

논문 2006-43SD-11-9

다중고장에 대한 효율적인 자가치유시스템 설계

(Design of efficient self-repair system for multi-faults)

최 호 용*, 서 정 일**, 유 충 호**, 우 철 종**, 이 재 은**

(Ho-Yong Choi, Jung-Il Seo, Chung-Ho Yu, Cheol-Jong Woo, and Jae-Eun Lee)

요 약

본 논문에서는 생명체의 구조를 모방하여, 다중고장에 대한 효율적인 자가치유시스템 구조를 제안한다. 기존의 자가치유시스템은 인공 셀의 데이터의 패스가 좌우상하로만 가능하여 열 단위 치유만이 가능했다. 또한, 인공 셀의 게놈블록의 설계에 있어서, gene정보 저장방식을 이용함으로써 하드웨어 오버헤드가 큰 문제점이 있다. 본 논문에서는 인공 셀의 데이터의 패스를 대각선으로도 가능하게 설계하여, 열 단위가 아닌 셀 단위로 고장 셀의 용이한 대체가 가능하고 다중고장에 대해서도 효율적으로 자가치유가 가능하다. 또한, 게놈블록은 로직기반 생성방식을 이용하여 설계함으로써 저면적 설계가 가능하다. 2진 up/down 카운터를 예제로 하여 설계한 결과 하드웨어 오버헤드를 약 40.3% 개선했다.

Abstract

This paper proposes a self-repair system which is able to self-repair in cell unit by imitating the structure of living beings. Because the data of artificial cells move even diagonally, our system can self-repair faults not in column unit, but in cell unit. It leads to design an efficient self-repair system for multiple faults. Moreover, in artificial cell design, the usage of logic-based design method has smaller system size than that of the previous register-based design method. Our experimental result for 2-bit up/down counter shows 40.3% reduction in hardware overhead, compared to the previous method [6].

Keywords : self-repair, artificial cell, spare cell, multiple faults, genome block

I. 서 론

최근, 반도체 공정기술과 고집적회로기술의 발달로 회로가 대규모화됨에 따라 전자시스템들의 고장 발생은 더욱 증가되고 있다. 따라서 회로의 지속적이고 신뢰성 있는 동작이 중요하게 되어지고 이를 만족하는 고장포용(fault tolerance)시스템 설계기술 또한 더욱 더 중요해지고 있다. 이러한 고장포용기술은 항공기 혹은 우주선과 같이 극도의 안전을 요하는 시스템에서 그 중요성

이 더욱 부각된다. 종전의 고장포용기술들은 Active, TMR (triple modular redundancy), 고장검출 및 정정 코드, 고장자가검사회로에 의한 방법들이 있다^[1-3]. 그러나 이러한 방법들은 하드웨어 오버헤드가 커, 비용이 비싼 단점이 있다.

최근, 고장포용시스템을 이용하여 비교적 적은 하드웨어로 높은 신뢰성과 지속성을 유지하는 면역전자학(immunotronics)과 태생전자학(embryonics)기술이 활발히 연구되고 있다^[4-7].

면역전자학(immunotronics)은 외부 바이러스의 침입으로부터 자신을 보호하고 감염에 대해 스스로 치유할 수 있는 인간의 면역시스템을 모방하여, 고장을 스스로 검출하고 고립시키며 치유할 수 있는 디지털시스템을 구현하는 기술이다^[4].

태생전자학은 생명체의 구조와 특징을 이용하여 디지털시스템으로 구현하는 기술이다^[5-7]. 태생전자학에서는 시스템을 생명체와 유사하게, 인공 셀(artificial cell)

* 평생회원, 충북대학교, 전기전자컴퓨터공학부
(School of Electrical and Computer Eng.,
Chungbuk National University)

** 학생회원, 충북대학교, 반도체공학과
(Semiconductor Eng., Chungbuk National
University)

※ 이 논문은 2005학년도 충북대학교 학술연구지원사업의 연구비지원과 IDEC의 틀지원에 의하여 연구되었음.

접수일자: 2006년8월1일, 수정완료일: 2006년10월25일

로 시스템을 구성하고 여분의 셀(spare cell)을 이용하여 고장치유를 하는 고장포용시스템을 구현한다. 기존의 자가치유시스템은 자가검출과 자가치유는 가능하지만, 치유를 하기 위해 고장 난 인공 셀이 포함된 열전체를 대체하는 열 단위 치유를 하고 또한, 게놈 블록이 자신의 gene정보와 이웃의 gene정보를 모두 레지스터에 저장하고 있기 때문에 하드웨어 오버헤드가 상당히 증가한다는 문제점을 가지고 있다^{5, 6}.

