

고밀도 PLA의 자동 Layout System의 구성 (Automatic Layout of High Density PLA)

李 濟 賢^{**}, 慶 宗 旻^{*}

(Je Hyun Lee and Chong Min Kyung)

要 約

고밀도 PLA(Programmable Logic Array) layout의 생성, 간소화 및 검증은 자동화한 일련의 유용 프로그램을 개발하였다. 이에는 논리 함수로부터 진리표를 만들어 내는 프로그램, 논리 간소화 프로그램, PLA 재배열 프로그램, stick diagram을 그릴 수 있는 화일을 만들어 내는 프로그램, dynamic CMOS PLA의 layout 생성 프로그램, 그리고 bipartite row folded CMOS PLA layout 생성 프로그램이 포함된다. 크기의 최소화는 주로 논리 간소화 프로그램과 bipartite row folding 프로그램에 의해 수행되며, 최대 지연시간은 재배열 프로그램에 의해 작아진다. 자동으로 생성된 layout에 대한 정보는 CIF(Caltach Intermediate Form)로 저장된다. 각 프로그램은 C언어로 작성되었으며, VAX-11/750 (UNIX)에서 수행되었다.

Abstract

A set of utility programs for automatic generation, minimization and verification of high density PLA layout was developed, which includes equation-to-truth table translator, logic minimizer, PLA product term sorter, file generator for plotting stick diagram, dynamic CMOS PLA layout generator and bipartite row folded CMOS PLA layout generator.

Size reduction is performed mainly by logic minimizer and bipartite row folder, and the maximal delay is reduced by sorter.

The file for automatically generated layout is stored in CIF. Each program was written in C language, and was run on VAX-11/750 (UNIX).

I. 序 論

모든 디지털 시스템은 기억부분과 논리 부분의 두가지 기본 형태의 회로로 이루어져 있다고 볼 수 있다. VLSI의 논리 시스템으로서 규칙적인 구조를 가지는 PLA는 random logic network에 비해 설계시간이 적

게 들며, 설계 검증 및 수정이 쉬운 장점으로 인해 근래의 IC 설계에 많이 쓰이고 있다. 그러나 random logic보다 칩 면적이 무척 크며, 또 이로 인한 기생 커패시턴스의 영향으로 동작 속도가 떨어지는 단점을 가지고 있다. 일반적으로 conventional PLA의 트랜지스터가 차지하는 면적의 전체 면적에 대한 비율(칩 이용율)은 AND 평면에서 약10%, OR 평면에서 약4%로 면적의 낭비가 큰 것으로 알려지고 있어¹⁾ 이를 극복하기 위한 연구가 활발히 진행되어 왔다.

여기에는 PLA의 논리함수의 최소화를 행하는 방법과 입력을 몇 그룹으로 나누고 각 그룹의 입력선에 decoder를 사용하여 decoded PLA를 구성하는 방법, 그리고 몇 개의 PLA로 설계하는 방식인 partitioned PLA, macro PLA 등이 있다. 또한 최근에 많이 연구되

^{*}正會員, 韓國科學技術院 電氣 및 電子工學科
(Dept. of Elec. Eng, KAIST)

^{**}正會員, 韓國電子通信研究所
(Electron. & Telecommunication Research Institute)

接受日字: 1985年 4月 1日

고 있는, 입력이나 출력 변수들의 독립성에 착안하여 여러 방향에서 변수를 입출력시켜 programming 평면의 집적도를 높이는 방법인 folding에 의한 방법이 있다.¹⁾ 한편, 우리나라에서는 NMOS static PLA의 설계를 CIF로 발생하는 프로그램 등의 개발이 최근에 이루어 졌으며¹²⁾ folding, 고장검사 등의 방법이 연구되고 있다.

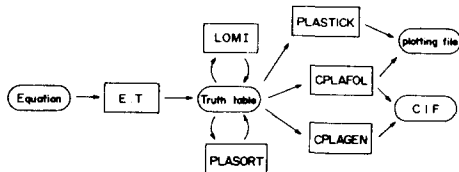


그림1. 전체 system개요도
Fig. 1. Flow chart of overall system.

