

논문 2010-47SD-5-7

회로 분할을 사용한 저비용 Repair 기술 연구

(Low-Cost Design for Repair by Using Circuit Partitioning)

이성철*, 여동훈*, 신주용*, 김경호**, 신현철***

(Sungchul Lee, Donghoon Yeo, Juyong Shin, Kyungho Kim, and Hyunchul Shin)

요약

반도체 설계기술의 발달로 구현 회로가 복잡해지고, 동작속도가 크게 증가함에 따라, 반도체 이후 (post-silicon) 설계 단계에서 repair를 위한 기간 및 비용이 크게 증가하고 있다. 본 논문에서는 예비 셀을 이용한 repair 방법을 통해 설계 오류로 인한 repair시 혹은 설계 변경 시에 전체 재설계를 최소화하는 방법을 제안하였다. 또한 예비 셀을 이용한 설계 변경 과정에서 repair layer에 설계 변경을 국한하여 mask 비용과 time-to-market을 줄이는 방법을 개발하였다. 또한 회로 분할을 통해 repair 과정에서 사용하는 예비회로의 비용을 줄일 수 있도록 한다.

Abstract

As the complexity and the clock speed of semiconductor integrated circuits increase, silicon validation becomes important. In this research, we developed new post-silicon repair & revision techniques to reduce cost and time-to-market. Spare cells are fabricated with the original design and are used for repair when necessary. The interconnections are modified by repair layer revision. The repair cost can be reduced by logic partitioning. Experimental results show that these techniques are effective for low-cost and fast turnaround repair.

Keywords: repair, low-cost, time-to-market, pin-extension, complex type spare cell/block

I. 서론

반도체 설계기술의 발달로 회로의 크기가 커지고, 동작속도가 크게 증가함에 따라, 반도체 개발 및 설계기간 비용이 크게 증가하고 있다. 특히 설계집적도 증가로 인한 복잡도 증가로 반도체 이전 (pre-silicon) 설계 단계에서 발생할 수 있는 모든 오류의 검증이 어려워졌으며, 설계 오류 발생 원인이 다양해지면서, 그림 1에서와 같이 설계 검증이 차지하는 비중이 크게 증가하였고^[1], 그림 2와 같이 반도체 설계에서 초기 설계가 오류를

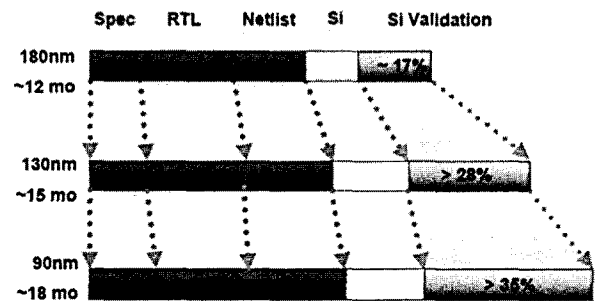


그림 1. 개발기간에서 반도체 설계 검증이 차지하는 비중^[1]

Fig. 1. Silicon validation vs development cycle.^[1]

가질 확률 또한 크게 증가하고 있다.^[2]

따라서 발생한 오류를 반도체 이후 (post-silicon) 설계 단계에서 수정해야 하며, 이는 반도체 개발 기간을 증가시키고, mask 제작비용을 크게 높일 수 있다.^[2]

그림 3은 공정 기술 개발에 따른 mask 제작비용의 증가를 나타내는 것이다. 공정기술이 발달할수록 mask

* 학생회원, 한양대학교 전자전기제어계측공학과
(Dept. of Electronics Engineering, Hanyang University)

** 정회원, 삼성전자부
(Samsung Electronics Co., Ltd.)

*** 정회원, 한양대학교 전자컴퓨터공학부
(School of Electrical and Computer Engineering, Hanyang University)

접수일자: 2009년11월19일, 수정완료일: 2010년4월12일

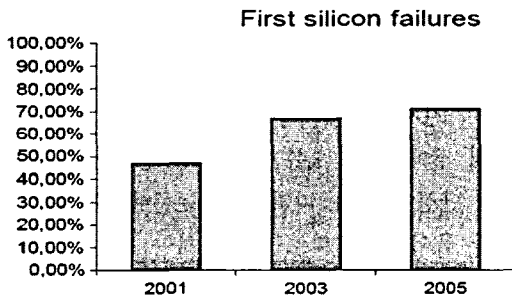


그림 2. 반도체 초기 설계가 오류를 가질 확률^[2]
 Fig. 2. First silicon IC failure.^[2]

