

# CIF를 이용한 PLA의 Layout 자동화

## (Automated Layout of PLA using CIF)

鄭泰聖\*, 梁榮日\*, 慶宗旻\*\*

(Tae Sung Jung, Yeong Yil Yang and Chong Min Kyung)

### 要 約

PLA(programmable logic array)의 논리함수 표현으로 부터 CIF(Caltech Intermediate Form) file을 만드는 PLACAD라는 프로그램과 CIF file을 graphics file로 바꿔주는 CIFPLOT라는 프로그램과 CALPLOT과 TEKPLOT이라는 graphic software를 이용하여 PLA의 몇 가지 mask layout을 각각 plotter와 terminal screen에 그려 내었다. 이 논문에서는 depletion load NMOS 기술을 예로 들어 6개의 기본 layout cell을 이용하여, PLA 논리 방정식에 맞도록 mask layout을 자동적으로 합성해 내는 과정을 보였다. 관련된 대개의 프로그램들은 PASCAL 언어로 작성하여 VAX 11/780에서 수행되었다.

### Abstract

Several mask layout patterns for PLA's were generated using PLACAD, a program which transforms logic description for PLA into CIF (Caltech Intermediate Form) file, and CIFPLOT, a program which transforms CIF file into graphics file which is in turn, used by CALPLOT and TEKPLOT to produce the drawings on plotter and terminal screen, respectively. The depletion load NMOS technology was chosen in this paper as an example to demonstrate the capability for automatic synthesis of the mask layout from the PLA logic equation set using six basic layout cells. Most of the routines were written in PASCAL and run on VAX 11/780 with VMS operating system.

### I. 序 論

Microprocessor의 control block 등의 일반 논리 회로를 hardware로 실현하는 대표적인 방법으로는 random logic, micro-coded ROM, PLA(programmable logic array)가 있는데 설계와 debugging 작업의 소요 시간과 chip 상의 소요 면적의 관점에서 최근 PLA가 상당한 관심을 끌게 되었다. PLA는 gate array 방식, standard cell 방식과 더불어 대표적인 custom 설계방식으로 부각되어, 최근에는 소요면적,

propagation delay 및 주위 회로와의 연결 용이도를 개선시키기 위하여 decoded PLA, folded PLA 등의 PLA 구조를 최적화하는 연구가 진행되고 있다. 이론적으로는 PLA layout의 column 및 row folding을 통하여 active area를 25%로 줄일 수 있으며 논리식의 간소화 및 PLA의 계층적 구조의 개선에 의하여 routing area를 10%이하로 줄일 수 있다.<sup>1)</sup>

PLA가 ROM에 비금 같 정도로 매우 반복적인 구조를 가지고 있는 점에 착안하여, layout을 함에 있어서 computer program을 사용함으로써, 설계 및 debugging 작업을 용이하게 하는 것이 이 논문의 요지라고 할 수 있다. 주어진 논리함수 혹은 Boolean Table로부터 PLA의 AND, OR 평면의 각 dimension의 크기를 알아내고, 여기에서 input buffer, output

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

接受日字: 1984年 5月 28日

driver 등을 포함한 전체 layout의 frame을 만든후에 logic에 맞도록 각 cell에 pull down transistor를 넣어 주는 방식을 사용하였다.

각 단위 cell의 layout geometry data는 CIF(Caltech Intermediate Form)<sup>[1]</sup>를 써서 file내에 저장하고, 이를 program내에서 editing한 후에 (non-interactive mode) Tektronix graphic monitor 및 Calcomp plotter를 써서 layout을 그릴 수 있도록 하였다. 모든 프로그램은 PASCAL과 FORTRAN 77언어로 작성되었으며 VAX 11/780(VMS)에서 수행 되었다.