따라서, 본 논문에서는 그림 1과 같이 진리표를 입력으로 단적과 동작속도의 최적화를 행한 PLA의 layout을 CIF로 발생시켜 주는 program set을 개발함으로써 CIF를 중심으로한 설계 자동화의 영역을 확장하였다.

다음에는 각 프로그램에 대해 자세히 설명하기로 한다.

II. 논리함수 표현으로부터 진리표로의 변환

E. T(Equation-to-Truth table translator)는 Boolean 함수로 주어지는 논리를 PLA로 프로그래밍 할 수 있는 진리표(personality matrix)로 바꾸는 프로그램이다.

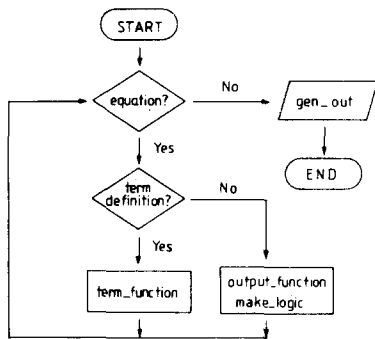


그림 2. E. T의 유통도
Fig. 2. Flow chart of E. T.

E. T는 약간의 dummy logic을 소개하며 그 흐름도는 그림 2와 같다. 자주 쓰이는 적항을 정의하여 쓸 수 있도록 하였으며 term function에서 처리한다. 입력변수의 보수는 변수 다음에“/”으로 표시한다. 구별

되는 2 개 이상의 입력 변수를 붙여 쓰면 AND logic으로 간주하며 필요에 따라 기호“&”를 쓴다. 어떤 입력 변수와 그 입력 변수의 보수가 AND되어 있는 항은 dummy logic으로 무시된다. 출력함수는 필요한 항들이 모두 정의된 후에 기술되며, make-logic에서 실제로 논리표를 구성한다.

함수 하나에 대해 논리표가 구성되는 순서는 다음과 같다.

- 1) 출력 변수를 인식하여 새로운 출력인지 검사한다. 새로운 출력 변수일때만 출력의 수를 증가시키고, 그렇지 않으면 오류를 발생한다.

```

T0=A1^A2^
T1=A1^&A2
T2=A3A3^
f1=T0+T1+T2
f2=ab
f3=a&b+T2
f4=A1A2A1^+A1A2^
f5=a2a2^
  
```

그림3. E. T 입력형태
Fig. 3. E. T input.

- 2) 입력변수를 인식하여, 새로운 입력 일때만 입력의 수를 증가시킨다. 또 적항을 인식하여 새로운 항이면 항의 수를 증가시키고, 이미 존재하는 항 일때는, 그 논리를 가져온다. 모든 함수에 대해 논리표가 만들어지면 진리표를 출력한다. 그림 3 과 같은 입력에 출력 진리표는 그림 4 와 같다. e(end command) 뒤의 문자들은 이 진리표가 입력으로 사용될 때는 무시된다.

```

i 5
o 4
p 5

00---;1000
01---;1000
--1--;0100
--11;0010
10---;10001

e
/* INPUTS IN ORDER
A1 ,A2 ,a,b ,a ,b
/* OUTPUTS IN ORDER
f1 ,f2 ,f3 ,f4
  
```

그림4. E. T 출력형태
Fig. 4. E. T output.

III. 논리 간소화

LOMI(Logic MInimizer)는 PLA 크기 결정의 가장 중요한 변수인 적항의 수를 줄여주며, 아울러gate(transistor)의 수도 줄여 PLA의 면적과 동작 속도의 최적화를 행하는 논리 간소화 프로그램이다. LOMI 는 heuristic direct one-pass method¹³⁾를 사용하므로 절

대 최적값소를 보장할 수는 없지만, 빠르고 최적에 가까운 해를 준다. LOMI의 흐름도는 그림 5와 같으며,

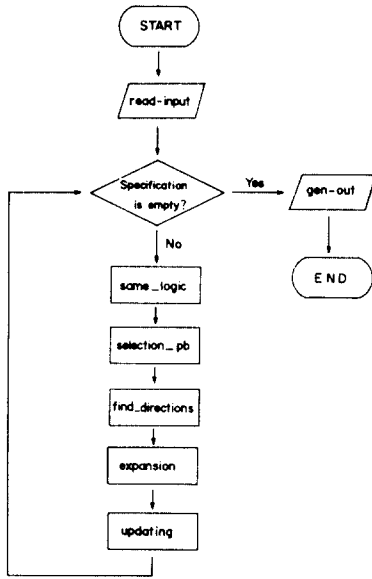


그림 5. LOMI의 유통도
Fig. 5. Flow chart of LOMI.