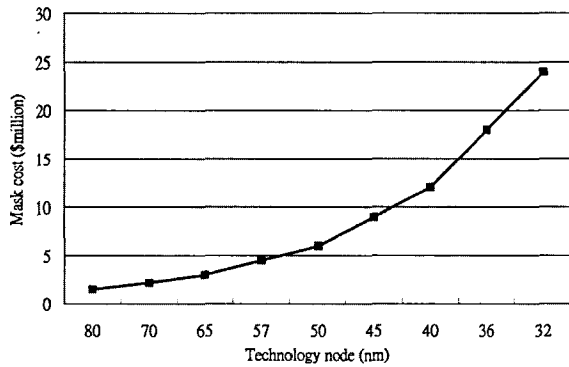


그림 3. mask 비용의 추정값^[2]
 Fig. 3. Mask cost.^[2]

비용이 증가하고, 이는 설계 오류 발생 시에 repair 비용 또한 증가하는 것을 의미한다. 따라서 반도체 이후 단계에서의 설계비용을 줄이기 위해 repair 기법들이 개발되고 있다. Repair는 보편적으로 반도체 이전 설계 단계에서 예비 셀 (spare cell) 혹은 예비 회로 (spare block)를 배치하고, 반도체 이후 설계 단계에서 배치한 예비 셀/회로를 사용하여, 오류가 발생한 논리회로를 대체하는 순서로 수행된다. 이때 오류가 발생한 논리회로를 예비 셀 혹은 예비 회로로 대체하기 위해 재배선 (re-routing) 과정이 필요하다. 이 재 배선 과정에서 다수의 mask 변경이 필요하게 된다.

본 논문에서는 반도체 이전 설계 단계에서 일반 셀 (standard cell) library 내의 셀을 예비 셀로 배치하고, 발생한 오류를 정정하는, 반도체 이후 설계 단계에서 interconnection의 변경을 통해 오류 회로를 예비 셀로 대체하여 설계 오류를 정정하는 방법을 제안한다. 또한 핀 확장 (pin extension) 방법을 통해 오류 정정 과정에서 mask 변경 비용을 최소화하며, 회로 분할을 통해 예비 셀의 사용 면적을 줄이는 방법을 제시한다.

II. 예비 셀을 사용한 repair 방법

Repair 시에 사용하는 예비 셀의 종류에 따라 일반 셀 (standard cell) library 내의 셀을 조합하여 사용하는 방법과, 재구성 가능한 (programmable) 예비 셀을 사용하는 방법으로 크게 나누어진다.

일반 셀을 예비 셀로 사용하는 방법은 회로 내에 일반 셀을 1 종류 이상 배치해 두고 오류 발생 시에 배치한 예비 셀로 오류 셀 혹은 회로를 대신한다. 일반 셀의 경우 그 종류가 매우 많기 때문에 반도체 설계 과정에서 사용한 모든 셀의 종류별로 예비 셀을 구성할 경우 예비 셀의 repair 효과가 매우 떨어지게 된다. 따라서 AND, OR, Inverter나 NAND, NOR, Inverter와 같은 특정 종류의 셀을 예비 셀로 사용하고, 다른 종류의 셀에서 오류가 발생할 경우 정해진 특정 예비 셀들을 조합하여 이를 대신하는 방법을 사용한다. 특히 오류를 repair하는 종합적인 방법을 제시한 Fogclear^[3] 방법의 경우 배치된 회로의 셀 구성에 따라 적합한 예비 셀 종류를 결정하고 이를 사용하는 방법을 사용한다. 하지만 일반 셀을 사용하는 방법은 오류가 발생한 셀과 예비 셀을 교환할 때는 유용하지만, 크고 복잡한 회로에서 다수의 셀에 걸쳐 오류가 발생할 경우 많은 예비 셀을 조합, 연결해야 하므로 효율성이 떨어질 수 있다. 특히 반도체 설계 이후 과정에서 다수의 오류 셀을 예비 셀로 대체할 경우 Focused Ion Beam (FIB)과 같은 재배선 방법을 쓰기 어려워지므로 metal layer 전체를 재설계하게 되며, 이는 repair 비용 및 개발 기간을 크게 증가시킨다.

재구성 가능한 예비 셀의 경우 look-up table을 사용하여 다양한 회로를 구현할 수 있는 셀을 사용할 수 있다. 이 방법은 한 종류의 셀로 repair를 수행하므로, 오류가 발생한 셀의 종류에 따라 예비 회로의 사용 효율이 바뀌는 경우가 적다. 또한 하나의 예비 셀이 다수의 오류 셀을 표현할 수도 있으므로 interconnection의 변경 또한 일반 셀을 예비 셀로 사용하는 방법에 비해 적어질 수 있다. 하지만 look-up table을 사용함으로써 인해 예비 셀 크기가 매우 크고, table을 설정하는 control이 복잡한 단점이 있다.