## II. CIF File의 그래픽 표현

CIF(Caltech Intermediate Form)은 Caltech의 Mead<sup>[2]</sup>에 의해 제안되어, 사용자의 편의와 layout으로의 변환, design rule check 및 layout으로 부터 회로의 추출 등을 편하게 하기 위한 언어로서 pattern generator file과 사용자 editing file의 중간 file을 구성하는 언어로 사용된다. CIF에서 사용되는 command를 살펴보면 mask geometry를 나타내는 command로 BOX, POLYGON, ROUNDFLASH, WIRE가 있는데, WIRE의 경우에는 그림1(a)와 같이 width가 W로 주어지며 P1, P2, P3, P4의 path로 이루어지는 도형으로 정의 되는데, 이 논문에서는 그림1(b)와 같이 여러개의 box로 분해하여 처리하였다.

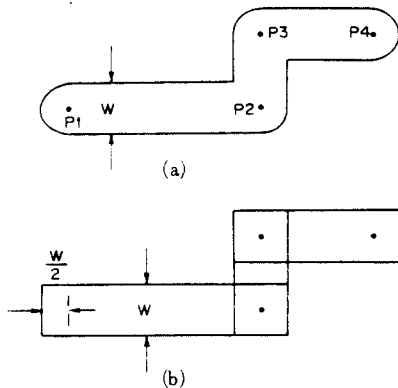


그림 1. (a) 반경이  $W/2$ 인 원의 path로 만든 wire  
(b) box로 구성한 wire의 도형  
Fig. 1. (a) Wire pattern generated by the movement of a circle with radius  $W/2$ .  
(b) Wire pattern approximated by multiple box patterns.

Roundflash는 원을 나타내는데, 이 논문에서는 data 양을 줄이기 위하여 정관각형으로 대체 하였다. 또한 CIF의 중요한 feature로는 layer의 이름을 표시하는

Layer Specification Command(LND, LNC, LNM, LNL, LNP, LNB, LNG)와 사용자의 편의를 위한 DEFINE, CALL, DELETE, COMMENT 등의 command가 있다.<sup>[1]</sup>

이 논문에서 작성한 CIFPLOT이라는 program에서 CIF file을 읽어 graphic device에 그림을 그려주기 위한 graphic file을 만들어 주는데, 그 처리하는 과정은 다음과 같다.

- Initial ; 모든 parameter의 초기값을 준다.
- Readline ; CIF file의 data를 읽어 들인다.
- Treatdata ; CIF file의 각 command line의 첫 문자에 따라 Box;Poly;R;Wire;Layer;Call;Define;E;C;로 인식하여, 각각 procedure에 보내어 그림을 그릴 평면상의 좌표값을 구한 뒤, layer specification과 함께 memory에 저장한다.
- Minimax ; 전체 layout에서의 X, Y 좌표의 최대값과 최소값을 구한다.
- Coutfile ; Memory에 저장된 값들을 graphic file에 적어 넣는다.
- End ; 프로그램의 수행완료를 표시한다.

Graphic file의 format을 보면 첫 부분에는 전체 drawing에 대한 minimum bounding box의 좌표값(Xmax, Xmin, Ymax, Ymin)과 box, polygon, roundflash의 갯수를 기록하고, box, polygon, roundflash의 순으로 각 도형의 layer type과 좌표값을 저장해 둔다. Graphic Terminal 혹은 plotting area에 맞게 X, Y 좌표값의 범위를 정한후, 이 범위내에 있는 graphic file의 data를 적절히 scaling하여 각 layer type에 대응하는 선의 종류를 택하여 terminal screen에 그려 주게 된다. 이때, 각 layer type을 나타내는 선의 종류는 그림 2와 같다. 그리고 그림 3에 이러한 notation에 의해 wire, polygon, box, roundflash(8 각형)의 여러 가지 도형을 그린 것을 보였다. Composite drawing이 아닌 경우에는 특정한 layer의 set만을 선택하여 display하는 것도 가능하게 하였다.

ND (diffusion)	—————
NC (contact)	.....
NP (poly-silicon)	-----
NI (implant)	-----
NM (metal)	-----
NB (buried contact)	—————
NG (overglass openings)	—————

그림 2. CIF의 각 층에 대한 기호  
Fig. 2. Representation for each layer in CIF.

