SoC 구현을 위한 안정적인 Power Plan 기법

서영호* · 김동욱**

Stable Power Plan Technique for Implementing SoC

Young-ho Seo* · Dong-wook Kim**

이 논문은 2011년도 광운대학교 교내 학술연구비 지원에 의해 연구되었음.

요 약

ASIC(application specific integrated circuit) 과정은 칩을 제작하기 위한 다양한 기술들의 집합이다. 일반적으로 RTL 설계, 합성, 배치 및 배선, 저전력 기법, 클록 트리 합성, 및 테스트와 같은 대표적인 과정들에 대해서는 많은 연구가 진행 되었고, 지금도 많은 연구가 진행 중이다. 본 논문에서는 이러한 ASIC 방법론에서 전력 플랜과 관련하여 경험적이고 실험적인 전력 스트랩 배선(power strap routing) 방법 기법에 대해서 제안하고자 한다. 먼저 수직 VDD 및 VSS와 수평 VDD 및 VSS를 위한 스트랩의 배선을 수행하고, 이 과정에서 발생하는 문제를 해결하기 위한 기법을 제안한다. 배선 가이드를 생성해서 의도하지 않는 배선을 방지하고, 차후를 위해서 배선 가이드에 대한 정보를 저장한다. 다음으로 불필요한 전력 스트랩을 제거하고, 매크로 핀에 대해 미리 배선을 수행한다. 마지막으로 배선 가이드를 이용하여 최종적인 전력 스트랩 배선을 완료한다. 이러한 과정을 통해서 전력 스트랩이 효율적으로 배선 되는 것을 확인하였다.

ABSTRACT

ASIC(application specific integrated circuit) process is a set of various technologies for fabricating a chip. Generally there have been many researches for RTL design, synthesis, floor plan & routing, low power scheme, clock tree synthesis, and testability which are widely researched in recent. In this paper we propose a new methodology of power strap routing in basis of design experience and experiment. First the power strap for vertical VDD and VSS and horizontal VDD and VSS is routed, and then after the problems which are generated in this process are analyzed, we propose a new process for resolving them. For this, the strap guide is inserted to protect the unnecessary strap routing and dumped for next steps. Next the unnecessary power straps which are generated the first inserting process are removed, and the pre-routing is performed for the macro cells. Finally the resultant power straps are routed using the dumped routing guide. Through the proposed process we identified the efficient and stable route of the power straps.

키워드

전력 플랜, 시스템온칩, 주문형반도체, 전력 스트랩, 설계, 레이아웃

Key word

power plan, SoC, ASIC, power strap, design, layout

* 종신회원 : 광운대학교

** 종신회원 : 광운대학교 (교신저자, dwkim@kw.ac.kr)

접수일자 : 2012. 06. 13

심사완료일자 : 2012. 09. 09

I. 서 론

ASIC 과정에서는 목표로 하고자 하는 성능을 얻기 위해서 많은 문제들에 부딪히게 된다[1][2]. 실제 대용량의 ASIC을 제작하는 과정에서는 실제적인 많은 문제들이 발생하게 되고 이들은 실제 현장에서 칩을 제작하고 있는 설계자들에게 주어진 것들이다. 그 중에서 가장 대표적인 것은 타이밍 위반이다. 타이밍 위반은 톨 간의 상관 차이로 발생하는 타이밍 위반의 대표적인 예이다. 최적화 및 라우팅 과정에서 모든 톨이 시간적 제약을 만족시키는 것을 최우선시 하고 있지만 아직도 상관성 문제는 극복되지 않았고 한 회사에서 제공되는 톨을 사용하지 않는 한 극복될 수도 없을 것이다[3][4]. 따라서 ASIC 엔지니어는 STA(static timing analysis)[5][6]결과를 참고하여 수동으로 타이밍 위반을 수정해야 한다[7]. 또 다른 문제 중의 하나는 클록 트리 합성의 문제이다.

시스템 온 칩(system on a chip, SoC)의 설계가 보편화되는 환경 하에서 수백 MHz 이상의 높은 성능과 신뢰성의 확보를 위한 중요한 요소가 클록(clock)의 동기화이다[8][9]. 디지털 회로는 클록 신호에 동기되어 동작하므로 클록 신호의 빠른 전달이 시스템의 성능 향상으로 직결된다. 클록 분배 네트워크(clock distribution network)라고 표현되는 클록 신호는 전체 회로 내에서 수백 수천 개의 플립플롭(flip-flop) 등으로 구성된 동기화 부하(load)들을 구동하며 칩 전체에 걸쳐서 배치된다. 클록 신호를 칩의 전체 플립플롭에 분배시키기 위해 trunk, mesh, symmetric H-tree, 그리고 버퍼 트리(buffered tree) 등 다양한 클록 네트워크 구조가 제안되었다[10][11].