예비 셀과 기존 설계 간의 연결을 위해서는 보통 배선(wire)을 재설계(re-spin)하는 방법을 많이 사용한다. 이는 다수의 metal layer를 재설계하게 되므로, metal mask 제작비용 및 개발 기간을 크게 증가시킨다. 이

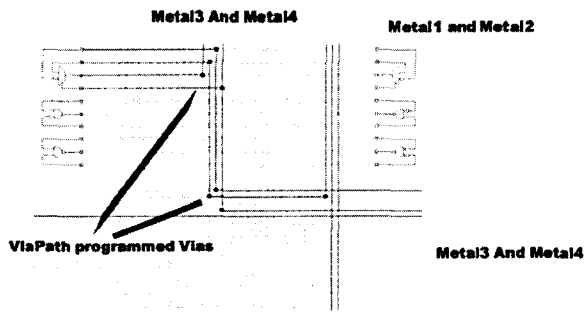


그림 4. 격자구조의 배선

Fig. 4. Interconnection fabric.

방법과 달리 routing 과정에서 미리 배선을 배치해 두어 repair 시에 via layer 1층만을 수정하는 방법으로 회로의 routing을 바꿀 수 있는 standard metal^[4] 방법이 있다. 그림 4와 같이 예비 셀의 입 출력 핀은 모두 metal 3과 4층에 연결되어 있으며, 각 셀의 interconnection은 metal 3과 4 사이에 있는 ViaPath programmable Via를 사용하여 변경하게 된다. 이는 다층의 metal layer를 전부 변경하지 않고, 1개의 via layer만을 변경하므로 mask 변경을 최소화할 수 있지만, 변경하고자 하는 interconnection의 pin 수만큼의 crossbar 형식의 격자(mesh) 구조의 배선을 수행해야 하므로 배선면적(routing area)의 비용(overhead)이 큰 단점이 있다. 따라서 이 방법은 재구성 가능한 예비 셀을 사용하여 예비 셀간의 변경해야 할 interconnection의 수를 줄이는 방법과 같이 사용할 때 유효하다.

본 논문에서는 반도체 이전 설계 단계에서 일반 셀(standard cell) library 내의 셀을 예비 셀로 배치하고, 반도체 이후의 repair는 두 층의 metal layer interconnection의 변경을 통해 오류 회로를 예비 셀로 대체하여 설계 오류를 정정하는 방법을 제안한다. 예비 셀의 사용면적 제약을 고려하고, 오류 발생시 바뀌어야 할 interconnection의 수를 줄이기 위해 회로 분할을 수행하며, 또한 핀 확장(pin extension) 방법을 통해 오류 정정 과정에서 변경하는 interconnection을 repair layer라는 특정 metal 층에 국한시켜 mask 변경 비용을 최소화한다.

III. 저 비용의 일반 셀과 배선 수정 (Standard Cell & 2 Metal Revision : SC2MR) 방식 Repair 방법

1. 일반 셀과 배선 수정 (SC2MR) 방식 예비 회로

본 논문에서는 논리적 오류를 repair 하기 위해서 일반 셀을 spare cell로 사용한다. 일반 셀의 경우 재구성 가능한 예비 셀에 비해 면적과 소모전력 면에서 좋은 결과를 가지지만 셀의 종류가 많아 논리 조합이 어렵고, 여러 셀에 걸쳐 오류가 발생할 경우 repair가 어려워 지는 단점이 있다. 따라서 본 논문에서는 예비 셀 여러 개를 묶어 예비 회로를 구성하는 방법을 사용한다. 예비 회로는 AND, OR, Inverter set 혹은 NAND, NOR, Inverter set과 같이 조합 회로(combination circuit)를 구성할 수 있는 예비 셀(cell)과 레지스터(register)와 같은 순차 회로(sequential circuit)를 구성할 수 있는 예비 셀을 나열한 것으로, 나열된 조합 회로 셀과 순차 회로 셀을 상황에 따라 재배선하여 원하는 회로를 구성한다. 순차 회로인 레지스터의 경우 조합 회로로 구현하는 것이 가능하나, 필요한 조합 회로 셀의 개수가 많고, clock skew와 같이 timing 제약 요소가 많으므로 조합 회로로 구현하지 않고 레지스터를 예비 셀로 사용한다.

반도체 이후 설계 과정에서 오류가 발생하지 않았을 경우, 혹은 예비 회로가 사용되지 않는 경우, 예비 회로 내의 셀들의 입력은 각각 VDD 혹은 VSS로 연결된다. 분할 된 회로에서 오류가 발생한 경우, 그림 5와 같이 예비 회로 내의 셀들은 오류가 발생한 부분을 구현하기 위해 재배선되고, 오류가 발생한 부분의 입력/출력 연결은 예비 회로로 연결되도록 재설계된다. 또한 오류가 발생한 부분의 입력/출력 연결은 그림 5의 (b)와 같이 VDD 혹은 VSS로 연결된다.

2. 저 비용을 위한 핀 확장 방법

일반 셀을 사용하는 방식의 경우 1개의 cell을 repair 하기 위해서는 1개 이상의 NAND, NOR, inverter 셀을 사용해야 한다. 또한 어떤 셀을 repair하는가에 따라 사용해야 할 셀의 수와 이들의 연결 interconnection 조합이 달라진다. 따라서 cell-to-cell 단위로 repair를 수행할 경우 재배선이 매우 복잡해지며, metal 배선 층을 전부 재배선할 수 있어 repair 비용이 크게 증가한다.