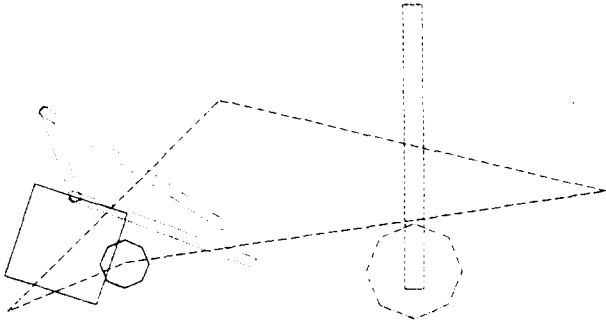


그림 3. 이 논문에서 사용하는 각 직선 기호로 나타낸 여러가지 도형

Fig 3. Various geometrical patterns represented with the convention in this work.

III. CIF를 이용한 PLA Cell의 설계

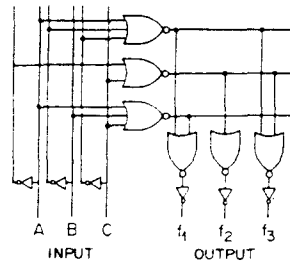
일반적인 PLA (programmable logic array)의 transistor level diagram 및 logic diagram은 그림 4와 같으며 PLA의 layout은 6 종류의 기본 cell들로 구성할 수 있다.

표 1. PLA layout에 사용되는 6 가지 기본 cell  
Table 1. Six basic cells for PLA layout composition.

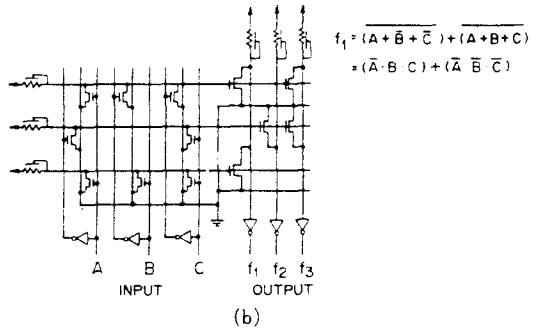
Cell Number	Size(unit : λ)	#Trans	Description
333	28×28	0	Cell Pair
351	32×28	0	Connect
361	46×28	2	Pullup Pair
371	28×20	0	Ground Cell
381	28×78	2	Input Buffer
391	28×90	2	Output Buffer

그림 4 (a), (b)의 NOR-NOR 연결은 Boolean 대수에 의해 AND-OR 연결로 바꾸어 생각할 수 있으며, 그림 4 (b)의 좌편을 AND 평면, 우편을 OR 평면이라고 부른다. 그림 5에 나타낸 이들 cell들은 표 1과 같다 (λ는 process technology에 의해 정해지는 최소 dimension을 나타냄). 한편, 주어진 논리함수를 실현하기 위하여 기본 cell pair에 channel 형성을 위한 diffusion을 특정된 곳에 첨가해 주어야 하는데, 이것을 실현하는 방법으로 첫째, 6 개의 cell로써 PLA를 구성하고, 논리 함수에 따라 필요한 곳에 diffusion이 첨가됨을 선언해 주는 방법과 둘째, 6 개의 기본 cell외에 모든 경우의 논리 함수에 대한 cell들을 정의하고 이들을 cell 333의 확장 cell로서 인용하는 방법이 있다.

본 논문에서는 CIF의 symbol definition 및 symbol call의 특징을 이용함과 아울러 process 변화등에 따른 cell updating의 용이함을 고려하여 후자의 방법을



(a)



(b)

$$f_1 = (A + \bar{B} + \bar{C}) + (A + B + C) = (\bar{A} \cdot B \cdot C) + (\bar{A} \bar{B} \bar{C})$$

그림 4. (a) PLA의 논리 개략도  
(b) Depletion NMOS를 이용한 PLA의 개략 회로도

Fig 4. (a) Logic diagram for PLA.  
(b) Depletion NMOS circuit schematic for PLA.

