

## 마이크로프로세서 설계 무작정 따라하기 part-III (5)

부제: 합성, 배치 및 배선

KAIST 전자전산학과 박사과정 배영돈(<http://www.donny.co.kr>)

이번 강좌로 ‘마이크로프로세서 설계 무작정 따라하기’의 연재가 끝을 맺습니다. 칩을 만들기 위한 최종 형태인 레이아웃을 만드는 배치 및 배선(place & route)과정을 설명합니다.

### 1. 후단계 설계 (Back-end)

일반적으로 반도체의 설계과정은 전단계(Front-end)와 후단계(Back-end)로 구분한다. 전단계는 원하는 설계를 Verilog/VHDL 로 설계하고 시뮬레이션과 합성을 하는 작업이며, 후단계는 합성된 netlist 를 반도체 제작을 위한 형태인 레이아웃(layout)으로 만들어 내는 과정이다. 많은 경우 ASIC 설계자들은 전단계까지만 설계하고, 후단계는 디자인하우스에서 작업한다. 그러나, 좋은 설계를 위해선 설계의 시작부터 이 후단계를 고려하는 것이 중요하다. 후단계 설계자역시 전단계를 이해하면 훨씬 더 좋은 결과를 만들어낼 수 있을 것이다.

후단계 설계는 다른 말로 배치 및 배선 (Place & Route)이라고 부른다. 말 그대로 회로를 구성하는 소자들을 위치시키고 서로 연결하는 과정이기 때문이다. 그림 1 은 후단계 설계과정을 나타내고 있다. 먼저 합성 결과는 netlist 파일로 저장이 된다. 이 때, 주로 gate-level Verilog 나 EDIF 등의 형식이 사용된다. 이 netlist 는 후단계 설계용 CAD 툴에 입력된다. 널리 사용되는 CAD 툴은 Avant!사(현재는 Synopsys 에 병합되었음)의 Apollo, Cadence 사의 Silicon Ensemble, Mentor 의 ICStation, 등이 있으며 본 강좌에서는 Apollo 를 사용한다.

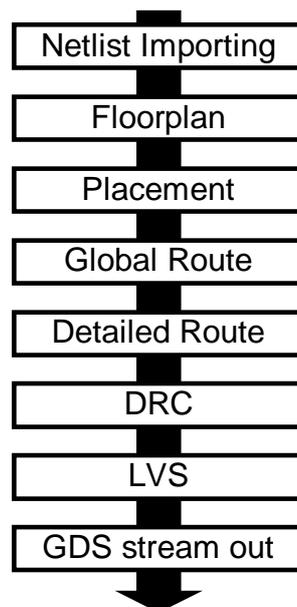


그림 1. 후단계 설계과정

netlist 를 가져온 후에는 floorplan 을 한다. floorplan 은 standard cell 과 macro 를 어떻게 배치/배선 할 것인지를 대략적으로 계획하는 작업이다. floorplan 을 어떻게 하는가에 따라 배치/배선작업이 간단해질 수도 있고, 불가능해질 수도 있으며 칩의 성능에도 영향을 미칠 수 있다. IO pad 의 위치, macro(SRAM, 등)의 pin 의 위치, standard cell 을 배치하기위한 core 영역, power/ground 의 연결, 등을 고려하면 좋은 결과를 얻을 수 있다.

placement 과정에서는 standard cell 이나 macro 들을 배치시키며, global route 와 detailed route 과정에서 standard cell, macro, pad 들을 서로 연결한다. DRC (Design Rule Check)는 각 layer(metal1, metal2, 등)들 간의 간격이나 두께가 공정에서 정해진 규격에 일치하는지 확인하며, LVS (Layout Versus Schematic)은 배치, 배선결과가 입력된 회로와 일치하는지 확인한다. 이 과정까지 완료되면 GDSII 형식으로 출력되어 공정과정에 들어가게 된다.