본 논문에서는 재배선 과정에서 발생하는 mask 변경 비용을 최소화하기 위해 모든 재배선 과정을 2개의 metal layer로 국한하는 배선 수정(2 metal revision) 방법을 제안한다. 배선을 수정하기 위한 layer를 repair layer로 정의하며, repair layer만으로 재배선이 가능하도록 하기 위해서, 핀 확장이라는 과정을 통해 분할된

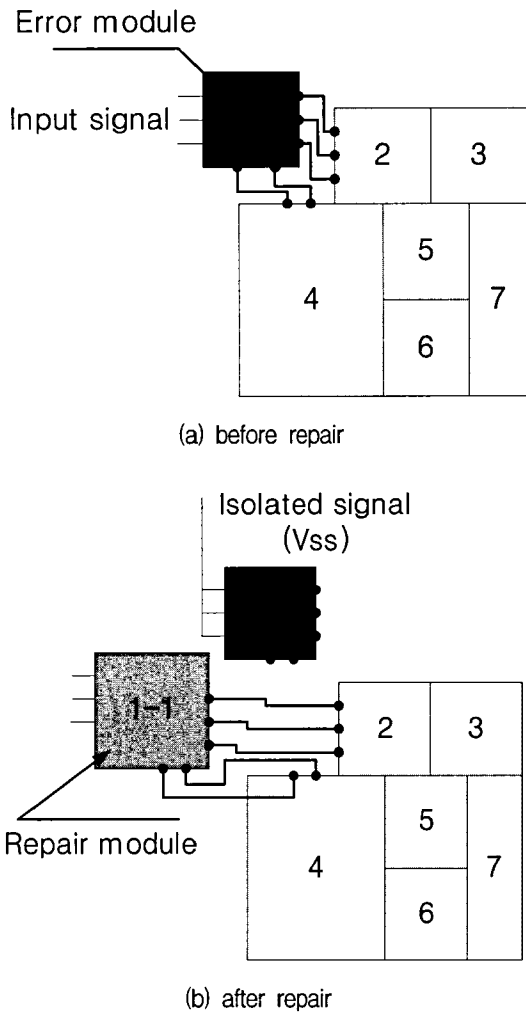


그림 5. 예비 회로를 사용한 repair
Fig. 5. Repair with spare block.

각 회로간의 연결의 전체 혹은 일부가 그림 6의 (a)와 같이 repair layer층을 한번 이상 지나도록 배선을 수행한다. 또한 예비 셀의 입/출력 핀 (pin) 들, 재구성 가능한 예비 회로를 구성하는 각 셀들의 입력/출력 핀들 역시 핀 확장 과정에서 repair layer까지 확장하여 재배선이 repair layer에 국한될 수 있도록 한다.

확장된 핀 수가 많을 경우 재배선 과정에서 복잡도가 크게 증가하게 되므로 이를 줄이기 위해 그림 5와 같이 cell-to-cell repair가 아닌 cluster-to-cluster repair 방법을 사용한다. Repair는 cluster 단위로 수행하게 되며, repair 시에는 그림 6 (b)와 같이 repair layer 내의 확장된 pin 에 연결된 배선을 재배선하여 그림 5 (b)와 같이 repair를 수행한다.