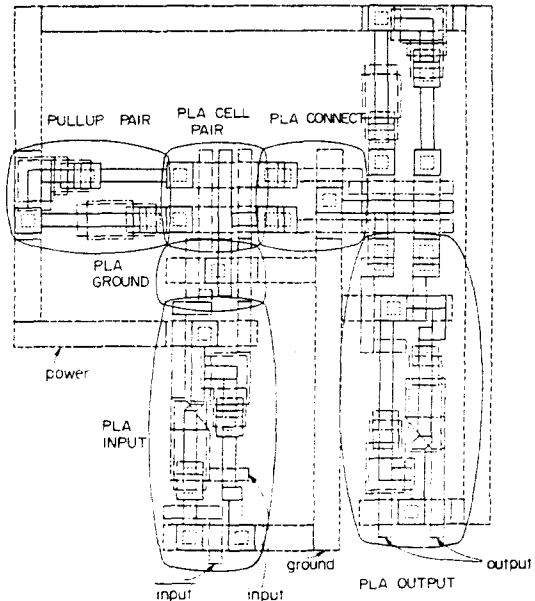


그림 5. PLA 합성에 필요한 6 개의 기본 cell의 layout  
Fig 5. Layout for six basic cells in PLA composition.

선택하였다. 이 논문에서는 PLA의 product항의 갯수와 output 항의 갯수를 모두 짝수가 되도록 하였다. 이것은 pullup pair가 면적의 효율적인 이용을 위하여 짝을 이루어 cell을 이루므로 이에 준하기 위한 것이다. 확장된 cell들은 기본적으로 333 cell과 같은 구조를 가지며 channel 형성을 위한 diffusion이 첨가되는 위치에 따라 다음과 같이 구분된다. 우선, AND plane에서 사용되는 cell들은

기본 cell ; 333

확장 cell ; 303, 313, 330, 331, 300, 301, 310, 311의 9가지가 있다. 세 digit으로 구성된 cell 번호에서 처음 숫자 3은 PLA cell pair 임을 말하며 두번째 및 세번째 숫자는 각각 cell pair 내의 왼쪽 product 항, 오른쪽 product 항에 대한 것으로서 아래와 같은 의미를 갖는다 (cell pair 내의 polysilicon line 입력 변수가 서로 complement이므로 한개의 product 항 밑에 두개의 channel이 동시에 형성될 수 없으므로 가지수는  $3^2=9$ 가 된다).

- 0 ; Complement input poly 밑에 diffusion이 첨가된 경우
- 1 ; Normal input poly 밑에 diffusion이 첨가된 경우
- 3 ; Don't care, 두 input poly 밑에 모두 diffusion이 첨가되지 않은 경우

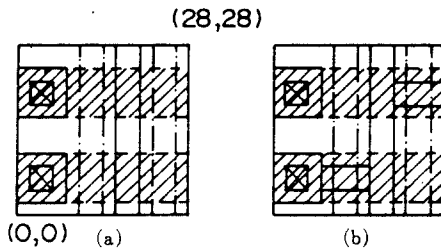


그림 6. (a) 기본 cell(333)의 layout  
 (b) 확장 cell(310)의 layout  
 Fig 6. (a) Layout of basic PLA cell(333).  
 (b) Layout of extended PLA cell(310).

예로써 cell 333에 대한 layout 및 cell 310에 대한 layout은 그림 6과 같다. 한편, OR plane에서 쓰이는 cell들은 입력 plane에서 쓰이는 cell들이 90° 시계 방향으로 회전된 것으로 생각할 수 있으며, 이때는 한개의 output 항 밑에 두개의 diffusion이 동시에 형

성될 수 있으므로 가지수는  $4^2=16$ 이 되어 다음과 같은 7개의 확장 cell들이 추가로 정의된다.

기본 cell ; ROT(333)

확장 cell ; ROT(303) ROT(313) ROT(330) ROT(331)  
 ROT(300) ROT(301) ROT(310) ROT(311)

추가된 확장 cell : 340 341 304 314 344 343 334

이때 ROT는 시계 방향으로 90°회전을 뜻하며 추가된 확장 cell 번호에서의 둘째, 셋째 자리의 숫자는 각각 cell pair 내의 오른쪽 product 항, 왼쪽 product 항에 대한 것으로서

- 0 : 왼쪽만 연결                      1 : 아랫쪽만 연결
- 3 : 모두 연결되지 않음          4 : 양쪽 모두 연결