이중에서도 버퍼 트리 방식은 칩 크기가 늘어나도 버퍼를 삽입하는 방식은 손쉽게 자동화할 수 있어 ASIC/SoC 설계에 적합하고 가장 널리 사용되는 방식이다[8][12]. 물론 논리 합성과 같이 잘 알려진 ASIC 과정에 대해서는 더욱 많은 논의가 있었고, 그에 대한 연구도 진행되고 있다. 논리 합성은 주어진 제약조건(constraint)을 최대한 만족시키면서 논리 사상과 최적화 등을 통하여 RTL 코드로부터 게이트-수준의 넷리스트(netlist)를 얻는 과정이다.

최적화는 RTL 코드 단계에서 이루어지는 것이 가장 효율적이다. 따라서 논리합성은 물리적 합성(physical implementation) 과정에서 가장 기본적이고 중요한 과정

이다. 논리합성의 결과는 주어진 제약조건과 합성 방법에 매우 종속적이다. 이들에 의해서 설계의 면적 및 타이밍이 크게 변화하므로 우리는 제약조건과 합성방법을 철저하게 고려하여야 한다. 아울러 제약조건을 매우 신중히 작성할 수 있어야 한다[13].

전력 플랜(power plan)은 매크로(macro) 혹은 표준셀(standard cell)에 충분한 전원을 공급하기 위하여 전력 스트랩(power strap)을 배선하는 과정을 의미한다. 칩에 충분한 전원이 인가되도록 최상위 설계자는 일정 공식에 의거해서 전력 플랜을 수립한다. 블록 설계자는 이러한 계획에 의해서 최상위 설계자가 정해주는 수 만큼의 전력 스트랩(power strap)을 배선한다.

전력 스트랩은 배선에 큰 영향을 미친다. 특히 전력 스트랩과 매크로사이의 비아(via)로 인해서 매크로 핀 배선(macro pin routing)이 영향을 받는 경우가 많다. 따라서 단순히 주어진 간격대로 power strap을 배선하는 내용 외에 배선을 고려한 전력 스트랩 배선 과정을 제안한다.

앞서 설명한 것과 같이 ASIC은 다양한 기술들에 의해서 수행되는 과정에 대한 집합이다. 일반적으로 RTL 설계, 합성, 배치 및 배선, 저전력 기법, 클록 트리 합성, 및 테스트와 같은 대표적인 과정들에 대해서는 많은 논의가 있었고, 지금도 많은 연구가 진행 중이다. 그러나 전력 플랜과 같은 과정에 대해서는 크게 부각되어 다루어지지 않고 있고, 이에 관한 국내 연구도 찾아보기 어려운 것이 사실이다. 본 논문에서는 전력 플랜과 관련하여 경험적이고 실험적인 전원 배선 방법 기법에 대해서 제안하고자 한다.

본 논문은 다음과 같이 구성된다. 먼저 2장에서는 ASIC 과정에 대해서 간략히 논의한다. 다음으로 3장에서는 전력 플랜을 위한 수직 및 수평 방향의 전원 배선 방법에 대한 이론과 실험을 소개하고 4장에서는 제안하고자 하는 전력 플랜 기법과 그 실험 결과에 대해서 나타낸다. 그리고 마지막으로 5장에서 결론을 맺는다.

II. ASIC 설계 과정

본 절에서는 ASIC 설계 과정을 전체적으로 이해함으로써 다음 장에서 소개될 제안한 CTS의 기법에 대한 기본적인 이해를 높이도록 한다. ASIC 설계과정은 RTL 설

계, 논리 합성(logical synthesis), 배치&배선(placement & routing, P&R), 설계 검토 및 준비(design setup), 플로어플랜(floor plan), 전력 플랜(power plan), 클록트리합성(clock tree synthesis, CTS), 고팬아웃네트합성(high fanout net synthesis, HFNS), 그리고 타이밍클로저(timing closure) 등의 과정으로 나뉜다. 아래에 주요 과정에 대해서 간략히 설명하였다.