주요 함수에 대한 설명은 다음과 같다.

- 1) read-input ; PLA logic을 나타내는 입력 진리표를 읽어 specification을 만든다.
- 2) same-logic
다른 항에 의해 포함되는 항을 소거한다.
- 3) selection-pb
확장(expand)을 하기 위한 기본 항을 선택한다. 본문에서는 다른 항에 의해 포함될 확률이 가장 작은 항을 우선적으로 선정한다.
- 4) find-directions

기본 항의 가능한 확장방향(expansion direction)을 구한다.

5) expansion & updating

LOMI는 tree 확장 algorithm¹⁾을 사용하며, updating은 완전히 확장한 기본 항을 해에 넣고, 이 항으로 되는 모든 항을 specification 내에서 제거한다.

그림 6과 같은 진리표를 입력으로 LOMI를 사용하면 그림 7과 같이 적항의 수가 줄어든 출력 진리표를 얻을 수 있다.

IV. 적항의 재 배열

PLASORT는 진리표를 읽어, PLA의 최대 지연시간을 줄이도록 적항을 재 배열한다. 즉 각 적항의 전달 지연 시간을 대략 계산하여, 큰 지연시간을 갖는 항부터 입력/출력 buffer 쪽에 가까이 재 배치한다.

다시 말하여 입력으로부터 보다 먼 쪽의 출력을 구동하거나 보다 많은 출력 함수에서 사용되는 적항을 그렇지 않은 항보다 입/출력 쪽에 가까이 배치하므로써 지연 시간을 줄이는 것이다. 실제적으로 각 적항에 대해 다음과 같은 평가 함수로부터 계산한 값을 큰 순서대로 배열한다.

$$f = \text{length} \times (\text{gweight} \times \text{noutputs} + \text{lweight} \times \text{length})$$

여기서 length는 이 적항이 구동하는 가장 먼 출력의 AND 평면과 OR 평면의 경계로부터의 거리이며, noutputs는 적항에 의해 구동되는 출력단의 수이며, gweight, lweight는 각각 이들에 대한 weight로서 PLA 설계자가 정해지게 되는 값이다.

위의 함수 f는 근사적으로 지연시간 변수 R×C를 나타낸다고 볼 수 있다. (length는 저항 R을, 괄호안의 항은 기생용량과 출력 증폭단의 입력 용량을 합한 전체 용량 C라고 볼 수 있다)

PLASORT는 배열 알고리즘으로 MERGESORT 방식을 사용하였다.¹⁴⁾ 그림 8과 같은 진리표를 입력으로

```

i 4
o 3
p 11

0000;111
00-1;100
0010;111
0100;001
-001;100
010-;011
-110;011
01-0;011
1000;110
10-0;011
1011;100

e
  
```

그림 6. LOMI의 입력 file
Fig. 6. LOMI input file.

```

i 4
o 3
p 7

-0-1 ; 100
1000 ; 110
00-0 ; 111
010- ; 011
-110 ; 011
01-0 ; 011
10-0 ; 011
10-0 ; 011

e
  
```

그림 7. LOMI의 출력형태
Fig. 7. LOMI output.

```

i 4
o 3
p 11

0000;111
00-1;100
0010;111
0100;001
-001;100
010-;011
-110;011
01-0;011
1000;110
10-0;011
1011;100

e
  
```

그림 8. PLASORT의 입력 file
Fig. 8. PLASORT input file.

```

i 4
o 3
p 11

0000 ; 111
0010 ; 111
010- ; 011
-110 ; 011
01-0 ; 011
10-0 ; 011
0100 ; 001
1000 ; 110
00-1 ; 100
-001 ; 100
1011 ; 100

e
  
```

그림 9. PLASORT의 출력
Fig. 9. PLASORT output.