을 뜻한다. 위의 cell 중 기본 cell 6개만이 library file에 저장되며 확장 cell 들은 program이 수행되는 동안 출력 file에 써지게 된다. 이때 cell들의 layout은 CIF의 text로 표현되어 저장된다. 예를 들어, 위의 cell 333에 대한 CIF file의 내용은 아래와 같다.

```

DS333 ;
LND ;
B 8 8 4 6 ;
B 8 8 4 20 ;
B 4 28 18 14 ;
LNP ;
B 4 28 12 14 ;
B 4 28 24 14 ;
LNM ;
B28 8 14 20 ;
B28 8 14 6 ;
LNC ;
B 4 4 4 20 ;
B 4 4 4 6 ;
DF ;
    
```

#### IV. Cell Base를 이용한 전체 PLA의 설계

PLA를 설계하는 전체 system을 도식화하면 그림 7과 같으며, 각 program module의 기능은 아래와 같다. PLATRANS: Equation 형태의 입력을 truth table 형태로 변환한다.

PLACAD : Cell library를 이용하여 PLA형태에 준하는 논리함수에 따라 전체 layout에 대한 CIF file을 만든다.

PLA의 설계를 위한 입력 data에는 표 2와 같은 두 가지 형태가 있다. 표 2의 (a)는 결국 program 내에서 (b)와 같이 변환되어 쓰이나 사용자의 편의에 따라 선택 될 수 있다. 표 2 (a)에서 P 및 Y는 product항 및

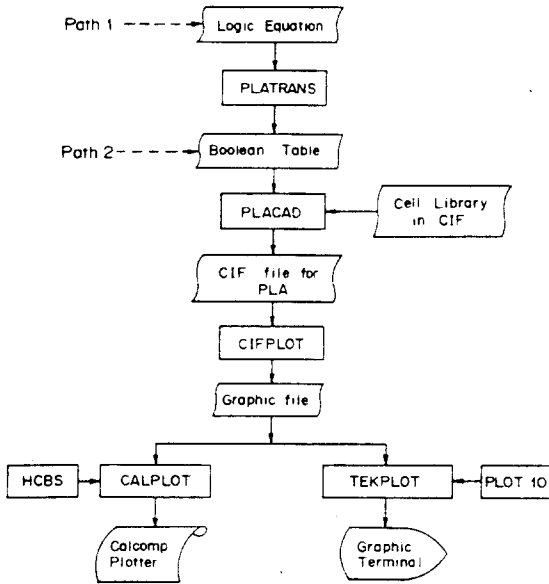


그림 7. PLA layout 합성의 유통도  
Fig 7. Flowchart of PLA layout composition.

PROCEDURE TRUTH\_TABLE\_TO\_CIF :

```

BEGIN
  Copy_Selected_Library_Cells :
  Write_Extended_Cells :
  ROW:= Cell_371_Y+Cell_381_Y+P_Cell
    _Number+Cell_333_Y ;
  Insert_Pullup_in_OR_PLANE :
  ROW:= ROW+Cell_333_Y ;
  Insert_Pullup_in_AND_PLANE :
  Determine_First_Row_Cells :
  Insert_Extended_Cells_Determined :
  FOR LOOP:=2 TO P_Cell_Number DO
  BEGIN
    ROW:= Cell_371_Y+Cell_Y+Cell
      _333_Y+(P_Cell_Number
        _LOOP) ;
    Insert_Pullup_in_AND_PLANE :
    Determine_Next_Row_Cells :
    Insert_Extended_Cells_Deter-
      mined :
  END ;
  Insert_Input_Output_GND_Cells_and
    _Powerline
END ;

```

그림 8. PLACAD의 유통 algorithm  
Fig 8. Flow algorithm of PLACAD.

output을 나타내며 A부터 F는 입력변수를 ~는 complement를 나타낸다. 표2(b)에서 I, O, P는 각각 입력

표 2. PLACAD의 두가지 입력방식

- (a) 방정식에 의한 형태
- (b) Boolean table에 의한 형태

Table 2. Two different input methods for PLACAD ;

- (a) Equation form.
- (b) Boolean table form.

