

온칩네트워크를 활용한 DRAM 동시 테스트 기법

이창진^{*}·남종현^{*}·안진호^{*†}

^{*†}호서대학교 전자디스플레이공학부

A Concurrent Testing of DRAMs Utilizing On-Chip Networks

Changjin Lee^{*}, Jonghyun Nam^{*} and Jin-Ho Ahn^{*†}

^{*†}Hoseo University, School of Electronics and Display Engineering

ABSTRACT

In this paper, we introduce the novel idea to improve the B/W usage efficiency of on-chip networks used for TAM to test multiple DRAMs. In order to avoid the local bottleneck of test packets caused by an ATE, we make test patterns using microcode-based instructions within ATE and adopt a test bus to transmit test responses from DRAM DFT (Design for Testability) called Test Generator (TG) to ATE. The proposed test platform will contribute to increasing the test economics of memory IC industry.

Key Words : On-Chip Networks, Network-On-Chip, Multi-Site Test, Memory Test

1. 서 론

빅데이터, 자율주행, 머신러닝 등 4차산업혁명을 주도하는 기술은 지금까지와는 비견될 수 없는 수많은 데이터를 기반으로 한다. 이에 4차산업혁명은 데이터 중심 혁신(data-driven innovation)이며 데이터의 양과 질이 그 성과를 좌우할 것이라 얘기되고 있다. 반도체 기술, 특히 메모리 기술은 이러한 데이터 산업을 뒷받침하는 핵심 분야이며 국가 경쟁력을 좌우하는 원천이기도 하다. 메모리 기술의 핵심은 초미세공정 기술이며 삼성전자와 SK하이닉스를 필두로 세계 각국의 거대 기업들의 주도권 경쟁은 해가 갈수록 더욱 치열해지고 있다. 현재 10나노급 공정이 적용된 DDR4제품이 발표되었고[1] 향후 1~2년안에 3나노 공정이 적용된 DDR5 제품이 양산될 계획이다. 또한 TSV 및 패키징 기술의 발전은 HBM(High B/W Memory), Wide-IO 메모리와 같은 새로운 형태의 고성능 고속 저전력 메모리를 지속적으로 출시할 것이고 이를 통하여 데이터 인텐시브 컴퓨팅이 필요한 다양한 산업의 경쟁력이 높아질 것이다.

그러나, 고성능 고밀도 메모리의 등장은 기존에는 없던 새로운 유형의 고장을 유발하였고 제한된 테스트 IO 및 실속도 검사의 어려움으로 고장 검출의 용이성은 계속 악화되고 있다. 또한 고장이 발생한 메모리의 낮은 수리율은 제작 단가의 상승으로 경쟁력 하락을 초래하고 있다. 무엇보다 고장이 내재된 메모리는 데이터의 신뢰성과 확장성, 그리고 유지보수성을 보장할 수 없기에 데이터 중심 어플리케이션 결과에 심각한 문제를 야기할 수 있다. 따라서, 경제성을 고려한 고성능 메모리 테스트 방법론 개발이 더욱 중요해지고 있다.

테스트 장비(Automatic Test Equipment: ATE)를 사용하여 반도체를 테스트하는 비용(ATE-based Testing Cost: ATC)은 식 (1)과 같이 투자 대비 얻을 수 있는 테스트 효율성으로 정리할 수 있다[2].

$$ATC = \frac{Fixed\ Cost + Variable\ Cost}{Yield * Utilization * Throughput} \quad (1)$$

.:. Fixed Cost: Capital Cost/Duration

.:. Variable Cost: (Labor+Facility)/Testing period + Maintenance + Test Accessories