PLASORT를 사용하면 그림 9 와 같이 재 배열된 진리 표를 얻을 수 있다. 여기서 입/출력을 뒷 방향으로 하는 것을 가정했다.

V. Dynamic CMOS PLA

근래에 와서 CMOS 공정은 소비전력이 낮고 잡음 여유도가 크다는 특징 때문에 급속히 실용화 되고 있는 경향이다. 또한 CMOS의 잇점으로는 전원 전압의 범위가 넓고 출력 임피던스가 낮은 것을 들 수 있다. 그러나 NMOS 공정에 비해 칩 면적이 많이 들고, 속도가 느린 등의 단점이 있어서, 이를 타개하기 위한 여러가지 구조의 연구가 활발히 행해지고 있다.

본 논문에서는 PLA를 conventional CMOS logic gate로 실현하는데 생기는 트랜지스터의 Z/L, 즉, con-tance matching 및 속도저하의 문제점을 피하기 위하여 precharge를 사용하는 dynamic CMOS logic을 사용하였다.

그림 10은 CPLAGEN(CMOS PLA GENERator)에 의해 만들어지는 2-phase clock에 의해 구동되는 dynamic CMOS PLA의 schematic diagram이다.

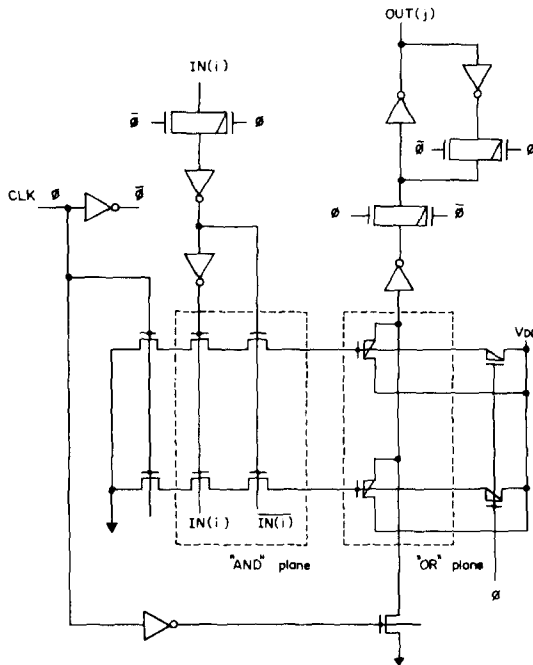


그림10. Dynamic CMOS PLA의 개략도
Fig. 10. Schematic diagram of dynamic CMOS PLA.

보통 PLA는 AND-OR 회로로 이루어지기 보다는 등

가회로인 NAND-NAND나 INVERTER-NOR-NOR-INV-ERTER로 이루어진 경우가 많은데, 여기서의 PLA는 면적을 최소화 하고 PMOS 트랜지스터가 NMOS 트랜지스터보다 느린점을 감안하여 AND 평면은 NMOS가 직렬로, OR 평면은 PMOS가 병렬로 연결되어 있는 NAND-NAND logic으로 설계되어 있으며, one clock 으로 구동된다. 따라서, clock 이 low일때 입력 read 동작이 이루어지고 high일때 출력이 발생되며, 다음 clock의 high때까지 latch된다. 몇가지의 입출력 수에 대해서 SPICE simulation을 하여 표 1 과 그림11에 보였다. 여기서 적항수는 입력수+출력수로 하였으며, PMOS와 NMOS에 대한 모델 파라미터 값은 참고문헌⁶⁾에 준하였다.

표 1. 참고문헌⁶⁾의 CMOS model parameter를 사용한 PLA delay time의 SPICE simulation 결과 Vdd=4.5V, T=100°C (tcklw/td; tcklw 은 그림11에서 product 항의 minimum delay, td는 output 항의 maximum delay를 나타낸다.)

Table 1. SPICE simulation result of PLA delay times using CMOS model parameters from ref.⁶⁾. Vdd=4.5V, T=100°C. (tcklw/td; tcklw denotes product term's minimal delay, and td denotes output term's maximal delay in Fig.11)

Output No. \ Input No.	10	20	30
5	4.0/24.5	11.0/29.5	19.6/34.6
10	6.1/25.5	12.7/31.0	21.7/37.7
20	9.0/28.8	16.2/35.4	26.1/43.9

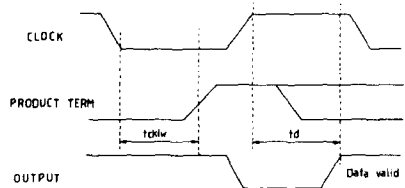


그림11. PLA 동작 파형
Fig. 11. Waveforms of PLA operation.

