

## Module Generator

鄭淵激

現代電子産業(株) 半導體研究所

Application specific integrated circuit (ASIC) 기술 발전에 중요한 기본개념은 미리 설계 되어진 기본 회로들에 대한 모임과 그들의 관계 특성을 미리 확인하여 모아둔 이론(library 개념)을 도입함에 있다고 하여도 과언은 아닌 듯 싶다.

지나간 20여년 간에 걸친 ASIC, 특히 gate array 또는 standard cell에 대한 실제 자동화 과정은 library 개념 도입에 의하여 그 발전속도가 가속화 되지 않았다 하고 생각된다. 이러한 library 개념에 대한 계속적인 ASIC 설계 기술발전과 그 사용자들에 의한 요구사항에 의하여 최근 수년동안 급진적으로 발전한, 설계 자동화 동향은 “한 특수기능을 행할 수 있는 어떤 회로의 집합(module)을 자동적으로 그 기능에 적합한 기본 논리회로들을 추출하고, 논리회로의 상호간에 연결을 하여 최종적으로 전 module에 대한 layout을 제공하는 software(module generator) 개발을 지나간 수년동안 연구 개발하여 오고 있다”라고 말할 수 있다.

### I. 서 론

설계 자동화에 대한 언어로서 “silicon compilation”라는 용어는 미국 Caltech 대학에 있는 Dave Johannsen에 의하여 처음으로 1979년도에 쓰여졌다.<sup>1)</sup> 상기 용어에 대한 그 당시의 의도는 하나의 layout module을 어떻게 부분화하여, 각 부분을 변수화할 수 없을 까하는 개념을 표현하는 언어로서 발표되어졌다. 그러나 오늘날 그 용어에 대하여서는 쓰이는 곳에 따라 그 내용이 다르며, 사용시에 그 주위 환경에 대한 정의를 정확히 하지 않으면 진정한 뜻의 내용을 잃어 버리기 쉽지만 일반적으로 그 의미는 2 가지로

분류할 수 있다.

하나는 좁은 뜻에서의 silicon compilation, 다른 하나는 넓은 뜻에서의 silicon compilation이다. 좁은 뜻에서의 silicon compilation은 간단한 논리 회로를 논리 및 전기적 특성을 결정하고 그에 대한 mask layout을 수행하는 것을 말하며, 더 나아가서는 종전에 transistor transistor logic (TTL)에 small size integration (SSI) IC기능을 제공하는 것을 말한다. 한편 넓은 뜻에서의 silicon compilation은 high-level description<sup>2)</sup>을 번역하여 그에 해당하는 mask layout을 제공하는 것을 의미한다. 여기서 말하는 high-level description은 사용자에게는 자세히 설명치 않는, 상당히 개념적으로 표현화된 논리회로의 기능을 말한다.

지금 여기에서 논의하고자 하는 module generator에 대한 개념도, 많은 사람들은 module compiler라고 한다. Module generator는 상기에 표현한 silicon compiler에 대한 부분집합이라고 생각하면 거의 잘못된 것이 아닐 것이다.

### II. 본 론

Module generator는 그림 1에서 보여주는 4개의 기능별 분류 flowchart를 형성한다. 여기서 “사용자의 module specification”이란, 생성하고자 하는 module의 기능 명세서이며, 그 예로써, PLA module을 생각할 경우, module의 기능 명세서는 AND-OR plane의 truth table이 될 것이다. 이 truth table은 그림 2에서 보여주고 있다. 여기에서 “-”표시는 논리에 “Don't care” 항목이 될 것이다. 이 truth table로부터 우리는 바로 입력단이 몇이 되고 출력단이 몇이 될 것인지 예측할 수 있을 것이다.

