

논문 2007-44SD-11-9

# CMOL FPGA 자동 레이아웃 설계

## ( Automatic Layout Design of CMOL FPGA )

김 교 선\*

( Kyosun Kim )

## 요 약

본 논문은 CMOS-나노 혼성 소자들 중 실현 가능성이 높은 CMOL 회로 설계 자동화 전용 시스템을 최초로 제안한다. CMOL 아키텍처는 NOR 게이트를 이용하여 조합회로의 구현한다. 이 CMOS-나노 혼성 구조는 논리 함수 및 연결 구조가 동일 나노 소자 하드웨어 자원을 공유하는 특징을 가진 새로운 설계 자동화 문제를 제기한다. CMOL의 물리 설계 자동화를 위해 먼저 CMOL 하드웨어 아키텍처 모델을 개발하고 이 CMOL 구조를 위한 배치 배선 문제에서 CMOL만의 특수한 제약조건을 도출하였으며 배치 알고리즘과 게이트 지정 알고리즘을 루프 형태로 조합하여 해를 구하는 접근 방법을 제안하였다. 제안된 접근 방법을 사용하여 CMOL 나노 전자 아키텍처를 위한 전용 설계 자동화 시스템을 개발하였으며 이 시스템을 이용하여 마이크로프로세서 설계에 실제 사용되는 모듈들을 구현함으로써 그 효용성을 증명하였다.

## Abstract

We developed the first automatic design system targeting a promising hybrid CMOS-Nanoelectronics Architecture called CMOL<sup>[2]</sup>. The CMOL architecture uses NOR gates to implement combinational logic. In this hybrid CMOS-nanoelectronics architecture, logical functions and the interconnections share the nanoelectronics hardware resource. Towards automating the CMOL physical design process, we developed a model for the CMOL architecture, formulated the placement and routing problems for the CMOL architecture subject to the unique CMOL specific constraints, and solved it by combining a placement algorithm with a gate assignment algorithm in a loop. We validated the proposed approach by implementing several industrial strength designs.

**Keywords :** CMOL, FPGA, Automatic Layout Design

## I. 서 론

최근 연구에 의하면 현재 리소그래피 식각 및 CMOS 회로, 그리고 부울 대수를 기초로 하는 VLSI 패러다임을 수 nm 영역 이하까지 지속하는 것이 거의 불가능할 것이라는 전망이다<sup>[1]</sup>. 생산에 있어 공정 변동은 피할 수 없는데 게이트 길이가 10 nm 이하에서는 전계 효과 트랜지스터 (MOSFET)의 파라미터, 특히 게이트 문턱 전압이 공정 변동에 따라 지수 함수적으로 영향을 받게

된다는 것이 주요 원인이다. 이것을 해결하려면 게이트 길이를 수 Å 내외로 제어해야 하는데 이는 반도체 산업 발전 속도를 장기적으로 전망해 보아도 기대할 수 없는 정확도이다. 그 같은 정확도가 복잡한 패터닝을 통해 기술적으로는 혹시 구현된다 하더라도 현재도 지수 함수적으로 증가하고 있는 공정 장비 비용을 천정부지로 치솟게 할 것이며 10년 이내에 무어의 법칙은 그 수명을 다하는 결과를 가져올 것이라<sup>[2]</sup> 한다. 마이크로일렉트로닉스 기술 진보 상에 임박한 위기를 해결하기 위해서는 기존에 리소그래피를 기본으로 했던 하향식 공정 방식에서 상향식 접근 방법으로 선회하는 급진적 패러다임 변화가 효과적일 수 있다는 데 의견 일치가 급속히 증가하고 있다. 상향식 접근 방법은 근본적으로 재현성을 보장하는 특수한 방법으로 최소형의 능동 소

\* 정희원, 인천대학교 전자공학과

(Department of Electronic Engineering, University of Incheon)

※ 이 논문은 인천대학교 2006년도 자체 연구비 지원에 의하여 연구된 것임.

접수일자: 2007년3월27일, 수정완료일: 2007년10월17일

자를 형성시킬 수 있어야 한다. 그 같은 소자의 가장 대표적인 예가 특수 설계하여 화학적으로 합성한 분자이며 흔히 일이백 개의 원자들로 구성된다. 이는 단전자 섬 역할을 하는 역셉터 및 터널 접합 역할을 하는 비전도성 단편과 같은 기능부를 포함하며, 미리 제조된 전극(티올 또는 이소시아나화물) 상에 분자의 자기 조립(self-assembly)을 화학적으로 유도할 수 있을 뿐만 아니라 상온에서 분자의 정밀성 및 안정성을 충분히 보장한다.

불행히도, 분자 소자로 구성된 집적 회로는 소자의 기능성이 제한적이기 때문에 그 자체만으로는 거의 실용성이 없다. 예를 들어, 알려진 물리적 효과(전계 효과, 양자 간섭, 단전자 충전)를 기준으로 1 nm 단위의 트랜지스터 전압 이득은 1을 초과하기 어렵다. 전압 이득이 1을 초과해야 한다는 것은 거의 모든 아날로그 혹은 디지털 능동 회로 동작에 필요한 기준이다. 따라서 고 성능 나노 전자 회로를 구현하는 방법들 중 희망이 보이는 유일한 방법은 분자 소자와 이를 연결하는 나노 와이어를 CMOS 회로와 집적시키는 것이라 할 수 있다. 비교적 큰 전계 효과 트랜지스터가 추가적으로 필

요한 기능성, 특히 높은 전압 이득을 제공할 수 있기 때문이다.