## 2. Netlist 의 준비

지난 강좌에서 SimpleCore 의 Verilog 코드를 합성하여 gate-level 의 netlist 를 만들었다. 실제 칩을 만들기 위해선 IO pin 에 pin 번호를 지정하고, netlist 에 pad cell 을 추가하는 과정이 필요하다. 칩을 제작하고자 하는 공정과 die 의 크기에 따라 사용할 수 pin 의 개수가 결정되므로, Verilog 코드상에 존재하는 IO port 의 개수와 전원을 공급하기위한 power/ground pin 을 수용할 수 있어야 한다. 본 강좌에서는 IDEC MPW Hynix 0.35μm 공정을 사용한다. 이 공정은 4mmx4mm 크기의 die 에 100 개의 pin 을 제공한다. 그림 2 는 pin 번호를 할당한 것을 보여준다.

Verilog netlist 에 pad cell 을 추가하는 방법은 다음과 같다.

```

pv0f    pad12(vss);
pc3c04  pad13(.cclk(clk ), .cp (pclk ));
pc3d01  pad14(.pad(nreset), .cin (pnreset))
pvdf    pad15(vdd);
    
```

pad cell name 은 library databook 을 보고 적당한 종류를 고르고, instance 의 이름은 해당 공정에 정해진 방식(naming convention)을 따른다. instance 의 이름은 pad instance 를 자동으로 place 하기위해 사용된다.

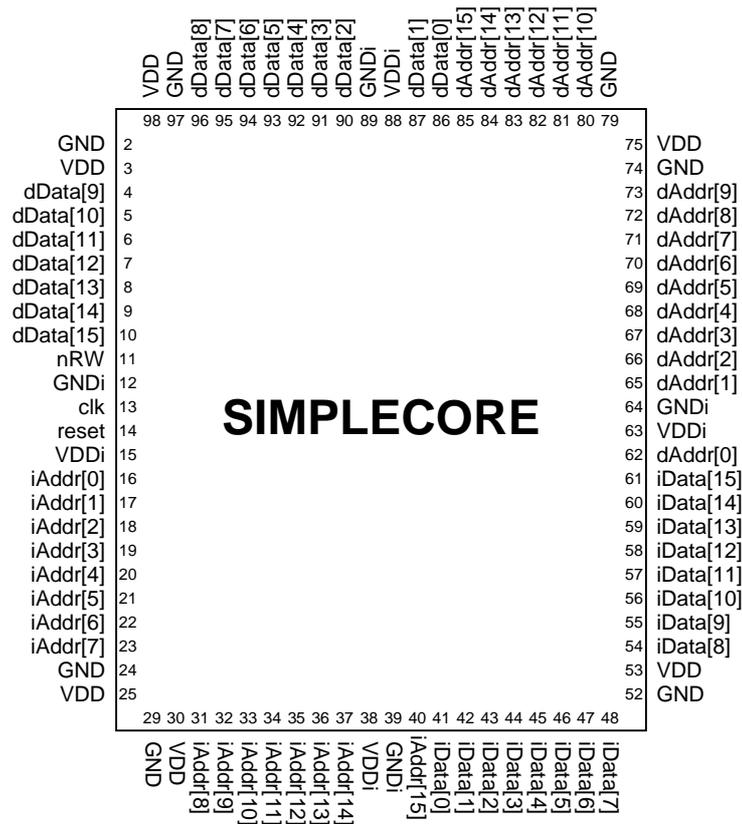


그림 2. Pin diagram

### 3. Netlist 불러오기 (Milkyway)

Milkyway 는 여러가지 형태의 파일을 Apollo 용 자료형태로 변환하거나 Apollo 의 작업결과를 다른 형태로 출력하는 기능을 한다. Pad cell 이 추가된 Verilog netlist 를 Apollo 에서 사용하기 위해 읽어 들이는 과정을 알아보자.

#### 3.1 Milkyway 의 실행

\$ Milkyway &

#### 3.2. 라이브러리 생성

해당 공정의 technology file (예제의 Apollo.tf)이 필요하다. ‘Library→Create...’메뉴를 실행한다.

