interposer wafer with TSVs are tested as well to ensure it does not have systematic defects. Then, ubump process is performed and prepared for die assembly; 3) top dies are assembled on the interposer and ubump underfill process is executed; 4) after the underfill cure, the TSV reveal process is executed for the dies and interposer wafer stack to expose TSVs. This is followed by creation of C4 bumps; 5) The wafer dicing process is executed and stacked dies are obtained. The stacked dies are attached on substrates using the standard mass reflow process and then C4 bump underfill process is executed; 6) Finally, the standard flip-chip ball grid array (BGA) assembly process is used for packaging the chip-on-chip module in production.

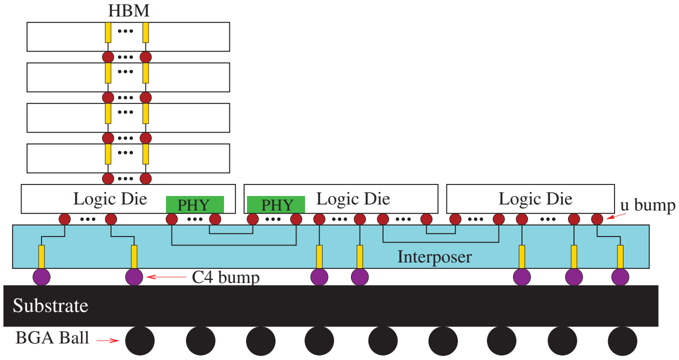

Fig. 1: An exemplary example of HBM and logic dies integrated using CoWoS assembly process.

Fig. 1 shows an exemplary example of high-bandwidth memories (HBM) and logic dies integrated using the CoWoS assembly process. An inter-die interconnect is constituted with metal lines in the dies, ubumps, and metal lines in the interposer. Among the manufacturing defects, common defects including shorts between adjacent ubumps, ubump opens, and interposer metal line shorts and opens [1], [18]. Therefore, the testing of open and short defects is a must for the testing of inter-die interconnects. Furthermore, the scaling index of the advanced packaging technology is the 3D interconnect density (3DID), defined as the product of the number of horizontal interconnects per mm and the number of vertical interconnects per mm^2 [3]. The 3DID is expected to double every two years and currently can reach a density of $10^6/mm^3$ [3]. Conventional interconnect testing techniques using scan-based DFT circuits is not time efficient, since the test application time is proportional to the number nets under test. Clearly, if the scan-based DFT method is used for the testing of inter-die interconnects in chiplet-based ICs, the test application time will be very long. Efficient test techniques for the testing of inter-die interconnects of chiplet-based ICs are therefore imperative.

B. Testing Techniques for Inter-Die Interconnects

The open and short defects in the interconnects can be modeled as stuck-at faults, wired-OR and wired-AND bridging faults. An efficient test algorithm for the testing and diagnosis of stuck-at and bridging faults is the true/complement binary counting sequence (TC-BCS) algorithm [10]. The number of required test patterns of TC-BCS algorithm for n interconnects is $2\lceil log_2 n\rceil$. Each test pattern of the TC-BCS algorithm is unique and has an equal number of “1” bits and “0” bits. If two interconnects have a wired-OR (wired-AND) bridging fault, the number of “1” bits (“0” bits) of the corresponding outputs will increase. As a result, any interconnect with bridging faults have erroneous outputs with an unequal number of “1” and “0” bits in the test response. Also, the fault location can be identified [11].

Fig. 2 shows an example of the TC-BCS algorithm for 8 interconnects equipped with scan-based DFT circuits. As the figure shows, six test patterns $(v_1, v_2, \ldots, v_6)$ are needed. Each test pattern and the corresponding test response can be applied and exported through the scan input (scan in) and scan output (scan out) sequentially, respectively. If the interconnect n_3 has a stuck-at-0 fault (SA0), the corresponding response will be all-0. The fault can be detected and located through the exported test response. Similarly, if a wired-OR bridging fault exists between the interconnects n_3 and n_4, the corresponding response captured at these two interconnects will be 011101. Again, the bridging fault can be detected and located.

Although the binary counting sequence-based test algorithm is efficient in terms of the number of test patterns, the scan-based test application approach results in high test application time if the number of interconnects under test is high. For example, if IEEE 1500 or IEEE 1149.1 is used for the test application, which is typically used in existing works, the test application time is proportional to the number of nets under test. As aforementioned, unfortunately, the number of inter-die interconnects of chiplet-based ICs is very high and will be drastically increased with the scaling of technology. Two methods can be used to reduce the test time: 1) reduce the

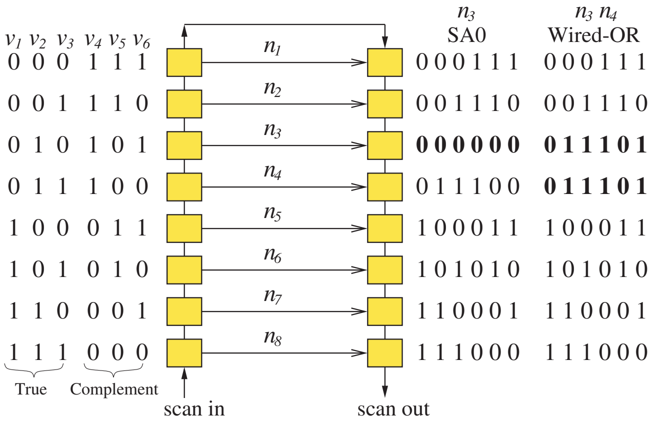

Fig. 2: An example of the true/complement binary counting sequence is for 8 interconnects equipped with scan-based DFT circuits.

the number of required test patterns and 2) reduce the test application time. In [14], for example, the number of test patterns is reduced by grouping the inter-die interconnects according to the physical location of ubumps. Similarly, in [15], [17], the number of test patterns is reduced by fixing the relative locations where shorts between ubump and adjacent ubumps are likely to occur. However, these previous approaches assume the use of a scan-based test application method. In [12], [13], the TSVs between dies in a 3D IC are logically arranged as an array and the test patterns are applied row by row such that the test application time is reduced.