본 논문에서는 태생전자학기술을 기반으로, 대각선으로도 데이터의 패스가 가능한 인공 셀을 설계하여 열 단위 치유가 아닌 셀 단위 치유가 가능하게 함으로써 다중고장에 대해 효율적으로 치유하고, 게놈블록을 로직기반 생성방식으로 설계함으로써 하드웨어 오버헤드를 개선한 효율적인 자가치유시스템을 설계한다.

본 논문의 구성은 다음과 같다. II장에서는 기존 자가치유시스템을 살펴보고, III장에서는 다중고장에 대한 효율적인 자가치유시스템에 대해 기술한다. IV장에서는 제안하는 구조를 이용하여 설계된 2-bit 업다운 카운터의 시뮬레이션 결과를 설명한다. V장에서는 실제 제작한 칩 설계의 테스트 결과를 설명하며, 끝으로 VI장에서 결론을 맺는다.

II. 기존의 자가치유시스템

그림 1은 기존의 일반적인 구조의 자가치유시스템이다.

자가치유시스템은 재구성 가능한(reconfigurable) 인공 셀과 치유를 위한 여분 셀로 구성된다. 배열내의 모든 인공 셀은 자신의 제어레지스터뿐만 아니라, 고장이 발생하였을 경우 그 기능을 대신할 수 있도록 이웃의 제어레지스터까지 가지고 있다.

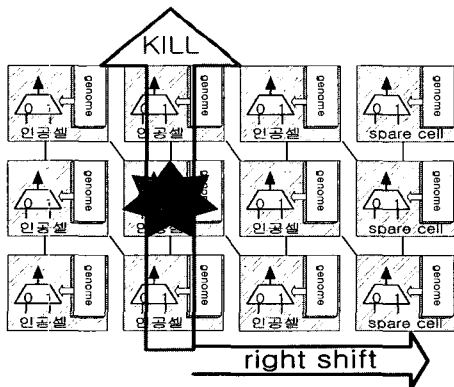


그림 1. 기존의 자가치유시스템 구조
Fig. 1. The structure of previous self-repair system.

인공 셀의 기능은 배열내의 자신의 위치에 따라 제어 레지스터를 선택함으로써 이루어진다.

만일 시스템에 임의의 인공 셀에서 고장이 발생하면 고장 검출기에 의해 고장이 검출되고, 해당 인공 셀이 속한 열의 모든 인공 셀은 바이패스모드로 전환된다. 그리고 고장이 난 인공 셀의 우측에 위치한 인공 셀 열은 고장 난 셀이 속한 인공 셀 열의 동작을 수행하게 되고, 여분의 셀 열을 이용하여 시스템을 재구성하는 열 단위 자가치유 메카니즘을 갖는다. 따라서, 기존의 자가치유시스템은 한 열에서 하나의 인공 셀에만 고장이 발생해도 한열의 여분의 셀 모두가 필요하여 비효율적으로 치유할 수밖에 없다. 또한 게놈블록을 레지스터 기반의 저장방식으로 설계하여 하드웨어 오버헤드가 커지는 문제점이 있다.

III. 다중고장에 대한 효율적인 자가치유시스템

1. 자가치유시스템의 구조 및 설계

본 절에서는 열 단위 치유가 아닌 셀 단위 치유를 하고 로직기반의 생성방식 게놈블록을 사용하여 하드웨어 오버헤드문제를 개선한 효율적인 자가치유시스템의 구조를 제안한다.

(1) 자가치유시스템의 구조

그림 2는 제안하는 자가치유시스템의 구조를 보여준다.

자가치유시스템은 acell(인공 셀)의 배열들과 최우열(最右列)의 scell(여분 셀) 배열로 구성된다.

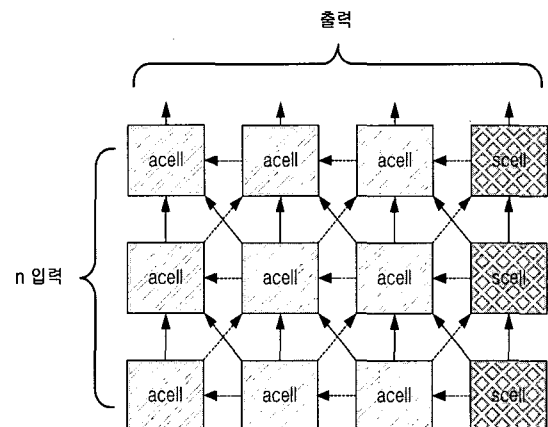


그림 2. 제안하는 자가치유시스템의 구조
Fig. 2. The structure of proposed self-repair system.

acell 및 scell은 좌우상하와 대각선의 데이터 패스가 가능하고, 정상 동작 시 실선의 데이터 패스를 유지하며 고장 동작 시에는 점선의 데이터 패스로 재구성하여 시스템이 치유된다.

제한하는 자가치유시스템은 인공 셀의 데이터 패스를 대각선 방향으로도 가능하게 함으로써, 고장이 발생할 경우 열 단위 치유가 아닌 셀 단위 치유가 가능하다.