다음의 “Module Compiler” 항목은 입력으로 들어온

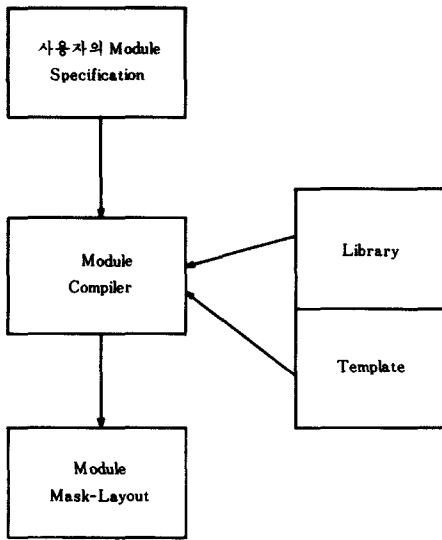


그림 1. Module generator

```

----1--10- 0000000000001000000
----1010- 0010110101000101110
----11-01- 0000000000001000000
00--0001- 0000110101011000000
-00--0001- 0010000101011000000
-011--01- 0000000000010000000
1100-0001- 0000110101010000000
0110-0001- 0010110111011000000
1001-0001- 0000111010000000000
1010-0001- 0010011111011000010
0101-0001- 0011110111011000000
  
```

그림 2. PLA의 truth table

AND-OR plane의 truth table을 받아 각각 AND와 OR에 들어갈 논리식을 최소화 한 후, 사용자에게 의하여 선택된 architecture에 맞는 template를 추출한다.

그러면 먼저 PLA 회로에 기본 회로에 대한 개념을 이해하고 다시 각 기능별 내용을 논의하자. 일반적인 PLA의 architecture는 그림 3에서 보는 바와 같이, 왼쪽의 아래에서 입력단이 들어와 그것을 AND plane에서 그 중간 결과를 결정하여 그림 오른쪽의 OR plane으로 보내 다시 최종적인 결정을 하여 그림 오른쪽 하단에 있는 출력단으로 내 보내는 것이다. 여기서 만약 사용자가 설계하고자 하는 어떤 회로에 지금 만들어 낼 PLA architecture가 서로 잘

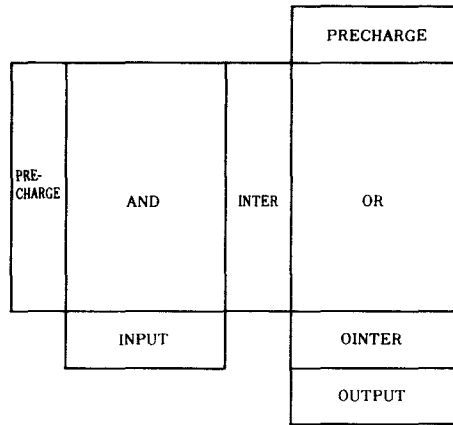


그림 3. 일반적 PLA에 architecture

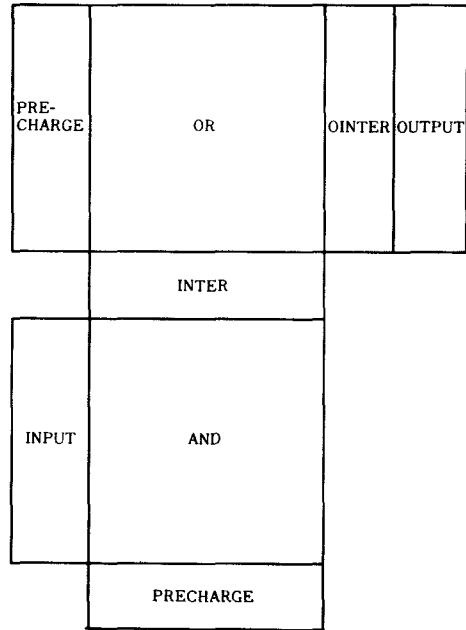


그림 4. 다른 모양에 architecture를 가진 PLA

맞지 않아 (mask상에서의 pitchmatch라든가 입력 및 출력에 위치 등등) 다른 architecture를 생각할 수도 있다. 그에 대한 조건은 많으나 여기서는 일단 그림 4에서와 같이 수직 방향으로 배열된 PLA 형태를 생각할 수 있다.

이와 같이 PLA의 형태를 사용자들이 일반적으로 어떠한 형태를 필요로 하는지를 파악하여 그에 따라

미리 필요한 PLA 형태를 선정할 수가 있다. 따라서 사용자는 자기가 설계할 회로에 맞는 PLA 형태를 선정하여, 그에 해당하는 template을 선정하도록 module compiler에게 명하게 된다. 이와같이 template가 결정되면 그 template에 알맞는 기본 회로들이 library로부터 추출되어 하나의 PLA module을 형성시킨다.