III. Proposed Alternating Row and Column Stripe Test Scheme

Since existing binary counting sequence-based test algorithms are efficient in terms of the number of test patterns, an efficient test application method for a large number of inter-die interconnects should be developed to reduce the test application time. We propose an alternating row and column stripe (ARCS) test scheme for the inter-die interconnects in chiplet-based ICs. Firstly, an ARCS test algorithm is proposed to detect the stuck-at and bridging faults of inter-die interconnects arranged as a two-dimension (2D) array logically. Secondly, the ARCS test patterns for the 2D array of inter-die interconnects can be compressed as compressed test patterns due to their homogeneous property. Each compressed test pattern consists of a row vector and a column vector. Thirdly, each ubump is wrapped with a wrapper cell which has a decompression logic gate. Lastly, the compressed test patterns are applied and decompressed by the wrapper cells. Since only the compressed test patterns are applied, the test application time is reduced.

A. Alternating Row and Column Stripe Test Algorithm

Without loss of generality, we assume that the inter-die interconnects between two chiplets are logically arranged as a $h \times w$ array, where h and w denote the number of rows and columns of the array, respectively. In this case, each chiplet has a corresponding $h \times w$ ubump array. Before the introduction of the proposed ARCS test algorithm, we first define some terminologies for the description of the test algorithm:

1) 2^i row stripe-1/0 patterns: 2^i row(s) applied with “1”s alternating with 2^i row(s) applied with “0”s.
2) 2^i row stripe-0/1 patterns: 2^i row(s) applied with “0”s alternating with 2^i row(s) applied with “1”s.
3) 2^i column stripe-1/0 patterns: 2^i column(s) applied with “1”s alternating with 2^i column(s) applied with “0”s.
4) 2^i column stripe-0/1 patterns: 2^i column(s) applied with “0”s alternating with 2^i column(s) applied with “1”s.

Algorithm 1 shows the pseudocode of the proposed ARCS test algorithm for a $h \times w$ ubump array. We can see that the ARCS test algorithm applies 2^i row(s) stripe-1/0 and row(s) stripe-0/1 patterns in a row-wise way for $0 \leq i \leq \lceil log_2 h \rceil - 1$. Then, 2^i column(s) stripe-1/0 and column(s) stripe-0/1 patterns in a column-wise way for $0 \leq i \leq \lceil log_2 w \rceil - 1$ are applied. Therefore, the number of required test patterns of ARCS test algorithm for a $h \times w$ ubump array is $2(\lceil log_2 h \rceil + \lceil log_2 w \rceil)$. Please note that the sequence of the test patterns are independent.

Algorithm 1 Pseudocode of ARCS Test Algorithm

```
Require: $h \times w$ ubump array
for $(i = (\lceil \log_2 h \rceil - 1); i \leq 0; i - -)$ do
    //Apply row stripe test patterns
    Apply $2^i$ row stripe-1/0 test patterns;
    Apply $2^i$ row stripe-0/1 test patterns;
for $(j = (\lceil \log_2 w \rceil - 1); j \leq 0; j - -)$ do
    //Apply column stripe test patterns
    Apply $2^j$ column stripe-1/0 test patterns;
    Apply $2^j$ column stripe-0/1 test patterns;
End test;
```

An example is illustrated to explain the ARCS test algorithm. Assuming a 4×4 ubump array is under test, as shown in Fig. 3, 8 test patterns are generated: 2 rows stripe-1/0 pattern, 2 rows stripe-0/1 pattern, 1 row stripe-1/0 pattern, 1 row stripe-0/1 pattern (as shown in Fig. 3(a)), 2 columns stripe-1/0 pattern, 2 columns stripe-0/1 pattern, 1 column stripe-1/0 pattern, and 1 column stripe-0/1 pattern (as shown in Fig. 3(b)).