VI. CIF를 이용하는 Dynamic CMOS PLA의 Layout

CPLAGEN은 진리표를 입력으로 하여 one clock으로 구동되는 dynamic CMOS PLA의 layout을 CIF로 만들어 내는 프로그램이다. CPLAGEN은 그림12의 흐름도에서 보이는 바와 같이 크게 기본 cell발생 과정과

PLA발생 과정의 두 함수로 되어 있다.

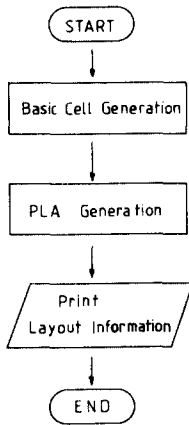


그림12. CPLAGEN의 유통도
Fig.12. Flow chart of CPLAGEN.

주요 함수는 다음과 같다.

1. CIF Command Function⁷⁾

본 논문에서는 CIF command들이 함수로 되어있고, 이 함수들이 call되면 그에 상응하는 command가 출력 file에 쓰여지는 방식을 취하고 있다.

2. 기본 Cell Function

CIF command function을 이용하여 PLA를 구성하는데 필요한 기본 cell을 만든다. 기본 cell들을 설계할 때 설계규칙은 ETRI CMOS 3um rule을 따랐다.⁷⁾

3. PLA발생 Function

기본 cell들을 적당한 위치에 배치시키고, 그 다음 power line이나 clock line을 연결하여 PLA를 구성하는 함수이다. 또 구성되는 PLA의 주요 좌표값도 얻는다. 그림13과 같은 진리표로부터 얻어지는 PLA의 layout은 그림14와 같다.

```

i 4
o 3
p 10

10-1;101
0010;110
-101;011
1-11;110
-11;111
0001;010
01--;110
0101;001
1100;011
-1-0;110
    
```

그림13. CPLAGEN의 입력진리표
Fig.13. Input truth table for CPLAGEN.

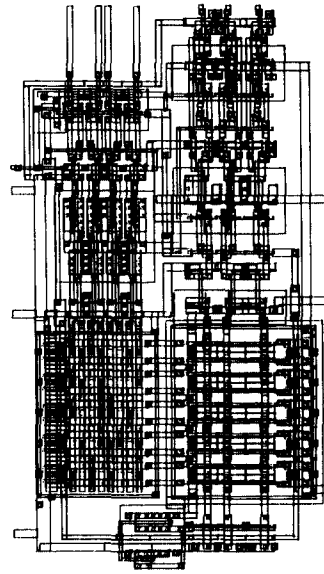


그림14. 그림13의 truth table로 부터얻어지는 CMOS PLA의 layout
Fig.14. Layout of CMOS PLA obtained from truth table of Fig.13.

VII. Bipartite Row Folding

CPLAFOL(CMOS PLA FOLder)은 진리표를 입력으로 하여 bipartite row folding을 한, 2-phase clock으로 구동되는 dynamic CMOS PLA의 layout을 CIF로 만들어 내는 프로그램이다. 또한 stick diagram을 그릴 수 있는 file도 만들어 낸다.

Optimum folding pair을 구하는 문제는 NP-complete

```

1 8
o 6
p 17

0--0----;011000
----1--1;000001
1--1----;001000
----0-1-;000011
0-1-----;110000
----0-1;000101
10-0-----;101000
----11-;000010
0--1----;010000
----1-0-;000101
--1-----;100000
----11-0;000010
1-0-0---;101010
-1-0-1-1;110100
10--00--;011001
-1-10-0-;001010
0-1--00-;010010
    
```

그림15. CPLAFOL의 입력표
Fig.15. Input table for CPLAFOL.

로 알려지고 있으며, 본 논문에서는 graph 표현을 이용하여 near-optimum 해를 찾는 방법을 채택하였다.