2.1. RTL 단계

가장 먼저 수행되는 단계는 설계 사양을 정하는 것이다. 설계자는 회로를 설계하기 전에 먼저 설계 사양을 결정한다. 설계 사양은 회로의 기능 및 타이밍에 대한 요구 조건을 명확하고 상세하게 기술하는 것이다. 설계 사양이 준비되면 설계자는 RTL(register transfer level) 코딩을 수행하고, 기능 시뮬레이션(functional simulation)을 수행하여 기능을 검증한다.

2.2. 설계 검토 및 준비

RTL 코드와 설계 사양을 바탕으로 해서 합성 및 DFT(design for test) 전략을 수립하고, 사용될 툴들을 확정한다. 이들에 필요한 스크립트(script)와 환경을 구축한다. 이를 바탕으로 pre-RTL 합성을 수행하여 RTL 코드의 문제점을 발견하고 수정한다. 다음으로 평면화(floorplan)과 빠른 배치(quick placement)를 수행하여 CWLM(custom wire load model)에 필요한 배선 기생요소(wire parasitic)를 생성한다. 이 과정은 모든 설계에 대해서 수행할 필요는 없다. 배선 기생요소를 이용하여 CWLM을 생성하고 앞서 결정된 DFT 방식에 따라서 메모리에 대한 BIST(built in self test) 등의 회로를 삽입한다[14].

2.3. 합성

RTL 코드가 확정되면 정상(normal) 또는 1-패스 스캔 합성을 하여 네트리스트(netlist)를 추출한다. 합성 과정에 따라서 CWLM이 사용되기도 한다. 합성을 완료하면 스캔(scan)과 경계(boundary) 스캔 회로를 네트리스트에 추가한다. Incremental 최적화를 통해 타이밍과 설계규칙위반 문제를 해결한다. DFT가 추가된 네트리스트는 기능의 등가 확인(functional equivalence checking)과 STA(static timing analysis)를 통해서 검증한다. 이 과정이 끝나면 초기 ATPG(automatic test pattern generation)을 통

해서 테스트 벡터를 생성한다. 이를 이용해서 네트리스트의 기능 및 타이밍에 대한 시뮬레이션을 수행한다[15-17].

2.4. 레이아웃

합성을 통해 네트리스트가 준비되면 타이밍과 수행 시간 등을 고려하여 타이밍-유도(timing-driven) 혹은 밀집-기반(congestion-based) 배치를 수행한다. 배치 이후에 CTS를 수행하게 된다[18-21]. 이와 함께 배선(routing)을 수행하고, 이를 통해 발생하는 타이밍과 설계규칙위반의 문제를 해결해야 한다. 배치와 배선 과정에서는 안정적인 전력 공급과 배분을 위해서 전력 플랜(power plan) 과정을 수행한다.

결과 네트리스트는 기능적 등가 확인과 배선 기생요소 정보를 이용하여 STA를 다시 수행하여 검증(verification) 과정을 진행한다. 검증된 네트리스트에 대해서 최종적인 ATPG를 수행하여 최종 테스트 벡터를 확정한다. 레이아웃 이후에 존재하는 타이밍과 설계규칙 위반은 후-레이아웃 최적화 과정을 통해서 해결한다. 또한 최적화로 인해 변경된 네트리스트는 기능적 등가 확인을 통해서 검증하고, 최종적인 후-레이아웃 네트리스트를 최종 테스트 벡터를 이용하여 기능 및 타이밍에 대해 시뮬레이션을 수행한다[22].

III. 전원 배선 방법

본 논문에서 전력 플랜을 위한 실험 환경은 EDA 툴로 Astro와 Jupiter XT의 tool을 사용하고, 플로어플랜이 완료된 P&R 라이브러리를 사용하는 것으로 하였다[23]. 전원 배선 방법은 수직과 수평 방향으로 수행된다. 일반적으로 블록 설계 시 전력 스트랩의 종류는 크게 네 가지이다. 이 네 가지는 수직 VDD 및 VSS와 수평 VDD 및 VSS를 위한 전력 스트랩이다. 이들의 배선 순서는 중요하지 않다. 다른 네트(net)의 배선을 방해하지 않도록 하기 위해서 전력 스트랩 배선 시에는 되도록 높은 메탈층(metal layer)을 사용한다.