Subsequently, we will show that the ARCS test algorithm can fully cover stuck-at and bridging faults. It is intuitive that the ARCS test algorithm can detect stuck-at-0 and stuck-at-1 faults, since every ubump can undergo 0 and 1 test data. To detect a wired-OR or a wired-AND bridging fault of two interconnects, complement data pattern should be applied to the sources of the two interconnects. If the two interconnects have a wired-OR or a wired-AND bridging fault, all-1 or all-0 response will be received at the destinations of the two interconnects, respectively. The ARCS test algorithm consists of row stripe and column stripe test patterns. The row stripe-1/0 and stripe-0/1 test patterns, e.g., as Fig. 3(a) shows, make sure any two ubumps in two individual rows undergo complement data pattern. Thus, all possible inter-row bridging

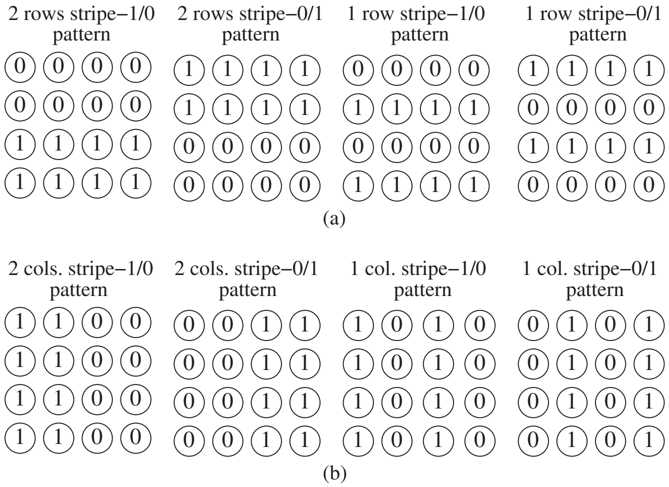

Fig. 3: An example of ARCS test algorithm for a 4×4 ubump array when (a) row stripe test patterns and (b) column stripe test patterns are applied.

faults can be detected. After the row stripe test patterns, the column stripe-1/0 and stripe-0/1 are applied. The column stripe-1/0 and stripe-0/1 test patterns, e.g., as Fig. 3(b) shows, make sure any two ubumps in a row undergo complement data pattern. Thus, the intra-row bridging faults can be detected. Consequently, the ARCS test algorithm can detect all possible bridging faults in an array of ubumps since it can fully detect both inter-row and intra-row bridging faults.

Furthermore, the ARCS test algorithm also has self-diagnosis capability. If the test set generated by a test algorithm is applied to test interconnects and every faulty interconnect can be identified by checking its output response , the test set (i.e., the test algorithm) has self-diagnosis capability [11]. A test set can be used for self-diagnosis iff it is independent [11]. If a test set is applied to interconnects, each interconnect receives a corresponding bit stream whose length (i.e., the number of bits) is equal to the number of test patterns of the test set. As Fig. 3 shows, if the test patterns of the ARCS test algorithm are applied to the 4×4 ubump array, each ubump receives an 8-bit bit stream since 8 test patterns are generated. As Fig. 3(a) shows, if the row stripe-1/0 and stripe-0/1 test patterns are applied, ubumps in different rows receive different 4-bit bit streams and every ubump in a row receive the same bit stream. As Fig. 3(b) shows, if the column stripe-1/0 and stripe-0/1 test patterns are applied, every ubump in a row receives different 4-bit bit streams. Thus, when the row and column stripe test patterns are applied, each ubump receives an unique 8-bit bit stream. Also, every bit stream has the same number of "0"s since every ubump receives the same true/complement test data (because 2^i row stripe-1/0 test pattern is the complement of 2^i row stripe-0/1 test pattern and 2^i column stripe-1/0 test pattern is the complement of 2^i column stripe-0/1 test pattern). For a test set, if every bit stream is unique and has the same number of "0"s, no bit stream is covered by any other bit stream. Thus, the test set is independent [11]. Since the test set generated by the ARCS test algorithm is independent, the ARCS test algorithm has the self-diagnosis capability.

According to the description above, we can conclude that the detection and diagnosis capability of stuck-at and bridging faults of the ARCS test algorithm is the same as that of true/complement binary counting sequence-based test algorithms [9], [10].

B. Test Pattern Compression and Decompression

The ARCS test algorithm generates row stripe and column stripe test patterns which have the homogeneous property (i.e., all-0 or all-1 pattern in a row or a column). The homogeneous property makes the test compression easy and efficient. As Fig. 3 shows, for example, each row and column under the row stripe test patterns and column stripe test patterns has either all-0 or all-1 test pattern. In this case, each row (column) can be represented using 1-bit test data. For a $h \times w$ ubumps, therefore, each $h \times w$-bit test pattern generated by the ARCS test algorithm can be compressed into a h- or w-bit compressed test pattern.

Fig. 4 shows an example of the test compression of 2 rows stripe-1/0 test pattern and 2 columns stripe-0/1 test pattern for a 4×4 ubump. The 2 rows stripe-1/0 and 2 columns stripe-0/1 test patterns are compressed as $\{1100\}$ and $\{0011\}$. Thus, the original two 16-bit test patterns are compressed into two 4-bit test patterns. This can reduce the test application time. In a similar way, the row stripe and column stripe test patterns generated by the ARCS test algorithm for $h \times w$ ubumps can be compressed as $2\lceil log_2 h \rceil$ h-bit compressed row stripe test patterns and $2\lceil log_2 w \rceil$ w-bit compressed column stripe test patterns, respectively.

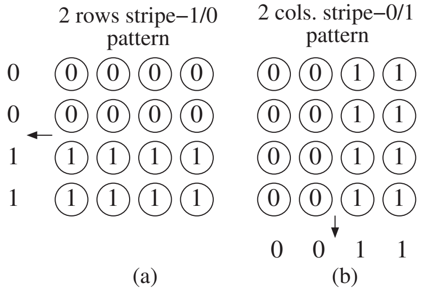

Fig. 4: For a 4×4 ubumps, (a) 2 rows stripe-1/0 test pattern is compressed as a compressed 2 rows stripe-1/0 test pattern and (b) 2 columns stripe-0/1 test pattern is compressed as a compressed 2 columns stripe-0/1 test pattern.