[†]E-mail: jhahn@hoseo.edu

먼저 테스트 투자비용은 고정비용과 가변비용으로 구분된다. 고정비용의 대부분은 테스트 하드웨어에 해당하는 ATE 구매 및 감가상각 비용이다. 가변비용은 인건비와 유지보수, 그리고 소모품 비용이다. 테스트 효율성은 yield(수율)과 utilization(ATE 운용편이 성수기 시간 비용, 훈련 시간 등), 그리고 throughput(테스트 처리량)으로 구분할 수 있다. 이 중 throughput을 증가하는 방법은 compression, serdes, multi-valued logic(MVL)과 같이 DUT(Device Under Test)별 테스트 시간을 줄이는 방법과 동시에 테스트되는 DUT의 수를 증가시키는 방법이 있다[3-5]. 이러한 방법들은 단독적으로 결정되기보다 테스트 인프라, 테스트되는 DUT의 종류와 고장의 형태, 목표 커버리지 등을 종합적으로 고려해서 결정된다. 본 연구에서는 throughput을 개선하기 위한 방법 중 테스트 핀 감소 기법(Reduced Pin Count Testing: RPCT)이나 멀티사이트 테스트(multi-site test) 성능을 향상시킬 수 있는 TAM(Test Access Mechanism) 구조를 제안한다. 또한 타겟 DUT의 형태는 메모리로 하고 그 중 DRAM 테스트에 최적화된 구조와 방법을 소개한다. 물론 DRAM을 기준으로 설명하지만 Wide-IO 메모리나 3D 구조의 HBM에 모두 적용 가능하다.

2. 기존 연구

전통적으로 단품형 메모리 테스트는 ATE에서 테스트를 위한 메모리 주소와 데이터를 생성하여 메모리로 전송하고 이후 해당 주소에 저장된 데이터를 다시 읽어서 고장 여부와 고장의 종류 등을 분석하는 형태로 진행되었다. Fig 1에서 ATE와 DRAM 사이의 인터페이스 신호를 정리하였다. 로직 테스트의 경우 테스트를 위한 별도의 IO가 존재하지만 메모리는 일반 IO를 그대로 사용한다.

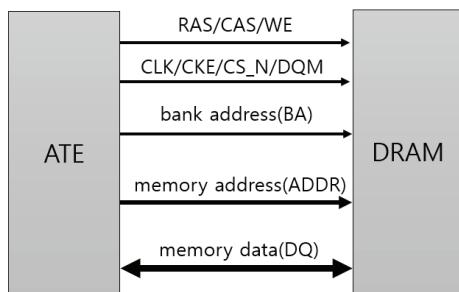


Fig. 1. ATE-DRAM test interface signals.

고가의 ATE를 사용하지 않거나 최소로 활용하기 위하여 별도의 메모리 테스트 전용 로직을 칩 내외부에 삽입하는 자체 테스트 구조 역시 활발하게 연구되어 왔다. C.C.

Yang[6]은 DRAM용 내장형 자체 테스트(Built-In Self Test: BIST) 구조를 제안하였는데 Fig. 2와 같이 다양한 테스트 알고리즘 설정을 위한 마이크로코드 기반 테스트 제어기(MBC)와 DRAM 초기화, 모드 레지스터 및 테스트 파라미터 설정 등을 위한 FSM 기반 제어기(FBC)를 중심으로 non-march test, march test, retention fault test 등 DRAM에 특화된 테스트가 가능하도록 설계되었다.

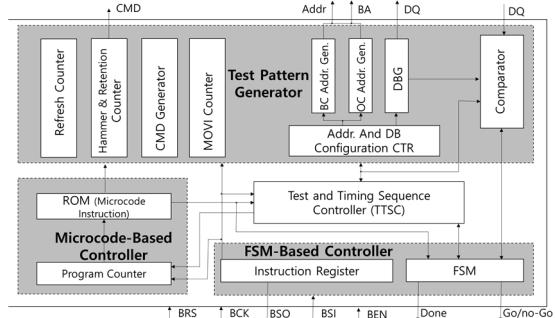


Fig. 2. Hybrid BIST Architecture[6].