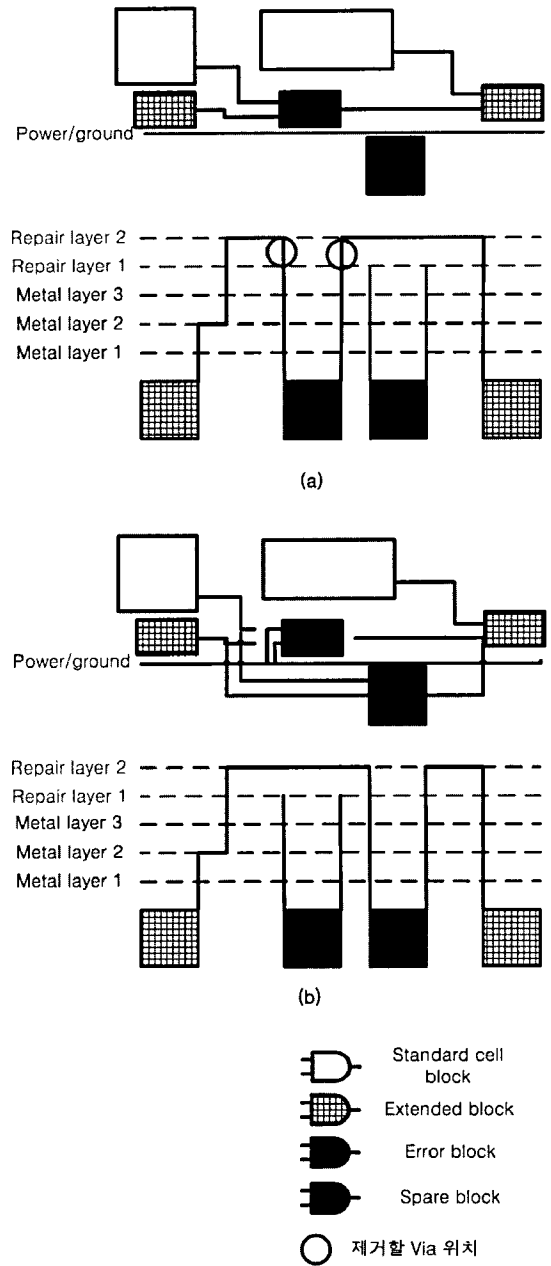


그림 6. 핀 확장 방법
Fig. 6. Pin extension method.

3. Repair 효율을 높이기 위한 회로 분할

배선 수정 방법의 경우 예비 회로 내의 모든 셀은 repair layer까지 입/출력 핀이 확장 되어 있다. 하지만 회로의 복잡도와 성능 최적화로 인해 일반 회로내의 모든 셀의 배선을 repair layer에 집중시키는 것이 불가능하다. 또한 예비회로의 면적 제약요소 등으로 인해 오류가 발생한 회로를 구현할 수 있을 만큼 예비 회로의 크기가 충분하지 않은 경우 회로를 repair하는 것이 불가능하다. 따라서 그림 7과 같이 SC2MR 과정에서 전

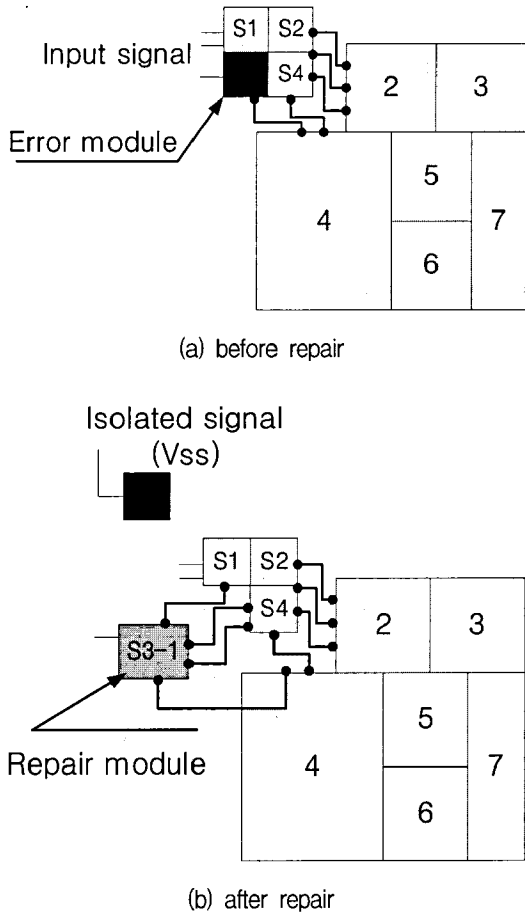


그림 7. 회로 분할을 사용한 repair
 Fig. 7. Repair with partition.

채 일반 회로를 예비 회로가 구현할 수 있는 크기로 분할 (partition) 하고, 분할된 회로의 입/출력 핀을 repair layer까지 핀 확장을 수행하여 확장된 핀 수와 배선 복잡도를 줄인다.

이때 register와 같은 순차(sequential) 회로를 NAND, NOR, Inverter의 논리 (combinational) 회로 조합만으로 구현할 경우 회로의 크기가 매우 커지게 되며, 예비회로로 구현된 순차회로의 경우 레이아웃 단계에서의 셀 내부 최적화를 할 수 없으므로 clock skew와 같은 제약요소를 만족시키기 어려워진다. 따라서 그림 7과 같이 회로를 분할하기 이전에 회로 내부의 순차회로와 논리회로를 구분하여 전체 회로를 2개로 분할하는 1단계 회로 분할과, 1단계 회로 분할 이후에 각각에 대해 repair 효율을 높이기 위한 회로 분할을 수행하는 2단계 과정을 거친다. 1단계 회로 분할 이후 생성된 각각의 그룹이 예비 회로로 구현하기에 너무 클 경우, 2단계 회로 분할을 수행하며, 이때 회로 분할은 min-cut 회로 분할 방법 중 하나인 agcep^[5] 회로 분할 방법을