PLA의 각 row를 vertex로 하고, 두개의 row가 같은 column상에 cares (0나1)를 가지는 경우가 있을때, 이 두 row(vertex)사이에 edge를 그어 만든 row intersection graph¹⁾로부터 disjoint한 vertex들 사이에 alternating cycle이 생기지 않도록 하면서 directed edge를 첨가해 나간다. 이때 directed edge의 방향에 따라 row가 배치되는 평면이 결정되므로 한 vertex에 두개 이상의 directed edge가 연결될 수 없다.

그림15와 같은 진리표를 입력으로 하여 CPLAFOL로부터 얻는 stick diagram file을 plotting하면 그림 16과 같다.

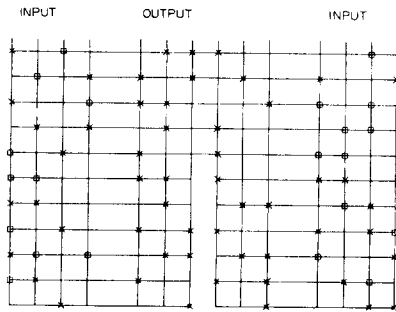


그림16. Folding 된 PLA의 stick diagram
Fig. 16. Stick diagram of folded PLA.

Ⅶ. 結 論

PLA의 입력 자료인 진리표를 다루어 면적의 감소와 동작 속도의 증가를 가져 오게 하는 프로그램들을 개발하였다. 즉, 논리 간소화 프로그램과 PLA 적향선을 새 배열하는 프로그램을 작성하였다. 아울러 논리 함수 표현으로부터 진리표를 얻을 수 있게 하였다.

앞으로 stick editor와 같은 프로그램을 개발하여 그래픽 입력으로 부터 진리표를 얻는 방법과, 고급언어 표현으로 부터 PLA logic을 얻는 방법²⁾의 연구가 수행될 수 있겠다.

본 연구에서 CPLAGEN에서는 설계규칙의 값을 변수로 하여 (technology file) technology에 따라 새로

cell을 설계 하지 않게 하는 방법이 진행되어야 하겠으며, 이때는 아울러 PLA가 optimal크기를 가지게 하기 위한 여러 함수들도 필요하다.

CPLAFOL은 bipartite folding을 함으로써 layout의 어려움을 해결하였으나, 다시 많은 불필요한 면적이 생겼다. 따라서 CMOS technology로서 folding을 할 때는 two layer 이상의 metal, polysilicon process가 뒷받침되면 좀 더 고밀도의 PLA layout을 얻을 수 있겠다.

參 考 文 獻

- [1] Saburo Muroga, *VLSI System Design*. Ch. 7, John Wiley & Sons, New York, 1982.
- [2] J.F. Martinez-Carballido and V.M. Pewers, *PRONTO: Quick PLA Product Reduction*. IEEE 20th Design Automation Conf., pp. 545-552, 1983.
- [3] V.T. Rhyne, et al., "A new techniques for the fast minimization of switching functions," *IEEE Trans. on Computers*, vol. 26, no. 8, pp. 757-764, Aug., 1977.
- [4] E. Horowitz and S. Sahni, *Fundamentals of Computer Algorithms*. Ch. 3, Computer Science Press, 1978.
- [5] S. Kang, W.M. Van Cleemput, *Automatic PLA Synthesis From DDL-P Description*. IEEE 18th DA Conf., pp. 391-397, 1981.
- [6] MOS Technology Development에 관한 연구, 2장, 한국전자기술연구소, 1984.
- [7] C. Mead and L. Conway, *Introduction to VLSI Systems*, Ch. 4, Addison-Wesley, 1980.
- [8] M. Luby, U. Vazirani, V. Vazirani and A. Sangiovanni-Vincentelli, *Some Theoretical Results on the Optimal PLA Folding Problem*. Submitted for publication.
- [9] G.D. Hachtel, *Techniques for Programmable Logic Array Folding*. IEEE 19th Design Automation, Conf., pp. 147-155, 1982.
- [10] 정대성, 양영일, 경중민, "CIF를 이용한 PLA의 Layout 자동화," 전자공학회지 22권 1호 pp. 14~21, 1985년 1월.