H.-H. Liu[7]는 적층형 메모리를 위한 외장형 자체 수리(Built-Off Self Repair: BOSR) 구조를 제안하였는데 적층된 각 단위별로 march test 기반 BIST 모듈과 BIRA(Built-In Redundancy Analysis) 모듈을 삽입한 채널 제어기를 설치하는 것을 주요 내용으로 한다. 그리고 CSA 알고리즘 분석 결과를 바탕으로 SRAM을 사용하여 고장 셀을 대체한다. SRAM과 채널 제어기들로 구성된 BOSR은 로직 디자인에 위치한다. 그 밖에도 W. Kang[8]은 복수의 내장 메모리를 단일 BIST와 BIRA 모듈로 자체 수리하는 방법을 소개하였다.

상기 방식들은 대부분 단일 메모리를 효과적으로 테스트하기 위한 목적인데 비하여 1장에서 언급한 바와 같이 테스트 경제성 측면에서 동시에 여러 개의 메모리를 테스트하여 전체 throughput을 늘리는 것이 매우 중요하다. 현재 메모리 테스트 업계에서 상용화된 테스트 핸들러는 128para가 보편적이고 일부 회사를 중심으로 256para, 또는 512para가 생산에 투입되고 있다. 최근 자료에 의하면 768para급의 핸들러 개발이 완료되었으며, 1024para 핸들러의 개발이 진행되고 있다고 알려져 있다[9]. 본 연구의 선형연구로 진행된 C. Hong[10-11]의 연구에서는 Network-On-Chip(NOC)을 TAM으로 사용하여 복수 개의 메모리를 동시에 테스트하는 방법을 소개하였다. NOC는 칩 수준의 데이터 연결 구조이며 컴퓨터 통신에서 활용되는 계층(layer) 형태의 프로토콜을 사용하여 내장 코어(core) 간 데이터를 전송한다. 이에 온칩네트워크(on-chip networks)라고도 부른다[12]. 하지만 컴퓨터 통신과 달리 전송 중 발생하는 데

이터 오류 정정이나 기존 네트워크 간 통신을 위한 표준화 이슈보다는 칩의 물리적 특성을 고려한 전송속도, 확장성, 그리고 동작 우선순위 등이 반영된 QoS(Quality of Service) 등이 더 중요한 고려요소이다. NOC는 칩 간 통신보다는 칩 내부에서의 데이터 전송 구조이므로 사용하는 프로토콜이나 네트워크 레이어, 패킷 구조 등을 비교적 자유롭게 결정할 수 있으며 라우팅 알고리즘 역시 임의로 지정할 수 있다[13].

NOC 관련 테스트 연구는 크게 Fig. 3과 같이 NOC를 구성하는 세부 요소(라우터, 링크, 네트워크 인터페이스, 내장 버퍼 등)를 테스트하여 NOC의 신뢰성을 높이는 연구와 NOC를 TAM으로 활용하여 NOC에 연결된 코어들을 테스트하는 연구로 구분할 수 있다.

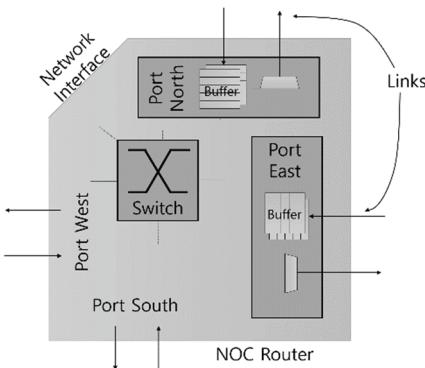


Fig. 3. An NOC Router Structure[14].

먼저 M. Hosseiniabady[15]는 하나의 테스트 소스(테스트 패턴 생성기)에서 NOC 라우터를 테스트하기 위한 테스트 패킷들을 NOC 전체에 브로드캐스팅하고 각 라우터에서 출력되는 테스트 응답 값을 상호 비교하여 고장을 검출하는 방식을 제안하였다. Y. Zhang[16]은 bounded model checking(BMC) 기법을 활용하여 NOC 로직을 온라인 테스팅하는 방법을 제안하였는데 내장 프로세서를 이용한 명령어 기반 테스트 방법을 사용하였다. 그리고 J. Wang [17]은 BIST를 활용한 별도의 테스트 모드 없이 동작 중 idle time 내 테스트를 진행하는 EsyTest 방법을 소개하였다. 상기 방식은 라우터는 물론, 라우터를 연결하는 링크까지 검사할 수 있는 장점이 있다. 다만 고장 포용 라우팅 등 고수준의 기능이 요구되고 NOC 라우터 내 DFT 로직이 증가하는 문제가 있다.