CMOS-나노 혼성 소자들 중 실현 가능성이 높은 것 중에 하나가 CMOL 회로<sup>[2]</sup>다. CMOL은 CMOS/nano wire/MOLecular 혼성 구조를 줄인 말이다. 대부분의 혼성 소자들과 마찬가지로 CMOL 회로에서 나노 소자는 크로스바 배열의 각 교차점에 형성되며 CMOS와 나노 소자 간 접속 문제가 존재한다. 이 접속 문제를 극복하기 위해 CMOL 회로에서는 그림 1(a)와 같이 하부에 위치한 CMOS 회로 층 전면에 핀들을 분포시켜 상부에서 크로스바를 형성하는 나노 와이어와 연결한다. 그림 1(b)와 같이 서로 수직인 CMOS 신호선 쌍에 전압을 인가하면 두 개의 핀이 두 개의 나노 와이어 쌍을 경유하여 접속되도록 할 수 있다. 그림 1(c)와 같이 핀들은 간격이  $2\beta F_{CMOS}$  인 ( $F_{CMOS}$ 는 CMOS 셀의 반 간격이고  $\beta$ 는 CMOS 셀의 복잡도에 따른 비례 상수) 정사각형 배열 형태를 이룬다. CMOS 핀 배열은 나노 와이어 크로스바에 대해 각도  $\alpha = \arcsin(F_{nano}/\beta F_{CMOS})$  ( $F_{nano}$ 는 나노 와이어의 반 피치) 만큼 회전한다. 이와 같이 회전 시킴으로써  $F_{nano} \ll F_{CMOS}$ 인 경우에도 각 나노 소자를 따로따로 접근하는 것이 가능하게 된다. 따라서 각 나노 와이어는 하나의 핀에만 직접 연결되며 한 핀에서 바로 옆 핀까지는 그림 1(c)와 같이  $r$ 개의 나노와이어가 지나게 된다.  $r$ 은 CMOS 집적도와 나노와이어 집적도 간의 비율 및 각도  $\alpha$ 를 고려하여 결정될 수 있다.

만약 나노 소자들이 일반적인 다이오드처럼 예리한 문턱 전압 전류 특성을 가진다면 각각을 검사할 수 있다. 더구나, 만약 그 같은 다이오드가 단전자 래칭 스위치처럼 두 가지 내부 상태 간에서 천이한다고 할 때 두 나노 와이어 쌍 간에 전압  $\pm V_w$ 를 인가하면 이들 사이에서 유일하게 선택된 소자에는  $V = \pm 2V_w$ 가 인가되어 켜지거나 꺼지도록 할 수 있고 다른 소자에는  $V = \pm V_w$ 가 인가되어 영향을 주지 않도록 할 수 있다.

CMOL 회로는 테라 비트 규모의 메모리, 그 구조가 소위 FPGA를 연상시키는 부울 논리 회로, 그리고 혼성 신호 신경망 회로를 포함하는 응용 분야에 적용될 수 있으며 이들은 재구성(reconfiguration)이 가능하고 결합 내성이 높다고 보고<sup>[2]</sup>되었다. 이들 중 CMOL FPGA 회로는 그림 2(b) 같이 두 개의 패스 트랜지스터와 하나의 인버터를 포함하는 CMOS 기본 셀이 두 개의 핀을 통하여 나노 와이어 / 분자 소자 구조에 연결된다. 재구성 과정에서 인버터를 꺼 놓으며 패스 트랜지스터를 사용하여 각 분자 소자들의 상태를 지정한다. 반대

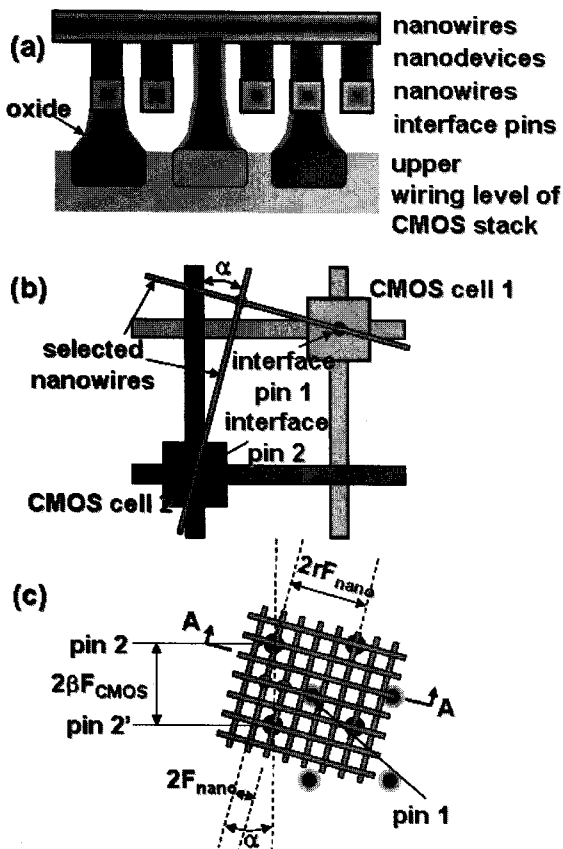


그림 1. CMOL 회로의 하드웨어 아키텍처  
Fig. 1. Low-level structure of the generic CMOL circuit.

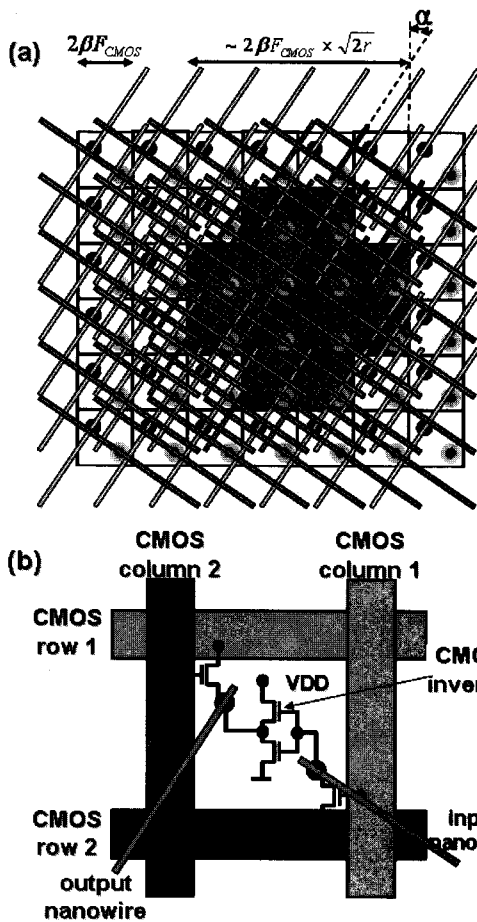


그림 2. CMOL FPGA: (a) 회로의 일반적 구조, (b) CMOS 셀 구조  
 Fig. 2. CMOL FPGA: (a) the general structure of the circuit, (b) a single CMOS cell.