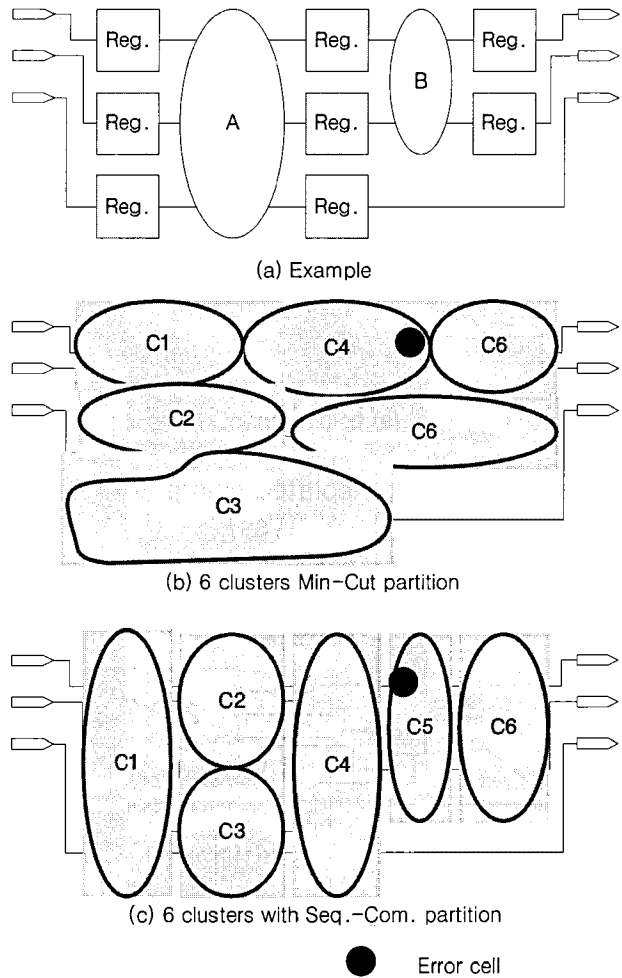


그림 8. 레지스터와 논리회로를 구분한 회로 분할을 사용한 repair
 Fig. 8. Repair with sequential-combinational partition.

사용하였다. 순차회로와 논리회로를 구분하여 회로를 분할할 경우 논리회로는 예비 셀 NAND, NOR, Inverter의 논리회로 조합만으로, 순차회로는 D-FlipFlop과 같은 정해진 특정 예비 셀을 조합하여 repair를 수행할 수 있어 예비 셀 사용면적을 줄일 수 있다. 예를 들어 그림 8 (b)와 같이 min-cut 회로 분할을 수행할 경우 레지스터가 분할된 각 그룹(cluster)에 걸쳐져 있게 되며, 오류가 발생한 C4 그룹을 repair 하기 위해 반드시 레지스터를 repair 해야 한다. 이는 예비 회로의 사용 면적을 증가시키는 요인이 되므로, 본 연구에서는 회로를 repair 하는 과정에서 레지스터와 같은 순차 회로의 경우 다른 논리 회로와 같이 NAND, NOR, Inverter 예비 셀로 repair를 수행하지 않고, 따로 마련된 D-FlipFlop 레지스터 예비 셀을 사용하여 repair를 수행하는 방법을 사용하였다.

그림 8 (b)와 같이 조합논리회로와 레지스터를 구별하지 않고 분할하면 B 블록에 오류가 있을 경우 레지스터를 포함하는 C4, C5, C6를 모두 repair 해야 한다. 따라서 순차회로와 논리회로로 회로를 먼저 분할하며, 이때 순차회로는 그림 8 (c)의 C1, C4, C6를 포함하는 하나의 그룹을 형성하고, 논리회로 역시 C2, C3, C5를 포함하는 하나의 그룹이 된다. 논리회로와 순차회로를 분리한 후 각각에 대해 회로 분할을 수행하면 그림 8 (c)와 같이 되며, 이 경우 논리 회로에서 발생하는 오류는 논리회로 예비셀 만으로, 순차회로에서 발생하는 오류는 순차회로 예비셀 만으로 repair가 가능하게 되며, 이로 인해 예비 회로 사용면적을 줄일 수 있다.

IV. 실험

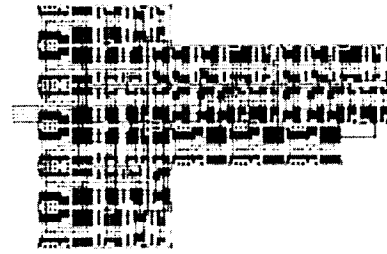
다음의 표 1은 ISCAS85 C17 benchmark 예제에 대해 repair를 수행한 결과이다. C17의 NAND2 6개중 2개의 셀에서 오류가 발생했다고 가정하고 같은 배선 수정 방법으로 repair를 수행한 결과이며, programmable한 Via Patterned Gate Array (VPGA)^[6]에 비해 1/4의 면적으로 repair가 가능하다.

표 1. 면적 결과 : VPGA vs. SCMR
Table 1. Area results : VPGA vs. SCMR.