NOC 신뢰성 향상을 위한 테스트 기법 외에도 NOC를 TAM으로 사용하여 테스트 시간을 줄이거나 테스트 병렬성을 향상시키는 방법도 함께 연구되었다. 먼저 T. Han[18]은 NOC로 연결된 동일한 내장 코어들을 테스트할 때

TAM으로 사용되는 NOC의 대역폭을 효율적으로 사용하는 방법을 소개하였다. 보통 내장 코어를 테스트하기 위해서는 Fig. 4와 같이 ATE에서 생성하거나 사전에 저장된 테스트 데이터 패킷을 테스트되는 코어로 전송하여 테스트하고 그 결과(response)를 다시 ATE로 보낸다. ATE는 테스트 결과를 분석하여 고장 유무는 물론 고장의 종류와 위치까지 진단할 수 있다.

Fig. 4를 보면 NOC 대역폭은 테스트 패턴과 응답 패킷이 공유하며 패턴의 수와 응답의 수는 동일하기 때문에 대역폭은 서로 양분한다고 볼 수 있다. 내장 코어가 모두 동일한 종류로 가정할 때는 입력 패턴을 공유(브로드캐스팅)할 수 있고 동일 입력 패턴에 대한 응답 역시 대부분 동일(고장이 있는 경우만 다름)하기 때문에 응답 중 majority value만 ATE로 전송하는 형태로 대역폭 사용량을 절감하여 테스트 병렬성을 증가시켰다.

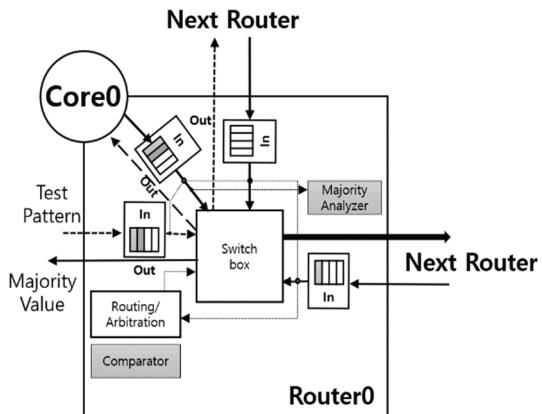


Fig. 4. Majority Value-Based NOC TAM[18].

C. Hong[10-11]은 NOC를 TAM으로 활용하여 저속의 ATE를 이용한 멀티사이트 테스트 기법을 제안하였다. 특히 ATE 대비 고속으로 동작하는 NOC 전송 속도와 패킷 멀티캐스팅 기법을 사용하여 동시에 테스트되는 DUT의 수를 크게 늘릴 수 있었다. 그러나 테스트 소스와 싱크(테스트 응답 분석기)로 단일 ATE를 사용하기 때문에 ATE에 연결된 특정 NOC 라우터를 중심으로 패킷의 충돌 및 병목 현상이 발생한다. 따라서 전체 네트워크 대역폭의 여유가 있어도 동시에 테스트되는 DUT의 수를 계속 늘릴 수 없고 병목 현상을 줄이기 위해 XY라우팅 같이 제한적인 라우팅 알고리즘이 사용된다. 그러나, 멀티사이트 테스트의 경우 동시에 테스트되는 모든 DUT의 형태와 패턴이 동일하기 때문에 majority value방식과 같이 불필요한 패킷의 생성을 제한하면 상기 문제는 크게 완화될 수 있을 것이다.