그러므로 n행으로 구성된 회로의 경우 하나의 여분 셀 열로 행 당 1개씩의 고장 치유가 가능하며, n개까지의 고장 셀에 대해 치유가 가능하다 (그러나 같은 행에 m개의 고장이 났을 경우에는 여분 셀 열을 m개를 가져야만 치유할 수 있다).

이는, 기존구조^[6]에서 하나의 고장 셀만을 치유할 수 있었던 것에 비해 여분 셀 열을 가지고 n개까지 치유할 수 있는 효율적인 구조가 된다.

(2) 정상동작 시의 데이터패스

그림 3은 고장이 없는 정상동작 시 인공 셀들 간의 데이터패스를 보여준다. 정상동작 시 각 인공 셀들은 셀의 남쪽과 남동쪽에 위치한 인공 셀들로부터 데이터를 받아들이고, 북쪽과 북서쪽으로 데이터를 내보낸다.

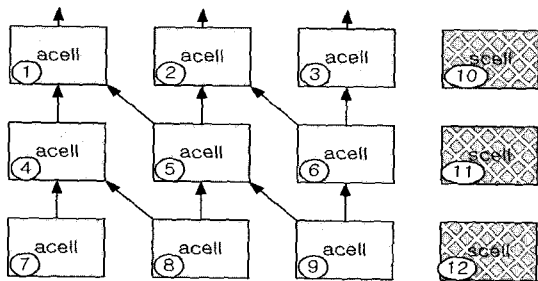


그림 3. 정상동작 시 인공 셀들 간의 연결구조
Fig. 3. Interconnection of acells at normal operation.

2. 고장동작시의 자가치유 과정

자가치유 동작은 단일 고장에 대한 자가치유와 다중 고장에 대한 자가치유로 나누어진다.

(1) 단일고장시의 자가치유과정

1) 최하위 행을 제외한 행에서 고장이 발생한 경우

그림 4는 임의의 행에서 하나의 인공 셀이 고장이 발생한 경우의 자가치유 과정을 보여준다. 그림에서 ⑤번 인공 셀이 고장 나면, 고장 난 ⑤번 인공 셀은 내부 고장 검출기에 의해 스스로 고장이 검출된다. 그 후, 고

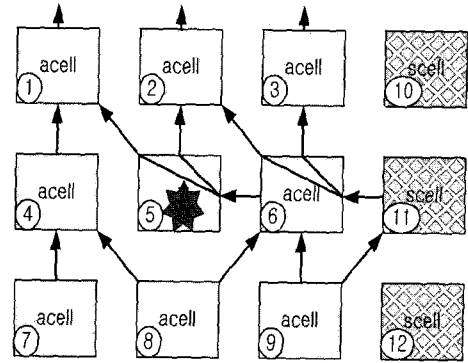


그림 4. 임의의 행에서 인공 셀이 고장이 발생한 경우의 연결구조

Fig. 4. Interconnection of acells in case of fault at arbitrary row.

장 난 인공 셀이 속한 행의 우측에 위치한 인공 셀들은 자기자신을 통과(bypass)모드로 전환하여 자신의 바로 우측에 위치한 인공 셀의 데이터를 자신의 출력으로 통과시키고 입력을 기존의 남, 남동에 위치한 인공 셀의 출력에서 남서, 남에 위치한 인공 셀들의 출력으로 바꾸어 받아들이도록 재구성 한다. 또한, 최 우측의 여분의 인공 셀은 자신의 좌측에 위치한 인공 셀의 기능을 대행한다. 이러한 재구성 과정과 여분의 인공 셀에 의해 자가치유시스템은 고장을 자가치유한다.

2) 최하위 행에서 인공 셀이 고장이 발생한 경우

최하위 행에서 인공 셀이 고장이 발생한 경우, 고장 난 인공 셀의 우측에 위치한 인공 셀들은 바로 자신의 우측에 위치한 인공 셀로 그 기능을 복제하기 때문에 자신의 입력비트 데이터들 역시 우측으로 복제되어야만 한다. 그림 5는 최하위 행에 속한 인공 셀들의 입력비트 데이터 연결을 보여준다. 최하위 행에 속한 인공 셀들의 자가치유과정은 1)의 자가치유과정과 동일하며 고장 발견 시 자신의 입력비트 데이터를 우측에 위치한 인공셀의 입력으로 연결만 해주면 쉽게 기능이 복제될 수 있다.

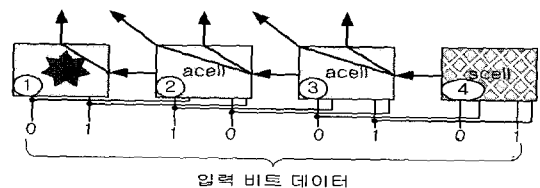


그림 5. 최하위 행에 고장난 경우의 인공셀들의 연결구조

Fig. 5. Interconnection of acells in case of fault at bottom row.