During the test application phase, the compressed test patterns must be decompressed. To reduce the test application time, decompression should be performed in parallel such that all the ubump can receive the corresponding test data in one clock cycle. Since the row stripe and column stripe test patterns have the homogeneous property, the compressed row stripe and column stripe test patterns can be easily decompressed by broadcasting the compressed test data to the ubumps in the corresponding rows and columns using row lines and column lines, respectively. However, the row stripe and column stripe test patterns are applied in sequence. Therefore, a two-input selection logic gate (i.e., decompression logic gate) should be added for each ubump. The inputs of every decompression logic gate are connected to corresponding row line (RL) and column line (CL). When the compressed row (column) stripe test pattern is decompressed, it is applied to the row (column) lines and one column (row) control pattern is applied to the column lines.

An example is used for a further explanation of the proposed test decompression scheme. A 4×4 ubump array under test is considered. As Fig. 5(a) shows, the compressed row stripe test pattern $\{1100\}$ is decompressed into 2 rows stripe-1/0 test pattern by using XOR decompression logic gate together with a column control pattern $\{0000\}$. Similarly, the compressed column stripe test pattern $\{0011\}$ is decompressed into 2 columns stripe-0/1 test patterns using row control pattern $\{0000\}$ as shown in Fig. 5(b).

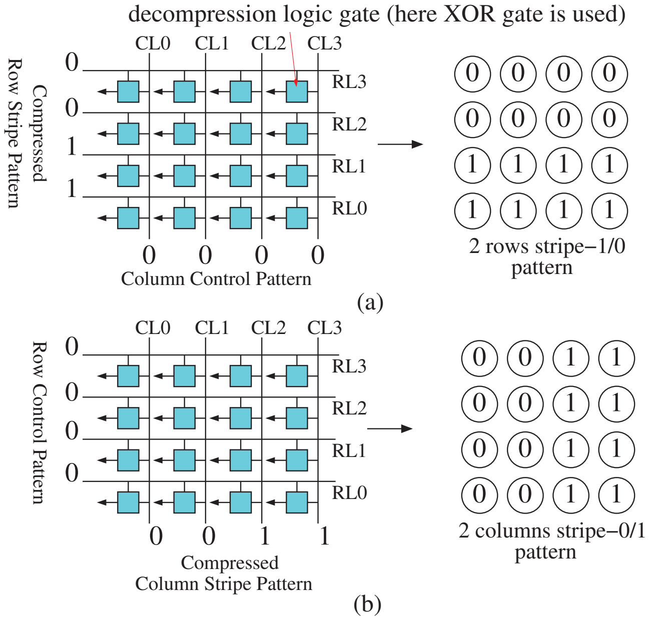
decompression logic gate (here XOR gate is used)
CL0
CL1
CL2
CL3
Compressed Row Stripe Pattern
0
0
1
1
RL3
RL2
RL1
RL0
0 0 0 0
0 0 0 0
1 1 1 1
1 1 1 1
0 0 0 0
Column Control Pattern
2 rows stripe−1/0 pattern
(a)
Row Control Pattern
0
0
0
0
0 0 1 1
0 0 1 1
0 0 1 1
0 0 1 1
0 0 1 1
Compressed Column Stripe Pattern
2 columns stripe−0/1 pattern
(b)

Fig. 5: An example of test decompression for a 4×4 ubump array. (a) decompression of the compressed row stripe test pattern and (b) decompression of the compressed column stripe test pattern.

Please note that different logic gates can be used as the decompression logic gate and the row/column control pattern is the non-controlling value of the used logic gate. For example, one can use 2-input NOR gate as the decompression logic gate and the row/column control pattern will be all-0 since the non-controlling value of the NOR gate is 0. Furthermore, if one input of the NOR gate is 0, the output will be the complement of another input. Therefore, the compressed row stripe and column stripe test patterns should be complemented to obtain correct decompressed test patterns.

IV. Proposed BIST Scheme

A. *TX BIST Circuit*

For the testing of inter-die interconnects in chiplet-based ICs, the inter-die interconnects comprise two ubump arrays which can be divided into transmitter (TX) and receiver (RX) ubump arrays. The proposed BIST scheme consists of a TX BIST circuit and a RX BIST circuit. The TX BIST circuit is responsible for test application and the RX BIST circuit is used for analyzing and exporting the test response. For brevity, the BIST scheme for a 4×4 ubump array is used to explain the TX and RX BIST circuits. Fig. 6 shows the block diagram of the BIST circuit for TX ubumps. The control signals for the BIST circuit include Clk, Rst, BIST_Start, Continue, and Detect/Locate. The Detect/Locate signal determines whether the TX BIST circuit is executed in fault detection or fault location mode. The Continue signal is a communication signal sent from the RX BIST circuit to the TX BIST circuit. Test pattern generation stops when Continue is 1 and continues when Continue is 0. We have the following two modes in our BIST scheme:

1) Fault detection mode: The TX BIST circuit generates a test pattern every clock cycle. Continue is set to 1 if a fault is detected in the RX BIST circuit, stopping test pattern generation early in the TX BIST circuit. Otherwise, Continue remains 0, allowing the next test pattern to be generated in the next clock cycle.