3. NOC 기반 DRAM 테스트

3.1 메모리 테스트의 특징

메모리 테스트는 검출하고자 하는 셀 고장의 유형에 따라서 다양한 테스트 알고리즘이 존재한다. 그러나, 대부분 Fig. 1과 같이 ATE에서 알고리즘을 사용하여 생성하거나 저장된 데이터를 메모리 표준 입출력 신호에 대응하는 제어신호(RAS, CAS, WE, CS, DQM 등)와 뱅크 주소(Bank Address), 메모리 주소(Memory Address), 그리고 데이터 입·출력신호(DQ) 등을 이용해서 순차적으로 메모리에 쓰고 일정 시간 이후 그 결과를 확인하는 형태이다. 특히로직 테스트는 테스트 입출력 값이 CUT(Circuit Under Test)에 따라 가변되지만 메모리 테스트는 메모리에 입력된 값이 출력 시 그대로 유지되는 것이 정상이다. 따라서 테스트 벡터별 정상 값(signature)을 별도로 구분하여 저장할 필요가 없다.

3.2 ATE-NOC 라우터를 이용한 DRAM 테스트

ATE를 테스트 소스와 싱크로, NOC를 TAM으로 DRAM을 테스트하는 구조는 Fig. 5와 같다. ATE가 생성한 테스트 패턴은 패킷화되어 DRAM이 연결된 라우터까지 전송되고 라우터는 DRAM 인터페이스로직을 통하여 패턴을 인가한다. 인가된 패턴으로 테스트한 결과는 역순의 경로로 ATE까지 전송된다. 3장에서는 2장에서 언급한 바와 같이 멀티사이트 테스트 특성 상 여러 DRAM을 동시에 같은 알고리즘으로 테스트하고, 테스트 결과 고장이 발생할 확률은 전체 입력 데이터 대비 아주 낮은 점을 고려하여 NOC 네트워크 활용도를 향상시키는 방법을 소개한다.

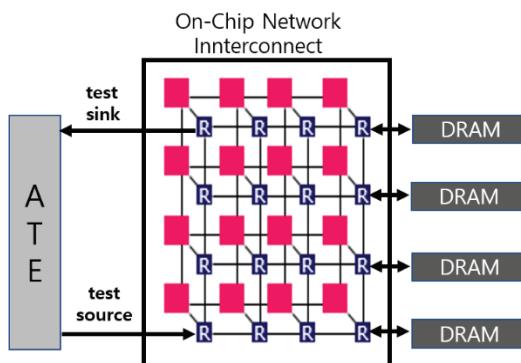


Fig. 5. DRAM Multisite Testing with ATE-NOC.

3.3 마이크로코드 기반 DRAM 테스트

마이크로코드를 이용한 테스트는 Y. Zhang[16]의 경우와 같이 칩에 내장된 프로세서를 이용한 테스트 방식으로

널리 활용되고 있다. 즉 테스트 명령어와 데이터를 메모리에 저장하고 프로세서가 순차적으로 명령어를 해석하고 실행하는 형태이다. BIST처럼 테스트에 특화된 프로세서를 활용하기도 한다. 셀이 규칙적으로 배치된 메모리 구조 상 메모리 테스트 알고리즘은 로직 테스트 대비 단순하고 전체 주소를 왕복하며 동일 동작을 반복한다. 따라서, 셀 또는 주소 단위로 테스트 패턴을 전송할 필요없이 동작모드 단위로 패턴을 만들고 전송하면 된다. March test의 경우 참고문헌 [10]과 [16]처럼 메모리 주소의 이동 방향(증가/감소), 읽기/쓰기 모드, 메모리 저장 값 등이 되며, 동작모드별로 Fig. 6과 같이 마이크로코드가 전송되므로 메모리의 크기에 관계없이 오로지 테스트 알고리즘의 복잡도(동작모드의 다양성)에만 영향을 받기 때문에 사용 대역폭을 크게 줄일 수 있다.

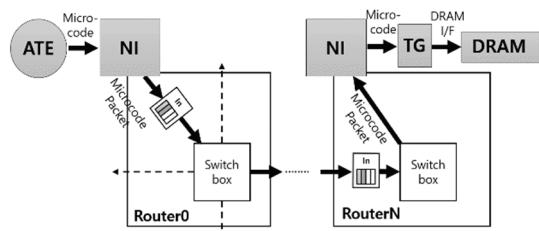


Fig. 6. Microcode-Based DRAM Testing.