로 동작 중에는 패스 트랜지스터를 꺼 놓는다. CMOS 셀의 각 핀은 나노 와이어-나노 소자-나노 와이어 연결 구조를 따라 그 핀 주변에 그림 2(a)에 직사각형 구간으로 어둡게 표시된 “연결 영역” 내에 있는  $M = 2r^2 - 2r - 2$ 개의 다른 셀들 각각에 연결될 수 있다<sup>[3]</sup>. 그림 3은 이 같은 조직이 어떻게 NOR 게이트를 구현하도록 재구성될 수 있는지 보이고 있다. 그림 3에서 A나 B가 둘 중 하나라도 1이면  $R_{pass}$ 에 전압이 걸려 인버터의 입력은 1이 되고 NOR 게이트의 출력 F는 0이 된다. A와 B 모두가 0이면 인버터의 입력은 0이 되어 “F”는 1이 된다.

기존에는 조합회로를 수작업으로 CMOL FPGA 조직에 매핑하거나<sup>[3]</sup> 또는 대학 연구용으로 개발된 CMOS FPGA 설계 자동화 방식과 소프트웨어를 약간 변형하여 CMOL FPGA 설계 자동화를 시도한 연구<sup>[4]</sup>가 있었다. 본 논문에서는 CMOL FPGA 전용 자동 레이아웃 소프트웨어 및 상용 논리 합성 시스템이 함께 집적된

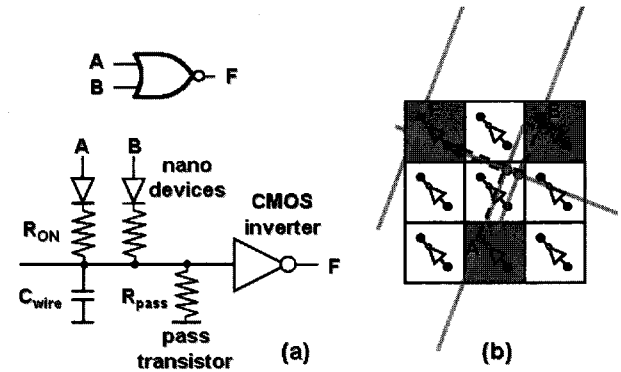


그림 3. NOR 게이트: (a) 등가 회로, (b) CMOL에서의 물리적 구현  
 Fig. 3. NOR gate: (a) equivalent circuit, (b) physical implementation in CMOL.

전용 설계 방식을 소개한다.

본 논문은 다음과 같이 구성한다. II장에서 CMOL 아키텍처 모델링에 관해 논의하고 논의된 내용을 기반으로 본 논문에서 제안하는 CMOL 설계 자동화 과정에 대해 III장에서 설명한다. 마지막으로 제안된 기법을 하드웨어 기술언어로 설계된 회로에 적용하여 실험한 결과를 IV장에서 보이면서 결론을 맺는다.

## II. CMOL 아키텍처 모델링 및 제약조건

앞장에서 설명한 CMOL 하드웨어 아키텍처를 기반으로 HDL 기술에서 CMOL FPGA 레이아웃 생성까지 이르는 설계 전 과정을 그림 4를 통해 살펴보자. 그림 4(a)는 1비트 비교기의 레지스터 전달 수준 회로 기술이며 이를 NOR 게이트를 기본으로 한 네트리스트로 합성하면 그림 4(b)가 된다. 기본 소자의 구성을 간단히 하기 위해 플립플롭은 이에 대응하는 NOR 게이트 회로로 대체하게 되고 그 결과 NOR 게이트로만 이루어진 회로 그림 4(c)를 얻을 수 있다.

CMOL 아키텍처에서 모든 셀들은 주어진 정밀도에 따라  $2\beta F_{CMOS}$ 로 결정되는 간격의 격자점으로 표현된다. 또한, 두 셀이 서로 상대의 연결 영역 안에 존재하는 한 나노와이어-나노소자-나노와이어 연결 구조의 유일한 경로를 통해 한 셀의 출력이 다른 셀의 입력으로 연결될 수 있기 때문에 별도의 배선이 필요 없다. 이 셀들 간의 대각 거리가  $2\beta F_{CMOS} \times \sqrt{2}r/2$ 보다 작다면 그들은 서로 연결 영역 안에 존재한다고 할 수 있다. 하지만 단일 연결영역을 벗어나는 원거리 연결도 필요하며 이때는 인버터 쌍이 삽입된다. 이것은 중간에 증계 게이트들을 배치하여 그들의 연결 영역이 이 원거리 연결

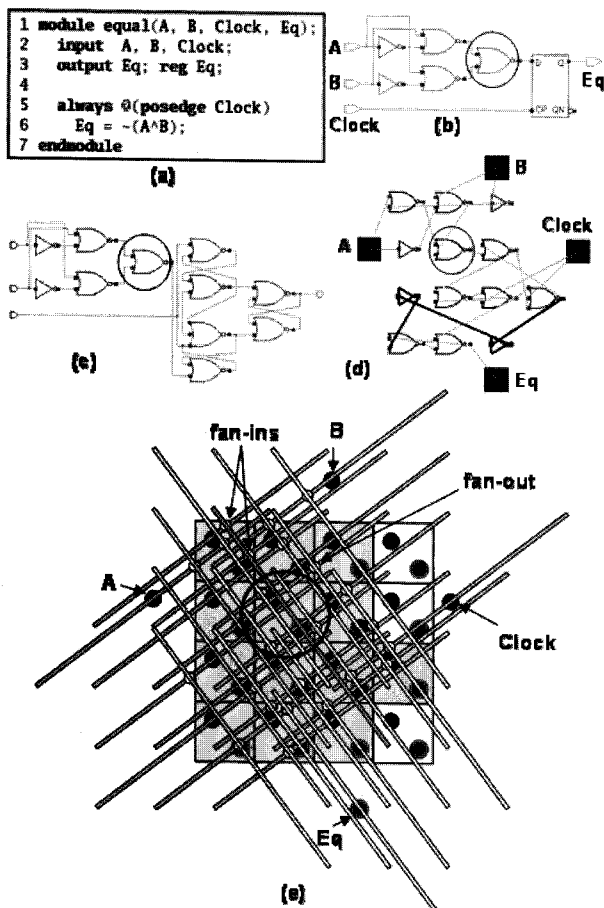


그림 4. CMOL 배열의 논리적 수준 모델과 유일성을 고려한 제약 조건들  
 Fig. 4. Logic level models of the CMOL Array and its unique constraints.