3.1. 수직 방향의 전력 스트랩 배선

본 절에서는 수직 방향의 전력 스트랩을 배선하는 과정에 대해서 설명한다. 전력 스트랩을 배치하는 과

정의 순서는 상관없지만 본 논문에서는 수직 방향의 VDD 라인과 VSS 라인을 먼저 배치한 후에 수평 방향의 VDD 라인과 VSS 라인을 배치하였다. 그림 1에 수직 방향의 VDD 트랩을 배선하기 위한 스크립트를 나타낸다.

```
avgCreateStraps
formDefault "Create Straps"
formButton "Create Straps" "DRC"
setFormField "Create Straps" "Use Fat Via If Width Meets Requirement" "1"
setFormField "Create Straps" "Extend for Multiple Connections" "1"
setFormField "Create Straps" "For Gap <" "0.5"
setFormField "Create Straps" "Start Y" "0"
setFormField "Create Straps" "Y Increment" "0"
setFormField "Create Straps" "Direction" "Vertical" → ①
setFormField "Create Straps" "Start X" "50"
setFormField "Create Straps" "Net Name(s)" "VDD"
setFormField "Create Straps" "Width" "3.3" → ②
setFormField "Create Straps" "Layer" "34"
setFormField "Create Straps" "Step" "70"
setFormField "Create Straps" "Extend to Low Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Extend to High Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Loop Count" "70"
;; (Core's X size - Start X for VSS)/X Increment
setFormField "Create Straps" "X Increment" "50" → ③
formOK "Create Straps"
```

그림 1. 수직방향의 전력 스트랩 배선을 위한 스크립트

Fig. 1 Script for the vertical power strap routing

그림 1의 스크립트는 백엔드(back-end) 설계자들에게 대부분 잘 알려진 내용이기 때문에 이 중에서 중요한 몇 가지 사항에 대해서만 설명한다. ①의 부분은 수직 방향의 스트랩을 형성한다는 명령에 해당한다. ②에서는 스트랩과 연결된 전력 포트의 크기가 3.3um라는 것을 나타내고, ③은 스트랩 배선의 반복횟수를 입력해주는 부분이다. 여기서는 (workcell width)/(X increment) 이 상 값을 입력한다. 그림 1의 확대된 부분에서 VDD에 대한 전력 스트랩에 전력용 핀이 생성된다는 것을 확인할 수 있다.

다음으로 수직 방향의 VSS를 위한 전력 스트랩 배치를 수행한다. 수직방향의 VSS 스트랩 배치를 위한 스크립트 VDD가 VSS로 바뀐 것을 제외하고 그림 1의 스크립트와 거의 동일하다. VDD에 대해서는 Start X가 50um이고 VSS에 대해서는 25um라는 것을 확인할 수 있다. 이때 VSS를 위한 전력 스트랩의 간격은 50um로

동일하다. 이와 같이 적용할 경우에 25um 간격으로 일정하게 VDD와 VSS의 전력 스트랩이 번갈아가면서 생성된다. 이는 그림 3의 결과에서 확인할 수 있다. 그림 3의 확대된 부분에서 VSS에 대한 전력이 생성된다는 것을 알 수 있다.

그림 2. 수직 방향의 VDD 스트랩 결과
Fig. 2 vertical VDD strap result

그림 3. 수직 방향의 VSS 스트랩 결과
Fig. 3 vertical VSS strap result

3.2. 수평 Power strap routing

다음으로 본 절에서는 수평 방향의 전력 스트랩을 배선하는 과정에 대해서 설명한다. 그림 4에 수평 방향의 VDD 트랩을 배선하기 위한 스크립트를 나타냈다. 그림 1의 스크립트에서 ①의 부분은 수직 방향의 스트랩을 형성한다는 명령에 해당한다. ②에서는 스트랩과 연결된 전력 포트의 크기가 3.5um라는 것을 나타내고, ③은 스트랩 배선의 반복횟수를 입력해주는 부분이다. 여기서는 (workcell width)/(X increment) 이상 값을 입력한다. 그림 2와 마찬가지로 그림 5의 확대된 부분에서도 수평 VDD에 대한 전력 스트랩에 전력용 핀이 생성된다.

```
axgCreateStraps
formDefault "Create Straps"
formButton "Create Straps" "DRC"
setFormField "Create Straps" "Use Fat Via If Width Meets Requirement" "1"
setFormField "Create Straps" "Ignore Parallel Targets" "1"
setFormField "Create Straps" "Extend for Multiple Connections" "1"
setFormField "Create Straps" "For Gap <" "0.5"
setFormField "Create Straps" "Start X" "0" :: reset
setFormField "Create Straps" "X Increment" "0"
setFormField "Create Straps" "Direction" "Horizontal" → ①
setFormField "Create Straps" "Start Y" "80"
setFormField "Create Straps" "Net Name(s)" "VDD"
setFormField "Create Straps" "Width" "3.5" → ②
setFormField "Create Straps" "Layer" "35"
setFormField "Create Straps" "Step" "70"
setFormField "Create Straps" "Extend to Low Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Extend to High Boundaries and Generate Pins" "1"
setFormField "Create Straps" "Loop Count" "70"
setFormField "Create Straps" "Y Increment" "100" → ③
formOK "Create Straps"
```