(2) 다중고장시의 자가치유과정

1) 이웃하지 않은 행에서 고장이 발생한 경우

이웃하지 않은 행에서 다중고장이 발생한 경우(그림 6)의 자가치유과정은 단일고장시의 최하위 행을 제외한 행에서 고장이 발생했을 경우와 동일한 과정에 의해 치유된다. 즉, 그림에서 ②, ⑧번 인공 셀이 고장이 발생하면, 고장이 발생한 ②, ⑧번 인공 셀은 내부 고장 검출기에 의해 스스로 고장이 검출된다. 그 후, 고장 난 인공 셀이 속한 행의 우측에 위치한 인공 셀들은 자기 자신을 통과(bypass)모드로 전환하여 자신의 바로 우측에 위치한 인공 셀의 데이터를 자신의 출력으로 통과시키고 입력을 기존의 남, 남동에 위치한 인공 셀의 출력에서 남서, 남에 위치한 인공 셀들의 출력으로 바꾸어 받아들이도록 재구성 한다.

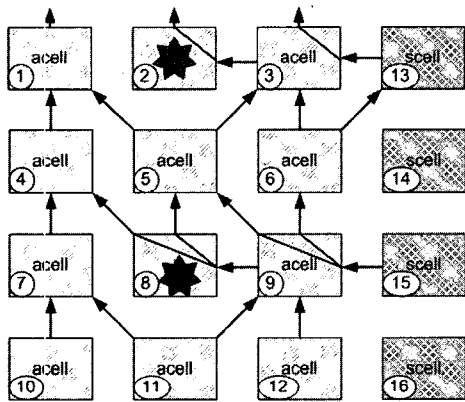


그림 6. 이웃하지 않은 행에서 인공 셀이 고장 난 경우의 연결구조
 Fig. 6. Interconnection of acells in case of faults at separated rows.

2) 이웃한 2행 이상에서 고장이 발생한 경우

그림 7은 이웃한 2행에서 인공 셀이 다중 고장이 발생한 경우의 자가치유 과정을 보여준다. 그림에서 ①번 인공셀과 ⑤번 인공 셀이 고장 나면, 고장 난 인공 셀들이 속한 열(column) 중에서 가장 우측에 위치한 고장 난 인공 셀 열까지는 1)번 행과 같이 인공 셀들의 입출력이 연결되고, 가장 우측에 위치한 고장 난 인공 셀이 속한 열에서부터 바로 인접한 우측의 열과 그 바로 위의 행들에 속한 인공 셀들의 입력은 정상동작시의 입력 연결로 되어야만 하며, 출력은 통과(bypass)모드로 되어야만 한다. 또한 고장 난 인공 셀이 속한 행들의 여분의 인공 셀들은 바로 좌측의 인공 셀들의 기능을 대행하여 자가치유가 된다.

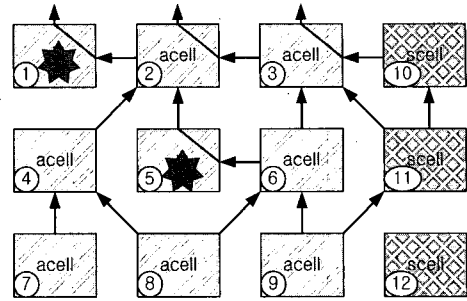


그림 7. 이웃한 2행 이상에서 인공 셀이 고장이 발생한 경우의 연결구조
 Fig. 7. Interconnection of acells in case of faults at neighboring rows.

3. 인공셀의 구조

그림 8은 본 논문에서 제안하는 인공 셀의 구조를 보여준다. 인공 셀은 크게 로직블록(logic block), 게놈블록(genome block), 고장검출기(fault detector), 스위치블록(SB)로 구성된다. 모든 블록은 DCVSL로 설계되어 고장을 검출하는데 용이한 구조를 갖는다^[6].

본 논문에서 제안한 자가치유 과정은 인공 셀의 게놈(genome)블록의 연산에 의해 로직블록을 재구성함으로써 이루어진다.

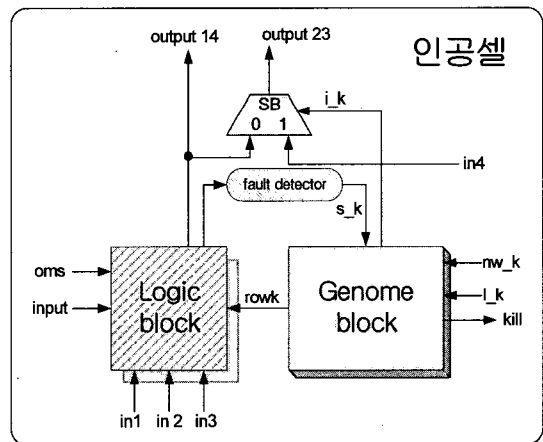


그림 8. 인공 셀의 블록도
 Fig. 8. Block diagram of asell.