여기서 이야기하는 library 내용은 다음과 같이 설명할 수 있다. 그림 3에서 보여주는 input단에 들어가는 회로는 input buffer가 될 것이고 AND plane의 왼쪽에 있는 precharge 회로는 CMOS 기술에 의한 회로일 때는 특정한 크기를 가진 p-channel MOS transistor가 되고, AND plane 자체는 논리식에 의거하여 n-channel MOS transistor가 존재(T cell)하거나 또는 존재하지 않은(N cell) 것으로 만들 수 있고, 그림 3 중앙에 있는 inter는 AND와 OR plane의 interface 회로이며 그 구성은 1개의 3-state-inverter 또는 1개의 pass-transistor로 구성시킬 수 있다.

OR plane에서의 "Precharge" 회로는 AND plane과 동일한 것이며, OR plane 자체도 n-channel MOS transistor의 존재 여부에(N cell 또는 T cell)의하여 만들어질 수 있다. OR plane의 "OINTER"는 "INTER"에서 사용한 회로를 동일하게 이용할 수 있다. 마지막으로 "OUTPUT" 회로는 간단한 inverter로 만들 수 있다.

상기에서 나열된 각각에 기본 회로의 layout은 그림 5에서 보여주고 있다. 즉 예를들어 10개의 입력과 10개의 출력을 가진 PLA를 만들때나, 20개의 입력과 50개의 출력에 PLA를 만드는 경우에, 필요한 기본회로는 그림 5에서 주어진 것 이외에 더 필요한 것이 없다.

먼저 주어진 template을 가지고 10개의 input buffer를 차례로 옆으로 붙여 나가면 10개의 입력을 형성하고, 같은 방법으로 다른 부분을 형성하여 그들을 모아 함께 붙이면 그에 해당하는 PLA을 만들 수 있다.

그림 6은 하나의 형성된 PLA을 보여주고 있다. 이 그림에서 볼 수 있는 바와 같이 결국 입, 출력단자와 precharge 부분등은 전부 간단히 나열하는 문제이고, AND나 OR plane 내부는 주어진 truth table로부터 "1" 또는 "0"에 의거하여 transistor을 넣을 것인가 또는 transistor을 넣지 않을 것인가의 결정이 될 것이다. 이 문제도 우리가 좀 더 깊게 생각하여 AND 및 OR에 해당하는 논리식을 최소화하여, 그에 해당하는 truth table로부터 상기 내용을 다시

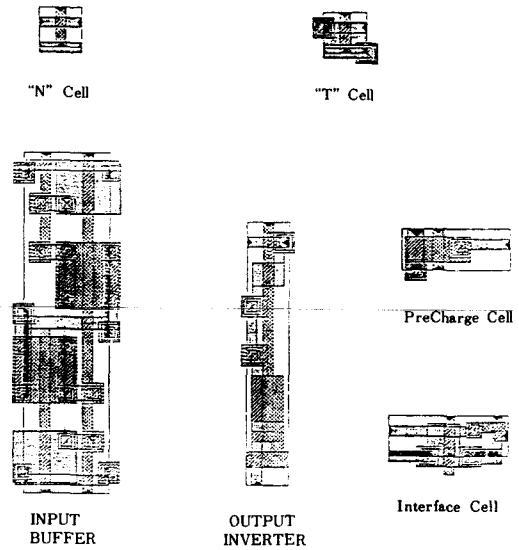


그림 5. PLA 기본 회로들

할 수 있는 방법론을 제시할 수가 있다.

위에서 언급한 바와 같이, 최소한의 library를 가지고 사용자에게 필요한 크기와, 필요한 형태의 PLA을 제공할 수 있는 간단한 software는 상당히 쉽게 만들어질 수 있으리라 생각된다. 그러나 여기서 우리가 조금 더 깊게 생각해야 할 것은 어떠한 회로 언어<sup>4)</sup>를 이용할 것인가 또한 그 회로 언어를 어떻게 software가 이해하도록 해야 할 것인가 등의 구체적인 방법 설정에 있다.