을 커버하도록 하는 것이라 할 수 있다.

만약 개략 배선기가 전면 배선 경로를 결정하면서 블록의 경계면에 외부 접속 핀들을 생성한다면 각 블록들은 상호 독립적으로 구현될 수 있다. 이렇게 구현된 블록을 간단히 조립함으로써 칩 전체 설계가 완성된다. 블록 설계이든 혹은 칩 전체 설계이든 이 외부 핀들은 내부 셀들로부터 나노와이어-나노소자-나노와이어 연결 구조를 통해 접근될 수 있어야 한다. 따라서 CMOS-나노 소자 경계에 해당하는 외부 핀도 역시 나노와이어 크로스바와 정렬되어야 한다. 그림 4(d)는 NOR 게이트로만 이루어진 1 비트 비교기를 4x4 CMOL 배열에 배치한 것이다. 그림 4(e)는 1-bit 비교기를 구현하기 위해 재구성된 4x4 CMOL 배열의 물리적 레이아웃 형태를 보여준다. 여기서 원 안의 셀은 그림 4(b-e)에서 원으로 표시한 NOR 게이트에 대응된다. 입력 측 두 개의 나노소자와 출력 측 한 개의 나노소자가 CMOS 인버터와 함께 NOR 게이트 하나를 구현

하며 주변과 연결되어 있다. 외부 입력 핀 A, B, Clock, 외부 출력 핀 Eq, 그리고 원 안에 표시된 NOR 게이트의 입출력 분기를 그림 4(b-d)의 논리 표현뿐만 아니라 그림 4(e)의 물리적 셀 배열 형태에서도 일관성 있게 표시하고 있다. 논리 레벨에서 셀 간 연결은 방사선(fly line)으로 표현하였으며 블록 간 연결을 담당하는 외부 입출력 핀들은 셀 배열의 외부에 짙은 색 상자로 표현하였다. 만약  $r = 4$  라면  $2\beta F_{CMOS} \times 4\sqrt{2}/2 = 2.82$ 이며 앞서 언급한 바와 같이 이 길이보다 긴 거리를 가진 셀 간 연결은 그림 4(d)에 굵은 선으로 표시한 것과 같이 인버터 쌍으로 증계해야 한다. 그림 4의 모든 도면은 본 논문에서 제안하는 CMOL FPGA 전용 설계 자동화 시스템에서 자동 생성된 것이다.

CMOL 아키텍처에서는 하부에 위치한 CMOS 회로층 전면에 분포된 나노와이어 크로스바와 나노소자들을 논리 게이트 구현 및 상호 연결에 동시에 사용하게 된다. CMOL 셀들을 배치하면 연결 영역 내에 존재하여 연결이 가능한 셀들은 그들까지 유일한 연결 경로가 스스로 결정된다. 하지만 연결 영역을 벗어나는 셀들의 연결을 증계하기 위해 인버터 쌍들이 필요하다. 이와 같은 원거리 연결로 인해 배치되어 연결되어야 할 추가적인 게이트 수가 상당히 늘어나게 된다. 각 인버터 쌍은 두 개의 기능적 게이트 셀을 소모한다. 사실상 25-30% 정도의 셀들이 이러한 원거리 연결을 증계하는데 사용된다. 이와 같은 문제는 별도의 배선 영역에 존재하는 멀티플렉서와 배선을 사용하여 논리 게이트와 플립플롭을 연결하는 전통적인 CMOS FPGA 기법과는 대조적이다.

CMOL 구조에서 인버터 쌍을 삽입해야 하는 원거리 연결은 모든 셀들의 배치가 완료된 후에나 확인된다. 이것은 논리 게이트들 및 이들 간 연결이 동일한 자원을 사용하기 때문이다. 반대로 전통적인 CMOS FPGA 배치 배선 문제에서는 배치 영역이 고갈되면 배선 영역

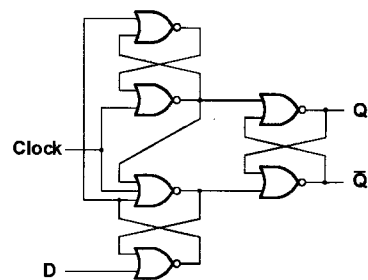


그림 5. NOR 게이트로만 구성된 플립플롭  
 Fig. 5. A CMOL flip-flop implemented using NOR gates.

이 남았다 하더라도 달리 쓰지 못하기 때문에 낭비되는 배선 영역은 무시된다. 반대로 배선 영역이 고갈되면 배치 영역이 무시된다. 이러한 차이는 CMOL 구조에서 배치와 배선 문제의 복잡도를 증가시키는 요인이 된다.

순차회로를 구현하기 위해서 플립플롭이 CMOL 셀 배열 상에 구현되어야 한다. 이러한 요구를 만족시키기 위해 Strukov와 Likharev<sup>[4]</sup>는 타일 배열을 제안하였다. 각 타일의 중앙에 CMOS로 구현된 플립플롭 하나를 포함한다. 이 플립플롭은 NOR 셀보다 네 배 크기이며 다른 12 개의 NOR 셀들로 둘러싸여 있다. 플립플롭들을 중심으로 회로를 분할하여 생성된 작은 모듈들을 각 타일에 배치하자는 것이다. 이와 같이 이중의 기능 셀들을 조합하는 것은 분할된 각 모듈이 요구하는 셀 수 및 종류와 배열 상에서 이들이 배치된 타일에서 공급되는 셀 수 및 종류의 불균형으로 인해 칩 사용률을 떨어뜨릴 수 있다. 본 논문에서는 그림 5와 같이 CMOL의 기본 NOR 게이트 6개를 사용하여 플립플롭을 구현하는 방안을 제안한다.