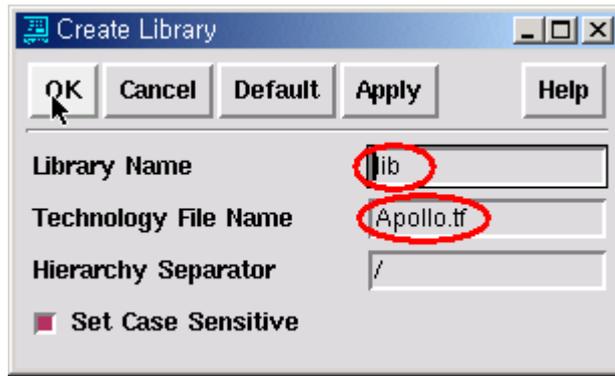


그림 3. 라이브러리의 생성

3.3. Verilog 파일 읽어오기

‘Netlist In→Verilog In...’메뉴를 실행한다.

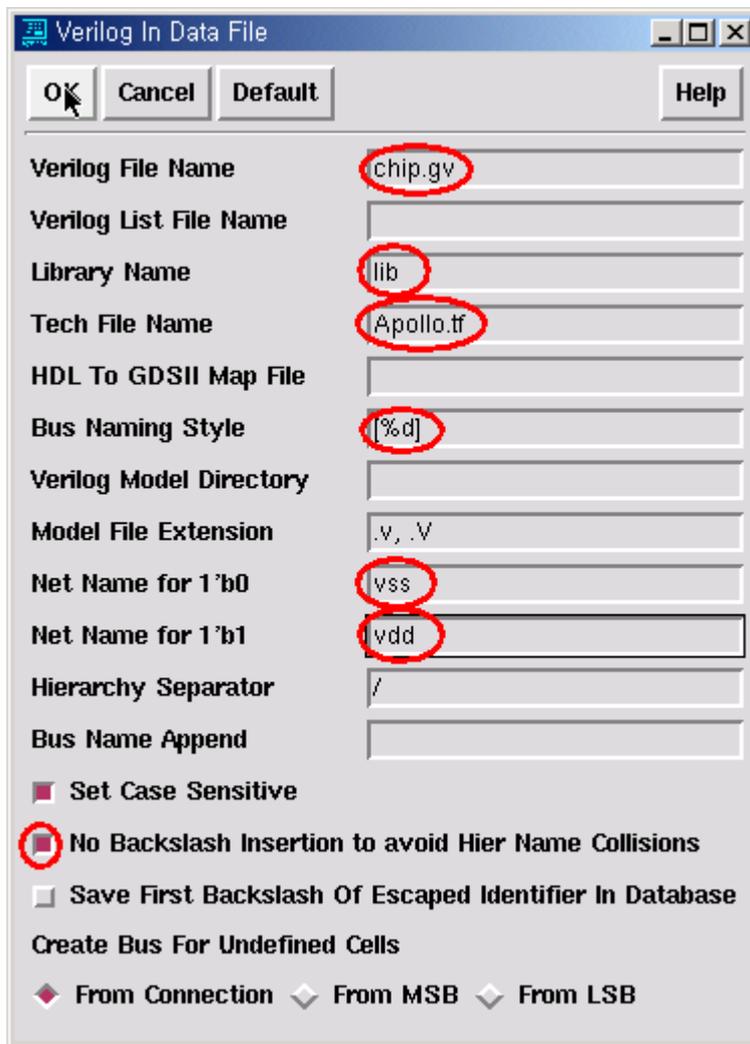


그림 4. Verilog 파일 읽어오기

3.4. Reference Library 추가

Standard cell 이 저장되어있는 라이브러리의 경로를 입력한다. netlist 파일에서 참조된 모든 cell 이나 macro 에 대한 reference library 를 추가해야 한다. ‘Library→Add Ref...’메뉴를 실행한다. Hynix 0.35μm 공정에서는 cb35os142d 와 cb35io122d 를 추가해야한다.