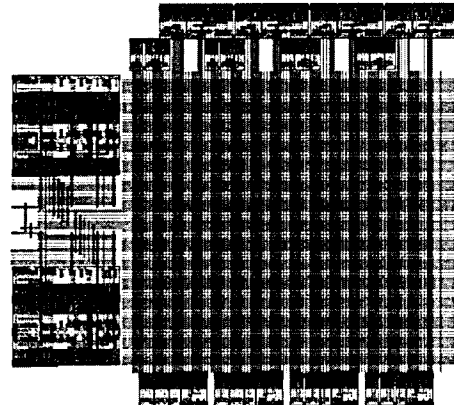
| 항목 | Cell area (um ²) | # of via | |
|-------|------------------------------|----------|---------------|
| | | In logic | In spare cell |
| SC2MR | 39.92 | 16 | 53 |
| VPGA | 159.67 (400%) | 16 | 35 (66%) |

표 2. 배선 길이 결과 : Standard Metal vs. SC2MR
Table 2. Wire length results : Standard Metal vs. SC2MR.

| 예제 | 방법 | Cell area (spare cell area : um ²) | Wire length(um) | | # of extended via | |
|-----|----------------|---|-----------------|-----------|-------------------|---------------|
| | | | Repair 이전 | Repair 이후 | In logic | In spare cell |
| e1 | SC2MR (ours) | 286.1 (136.36) | 73.04 | 248.49 | 20 | 44 |
| | standard metal | 651.9 (502.164) | 4,048.88 | 4,048.88 | 12 | 92 |
| e2 | SC2MR (ours) | 568.8 (226.2) | 132.6 | 223 | 4 | 20 |
| | standard metal | 4620 (2310) | 3197 | 7307.5 | 14 | 40 |
| e3 | SC2MR (ours) | 641.3 (60.5) | 9.2 | 105.7 | 14 | 12 |
| | standard metal | 707.9 (127.1) | 784.9 | 818.4 | 14 | 20 |
| e4 | SC2MR (ours) | 871.5 (13.3) | 2.1 | 8.5 | 6 | 3 |
| | standard metal | 913.6 (127.1) | 132.6 | 223.0 | 4 | 20 |
| S_A | SC2MR (ours) | 1539.4 (758.4) | 1780 | 4165 | 78 | 178 |
| | standard metal | 3997.7 (3146.3) | 685584 | 685584 | 63 | 414 |



(a) SC2MR방식



(b) standard metal 방식

그림 9. 예제 e1 : Standard metal vs. SCMR
Fig. 9. Example e1: Standard metal vs. SCMR.

표 2는 일반 회로에 대하여 각각 특정 repair를 수행한 결과이다. e1은 ISCAS89 S27 benchmark 예제에 대해 repair를 수행한 결과이다. 5개의 입력 신호가 S27 회로 2개에 각각 연결되어 있고, 2개의 S27 회로 중 하나에서 오류가 발생하였다고 가정하였다. 예제 e2와 e3는 각 회로에 saturation 회로를 추가한 것이고, e4는

표 3. Repair 결과 : Min-cut 회로분할
Table 3. Repair results : min-cut partition.

| | | Non partition | Partition 8 | Partition 12 | Partition 16 |
|-----|-----------------|---------------|-------------|--------------|--------------|
| S_A | Spare cell size | 106111 | 69187 | 69187 | 66193 |
| | Reduce | | 34.80 % | 34.80 % | 37.62 % |
| S_B | Spare cell size | 445740 | 158002 | 111101 | 87484 |
| | Reduce | | 64.55 % | 75.07 % | 80.37 % |
| S_C | Spare cell size | 352592 | 268431 | 268431 | 268431 |
| | Reduce | | 23.87 % | 23.87 % | 23.87 % |

회로의 출력단에 특정 gate를 추가한 것이다. S_A 예제는 FSM예제에서 state변경에 따른 repair결과이다. Standard metal 방법은 1 via layer만을 수정하여 재배선을 수행하는 방법으로 2 metal layer를 수정하는 배선 수정 방법에 비해 mask변경이 적지만 배선길이가 16배 더 길게 필요하다.

그림 9는 e1 예제에 대해 SC2MR 방식과 standard metal 방식을 비교한 것이다. 그림 9 (b)에서 격자로 배열되어 있는 배선이 standard metal을 구현하기 위해 미리 배치되어 있는 배선이며, 격자 배선의 위와 아래에 배치된 회로가 예비회로이다. 그림 9 (b)의 예비회로는 격자 구조 밑으로 배치될 수 있으나 보기 편하도록 그림에서는 격자의 위 아래로 배치하였다.

Standard metal 방법은 1 via layer만을 수정하여 재배선을 수행하는 방법으로, 2 metal layer를 수정하는 배선 수정 방법에 비해, mask변경이 적지만 배선길이가 이 예제에서는 16배 더 길게 필요하다.