그림 4. 수평 VDD 라우팅을 위한 스크립트
Fig. 4 Script for the horizontal power strap routing

다음으로 수평 방향의 VSS를 위한 전력 스트랩을 배치한다. 수평 방향의 VSS 스트랩을 배치하기 위한 스크립트에서는 VDD가 VSS로 바뀐 것과 스트랩의 시작점이 다르다는 것을 제외하고 그림 4의 스크립트와 거의 동일하다. 스트랩이 시작하는 위치는 VDD에 대해서는 80um이고 VSS에 대해서는 40um라는 것을 확인할 수 있다. 이때 VSS를 위한 전력 스트랩의 간격은 100um로 동일하다. 수직의 경우와 마찬가지로 50um 간격으로 일정하게 VDD와 VSS의 전력 스트랩이 번갈아가면서 생성된다. 이는 그림 6의 결과에서 확인할 수 있다. 그림 6의

확대된 부분에서 VSS에 대한 전력이 생성된다는 것을 알 수 있다.

그림 5. 수평 방향의 VDD 스트랩 결과
Fig. 5 Horizontal VDD strap result

그림 7은 1차 전력 스트랩을 위한 배선을 수행한 결과를 나타낸다. 그림 8에서는 중요 부분에 대해서 스트랩을 확대한 결과를 나타내고 있다.

3.3. 전력 스트랩 결과의 문제점

앞 절들에서 수직 및 수평 방향에 대해서 VDD와 VSS의 전력 스트랩을 배선하였다. 이와 같이 전력 스트랩을 배선한 후에는 아래와 같은 사항에 대해서 고려해야만 한다.

- ① 전력 스트랩 간격의 균일성
- ② 셀의 경계에서 전력 핀의 생성 여부
- ③ 모든 매크로에 대한 VDD, VSS의 전력 스트랩의 연결 여부
- ④ 매크로 링과 전력 스트랩 사이의 비아 배열의 생성 여부(그림 8(a))
- ⑤ 매크로 링과 전력 스트랩 사이의 단일 비아 배열의 생성 여부(그림 8(b))
- ⑥ 매크로 핀 위에 매크로 링과 전력 스트랩 사이의 via2,3 배열의 생성 여부(그림 8(c))

그림 6. 수평 방향의 VSS 스트랩 결과
Fig. 6 Horizontal VSS strap result

일반적으로 ①~③의 문제는 스트랩 생성 시에 주의를 기울여서 스크립트를 올바르게 작성하고 스트랩의 배선을 수행했다면 사전에 방지할 수 있다. 그러나 ④~⑥의 문제는 설계자가 의도하지 않았음에도 발생할 수 있다는 것을 실험적으로 확인할 수 있었다.

그림 7. 1차 전력 스트랩 배선 결과
Fig. 7 the first power strap routing results

④~⑥의 현상은 다음과 같이 분석할 수 있다. 그림 8(a)에서 매크로의 내부에는 metal1,2,3이 모두 사용되므로 매크로의 위로는 metal4 이상만 지나다닐 수 있다. 따라

서 매크로 링과 전력 스트랩 via3,4, via,4,5 등이 많은 영역에 걸쳐있다면 metal4,5 등이 그 부분을 지나갈 수 없으므로 배선에 지장을 준다. 따라서 이런 부분은 제거해야 올바른 전력 스트랩을 배선할 수 있다. 그림 8(b)에서 매크로 내부에도 많은 비아가 존재한다. Astro는 매크로 내부의 비아를 인식할 수 없다. 추후에 설계 룰 체크 (design rule check, DRC)시 매크로 내부의 비아와 외부의 비아가 서로 겹쳐져 있는 경우 문제가 발생한다. 그림 8(c)에서 매크로 핀은 metal1,2,3으로만 연결할 수 있으므로 매크로 핀에 배선을 하기 위해서는 metal1,2,3으로 연결한 후에 metal 4,5로 연결하는 수밖에 없다. 그러나 매크로 내부에서는 이와 같이 metal을 마음대로 연결할 수 없다. 따라서 우선 매크로 핀에 metal1,2,3으로 연결하고 매크로 밖에서 metal4이상으로 이어야 한다. 그러나 핀 위에 via2,3의 배열이 생성되면 매크로 배열을 연결하는데 지장을 받는다. 따라서 이러한 문제들은 제거되어야지만 올바른 전력 스트랩을 배선할 수 있다. 본 논문에서는 이러한 문제들을 제거하는 과정을 다음 장에서 실험과 함께 제안하고자 한다.