2) Fault location mode: The TX BIST circuit generates a test pattern in a clock cycle. Continue is set to 1 to hold the current test pattern while the RX BIST circuit exports the captured response during the subsequent cycles. Once the export is complete, Continue is set to 0, allowing the next test pattern to be generated in the following clock cycle.

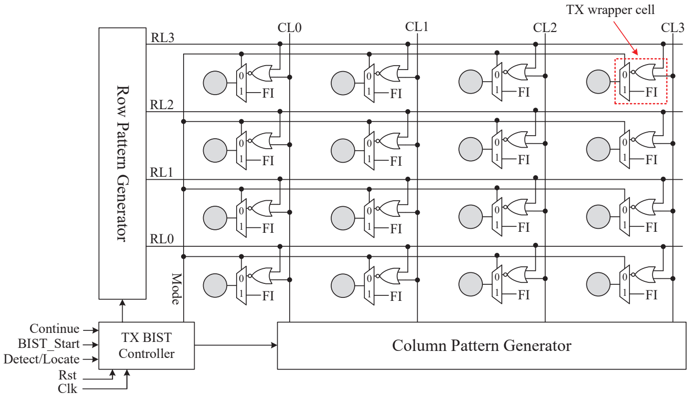

Fig. 6: Block diagram of the TX BIST circuit for a 4×4 ubump array.

The TX BIST circuit consists of a TX BIST controller, TX wrapper cells, a row pattern generator (RPG), and a column pattern generator (CPG). Each ubump is wrapped with a TX wrapper cell which consists of a multiplexer for the switching

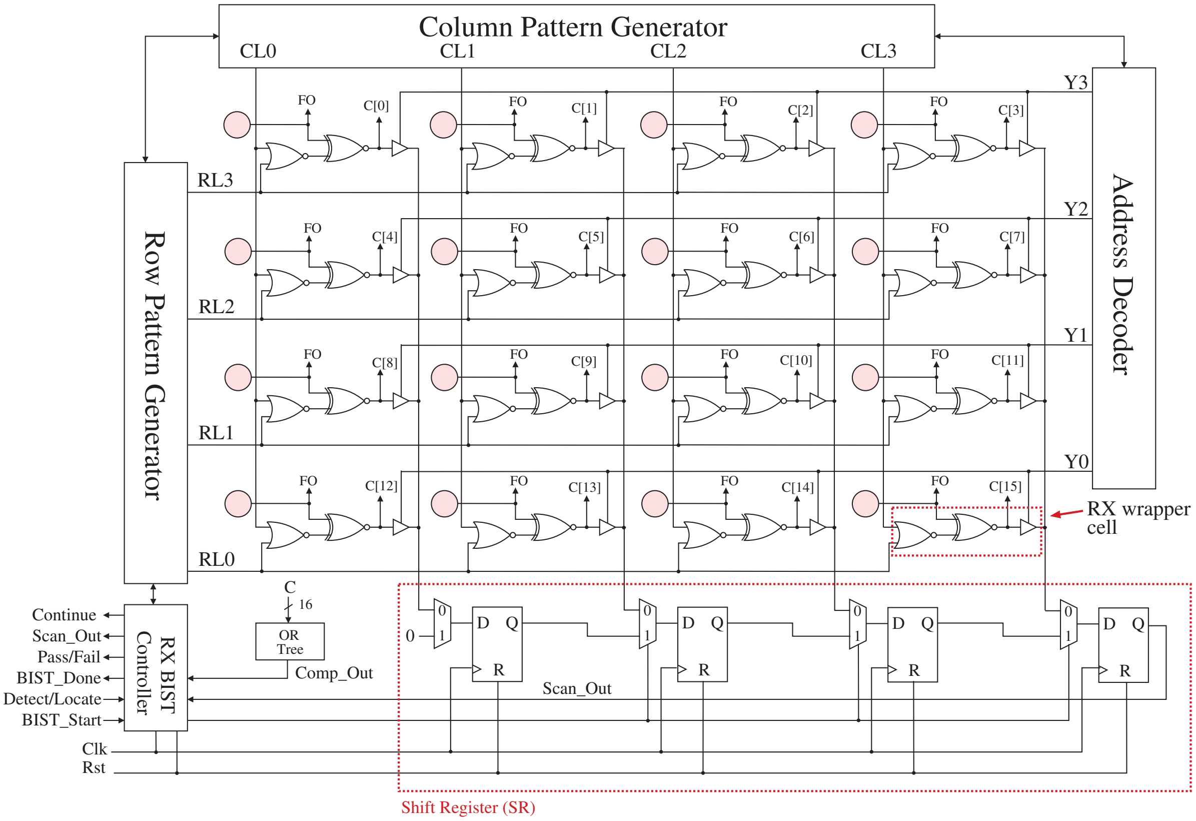

Fig. 7: Block diagram of the RX BIST circuit for a 4×4 ubump array.

between normal mode and test mode and a NOR logic gate for test decompression. The BIST controller generates control signals for the RPG and CPG. The RPG (CPG) can generate compressed row (column) stripe test patterns and row (column) control pattern. If the compressed row (column) stripe test patterns are generated by the RPG (CPG), the CPG (RPG) generates the column (row) control pattern. Then, the compressed row and column stripe test patterns are decompressed into the row and column stripe test patterns by the wrapper cells such that each ubump can receive the corresponding test data. Therefore, the test application time of one compressed test pattern only consumes 1 clock cycle. However, the hardware complexity of the RPG and CPG increases with an increase in the size of the ubump array, which slows down the operation speed. Nonetheless, this can be coped with by partitioning a large ubump array into multiple arrays. Furthermore, the hardwired RPG and CPG can only generate the designed test patterns. To enhance flexibility, users can choose to replace the RPG and CPG with shift registers to increase programmability of test patterns. However, if shift registers are used, the test application time is related to the length of the shift registers. Thus, there is a trade off between the test programmability and test time.