이 플립플롭은 일반적으로 널리 알려진 회로이나 면적 효율이 높은 고집적 CMOS 플립플롭이 사용되어 왔기 때문에 교육적 목적 이외에는 실제로 사용되지 않았다. 하지만 CMOL 아키텍처에서는 NOR 게이트의 입력을 형성하는 나노 소자가 풍부하기 때문에 인버터 구현 비용으로 NOR 게이트를 구현할 수 있으며 크로스바를 이용한 국부 배선도 별도의 소모비용이 들지 않는다. 비록 이러한 모델은 Strukov와 Likharev<sup>[4]</sup>의 방식과 비교하면 플립플롭 당 2개의 셀이 더 사용되나 이 플립플롭은 어느 위치에도 배치 가능하며 활용도는 항상 100%를 유지할 수 있기 때문에 자동화나 칩 활용도 측면에서 훨씬 유리하다. 또한, Strukov와 Likharev<sup>[4]</sup>의 플립플롭의 경우 클럭 신호를 연결하기 위한 핀이 존재하지 않으나 제안된 방법으로는 클럭은 물론 별도 비용 없이 간단히 NOR 게이트의 입력을 추가하면 비동기 셋 /리셋 기능도 추가할 수 있다.

CMOL 구조는 NOR 게이트들 간에 독특한 인접 관계를 부여한다. 한 게이트의 출력에 연결된 게이트들은 모두 그 게이트에 인접해야 하나 출력에 연결된 게이트들끼리는 인접할 필요가 없다.

### III. CMOL 설계 자동화

CMOL FPGA 설계 흐름도를 그림 6에 나타내었다. 논리 합성은 시놉시스 사의 Design Compiler를 사용하

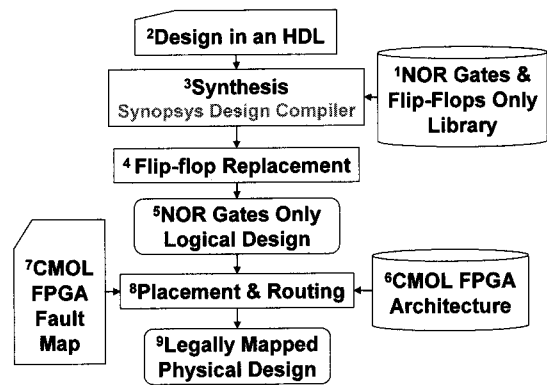


그림 6. CMOL FPGA 설계 흐름도  
Fig. 6. The CMOL FPGA design flow

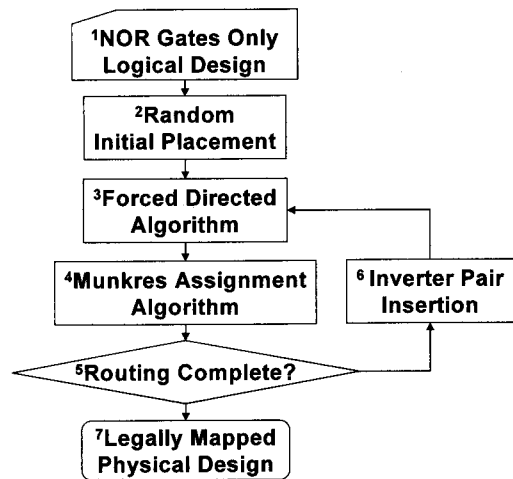


그림 7. CMOL FPGA 설계 전용 배치 배선 알고리즘  
Fig. 7. The algorithm for the placement and routing of the CMOL FPGA design.

였으며 레이아웃 설계를 위해서 CMOL FPGA 전용 배치 배선 소프트웨어를 개발하여 사용하고 있다. 먼저 NOR 게이트들 및 플립플롭들로만 구성된 라이브러리를 사용하면 하드웨어 기술 언어 (HDL)로 기술된 레지스터 전달 수준 (RTL) 설계로부터 게이트 수준의 네트리스트를 합성해 낼 수 있다. 이 때, 플립플롭들은 II장에서 제안한 바와 같이 NOR 게이트로 구성한 회로로 대체되어 레이아웃 설계 단계로 들어간다. 기본적인 CMOL 아키텍처를 그대로 이용하기 때문에 배열 구조의 규격 및 입출력 포트 위치만 지정하면 레이아웃을 진행할 수 있다. 추후, 대규모 회로 구현을 위해서 배치 계획 단계를 추가할 수 있으며 칩 영역을 분할하여 각 논리 블록을 이에 할당하는 형태로 작업이 단순해진다. 또한, 제조된 칩 상에 결함이 존재할 경우 레이아웃 설계에서 이를 제외한 부분만 사용하도록 하는 결함 내성 기능을 위해 결함 소자 정보가 추후 필요할 것이다.

CMOL FPGA 전용 배치 배선 소프트웨어의 수행 호

름도를 그림 7에 나타내었다. 먼저, NOR 게이트만으로 구성된 네트리스트를 입력으로 무작위 초기 배치를 한 후 후크의 법칙을 응용한 힘 지향 (Force-Directed, FD) 알고리즘<sup>[5]</sup>을 수행하면 게이트들 간의 최적 상대 위치를 얻을 수 있다. 이어, Munkres 지정 알고리즘<sup>[6]</sup>을 수행하면 이 상대 위치를 기준으로 가장 가까운 거리에 있는 셀 격자 위치를 배타적으로 확정한다. 계산량의 상한은 힘 지향 알고리즘이  $O(n^2)$ 이며 Munkres 알고리즘은  $O(n^3)$ 이다. 앞서 언급한 바와 같이 하나의 게이트 출력을 다른 게이트의 입력으로 연결하기 위해서는 크로스바를 경유해야 하는데 나노 와이어의 길이가 한정되어 있기 때문에 멀리 떨어진 셀 간에는 연결이 안 될 수 있다. 연결 영역에서 벗어나는 원거리 연결은 인버터 쌍으로 중계해야 하며 논리 구현에서 사용되는 게이트 자원을 같이 사용해야 한다. 먼저 연결영역을 벗어나는 길이의 연결을 찾아내고 해당 네트에 두 개의 인버터를 직렬로 삽입하는데 그 위치는 두 연결점의 대각선을 3등분하여 1/3 지점과 2/3 지점으로 결정한다. 대개의 경우 인버터가 놓일 위치를 다른 게이트가 이미 차지하고 있기 때문에 힘 지향 알고리즘과 Munkres 알고리즘을 재수행하여 배타적인 셀 배치가 되도록 해야 한다. 이 과정에서 연결 영역을 벗어나는 배치가 발생할 수 있으며 배선이 완료될 때까지 루프를 반복 수행해야 한다. 원거리 연결을 위한 인버터 쌍의 삽입은 셀 사용을 증가시키기 때문에 셀 활용도가 100%를 초과하여 자동 레이아웃이 실패할 수 있으며 이 경우 셀 격자 배열을 증가시켜야 한다.