(a)

(b)

(c)

그림 8. 1차 전력 스트랩 배선 결과의 확대
(a) ①, (b) ②, (c) ③

Fig. 8 The magnifying the image of the first power strap routing result (a) ①, (b) ②, (c) ③

IV. 제안한 전력 스트랩 배선 방법 및 실험

본 장에서는 1차로 수행한 전력 스트랩 배선 방법의 문제점들에 대해서 해결 방법을 실험과 함께 제안하고자 한다. 전력 스트랩을 위한 제안하는 방법은 다음과 같은 순서로 수행된다. 먼저 배선 가이드를 생성해서 의도하지 않는 배선을 방지하고, 차후를 위해서 배선 가이드에 대한 정보를 덤프한다. 다음으로 불필요한 전력 스트랩을 제거하고, 매크로 핀에 대해 미리 배선을 수행한다. 마지막으로 배선 가이드를 이용하여 최종적인 전력 스트랩 배선을 완료한다.

4.1. 배선 가이드의 생성

1차 전력 스트랩 결과를 근거로 하여 비아 배열이 생성되는 부분을 미리 알고 그 부분에 metal4이하는 배선이 되지 않도록 배선 가이드를 설정한다. 일반적으로 매크로 핀 배선은 문제를 발생할 수 있다. 따라서 전력 스트랩 배선 이전에 매크로 핀에 metal2로 연결해서 이를 매크로 밖으로 미리 뽑아 놓는다. 그림 9에 이러한 전력 스트랩의 배선 가이드를 위한 스크립트를 나타냈다.

```
axgAddRouteGuide
setFormField "Add Route Guide" "No Signal Route on Layer:" "1"
setFormField "Add Route Guide" "Poly" "1"
setFormField "Add Route Guide" "M1" "1"
setFormField "Add Route Guide" "M2" "1"
setFormField "Add Route Guide" "M3" "1"
setFormField "Add Route Guide" "No Automatic Preroutes on Layer:" "1"
setFormField "Add Route Guide" "PPoly" "1"
setFormField "Add Route Guide" "PM1" "1"
setFormField "Add Route Guide" "PM2" "1"
setFormField "Add Route Guide" "PM3" "1"
```

그림 9. 배선 가이드
Fig. 9 Route guide

그림 9를 이용하여 생성한 배선 가이드의 결과를 그림 10에 나타냈다.

그림 10. 배선 가이드 설정
Fig. 10 Route guide setup

4.2. 배선 가이드의 저장

앞에서 많은 매크로에 생성된 비아 배열을 제거하기 위한 배선 가이드가 생성되었다. 이들은 차후를 위하여 저장해 놓는다. 그림 11과 같이 배선 가이드를 저장하게 되면 아직 1차 전력 스트랩 결과들이 존재하므로 이들도 함께 저장될 것이다. 따라서 저장된 덤프 파일에서 VDD, VSS, 및 비아 등은 제거한다.

```
axgDumpRouting
setFormField "Dump Routing" "Output File Name" "dump/Power_Strap.rg"
formOK "Dump Routing"
define _cell (geGetEditCell)
axCreateRouteGuide _cell '(27.45 29.42) '(23.16 23.195) '(0 0 15 15 0)
::axCreateWire _cell "VSS" 33 0 0 3.5 18 0 3.95 '(476.89 30)
::axCreateWire _cell "VDD" 33 0 0 3.08 18 0 3.17 '(48.415 23.855)
::axCreateContactArray _cell "VSS" 9 0 18 '(30 263.025) 480 480 7 11 '0
::axCreateContactArray _cell "VDD" 9 0 18 '(30 263.025) 480 480 7 11 '0
```

그림 11. 배선 가이드의 저장을 위한 스크립트
Fig. 11 Script for dumping route guide

4.3. 전력 스트랩의 제거

기존에 배선된 전력 스트랩에서 필요없는 비아 배열의 위치를 확인한 후에 제거한다. 이것은 새로 전력 스트랩을 배선하기 위해서이다.

```
axgDelRouteType
formDefault "Delete Route Type"
setToggleField "Delete Route Type" "Clock" "ring" 1
setToggleField "Delete Route Type" "P/G" "ring" 1
setToggleField "Delete Route Type" "Clock" "strap" 1
setToggleField "Delete Route Type" "P/G" "strap" 1
formOK "Delete Route Type"
```