(1) 논리 블록의 설계

로직블록은 인공 셀의 로직기능을 담당하는 블록으로 그림 9에서와 같이 입력 mux (im), main mux (mm), Flip Flop (FF), 출력 mux (om)로 구성된다. 입력 mux는 남, 남동 방향에 위치한 인공 셀로부터 각각 in2와 in3를 입력으로 받고, 남서 방향에 위치한 인공 셀로부터 in1을 받는다. 입력 mux의 선택(select)단자

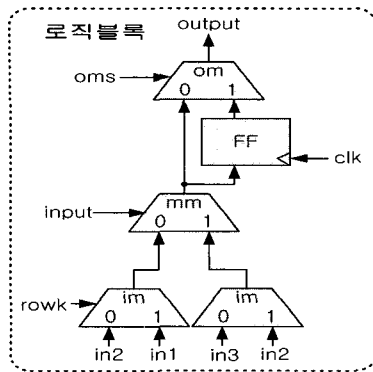


그림 9. 로직블록
Fig. 9. Logic block..

rowk는 자신의 좌측 인공 셀이 고장 났을 때 게놈블록의 제어로 선택된다. main mux는 로직블록의 주 기능을 담당하며, 입력(input)신호에 의해 입력 mux의 출력을 선택한다. Flip Flop은 인공 셀의 순차회로 동작을 담당하며, 출력 mux (om)의 출력모드신호(oms)의 선택에 의해 인공 셀의 순차회로 혹은 조합회로로의 기능이 결정된다.

본 논문에서 제시된 로직블록의 구조는 in1과 in3로 남서쪽과 남동쪽에 있는 인공 셀의 데이터를 입력으로 받음으로써 효율적인 셀 단위 치유가 가능하게 된다.

(2) 게놈블록 설계

기존 논문^[6]의 게놈블록은 gene정보를 레지스터에 저장하는 방식을 사용하고, 자신의 gene정보뿐 아니라 주변의 gene정보를 모두 저장하였기 때문에 하드웨어 오버헤드가 크게 증가하였다. 이를 개선하기 위해 본 논문에서는 레지스터 기반이 아닌 로직 기반으로 게놈블록을 설계함으로써 게놈블록의 오버헤드 문제를 보완한다.

게놈블록은 로직블록과 스위치블록의 정상동작 및 자가치유동작을 제어하는 신호를 생성하는 블록이다. 표 1은 게놈블록의 진리표를 보여주며, 그림 10은 게놈

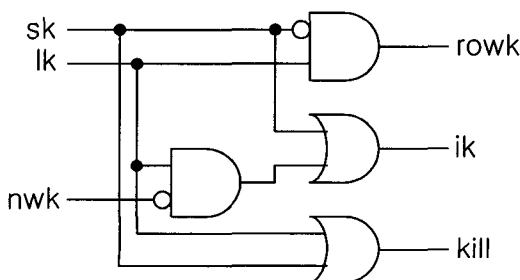


그림 10. 게놈블록의 제어신호회로
Fig. 10. Control circuit of genome block.

표 1. 게놈블록의 진리표
Table 1. Truth table of genome block.

sk	lk	nwk	rowk	ik	kill
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	1	1	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	-	1	1
1	1	1	-	1	1

블록의 제어회로를 보여준다. 게놈블록은 자신이 속한 인공 셀이 고장 난 경우(s_k), 좌측 인공 셀이 고장 난 경우(l_k), 북서쪽에 위치한 인공 셀이 고장 난 경우(nw_k)를 입력으로 받아 로직블록과 스위치블록의 rowk, i_k신호를 제어하고, 자신이 속한 인공 셀의 kill 출력을 내보낸다.

(3) 고장검출기(fault detector) 및 스위치블록

고장검출기는 DCVSL (differential cascode voltage switch logic)구조^[6]로 된 인공 셀의 출력을 XNOR 게이트를 이용하여 실시간으로 감시하고 고장을 검출한다. 인공 셀에 고장이 발견되면 즉, 로직블록의 출력이 동일한 값이 되면, 고장검출기는 로직값 '1'을 게놈블록에 전달하여 게놈블록이 적당한 출력을 연산하도록 한다. 또한, 스위치블록은 인공 셀의 출력을 통과(bypass) 모드로 전환해 주는 기능을 한다. 통과모드가 '0'인 경우 로직블록의 출력이 인공 셀의 출력으로 나가고, 통과모드가 '1'인 경우 동쪽에 위치한 인공 셀의 출력을 입력 in4로 받아 자신이 속한 인공 셀의 출력으로 내보낸다.