‘Library→Show Ref...’메뉴를 실행하여 결과를 확인할 수 있다.

```

-----
Reference Library List for lib
Library cb35iol22d (relative to main library)
Library cb35os142d (relative to main library)
End of Reference Library List
-----
Opened Reference Libraries for lib
Library /home/donny/simplecore/pnr/cb35iol22d
Library /home/donny/simplecore/pnr/cb35os142d
End of Opened Reference Libraries

```

### 3.5. Netlist Expand

netlist 파일에서 사용되는 cell 이 정의되어있지 않은 경우나 port 이름이 일치하지 않는 경우 에러를 발생한다. Expand 과정에서 global net 을 정의해준다.

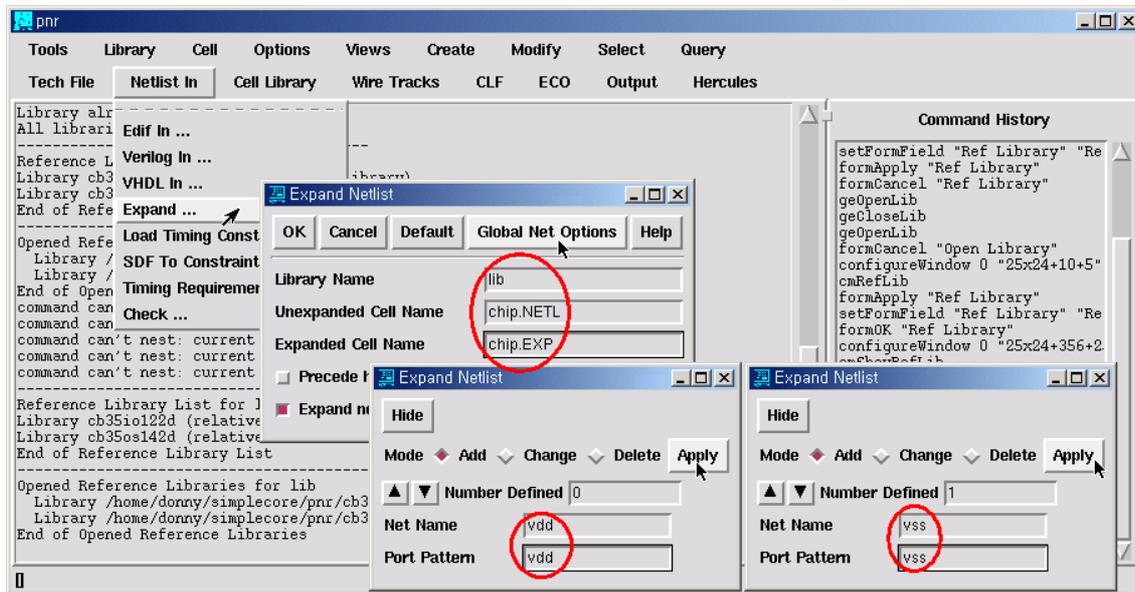


그림 5. Netlist Expand

### 3.7. Milkyway 종료

‘Tools→Quit’ 메뉴 실행

### 4. Place & Route

Milkyway 에서 준비된 netlist 를 가지고 Apollo 에서 레이아웃을 생성하도록 해보자.

#### 4.1. Apollo 의 실행

\$ Apollo &

4.2. ‘Library→Open...’ 메뉴를 실행하여 Milkyway 에서 생성했던 ‘lib’을 연다.

4.3. ‘Cell→Create...’ 메뉴를 실행하여 ‘chip’이름으로 cell 을 생성한다. ‘chip’은 Verilog netlist 의 pad 까지 포함한 module 이름이다.

4.4. ‘Design Setup→Netlist→Bind Netsit...’메뉴를 실행하여 Milkyway 에서 Expand 한 결과 netlist 인 chip.EXP 를 입력한다.

4.5. ‘Design Setup→TDF→Load TDF...’메뉴를 실행하여 padplace.tdf (라이브러리와 함께 제공하는 매크로파일)을 실행한다. 그 결과, power/ground 의 정보와 pad cell 들의 위치정보를 설정하게 된다.

4.6. ‘Design Setup→Floorplan→Set Up Floorplan...’메뉴를 실행한다.