그림 12. 전력 스트랩의 제거
Fig. 12 Remove of the first power strap

4.4. 전-배선 매크로 핀

전력 스트랩의 문제점을 제기하는 부분에서 언급했듯이 일반적으로 매크로 핀 배열시 핀위의 스트랩과 링 사이의 비아 배열은 문제를 야기한다. 따라서 그림 13과 같이 스트랩을 배열하기 전에 미리 매크로 핀을 metal2로 배열해 놓으면 매우 효율적이다. Fig. 14에서 매크로 핀의 차후 배열 패턴을 비교하였다.

```

setupleg
geNameSelect
setFormField "Name Select" "Name" ".*SRAM_I0"
setFormField "Name Select" "Type" "cell instance"
setFormField "Name Select" "pattern match" "1"
formOK "Name Select"
MRG "metal3"
MRG "metal1"
MPR "metal2"
RPRW
geDeselectAll
axDeleteAllRouteGuide
  
```

그림 13. 전-배치 매크로 핀의 생성을 위한 스크립트
Fig. 13 Script for generating the pre-route macro pin

그림 14. 전-배선 매크로 결과
(a) 매크로 핀 배선 (b) 배선 패턴
Fig. 14 Pre-route macro results
(a) Macro pin Routing (b) Routing pattern

V. 결 론

본 논문에서 효율적인 전력 스트랩의 배선을 위한 방법론을 제안하였다. 전원 배선 방법은 수직과 수평 방향으로 수행되는데 블록 설계 시 전력 스트랩의 종류는 크게 네 가지이다. 이 네 가지는 수직 VDD 및 VSS와 수평 VDD 및 VSS를 위한 전력 스트랩이다. 먼저 1차 전력 스트랩 배선을 수행하였고, 이 결과를 분석하여 일반적으로 쉽게 발생하는 문제점을 파악하였다. 이러한 문제점들을 해결하기 위한 전력 스트랩 배선 방법론을 제안하였다. 먼저 배선 가이드를 생성해서 의도하지 않는 배선을 방지하였고, 차후를 위해서 배선 가이드에 대한 정보를 저장하였다. 다음으로 불필요한 전력 스트랩을 제거하고, 매크로 핀에 대해 미리 배선을 수행하였다. 최종적으로 배선 가이드를 로딩하여 최종적인 전력 스트랩 배선을 완료하였다. 결과로 얻어진 전력 스트랩은 1차 전력 스트랩 배선에서 발생한 문제들을 효율적으로 제거할 수 있었다. 이와 같이 제안한 기법은 전력 스트랩 배선에 관한 많은 연구 결과가 없는 상황에서 ASIC 설계자들에게 좋은 가이드라인을 제시할 것으로 사료되고, ASIC 과정의 효율성을 높이는데 기여를 할 것으로 기대한다.

감사의 글

이 논문은 2011년도 광운대학교 교내 학술연구
구비 지원에 의해 연구되었음.

참고문헌

- [1] Olivier Coudert, "Timing and Design Closure in Physical Design Flows", ISQED 2002, pp.511-516, Mar., 2002.
- [2] Bart Vermeulen and Sandeep Kumar Goel, "Design for Debug: Catching Design Errors in Digital Chips", MDT2002, Vol.19, pp.35-43, May-Jun., 2002.
- [3] R. Arunachalam, K. Rajagopal, L. Pileggi, "TACO: Timing Analysis with Coupling", Proc. of ICCAD