- P 1 = ADE
- P 2 = BDE
- P 3 = A~B~DE
- P 4 = CD~E
- P 5 = C~D~E
- P 6 = A~BD~E~
- P 7 = AD~E~
- P 8 = B~D~E~
- P 9 = CDE~
- P 10 = C~DE~
- P 11 = A~BE~
- P 12 = C~D~F ;
- Y 1 = P 5 + P 6 + P 7 + P 8 + P 9
- Y 2 = P 3 + P 4 + P 5 + P 6
- Y 3 = P 3 + P 5 + P 7 + P 8 + P 10
- Y 4 = P 6 + P 7 + P 8 + P 9 + P 10
- Y 5 = P 4 + P 5
- Y 6 = P 1 + P 2 + P 3 + P 4
- Y 7 = P 9 + P 10
- Y 8 = P 11 + P 1
- Y 9 = P 12 + P 2 + P 1

(a)

.I	6		
.O	9		
.P	12		
	1 × × 1 1 ×	0 0 0 0 0 1 0 1 1	
	× 1 × 1 1 ×	0 0 0 0 0 1 0 0 1	
	0 0 × 1 1 ×	0 1 1 0 0 1 0 0 0	
	× × 1 0 1 ×	0 1 0 0 1 1 0 0 0	
	× × 0 0 1 ×	1 1 1 0 1 0 0 0 0	
	0 1 × 0 0 ×	1 1 0 1 0 0 0 0 0	
	1 × × 0 0 ×	1 0 1 1 0 0 0 0 0	
	× 0 × 0 0 ×	1 0 1 1 0 0 0 0 0	
	× × 1 1 0 ×	1 0 0 1 0 0 1 0 0	
	× × 0 1 0 ×	0 0 1 1 0 0 1 0 0	
	0 1 × × × 0	0 0 0 0 0 0 0 1 0	
	× × 0 0 × 1	0 0 0 0 0 0 0 0 1	

.E

(b)

수, 출력수 및 product 항의 수를 나타내며 table에서의 1, 0, ×는 각각 입력 변수 자체, 입력 변수의 com-

plement 및 don't care를 나타낸다. 한편 기본 cell pair 333이 두개의 product 항과(AND plane) 두개의 출력항(OR plane)을 함께 포함하게 되므로 product 항과 출력항의 갯수가 홀수개인 경우 사용되지 않는

row (AND plane)와 column(OR plane)이 하나씩 생기게 된다. 이때, product 항의 경우는 가장 위의 row를, 출력항인 경우는 가장 우측의 column을 dummy line으로 포함 시켰다. Library cell들을 이용하여 PLA