1. CMOL 레이아웃에서 게이트 배치

한 게이트의 출력 분기들은 각각 그 게이트 셀과는 인접성이 존재하지만 그들 상호간에는 인접성이 없기 때문에 FD 알고리즘 입력 네트리스트의 인접 행렬 (adjacency matrix)는 수정되어야만 한다. 원거리 연결을 담당하는 인버터 쌍의 삽입 이후의 FD 알고리즘의 수행 결과는 삽입 전과 많은 차이가 없어야만 하지만 불행히도 원본 FD 알고리즘의 반복적인 실행은 그림 8 (a)에서 굵은 선으로 나타낸 바와 같이 입력 및 출력 분기를 각각 한 개씩 보유한 인버터 쌍의 게이트 셀들을 밖으로 밀어 낸다. 원인을 파악하기 위해서 먼저 FD 알고리즘에서 힘을 계산하는 수식을 살펴보자.  $\Delta x_{ij} = x_j - x_i$ ,  $\Delta y_{ij} = y_j - y_i$  이고  $\Delta S_{ij} = |\Delta x_{ij}| + |\Delta y_{ij}|$ 이며 만약 M개의 게이트 셀이 존재한다면 x와 y 방향으로의 힘은 다음과 같이 수식 (1, 2)와 같이 계산된다.

$$F_x = \sum_{j=1}^M \left( -K_{ij} \Delta x_{ij} + R_{ij} \frac{\Delta x_{ij}}{\Delta S_{ij}} \right) \tag{1}$$

$$F_y = \sum_{j=1}^M \left( -K_{ij} \Delta y_{ij} + R_{ij} \frac{\Delta y_{ij}}{\Delta S_{ij}} \right) \tag{2}$$

여기서  $i = 1, 2, \dots, M$  이다. 괄호안의 첫 번째 항이 후크의 법칙에 의한 인력을 의미한다면 두 번째 항은 척력을 의미한다.  $K_{ij}$ 는 i 번째에서 j 번째까지의 연결된 게이트 수와 비례하는 상수며  $R_{ij}$ 는 연결된 게이트들 간 인력의 누적 값이 연결되지 않은 게이트들 사이에서 발생하는 척력의 누적 값과 같도록 결정되는 상수를 나타낸다. 척력의 궁극적인 목적은 게이트의 과잉 밀집을 이완하기 위한 목적으로 사용되기 때문에 그 힘은 밀집된 연결도를 보이는 지역에서 반대의 지역 쪽으로 방향성을 가지고 있다. 수식 (1, 2)를 통해 우리는 척력이 게이트간 거리에 독립적인 반면 셀들의 밀집도에 의존적인 것을 확인할 수 있다.

그림 8(b)에서 좌표  $(x_1, y_1)$  지점 게이트 셀이  $x(y)$ 의 양의 방향으로 작용하고 있는 척력은  $(x_1, y_1)$  지점을 기준으로 음의  $x(y)$ 방향 지역의 게이트 셀 수와  $x(y)$ 의 양의 방향 지역의 게이트 셀 수와의 비율에 따라 결정된다. 하지만 외곽지역에 비해 상대적으로 면적이 넓은 중앙 측은 높은 셀 밀도를 갖게 되며 이로 인해 과대평

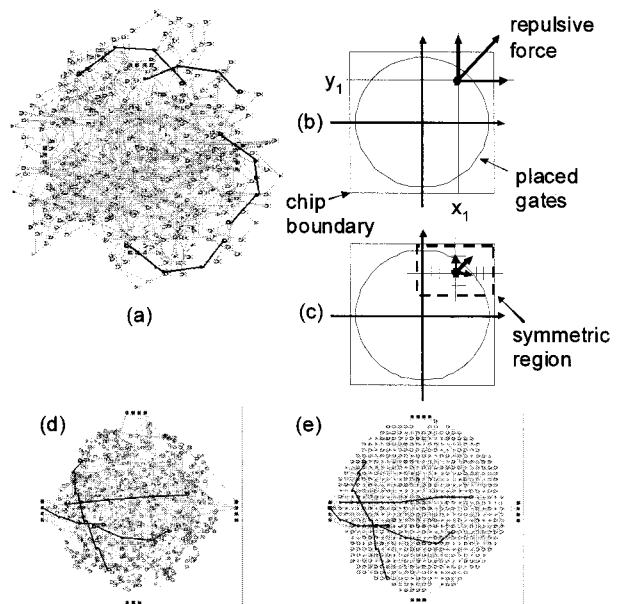


그림 8. 척력의 균형적 계산을 위한 대칭 영역  
Fig. 8. Symmetric region to fairly calculate repulsive force.

가된 척력은 칩 외곽에 있는 게이트 셀을 더욱 밖으로 밀어내는 역할을 하는 것이다. 중앙과 외곽 측 면적의 불균등으로 인해 발생하는 이런 문제를 극복하기 위해서는 각 셀의 척력 계산 시 오직 대칭 사각 영역 안쪽 게이트들만을 고려하는 방식 (그림 8(c))로 수식 1, 2를 수정해야 한다. 이를 위해서 먼저 칩 영역의 네 개의 모서리 중 해당 게이트와 가장 가까운 것을 선택하여 대칭 사각형의 한 쪽 모서리를 결정하고 이 모서리와 게이트 셀과의 동일 거리만큼 대각 방향으로 확장하여 대칭 사각형의 반대편 모서리를 결정한다. 제안된 기법이 적용한 수정된 FD 알고리즘을 이용하여 발생한 배치 결과를 그림 8(d)에 나타내었으며 인버터 쌍 네 개의 연결을 굵은 선으로 표시하였다. 이 셀들이 더 이상 외부로 밀리지 않은 것을 확인 할 수 있다. 또한 Munkres 알고리즘 수행 이후의 배치 결과도 그림 8(e)에 나타내었다. 그 결과를 그림 8(d)와 비교하면 위치 지정 시 셀들의 이동도 매우 적음을 확인 할 수 있다.