- ’’2000, Nov., 2000.
- [4] G. Stenz, B.M. Reiss, B.Rohfleisch, F.M. Johannes, “Timing Driven Placement in Interaction with Netlist Transformations”, in Proc. of ISPD’97, pp.36-41, 1997.
- [5] Himanshu Bhatnagar, Advanced ASIC Chip Synthesis Using Synopsys® Design Compiler™ and PrimeTime®, Kluwer Academic Publishers, 1999.
- [6] Synopsys, Prime Time User Guide Advanced Timing Analysis, 2005.
- [7] 서영호, 최현준, 유지상, 김동욱, “ASIC ECO 단계에서 효율적인 Timing Closure 방법론”, 한국정보통신학회, Vol.13, No.3, pp.522-530, 2009.3.
- [8] J. Minz, X. Zhao, and S. K. Lim. buffered clock tree synthesis for 3d ics under thermal variations. In Proc. Asia and South Pacific Design Automation Conf., Jan., 2008.
- [9] T.-Y. Kim and T. Kim. Clock tree embedding for 3d ics. In Proc. Asia and South Pacific Design Automation Conf., Jan., 2010.
- [10] X. Zhao, D. Lewis, H.-H. S. Lee, and S. K. Lim. Pre-bond Testable Low-Power Clock Tree Design for 3D Stacked ICs. In Proc. Int. Conf. on Computer Aided Design, Nov., 2009.
- [11] F. Liu. A General Framwwork for Spatial Correlation Modeling in VLSI Design. In Proc. Design Automation Conf., Jun., 2007.
- [12] 서영호, 최의선, 김동욱, “다중 클록 영역의 SoC를 위한 효율적인 버퍼삽입”, Vol.16, No.4, pp.643-653, 2012.4.
- [13] 서영호, 김동욱, “효율적인 SoC 논리합성을 위한 혼합방식의 설계 방법론”, 한국정보통신학회, Vol.16, No.3, pp.571-578, 2012.3.
- [14] G. D. Hachtel, F. Somenzi, “Logic Synthesis and Verification Algorithms”, Kluwer Academic Pub., 1996.
- [15] Jie-Hong (Roland) Jiang, Srinivas Devadas (2009). “Logic synthesis in a nutshell”. In Laung-Terng Wang, Yao-Wen Chang, Kwang-Ting Cheng. Electronic design automation: synthesis, verification, and test. Morgan Kaufmann. ISBN 9780123743640. chapter 6.
- [16] Gary D. Hachtel; Fabio Somenzi (1996). Logic synthesis and verification algorithms. Springer. ISBN 0792397460. also as published as softcover ISBN 0387310045 in 2006
- [17] Svoboda, A., D.E. White. Advanced Logical Circuit Design Techniques. Garland Press, New York, 1979.
- [18] J. Minz, X. Zhao, and S. K. Lim. buffered clock tree synthesis for 3d ics under thermal variations. In Proc. Asia and South Pacific Design Automation Conf., Jan 2008.
- [19] T.-Y. Kim and T. Kim. Clock tree embedding for 3d ics. In Proc. Asia and South Pacific Design Automation Conf., Jan., 2010.
- [20] X. Zhao, D. Lewis, H.-H. S. Lee, and S. K. Lim. Pre-bond Testable Low-Power Clock Tree Design for 3D Stacked ICs. In Proc. Int. Conf. on Computer Aided Design, Nov., 2009.
- [21] F. Liu. A General Framwwork for Spatial Correlation Modeling in VLSI Design. In Proc. Design Automation Conf., Jun., 2007.
- [22] H. Eisenmann, F. M. Johannes, “Generic Global Placement and Floorplanning”, in Proc. of 35th DAC, Jun., 1998.
- [23] Synopsys, Astro User Guide, 2005.

저자소개

서영호(Young-Ho Seo)

1999년 2월 : 광운대학교 전자재료 공학과 졸업(공학사)

2001년 2월 : 광운대학교 일반대학원 졸업(공학석사)

2004년 8월 : 광운대학교 일반대학원 졸업(공학박사)

2003년 9월 ~ 2004년 6월 : 한국전기연구원 연구원

2005년 9월 ~ 2008년 2월 : 한성대학교 조교수

2008년 3월 ~ 현재 : 광운대학교 교양학부 부교수

※관심분야: 2D/3D 영상 및 비디오 처리, 디지털 홀로그램, SoC 설계, 워터마킹/암호화

김동욱(Dong-Wook Kim)

1983년 2월 : 한양대학교 전자공학과
졸업(공학사)

1985년 2월 : 한양대학교 전자공학과
졸업(공학석사)

1991년 9월 : Georgia공과대학 전기공학과 졸업
(공학박사)

1992년 3월 ~ 현재 : 광운대학교 전자재료공학과
정교수

2000년 3월 ~ 2001년 12월 : 인티스닷컴(주) 연구원

2006년 3월 ~ 현재 : (사)실감미디어산업협회 이사

2009년 3월 ~ 현재 : 광운대학교 실감미디어 연구소
소장

※ 관심분야 : 3D 영상처리, 디지털 홀로그램, 디지털
VLSI Testability, VLSI CAD, DSP 하드웨어 설계