B. *RX BIST Circuit*

Fig. 7 depicts the block diagram of the RX BIST circuit for a 4×4 ubump array. The IOs of the RX BIST are Clk, Rst, BIST_Start, Continue, Detect/Locate, BIST_Done, Pass/Fail, and Scan_Out. In comparison with the TX BIST circuit, three addition outputs are needed, which are the BIST_Done indicating whether the BIST process is complete or not, the Pass/Fail denoting whether a fault is detected or not, and the Scan_Out being used to export the response in fault location mode. The RX BIST circuit comprises a RX BIST controller, RX wrapper cells, a RPG, a CPG, an OR tree, an address decoder, and a shift register (SR).

To achieve significant reduction in test time, the main objective is to apply test pattern in one clock cycle in the TX BIST and complete detection in the RX BIST within the same clock cycle in fault detection mode. Each ubump is wrapped with an RX wrapper cell consisting of a two-input NOR, an XNOR gate, and a tri-state buffer. Similar to the TX BIST circuit, the RPG and CPG generate golden data for the comparison with the received test data. Since the XNOR gate is used for the comparing function, the generated golden data should be the complement of the test patterns generated by the TX BIST circuit. Therefore, the XNOR gate outputs logic 0 if the corresponding inter-die interconnect is fault free. All the outputs of the XNOR gates of the ubump array under test are evaluated by the OR-tree whose output, Comp_O, which is connected to Pass/Fail to indicate whether a fault is detected or not in fault detection mode. In the fault location mode, the address decoder selects the tri-state buffers in a row each time and the corresponding XNOR outputs are captured to the first four bits of the SR. Then, the captured responses are exported through the Scan_Out. In the fault detection mode, Continue is 1 upon detecting a fault. In the fault location mode, Continue is 1 while exporting the captured response. Since the TX BIST circuit and RX BIST circuit are in different dies, we assume that the control of the two BIST circuits is through the IEEE 1500 test interface or IEEE 1149.1. As aforementioned, the Continue signal is the communication signal between the TX BIST and the RX BIST, which is a dynamic signal. In the IEEE 1500 test interface, we can implement a monitor register to monitor the Continue signal in the RX BIST circuit and send back to the TX BIST [19].

V. Simulation Results And Analysis

A. Analysis of the Area Cost

As the size of the ubump array increases, the area cost of DFT will be dominated by the wrapper cells. For each interconnect, IEEE 1500 requires two wrapper cells. Similarly, each of the TX and RX ubumps in the proposed BIST circuit has a wrapper cell. Our hardware implementation uses the TSMC 40nm CMOS standard cell library operating at a frequency of 500 MHz. A comparison of the area cost is shown in Fig. 8. Two IEEE 1500 one-storage wrapper cell has an area cost of 16.37 μm^2. On the other hand, the TX and RX wrapper cell of the proposed BIST circuit has an area cost of 6.05 μm^2, reducing the area cost by 63.0%.

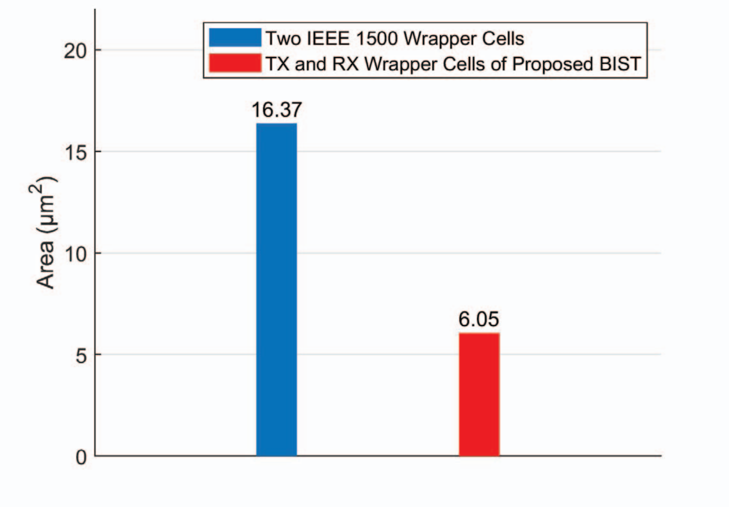

Fig. 8: Comparison of the area cost between two IEEE 1500 wrapper cells and the TX and RX wrapper cells.