## 2. CMOL에서 게이트 지정

본 논문에서는 겹치지 않으면서 총 이동 거리를 최소화 하도록 각 게이트 셀들을 셀 격자점에 지정하는 문제를 해결하기 위해 Munkres 지정 알고리즘<sup>[6]</sup>을 사용하였다. 불행하게도 이 알고리즘의 복잡도는  $O(n^3)$ 이다. CMOL 배치와 배선의 주된 목적이 배선 길이의 최소화가 아니라 배선 작업의 빠른 완료이기 때문에 연결 영

역 안에서는 게이트들의 자유로운 이동이 허용될 수 있으며 이 여유 공간 (slack)은 지정 문제를 분할하여 푸는 것을 가능하게 한다. 회로는 네 개의 모듈로 분할된다. 먼저 게이트들을 그들의 y 좌표를 기준으로 정렬하고 동일한 수의 상하 두 개의 그룹으로 분리한다. 그리고 또 다시 각 그룹을 내부 게이트 셀들의 x 좌표를 기준으로 정렬하고 좌우 두 개의 그룹으로 분리하는 것이다. 또한 CMOL의 셀 배열 역시 사분 구역 (quadrisection)으로 분할한다. 이후 각 게이트 그룹은 하나씩 대응되는 사분 구역에서 Munkres 지정 알고리즘을 통하여 위치가 지정되는 것이다. 즉, 문제를 네 부분으로 쪼개어 조각들을 순차적으로 푸는 것이다. 이러한 지정 과정은 게이트 간 연결도 검사나 좌표의 변환을 요구하지 않는다는 것이 특징이다. 유일한 제한 사항은 어느 한 그룹 안에 소속된 게이트가 다른 그룹의 구역으로 이동할 수 없다는 것이다. 하지만 수정된 힘 지향 알고리즘이 척력과 무게 중심을 정확히 계산해 내고 있어 적절히 분산되고 치우침이 없는 분포를 생성해 내기 때문에 이러한 이동이 거의 필요 없게 된다. 만약 한 그룹 내 게이트 수가 상수 값에 도달할 때까지 반복 분할한다면 이론적으로 복잡도는  $O(n^2 \log n)$ 로 감소된다.

## III. 실험 및 고찰

제안된 CMOL FPGA 자동 배치 배선 소프트웨어의

표 1. CMOL 배치 및 배선 결과

Table 1. Experimental results on placement and routing on CMOL.

Module	Array Size	Gate Count		Execution Time (sec)								
				Init		FD		Munkres		Total		Speed Up
		Single	Quad	S	Q	S	Q	S	Q	S	Q	
CONTROL	9x9	77+0 (95.0%)		0.11		0.22		0.23		0.56 (1)		-
FSM	13x13	169+0 (63.3%)	169+0 (63.3%)	0.77	0.94	0.77	0.11	0.13	0.33	1.38 (1)	1.67 (1)	0.8x
PRGRM	18x18	231+24 (71.3% +7.4%)	231+16 (71.3% +4.9%)	0.12	0.12	3.45	1.03	36.47	2.5	40.04 (3)	3.65 (3)	11x
MEM	31x31	488+158 (50.8% +16.4%)	488+140 (50.8% +14.6%)	0.17	0.15	9.47	7.36	990.05	15.3	999.69 (5)	22.81 (8)	44x
DATA	30x30	588+170 (65.3% +18.9%)	588+154 (65.3% +17.1%)	0.19	0.19	14.65	11.94	1247.92	14.93	1262.76 (7)	27.06 (6)	47x
INSTRN	34x34	684+244 (59.2% +21.1%)	684+270 (59.2% +23.3%)	0.19	0.19	18.64	12.99	2608.58	33.66	2627.41 (8)	46.84 (5)	56x
REG_FILE	48x48	944+474 (41.0% +20.6%)	944+528 (41% +22.9%)	0.17	0.17	37.59	38.39	18414.27	443.99	18452.03 (8)	482.55 (9)	38x

성능을 확인하기 위하여 RTL HDL로 기술된 마이크로 프로세서의 내부 모듈들 중 다양한 규모의 예제를 선택하여 CMOL FPGA로 구현해 보았다. 이 예제는 프로그램 계수기 (PRGRM), 메모리 컨트롤러 (MEM), 데이터 패스 (DATA)를 포함하며 표 1에 자동 배치 배선 결과를 정리하였다.

첫 번째 열은 실험에 사용된 모듈들이고 다음 열은 2차원 CMOL 배열의 크기를 나타낸다. 이때  $r$ 값은 16으로 하였으며 따라서 최대 대각 거리는  $16/\sqrt{2} = 11.3$ 보다 짧게 된다. 사분할 지정 기법 (Q)의 실험 결과는 셀의 수와 수행시간에 대해 사분할을 하지 않은 지정 기법 (S)과 각각 비교하였다. 괄호 안의 숫자는 루프의 반복 수행 횟수이며 CONTROL 모듈은 작아 사분할을 하지 않았다. 여기서 게이트의 수는 최초 네트리스트를 구성하는 게이트의 수와 원거리 연결을 위해 추가된 인버터의 수 형태로 나타내었다. 회로의 규모가 증가할수록 인버터의 비중이 증가함을 알 수 있다. 실행시간은 최초 무작위 배치부터 마지막 지정 기법 적용까지 누적하여 측정하였으며 루프 반복 수행 회수는 괄호 안에 기록하였다. 제안된 사분할 지정 기법의 성능이 사분할하지 않은 지정 기법에 비해 평균 40배가 향상됨을 확인할 수 있다.

그림 9는 PRGRM 모듈의 (a) 힘 지향 알고리즘 적용 후 및 (b) Munkres 알고리즘 적용 후 배치 결과를 보이고 있으며 두 알고리즘 간의 높은 상관관계를 나타내고 있다. 또한, 다수의 인버터가 원거리 연결의 중계를 위해 적절히 삽입되어 있음을 알 수 있다. 앞서 설명한 바와 같이 플립플롭은 NOR 게이트의 조합으로 구현되어 있다.

그림 10은 PRGRM의 배선 길이 당 네트 도수 분포를 각 수행 단계에 따라 나타낸 것이다. 배선 길이가 1 구간 ( $0 \sim 1 * 2\beta_{CMOS}$ )인 네트의 수부터 배선 길이가 23 구간 ( $22 * 2\beta_{CMOS} \sim 23 * 2\beta_{CMOS}$ )인 네트의 수

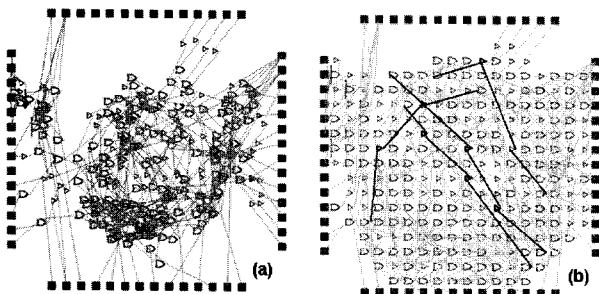


그림 9. 모듈 PRGRM의 배치 결과  
Fig. 9. Placement results of PRGRM.