IV. 예제 설계 및 시뮬레이션

제안된 자가치유시스템의 구조를 이용하여 2-bit 업다운카운터를 설계한다. BDD^[5]를 이용하여 2-bit 업다운카운터의 기능을 표현한 후 제안된 인공 셀과 여분의 인공 셀 배열로 대응시켜 효율적인 자가치유가 되도록 한다^[6].

그림 11은 제안된 자가치유시스템을 검증하기 위한 자가검출 및 자가치유기능이 내장된 2-bit 업다운카운터의 블록도를 보여준다.

제안된 자가치유시스템을 이용한 2-bit 업다운카운터의 입출력단자는 카운터 출력의 동기화를 위한 클럭

(clk), 시스템을 초기화 하는 리셋(reset)신호, 업카운트와 다운카운트동작을 제어하는 모드(mode)신호, 자가치유동작을 검증하기위한 고장주입(fault 1~3)신호, 칩테스트를 위한 테스트제어(test control)신호, 2-bit 업다운카운터의 출력 Q1과 Q0, 테스트결과를 관측하기위한 테스트출력(test output)신호가 있다. 여기서 고장주입은 그림10의 ①, ②, ③번 인공 셀 내부에 각 fault 1, 2, 3신호를 이용해 임의의 노드가 s-a-1이 되도록 주입하였다. 또한 칩 테스트를 위해 4-to-1 multiplexer 12개로 구성된 테스트블록(test block)을 내장하여 각 인공셀의 출력을 외부에서 관측할 수 있도록 하였다.

그림 12는 설계된 2-bit 업다운카운터의 고장이 없을 시의 동작을 검증하는 시뮬레이션 결과 파형이다. 시뮬레이션은 레이아웃 상에서 넷 리스트를 추출하고 HSPICE를 이용하여 83MHz의 동작주파수로 post-

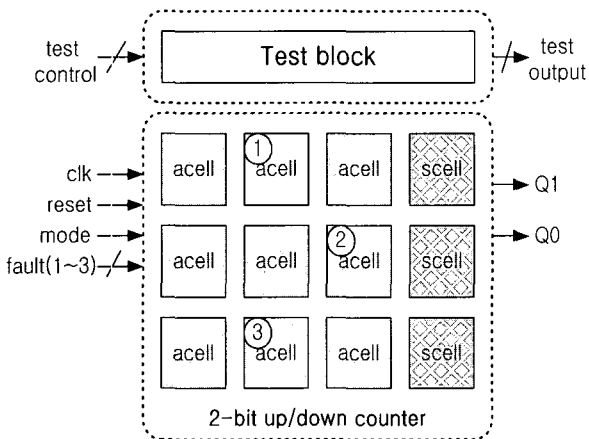


그림 11. 제안된 자가치유시스템을 이용한 2-bit 업다운카운터

Fig. 11. 2-bit updown counter using the proposed self-repair system.

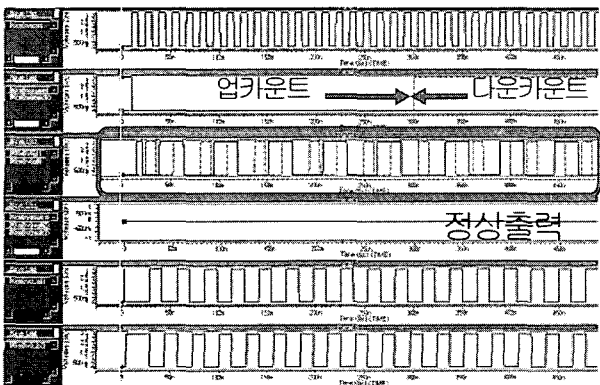


그림 12. 고장이 없을 시 2-bit 업다운카운터의 결과 파형

Fig. 12. Waveform of 2-bit updown counter with no faults.

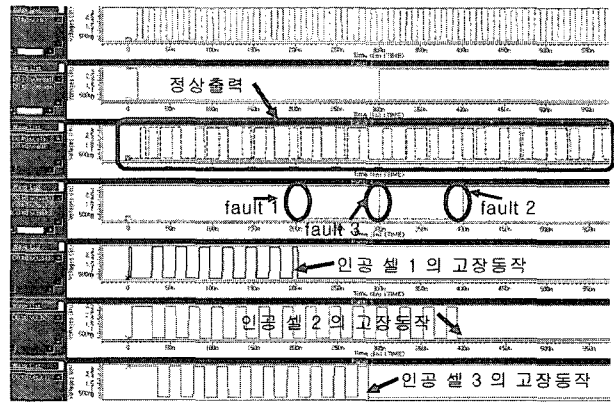


그림 13. 고장주입 후 2-bit 업다운카운터의 결과파형
Fig. 13. Waveform of 2-bit updowncounter with faults.

layout 시뮬레이션을 하였다. mode신호가 '0'인 경우 2-bit 업다운카운터는 업카운트 동작을 하며, mode신호가 '1'인 경우 2-bit 업다운카운터는 다운카운트 동작을 수행한다.