A comparison of the area cost between the proposed BIST circuit and the IEEE 1500 test scheme with respect to different size of ubump arrays is shown in Fig. 9. The dashed line depicts the area cost of the proposed BIST circuit that has under 32 rows. For a 32×64 ubump array, the IEEE 1500 test scheme has an area cost of 33517 μm^2. On the other hand, the proposed BIST circuit has an area cost of 15590 μm^2, reducing the area cost by 53.5%. It is important to note that the area cost for the IEEE 1500 test scheme shown in Fig. 9 only considers the area cost of the wrapper cell, neglecting the area cost of the test control interface of the IEEE 1500. In practice, the area cost of the IEEE 1500 test scheme is expected to be higher than depicted in Fig. 9. This implies that the proposed BIST circuit can effectively reduce a larger amount of area costs in practice. The proposed BIST circuit can save an average of 52.8% in area costs. Furthermore, with the increase of the ubump array size, we can save even more in area costs. In [15], [17], testing delay faults caused by weak shorts/opens requires wrapping both ends of each interconnect with IEEE 1838 two-storage die wrapper registers (DWRs). However, the proposed BIST circuit offers a smaller area cost compared to this approach.

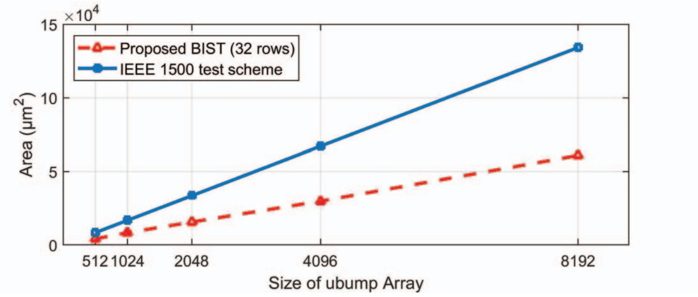

Fig. 9: Comparison of the area cost between the proposed BIST circuit and the IEEE 1500 test scheme with respect to different sizes ubump array.

B. Analysis of Test Time

Subsequently, the test application time of IEEE 1500 test scheme and the proposed BIST scheme is analyzed and compared. Assume a $h \times w$ ubump array is under test. For the IEEE 1500 test scheme, $(h \times w) \times (2 \times (\lceil \log_2 (h \times w) \rceil) + 1)$ cycles are required in the fault detection mode, where the "+1" indicates that the response exportation time of the last response. Conversely, the proposed ARCS test algorithm requires a significantly reduced test time of $(2 \times (\lceil \log_2 w \rceil + \lceil \log_2 h \rceil))$ cycles in fault detection mode. It has a test complexity of logarithmic function compared to a test complexity of square function in traditional methods. Test time can be significantly reduced in this case, as shown in Fig. 10. Using a 32×64 ubump array as an example, the IEEE 1500 test scheme requires 47104 cycles, while the ARCS test algorithm only requires 22 cycles. On average, the proposed BIST circuit can reduce detection time by 99.9%. For the proposed BIST scheme, however, the complexity of OR-tree for the evaluation of comparison results of ubumps is increased with the ubump array size. One approach to overcome this issue is to partition the OR-tree into multiple OR-trees and multiple clock cycles are needed to capture the response. Even if multiple OR-trees are used for test evaluation, the test time required by the proposed BIST scheme is still significantly shorter than that of the IEEE 1500 test scheme.

The test time required for the fault location in the IEEE 1500 test scheme is the same as the test time required for

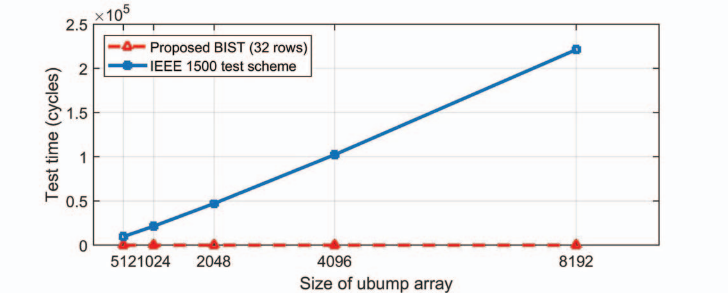

Fig. 10: Comparison of the test time between the proposed BIST and the IEEE 1500 test scheme with respect to different sizes of ubump array.

the fault detection. On the other hand, the test time required for the fault location mode in the ARCS Test Algorithm is $((1+1+w\times h)\times 2\times(\lceil\log_2 w\rceil+\lceil\log_2 h\rceil))$ cycles. The first "1" indicates that each time the test pattern is applied, one clock cycle is needed. The second "1" signifies the time required for the initial capture of the output response. Subsequent captures overlap with the time it takes to shift out the output response. As seen from the given equations, the test times required for fault location in both approaches are very similar. For a 32×64 ubump array, the IEEE 1500 test scheme requires 47104 cycles, while the ARCS test algorithm requires 45100 cycles. In [15], [17], only 16 test patterns are required to test each interconnect for hard or weak open/short defects. However, using a scan-based test application method will still demand a considerable amount of test application time.

VI. Conclusion

With the evolution of the advanced packaging technology, the 3D interconnect density is increased drastically. Conventional scan-based testing methods for the inter-die interconnects in chiplet-based ICs are, therefore, inefficient. We have proposed an efficient BIST scheme for the testing of inter-die interconnects in chiplet-based ICs. An ARCS test algorithm is proposed to detect and diagnose stuck-at and bridging faults of inter-die interconnects. The testing and diagnosis capability is the same as the widely-used true/complement binary counting sequence-based test algorithms. Furthermore, a simple BIST circuit can be realized to generate the ARCS test algorithm. In comparison with the typical IEEE 1500 test scheme, the proposed BIST scheme can achieve 99.9% test time reduction and save approximately 53.5% in area cost for a 32×64 ubump array.