먼저 Control Param 이 aspect ratio (default)인 상태에서 Core Utilization 을 0.9 정도로 입력하고 ‘Set’버튼을 클릭한다. 그러면 Core Width 와 Core Height 가 계산되어 나온다. 우리가 사용할 die 의 크기는 4mmx4mm 임을 고려하여 Core 크기에 따른 여백을 입력한다.

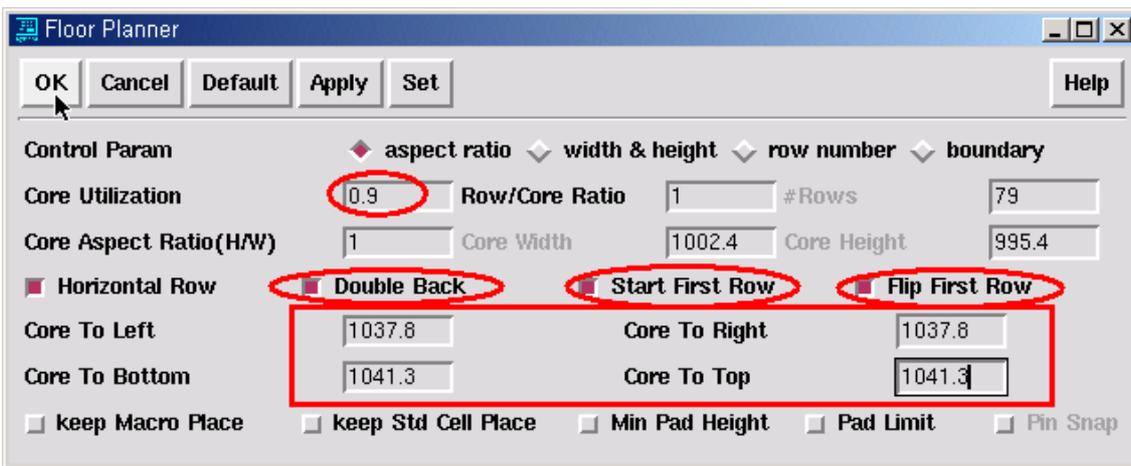


그림 6. Set Up Floorplan

그 결과 그림과 같이 Core 와 IO 영역이 정의된 것을 볼 수 있다. TDF 파일의 실행에 의하여 pad cell 이 함께 place 되었으며, 오른쪽에는 아직 place 되지않은 standard cell 들을 볼 수 있다.

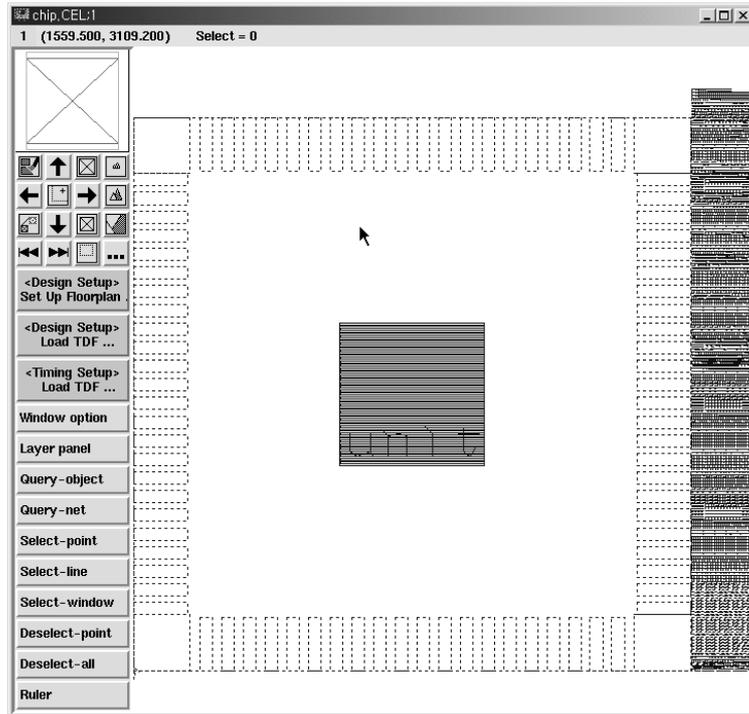


그림 7. Floorplan 결과

4.7. 'PreRoute→Rectangular Rings...'메뉴를 실행하여 Core 영역 주변에 power ring 을 추가한다. 본 예제에서는 50 $\mu$ m 의 width 와 5 $\mu$ m 의 offset 을 사용하였다.