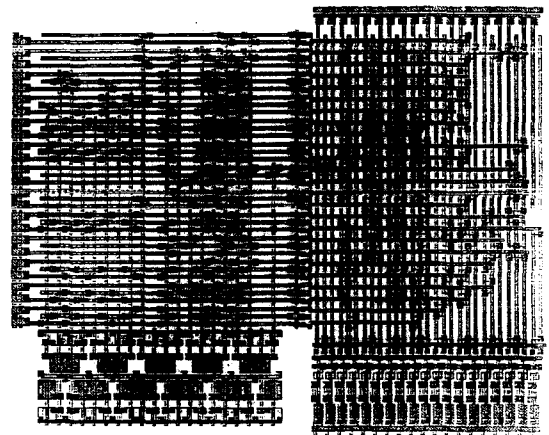


그림 6. PLA Generator에 의한 결과 예제

### Ⅲ. 결 론

본론에서 설명된 내용은 PLA module generator를 예를 들어 논의하였지만, 일반적인 module generator의 방향은 현시점까지는 주로 규칙성이 많은 module에 대하여만 개발되어 왔다. 그 예가 ROM, SRAM, PLA, 특수 data path 등이다. 그러나 앞으로는 규칙성이 없는 회로에도 적용되는 module generator가 발전되어야 한다고 생각된다.

또한 특히 중요한 내용인 각각의 기본회로 선택에 대하여는 본론에서는 별도로 강조하지 않았지만, 실제로는 특히 신중히 고려해야 할 점이라고 생각된다. 본 내용에서 예를 든 PLA만 하여도 여기에서 선택한 간단한 기본회로 이외의 다른 회로 구성이 가능하므로, 그에 대한 hardware도 신중히 고려되어야 한다고 생각되어진다.

마지막으로 한가지 더 고려 대상이 되는 것은, 형성하려는 module이 몇개의 module로 되어질 경우 module과 module을 간단하게 서로 접하게하여 module을 형성하는 방법이 있으나, 이는 실제적으로 상당히 어려운 문제를 야기시킨다. 간단한 예로, 서로 접하게 되는 부분에서 근접되는 module간에 정확한

pitch가 맞아떨어질 뿐만 아니라, 서로 교환할 신호 node도 정확한 위치에서 고려되어야 하므로 아주 정밀하게 layout을 해 주어야 하나, 만약에 개발된 system에 module과 module사이를 routing software를 이용하여 routing을 하여 주는 것도 생각하여 볼 수도 있다.

지금까지 논의된 내용을 살펴보면 module generator를 개발, 발전시키는 방법은 hardware 뿐만 아니라 software도 같이 동행하여 발전되어야 하며, 그 발전 방향은 되도록 사용자에게 사용하기 편하고 그들이 필요로하는 module의 다양성을 제공하는 방향으로 발전되어야 한다고 생각한다.

### 參 考 文 獻

- [1] D. Johannsen, "Bristle Blocks: a silicon compiler," Proc. 16th Design Automation Conf., June 1979.
- [2] D. Gajski and R. Kuhn, "New VLSI tools," *Computer*, vol. 16, no. 12, Dec. 1983.
- [3] I. Buchanan, "Modelling and Verification in structured IC Design," Ph.D. Thesis, University of Edinburgh, Scotland, 1980. 🌐

### 筆 者 紹 介



鄭 淵 激 (非會員)

1952年 4月 15日生

1976年 한양대학교 공과대학 졸업

1981年~1985年 Rhode Island 주립대 전자공학 석사 및 박사학위 취득

1976年~1979年 한국과학기술 연구소 반도체센터 반도체 공정 및 회로설계 담당

1979年~1981年 미국 Iowa 주립대학 Thermoluminescence 담당 연구원 근무

1982年~1984年 Rhode Island 주립대 VLSI Digital과 Analog 회로설계강의

1983年~1985年 Transcom Electronics 회사 Gate Array 회로 담당 고문

1986年~1988年 Bell Communication Research 근무

현재 현대전자 반도체 연구소 근무

관심분야 : 반도체 회로설계 및 VLSI CAD분야