Acknowledgment

This work was supported in part by the National Science and Technology Council (NSTC), Taiwan, R.O.C., under Contracts NSTC 113-2640-E-008-001, MOST 112-2221-E-008-096 and MOST 112-2218-E-002-025-MBK.

References

[1] R. Chaware and *et al.*, "Assembly challenges in developing 3D IC package with ultra high yield and high reliability," in *Proc. IEEE 65th Electronic Components and Technology Conference*, San Diego, May 2015, pp. 1447–1451.

[2] F.-C. Chen, M.-F. Chen, W.-C. Chiou, and D. C.-H. Yu, "System on integrated chips SoIC^{TM} for 3D heterogeneous integration," in *Proc. IEEE 69th Electronic Components and Technology Conference*, Las Vegas, May 2019, pp. 594–599.

[3] Y.-H. Chen, C.-A. Yang, C.-C. Kuo, M.-F. Chen, C.-H. Tung, W.-C. Chiou, and D. Yu, "Ultra high density SoIC with sub-micron bond pitch," in *Proc. IEEE 70th Electronic Components and Technology Conference*, Orlando, June 2020, pp. 576–581.

[4] P.-K. Huang and *et al.*, "Wafer-level system integration of the fifth generation CoWoS-S with high performance Si interposer at $2500mm^2$," in *Proc. IEEE 71st Electronic Components and Technology Conference*, San Diego, June 2021, pp. 101–104.

[5] S.-P. Jeng and M. Liu, "Heterogeneous and chiplet integration using organic interposer (CoWoS-R)," in *Proc. IEEE International Electron Device Meeting (IEDM)*, San Francisco, Jan. 2022.

[6] L. Lin, T.-C. Yeh, J.-L. Wu, G. Lu, T.-F. Tsai, L. Chen, and A.-T. Xu, "Reliability characterization of chip-on-wafer-on-substrate (CoWoS) 3D IC integration technology," in *Proc. IEEE 63rd Electronic Components and Technology Conference*, Las Vegas, May 2013, pp. 366–371.

[7] W. H. Kautz, "Testing for faults in wiring networks," *IEEE Trans. on Computers*, vol. c-23, no. 4, pp. 358–363, Apr. 1974.

[8] P. Goel and M. T. McMahon, "Electronic chip-in-place test," in *Proc. Int'l Test Conf. (ITC)*, 1982, pp. 83–90.

[9] P. Wagner, "Interconnect testing with boundary scan," in *Proc. Int'l Test Conf. (ITC)*, 1987, p. Paper 2.2.

[10] A. Hassan, J. Rajski, and V. K. Agarwal, "Testing and diagnosis of interconnects using boundary scan architecture," in *Proc. Int'l Test Conf. (ITC)*, 1988, pp. 126–137.

[11] W. T. Cheng, J. L. Lewandowski, and E. Wu, "Diagnosis for wiring interconnects," in *Proc. Int'l Test Conf. (ITC)*, 1990, pp. 565–571.

[12] Y.-J. Huang, J.-F. Li, J.-J. Chen, D.-M. Kwai, Y.-F. Chou, and C.-W. Wu, "A built-in self-test scheme for the post-bond test of TSVs in 3D ICs," in *Proc. IEEE VLSI Test Symp. (VTS)*, Dana Point, May 2011, pp. 20–25.

[13] Y.-J. Huang and J.-F. Li, "Built-in self-repair scheme for the TSVs in 3D ICs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 10, pp. 1600–1613, Oct. 2012.

[14] P.-Y. Chuang, F. Lorenzelli, S. Chakravarty, C.-W. Wu, G. Gielen, and E.-J. Marinissen, "Effective and efficienct testing of large number of inter-die interconnects in chiplet-based multi-die packages," in *Proc. IEEE VLSI Test Symp. (VTS)*, San Diego, Apr. 2023.

[15] P.-Y. Chuang, F. Lorenzelli, and E.-J. Marinissen, "Generating test patterns for chiplet interconnects: achieving optimal effectiveness and efficiency," in *IEEE International Test Conference in Asia (ITC-Asia)*, Matsue, Sept. 2023.

[16] P.-Y. Chuang, F. Lorenzelli, S. Chakravarty, C.-W. Wu, G. Gielen, and E.-J. Marinissen, "Effective and efficienct test and diagnosis pattern generation for many inter-die interconnects in chiplet-based packages," in *IEEE International 3D Systems Integration Conference (3DIC)*, Cork, May 2023.

[17] T.-H. Wang, P.-Y. Chuang, F. Lorenzelli, and E.-J. Marinissen, "Test and repair improvements for UCIe," in *Proc. IEEE European Test Symp. (ETS)*, Hague, May 2024.

[18] S. Chakravarty, "3D interconnect test challenge," in *Proc. IEEE European Test Symp. (ETS)*, May 2022.

[19] J.-F. Li, H.-J. Huang, J.-B. Chen, C.-P. Su, C.-W. Wu, C. Cheng, S.-I. Chen, C.-Y. Hwang, and H.-P. Lin, "A hierarchical test methodology for system-on-chip," *IEEE Micro*, vol. 22, no. 5, pp. 69–81, Sept./Oct. 2002.