표 3은 회로의 verilog code에서 조건식을 일부 변경한 회로의 예시이다. 표 3의 값은 Sample A, B, C에 대해 각각 verilog code내 조건 식에서 새로운 조건을 추가한 회로 변경을 하였으며, 각 예제에 대해 회로를 8/12/16개로 회로 분할한 후에 repair 한 결과이다. SC_size는 repair시에 필요한 예비 회로의 면적을, reduce는 회로 분할 없이 전체 회로를 repair 했을 때 필요한 예비 회로 면적 대비 회로 분할 이후 repair 할 때 예비 회로의 면적이 줄어든 비를 나타낸 것이다. 저비용 repair를 위해 min-cut 회로 분할을 사용할 경우 표 3의 3개 예제에 대해 사용해야 할 spare cell을 최대 80%까지 줄일 수 있었다.

V. 결 론

본 논문에서는 예비 셀을 이용한 repair 방법을 통해 설계 오류로 인한 repair시 혹은 설계 변경 시에 재설계 비용을 최소화하는 방법을 제안하였다. 또한 예비 셀을 이용한 설계 변경 과정에서 두 개의 metal repair layer에 설계 변경을 국한하여 mask cost를 줄이는 방법을 개발하였다. 이 방법은 repair cost 및 개발 기간을 크게 줄일 수 있으며, 하나의 via layer만 변경하는 standard metal 방법보다 성능에서 크게 최적화 된 결과를 보인다. 또한 순차회로와 논리회로를 나눈 회로 분할을 통해 예비 회로의 사용 면적을 상당히 줄일 수 있었다.

참 고 문 헌

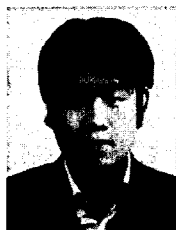
- [1] M. Abramovici, P. Bradley, K. Dwarakanath, P. Levin, G. Memmi and D. Miller, "A Reconfigurable Design-for-Debug Infrastructure for SoCs," in Proc. of Design Automation Conference (DAC), pp. 7-12, San Francisco, CA, July 24-28, 2006.
- [2] Collett International Research 2005.
- [3] K. Chang, I. Markov and V. Rertacco, "Automating Post-Silicon Debugging and Repair," in Proc. International Conference on Computer-Aided Design (ICCAD), Nov, 2007.
- [4] Z. Or-Bach, "Paradigm Shift in ASIC Technology," <http://www.easic.com> 2005.
- [5] 김남훈, 김충희, 신현철, "클러스터의 개선을 이용한 회로 분할 방법", 한국정보과학회 논문지 제25권 A편, 제1호, 94-99쪽, 1998년.
- [6] K. Y. Tong, V. Kheterpal, V. Rovner, L. Pileggi, H. Schmit, "Regular Logic Fabrics for a Via Patterned Gate Array(VPGA)," in Proc. of IEEE on Custom Integrated Circuits Conference, pp. 53-56, San jose, USA, Sept 2003.

저 자 소 개



이 성 철(학생회원)
 2001년 한양대학교 전자컴퓨터
 공학부 학사 졸업.
 2003년 한양대학교 전자전기 제어
 계측공학과 석사 졸업.
 2010년 현재 한양대학교 전자전기
 제어계측공학과 박사 과정.

<주관심분야 : CAD&VLSI, 반도체>



여 등 훈(학생회원)
 2006년 한양대학교 전자컴퓨터
 공학부 학사 졸업.
 2008년 한양대학교 메카트로닉스
 공학과 석사 졸업.
 2010년 현재 한양대학교 전자전기
 제어계측공학과 박사 과정.

<주관심분야 : 저전력설계, H.264, 반도체 설계>



신 주 용(학생회원)
 2009년 한양대학교 전자컴퓨터
 공학부 학사 졸업.
 2010년 현재 한양대학교 메카트로
 닉스공학과 석사 과정.

<주관심분야 : Design for Debug, 반도체 설계>



김 경 호(정회원)
 1984년 연세대학교 전자공학과
 학사 졸업.
 1987년 KAIST 전기전자공학과
 석사 졸업.
 1991년 KAIST 전기전자공학과
 박사 졸업.

현재 삼성전자 통신연구소 상무
<주관심분야 : 이동통신용모델링, 단말무선부품>



신 현 철(정회원)
 1978년 서울대학교 전자공학과
 학사 졸업.
 1980년 한국과학기술원 전기 및
 전자공학과 석사 졸업.
 1983년~1987년 U.C. Berkeley
 박사 졸업.

1983년~1987년 Fulbright scholarship
 1987년~1989년 MTS, AT&T Bell Lab's,
 Murray Hill N.J., USA.
 1997년~2008 IDEC 한양대학교 지역센터 센터장,
 2008년~2010 ITRC Multi-Core Design
 Methodology 연구센터 소장.
 1989년~현재 한양대학교 전자컴퓨터공학부 교수
 <주관심분야 : CAD&VLSI, Vision, 3D용 반도체
 설계, 저전력 설계>