그림 13은 2-bit 업다운카운터에 고장을 주입하였을 시의 시뮬레이션 결과 파형을 보여준다. 고장을 fault 1, 2, 3 순으로 각 200ns, 400ns, 300ns에 주입하였고, ①, ②, ③번 인공셀들의 출력이 로직값 '0'으로 고장동작 하였으나, 2-bit 업다운카운터의 출력은 자가치유되어 정상적인 동작을 유지하였다.

V. 칩 설계와 결과

DCVSL을 이용하여 고장검출 회로가 내장된 자가치유 2-bit 업다운카운터를 하이닉스 0.35um CMOS공정을 이용하여 완전주문형으로 설계하였다.

그림 14는 레이아웃도를 보여준다.

사용된 트랜지스터 수는 총 1,285개이고 코어면적은 $0.32 \times 0.22 \text{ mm}^2$ 이다.

기존구조^[6]에서 2,152개의 트랜지스터가 사용된 것에 비해 하드웨어 오버헤드가 약 40.3% 개선되었다.

Q1+과 Q2+는 일반 출력을 나타내고 그 밑의 출력은 고장을 주입한 인공 셀의 데이터를 보여준다.

2개의 인공 셀에 고장을 주입하여도 정상적인 데이터가 나옴을 확인할 수 있었다. 즉, 다중고장에 대해서도 1개의 여분 셀 열로 효율적으로 치유됨을 알 수 있었다.

그림 15는 테스트 구성도이고, 그림 16은 설계된 칩의 테스트 결과를 보여준다.

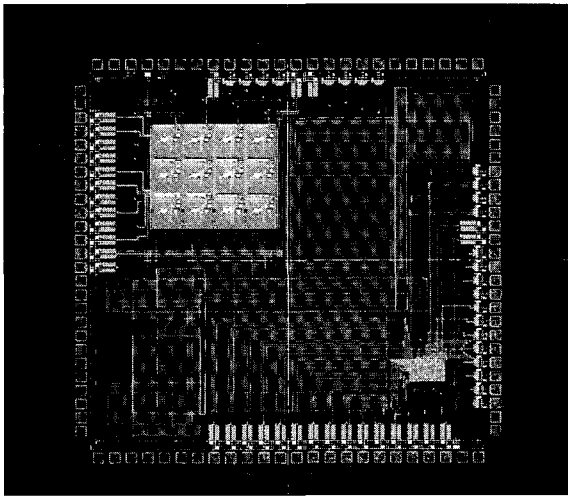


그림 14. 2-bit 업다운카운터의 레이아웃
Fig. 14. Layout of 2-bit updowncounter.

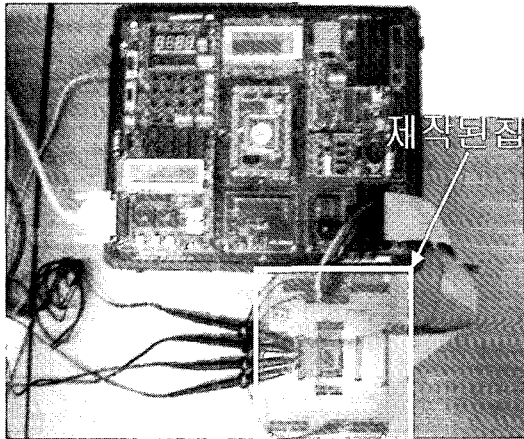


그림 15. 테스트 구성도
Fig. 15. Test configuration.

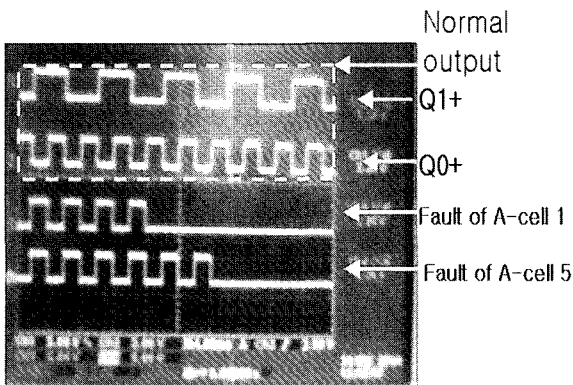


그림 16. 테스트 결과
Fig. 16. Test results.

VI. 결 론

본 논문에서는 기존 자가치유시스템의 열 단위 치유

로 인한 여분 셀의 낭비문제와 계층블록의 하드웨어 오버헤드 문제를 해결하기 위해 효율적인, 계층블록 제어 신호회로와 논리블록을 설계하였고, 이를 이용하여 다중고장에 대해 효율적으로 치유할 수 있는 자가치유시스템의 구조를 제안하였다. 제안된 구조 검증을 위해 인공 셀들과 여분 셀들의 배열로 다중고장에 대해 효율적인 치유가 가능한 2-bit 업다운카운터를 설계하였다.