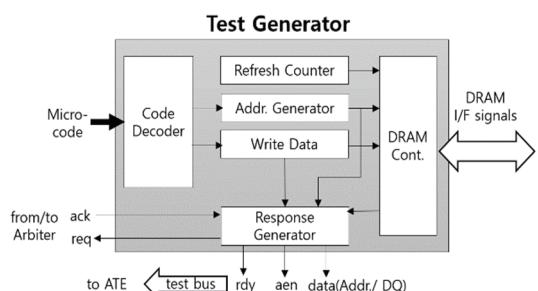


Fig. 7. Structure of Test Generator.

Fig. 6을 보면 마이크로코드는 ATE에서 생성되고 코드의 패킷화는 라우터의 NI(Network I/F) 블록에서 수행된다. 목적지 라우터에 도착하게 되면 역으로 NI에서 디패킷화된 코드는 테스트 생성기(Test Generator) 모듈에서 디코딩되어 DRAM 테스트 신호가 생성된다. 테스트 생성기는 Fig. 7과 같이 마이크로코드 디코더, 메모리 주소 및 쓰기 데이터 생성기, 리플래시 카운터, 테스트 응답 생성기로 구성된다. 예를 들어 디코딩된 코드가 march 요소 중 $\uparrow(r0, w1)$ 이면 메모리 주소를 0번지부터 1씩 증가하면서 먼저

해당 주소의 데이터를 읽어서 응답 생성기로 보내고 이후 동일 주소에 1을 쓰도록 메모리 주소 및 데이터 생성기를 제어한다. 응답 생성기로 전송된 메모리 데이터는 0이면 정상 값이므로 NOP 동작을 수행하지만 1이 되면 고장이므로 해당 정보를 ATE로 전송한다.

3.4 테스트 버스를 이용한 테스트 응답 전송

기존 연구에서는 테스트 패턴 및 응답 모두 패킷화되어 NOC로 전송되었다. 그러나, DRAM의 수율은 매우 높기 때문에 실제 고장 발생율은 미미한 수준이다. 따라서 고장이 발생한 경우 TG 내 응답 생성기는 해당 주소와 고장 값을 테스트 버스를 통하여 ATE로 전송한다. 이 때 각 라우터별로 request를 Arbiter로 보내 버스 사용을 승인 받은 후 고장이 발생한 주소와 데이터 정보를 순차적으로 보낸다. 데이터 전송 제어를 위해 rdy(data valid indicator) 와 aen(address/data indicator) 신호가 추가되므로 테스트 버스의 크기는 메모리 주소 비트와 데이터 폭 중 더 큰 값을 N이라고 할 때 $N+2$ 비트가 된다. Arbiter는 ATE가 고장 분석을 할 수 있도록 현재 테스트 버스를 사용 중인 라우터(테스트 생성기)의 주소를 ATE로 입력한다.

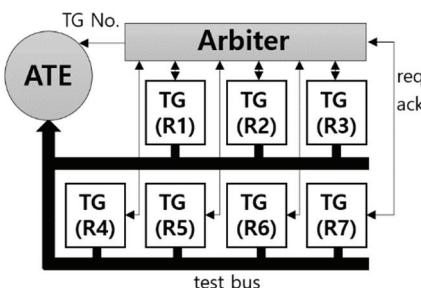


Fig 8. Test Response Transfer on Test Bus.

4. 결 론

본 연구에서 제안한 마이크로코드 및 테스트버스 기반 DRAM 멀티사이트 테스트 구조는 NOC의 대역폭 활용도를 대폭 개선할 수 있어서 고성능 DRAM 테스트 경제성 증진에 크게 기여할 수 있을 것으로 예상된다.