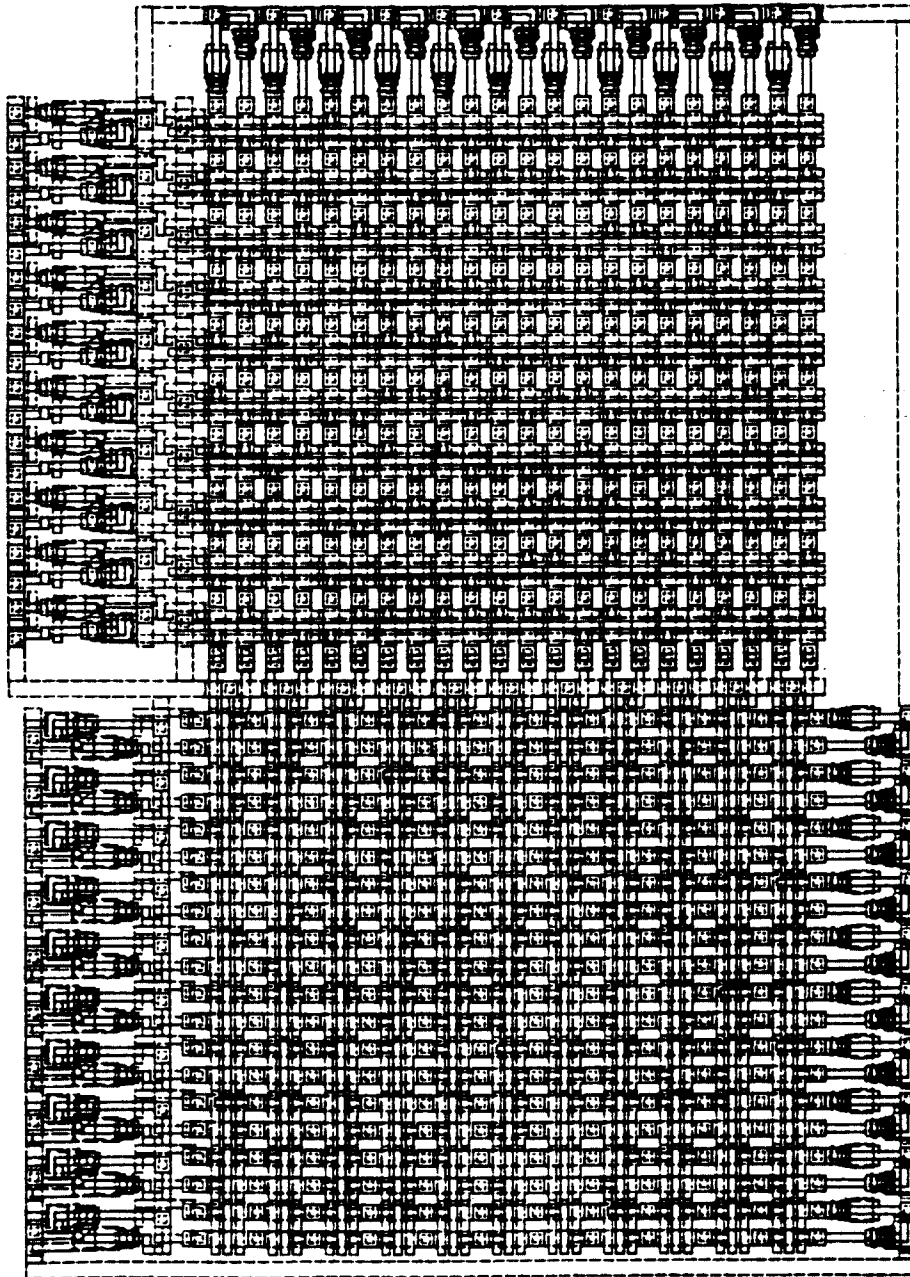


그림 9. PLACAD, CIFPLOT와 TEKPLLOT를 사용하여 4114 Tektronix terminal에 그려낸 신호등 조절기의 PLA layout

Fig. 9. PLA layout for modified traffic light controller generated using PLACAD, CIFPLOT and TEKPLLOT on 4114 Tektronix terminal.