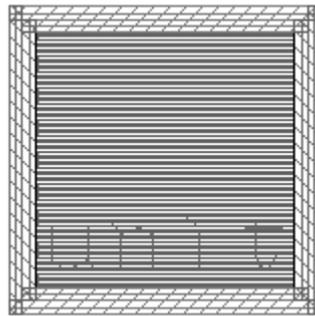


그림 8. Power Ring 의 추가

4.8. 'Place→Place Cells→Design Placemen...'메뉴를 실행하여 standard cell 을 place 한다.

4.9. 'PreRoute→Connect Ports to P/G...'메뉴를 실행하여 standard cell 과 power/ground net 을 연결한다.

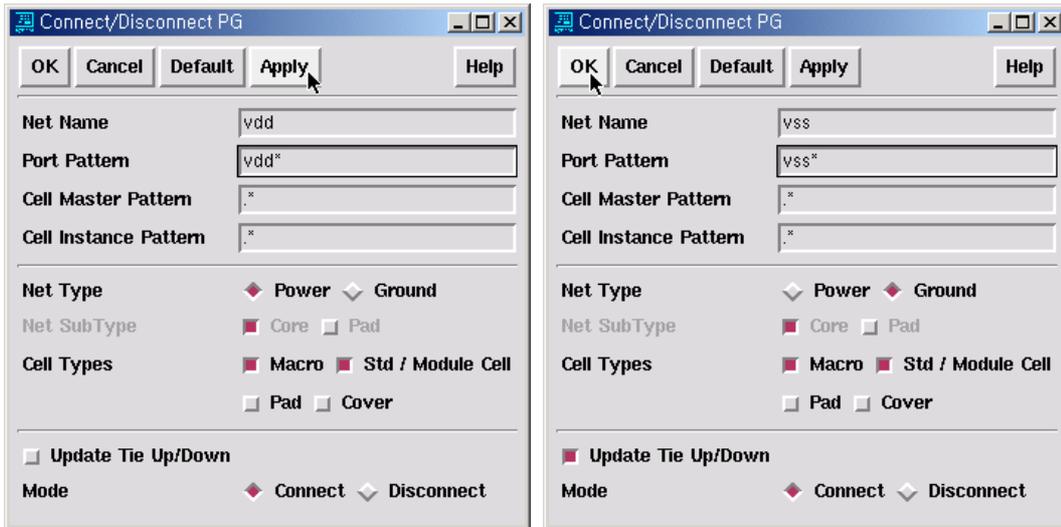


그림 9. Connect Ports to P/G

- 4.10. 'PreRoute→Macros/Pads...'메뉴를 실행하면 power ring 과 power/ground pad 가 연결된다.
- 4.11. 'PreRoute→Standard Cells...'메뉴를 실행하면 power ring 과 standard cell power/ground port 가 연결된다.
- 4.13. 'Route→ Global Route →Global Route...'메뉴를 클릭하여 global route 를 실행한다.
- 4.14. 'Route→Detail Route→Initial Detail Route...'메뉴를 클릭하여 detail route 를 실행한다.
- 4.15. 'Verify→DRC'와 'Verify→LVS'를 실행하여 회로를 검증한다.