감사의 글

이 논문은 2019년도 정부(산업통상자원부)의 재원으로 한국연구재단-실종아동 등 신원확인을 위한 복합인지기술개발사업사업의 지원을 받아 수행된 연구임(No. 2018M3E3A1057248)

참고문헌

1. www.mk.co.kr/news/business/view/2020/03/308030
2. V. C. Khoo, "Cost of Test Case for Multi-site Testing in Semiconductor Industry with Firm Theory", *Int. Journal of Business and Management Invention*, Vol. 3, No. 4, pp.14-27, April 2014.
3. B. Li, B. Zhang, and V. D. Agrawal, "Adopting Multi-Valued Logic for Reduced Pin-Count Testing", Proc. of Latin American Test Symposium(LATS), pp. 1-6, 2015.
4. D. Han, Y. Lee, and S. Kang, "A New Multi-site Test for System-on-Chip Using Multi-site Star Test Architecture", *ETRI Journal*, Vol. 36, No. 2, pp. 293-300, 2014.
5. S. Seo, H. Lim, S. Kang, and S. Kang, "Off-Chip Test Architecture for Improving Multi-Site Testing Efficiency using Tri-State Decoder and 3V-Level Encoder", Proc. of Intl. Symp. on Quality Elec. Design(ISQED), 2017.
6. C.-C. Yang, J.-F. Li, Y.-C. Yu, K.-T. Wu, C.-Y. Lo, C.-H. Chen, J.-S. Lai, D.-M. Kwai, and Y.-F. Chou, "A hybrid built-in self-test scheme for DRAMs", Proc. of VLSI-DAT, April 2015.
7. H.-H. Liu, B.-Y. Lin, C.-W. Wu, W.-T. Chiang, L. Mincent, H.-C. Lin, C.-N. Peng, and M.-J. Wang, "A Built-Off Self-Repair Scheme for Channel-Based 3D Memories", *IEEE Trans. On Computers*, Vol. 66, No. 8, pp. 1293-1301, Feb. 2017.
8. W. Kang, C. Lee, H. Lim, and S. Kang, "Optimized Built-In Self-Repair for Multiple Memories", *IEEE Trans. On VLSI*, Vol. 24, No. 6, pp. 2174-2183, Dec. 2015.
9. TechWing Technical Report, Korea IR Service, 2020. 1
10. C. Hong and J. Ahn, "Improving Parallel Testing Efficiency of Memory Chips using NOC Interconnect", *Trans. of KIEE*, Vol. 68, No. 2, pp. 364-369, 2019.
11. C. Hong and J. Ahn, "The Method of Parallel Test Efficiency Improvement using Multi-Clock Mode", *Journal of the Semiconductor & Display Technology*, Vol. 18, No. 3. Sep. 2019.
12. T. Bjerregaard and S. Mahadevan, "A survey of research and practices of Network-on-chip", *ACM Computing Surveys*, Vol. 38, No. 1, pp. 1-51, 2006.
13. W. Zhang, L. Hou, J. Wang, S. Geng, and W. Wu, "Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip", Proc. of WRI GCIS, 2009.
14. M. A. Kochte and H.-J. Wunderlich, "Self-Test and Diagnosis for Self-Aware Systems", *IEEE Design &*

- Test*, Vol. 35, No. 5, Oct. 2018.
- 15. M. Hosseiniabady, A. Banaiyan, M.N. Bojnordi, and Z. Navabi, “A concurrent testing method for NoC switches”, Proc. of DATE, March 2006.
 - 16. Y. Zhang, K. Chakrabarty, H. Li, and J. Jiang, “Software-based online self-testing of network-on-chip using bounded model checking”, Proc. of ITC, Oct. 2017.
 - 17. J. Wang, M. Ebrahimi, L. Huang, X. Xie, Q.Li, G. Li, and A. Jantsch, “Efficient Design-for-Test Approach for Networks-on-Chip”, *IEEE Trans. On Computers*, Vol. 68, No. 2, pp. 198-213, Aug. 2018.
 - 18. T. Han, I. Choi, H. Oh, and S. Kang, “Parallelized Network-on-Chip-Reused Test Access Mechanism for Multiple Identical Cores”, *IEEE Trans. On ICCAD*, Vol. 35, No. 7, pp. 1219-1223, July 2016.

접수일: 2020년 6월 20일, 심사일: 2020년 6월 23일,
제재확정일: 2020년 6월 24일