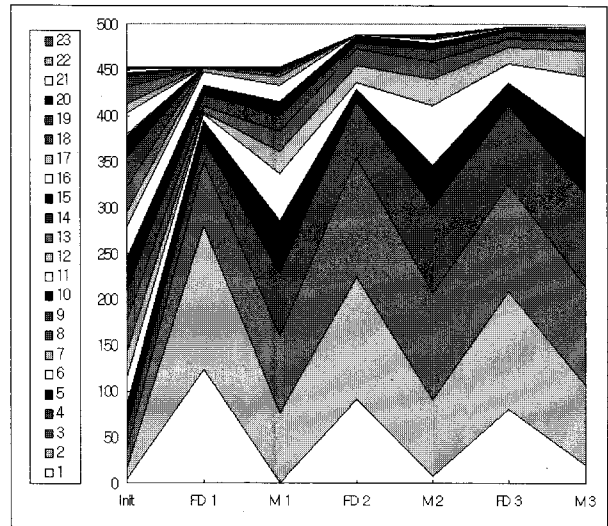


그림 10. 모듈 PRGRM의 배선 길이 당 네트 도수  
Fig. 10. Accumulated number of nets for each distance interval at each step.

표 2. 기존 방식과 면적 효율 비교  
Table 2. Area efficiency comparison with previous work.

회로	Strukov & Likharev <sup>[4]</sup>		본 논문	면적비
	tile 수	cell 수	cell 수	
ex5p	20x20	320x320	35x35	83:1
apex2	21x21	336x336	55x55	37:1
apex4	18x18	288x288	50x50	33:1

를 아래에서 위로 차례로 누적시켜 보이고 있다. 힘 지향 알고리즘 (FD) 적용 후 대체적으로 연결 거리가 짧아졌다가 Munkres 알고리즘 (M) 적용 후 다시 길어지는데 이것은 FD가 격자 간격 이내에 여러 셀의 배치를 허용하기 때문이다. 전체적으로 살펴보면 인버터 쌍의 삽입으로 네트의 총수는 약간 늘어나며 루프의 반복 수행에 따라 배선 길이가 12 구간 이상인 네트의 수가 상대적으로 감소하고 결국 모두 0이 되어 배선이 100% 달성함을 알 수 있다.

앞서 설명한 바와 같이 Strukov와 Likharev<sup>[4]</sup>는 순차 회로 구현을 위해 플립플롭을 내장시킨 타일 ( $4 \times 4 = 16$  개 셀 크기)의 배열을 사용하였는데 본 논문에서는 6개의 셀을 조합하여 플립플롭을 구현하는 방식을 제안하였다. 이 두 방법의 면적 효율을 비교하기 위해 표 2에 벤치마크 회로 3개에 대한 결과를 나타내었다. 조합 회로 및 순차 회로, 그리고 연결을 위해 동일한 자원을 사용하는 균질의 배열 구조가 수십 배의 면적 효율을 가져옴을 알 수 있다.



#### IV. 결 론

본 논문을 통해 CMOL FPGA의 설계 자동화 시스템을 개발하고 검증하였다. 이 시스템은 먼저 레지스터 전달 수준의 하드웨어 기술 설계를 입력으로 NOR 게이트만의 로직을 생성하고 레이아웃 생성을 위해 이 NOR 게이트 로직은 향상된 힘 지향 알고리즘 및 사분할 Munkres 지정 알고리즘을 두 단계로 레이아웃을 생성하게 된다. 개발된 CMOL 나노전자 아키텍처를 위한 설계 자동화 시스템은 본 논문에서 사용한 7가지 회로에 대해 좋은 수렴 특성을 확인하였으며 차후 CMOL 구조에 대한 결함 내성 논리 합성 및 물리적 합성 등의 검증을 포함한 다양한 설계 방식에 응용될 수 있다.

#### 참 고 문 헌

- [1] International Technology Roadmap for Semiconductors. 2003 Edition, 2004 Update. Available online at <http://public.itrs.net/>.
- [2] K. K. Likharev and D. B. Strukov, "CMOL: Devices, circuits, and architectures," *Introducing Molecular Electronics*, Chapter 16, Springer, Berlin, 2005.
- [3] Dmitri B Strukov and Konstantin K Likharev, "CMOL FPGA: a reconfigurable architecture for hybrid digital circuits with two-terminal nano devices," *Nanotechnology*, vol. 16, no. 6, pp.888-900, Institute of Physics Publishing, June 2005.
- [4] Dmitri B. Strukov, Konstantin K. Likharev, "A Reconfigurable Architecture for Hybrid CMOS/Nano device Circuits," *Proceedings of the International Symposium on Field Programmable Gate Arrays*, pp.131-140, 2006.
- [5] Neil R. Quinn, Jr., and Melvin A. Breuer, "A Forced Directed Component Placement Procedure for Printed Circuit Boards," *IEEE Transactions on Circuits and Systems*, vol. CAS-26, no. 6, June 1979.
- [6] Francois Bourgeois and Jean-Claude Lassale, "An Extension of the Munkres Algorithm for the Assignment Problems Rectangular Matrices," *Communications of the ACM*, vol.14, pp.802-804, December 1971.

#### 저 자 소 개



김 교 선(정회원)

1986년 연세대학교 전자공학과  
학사 졸업.

1988년 연세대학교 전자공학과  
석사 졸업.

1998년 Ph.D. Department of  
Electrical & Computer  
Engineering, University  
of Massachusetts, Amherst,  
U.S.A.

1988년~2003년 삼성전자 CAE Center 주임,  
선임, 책임, 수석연구원.

2007년 현재 인천대학교 공과대학 전자공학과  
부교수

<주관심분야 : 상위수준합성, Reconfigurable  
Computation, Fault-Tolerance, Network  
Embedded Systems, Low-Power Design,  
Nanoelectronic Architectures>