제안된 자가치유시스템은 기존의 방법과는 달리 인공 셀의 데이터패스를 대각선으로도 가능하게 설계함으로써 기존의 열 단위치유가 아닌 셀 단위치유가 가능하게 하였고 이로 인해 다중고장에 대해 효율적인 치유가 가능하게 한 결과, 행 당 1개씩의 고장을 치유할 수 있어, n행으로 구성된 하나의 여분 셀 열로 n개까지 치유가 가능하고, 계층블록을 레지스터기반의 저장방식이 아닌 로직기반의 생성방식으로 설계하여 기존의 계층블록이 가졌던 문제점인 하드웨어 오버헤드문제를 개선하였다.

설계는 하이닉스 0.35um 공정으로 full-custom으로 하였으며, 기존의 자가치유시스템 구조를 이용한 설계에 비해 하드웨어 오버헤드를 40.3% 개선하였다.

향후, 적은 하드웨어로 고장을 자가검출하고, 더 많은 인공 셀의 고장을 자가치유할 수 있는 자가치유시스템의 구조 연구가 필요하다.

참 고 문 헌

- [1] V. Shen and F. Shen, "Requirements specification and analysis of fault tolerant digital systems," *IEEE Transactions on Man and Cybernetics, Part A*, Vol. 32, No. 1, pp.149-159, Jan. 2002.
- [2] H. Fahmy, S. Ghoneim, and A. Pacha, "Fault tolerant communication with partitioned dimension-order routers with complex faults," *Proc. of 15th International Conference on Information Networking*, pp.361-366, Jan. 2001.
- [3] M. Yen, W. Fuchs, and J. Abraham, "Designing for concurrent error detection in VLSI: Application to a microprogram control unit," *IEEE J. Solid State Circuits*, Vol. 22, No. 4, pp.595-605, Apr. 1987.
- [4] D. Bradley and A. Tyrrell. "Immunotronics: Novel finite state machine architectures with built-in self-test using self-nonsel differentiation," *IEEE Transactions on Evolutionary Computation*, Vol. 6, No. 3, pp.227-238, June

2002.
 [5] D. Mange, "Embryonics: A new methodology for designing FPGA with self-repair and self-replicating properties," IEEE Transactions on VLSI Systems, Vol. 6, No. 3, pp.387-399, Sept. 1998.
 [6] 서정일, 성낙훈, 오택진, 양현모, 최호용, "자가검출 회로 내장의 자가치유시스템 설계" 전자공학회는

문지, 제42권 SD편, 제5호, 307-314쪽, 2005년 5월.
 [7] P.K. Lala and B.K. Kumar, "An Architecture for Self-Healing Digital Systems," Proc. of 8th IEEE International on On-Line Testing Workshop, pp.3-7, July 2002.
 [8] M. Rabaey, "Digital Integrated Circuits," Prentice-Hall Inc., pp.522-543, 1996.

저 자 소 개



최 호 용(평생회원)
 1980년 2월 서울대학교
 전자공학과 졸업(공학사)
 1982년 2월 한국과학기술원 전기
 및전자공학과 졸업
 (공학석사)
 1994년 3월 오오사카대학교 전자
 공학 졸업(공학박사)

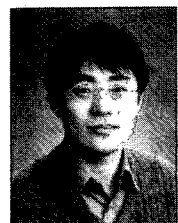
1980년 3월~1985년 7월 삼성반도체연구원
 선임연구원
 1985년 8월~1996년8월 부경대학교 부교수
 1996년 9월~현재 충북대학교 전기전자컴퓨터
 공학부 교수
 <주관심분야: DFT & Testing, 집적시스템설계>



서 정 일(학생회원)
 2004년 2월 충북대학교 전기전자
 공학부 졸업(공학사)
 2006년 8월 충북대학교 반도체
 공학과 졸업(공학석사)
 <주관심분야: DFT & Testing,
 Memory설계>



유 충 호(학생회원)
 2005년 2월 충북대학교 전기전자
 공학부 졸업(공학사)
 2005년 3월~현재 충북대학교
 반도체공학과 석사과정
 <주관심분야: DFT, LDI 설계>



우 철 중(학생회원)
 2005년 2월 충북대학교 전기전자
 공학부 졸업(공학사)
 2005년 3월~현재 충북대학교
 반도체공학과 석사과정
 <주관심분야 : DFT, RFID설계>



이 재 은(학생회원)
 2004년 2월 충북대학교 전기전자
 공학부 졸업(공학사)
 2006년 3월~현재 충북대학교
 반도체공학과 석사과정
 <주관심분야 : DFT, 저전력 &
 CAM 설계>