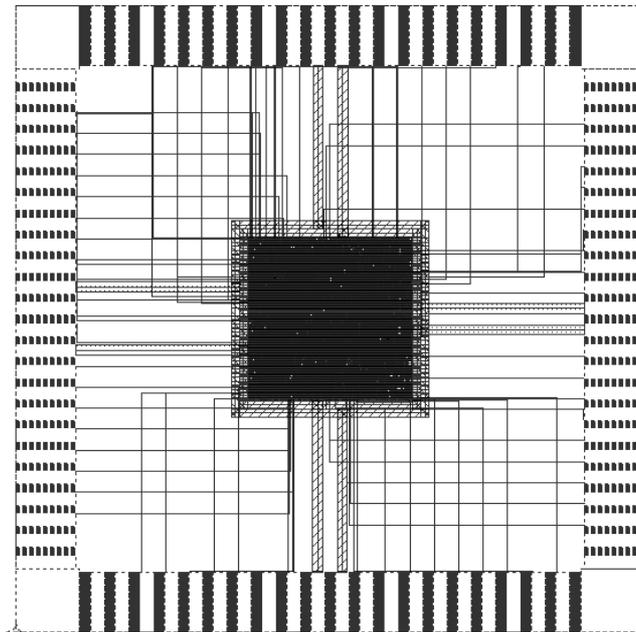


그림 10. 배치 및 배선 결과

5. 칩 제작

지금까지 설명한 것과 같이 배치 및 배선이 완료된 설계는 마지막으로 cell merge 과정을 거친다. 배치 및 배선에 사용된 standard cell 들은 입출력 정보들만 갖고있는 phantom cell 이다. 이것을 실제 칩 제작을 위한 physical cell 로 변환하는 과정을 거친다. physical cell 은 기밀에 해당하므로 일반 설계자들은 physical cell 에 접근 할 수는 없다. Physical cell 의 구조가 궁금한 경우에는 IDEC 에서 직접 개발한 라이브러리를 참고하기 바란다. 그림 11 은 cell merge 단계를 거쳐 실제로 제작된 칩의 현미경 사진이다.

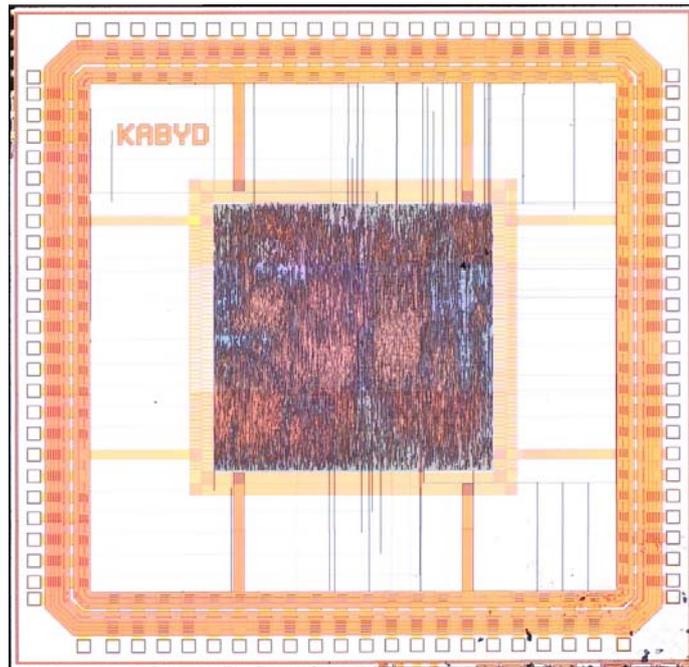


그림 11. 제작된 칩의 사진

지난 1 년 여 동안 반도체 설계를 공부하는 학생이나 마이크로프로세서 설계경험이 없는 설계자들이 실제로 마이크로프로세서를 설계할 수 있도록 쉬운 예제와 구체적인 설명으로 조금이라도 쉽게 전달하고자 노력하였습니다. 한정된 지면으로 충분한 내용을 담기엔 어려움이 있어서 아쉬움이 남습니다. 일천하지만 국내 반도체분야의 발전에 힘이 되었으면 하는 바람입니다.

본 강좌를 연재할 수 있도록 지면을 할애해주신 경종민 교수님과 IDEC 의 김명현씨 그리고 마이크로프로세서에 대해서 가르쳐주신 박인철교수님께 감사의 말을 전합니다.

2002 년 10 월 29 일 배영돈

본 강좌에 사용된 예제는 홈페이지(<http://www.donny.co.kr/simplecore>)를 통해 제공됩니다.