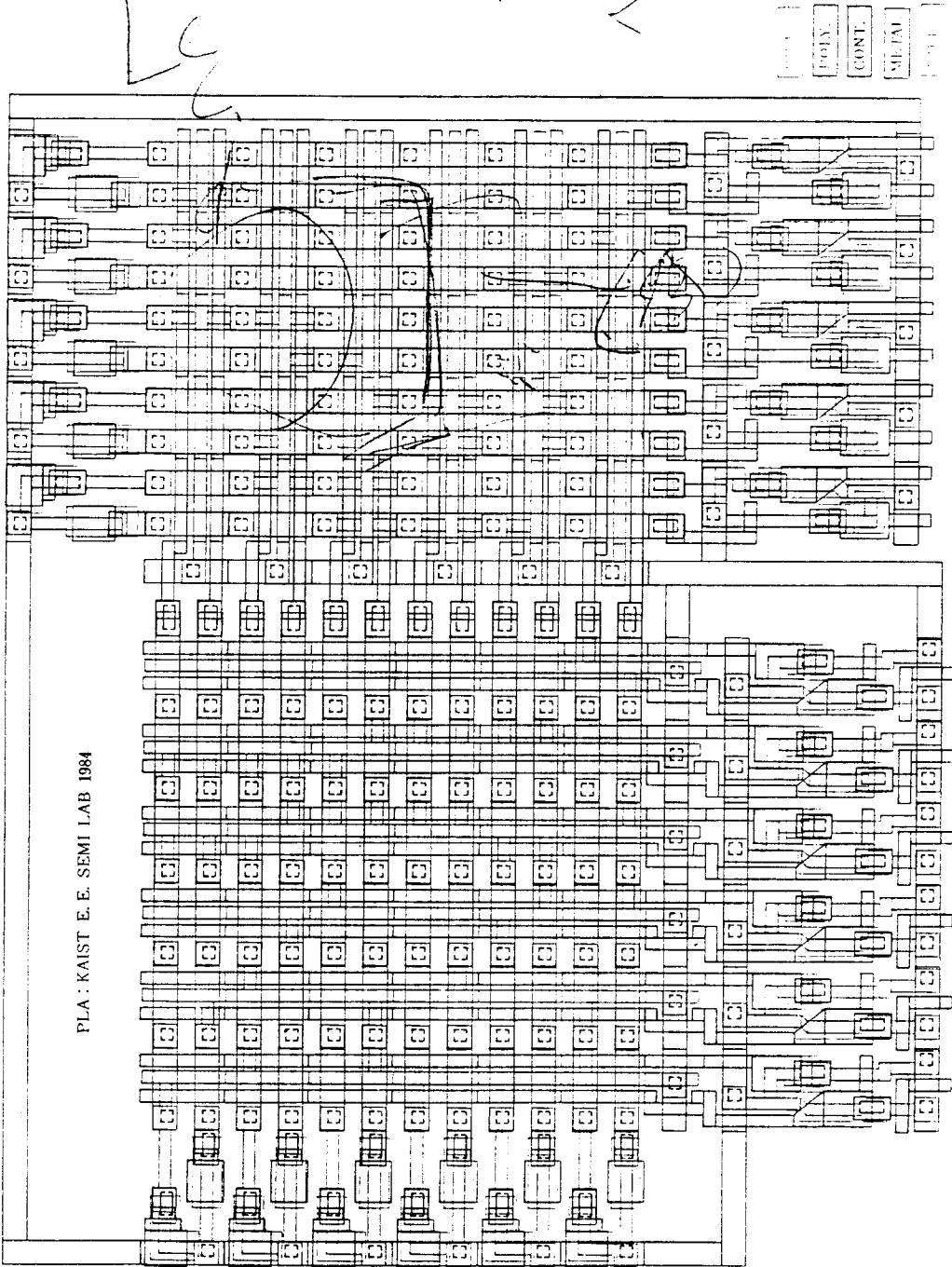


그림10. Calcomp plotter를 이용한 표 2의 논리 방정식에 대한 PLA layout  
 Fig. 10. PLA layout for the logic equation set shown in Table 2 using Calcomp plotter.

를 구성하는 program(PLACAD)의 algorithm은 그림 8과 같다.

앞에서 서술된 바와 같이 입력에 의한 product 항의 수가 홀수인 경우 처음 row (AND plane의 가장 윗측, OR plane의 가장 윗측)는 사용되지 않는 것으로 취급되므로 Determine-First-Row-Cells에서 이것을 결정하고 나머지 row는 주어진 logic 함수를 그대로 실현하게 된다. 한편 요구되는 전기적 특성에 따라 6개의 기본 cell들의 layout이 바뀌는 경우 program의 커다란 수정없이 재구성이 가능하므로 매우 효과적인 design path를 이룰 수 있다. 그림 9에 보인 것은 PLACAD, CIFPLOT과 TEK PLOT을 연계하여 Tektronix 4114 terminal에 그런 PLA layout으로서 참고문헌[2]의 traffic light controller에 대한 logic을 확장한 것이다. 또한, 표 2에 주어진 입력을 PLACAD, CIFPLOT과 CALPLOT을 연결하여 Calcomp color plotter에 그린 것을 그림 10에 보였다.<sup>(3,4)</sup>

## V. 結 論

집적회로의 layout, 회로검증등을 포함한 설계작업을 자동화하기 위하여 첫째, 여러가지의 design, simulation 및 verification을 위한 software를 개발하는 것과 둘째, 이러한 software의 입출력 medium이 되는 data base가 필요하다. 각 종류의 CAD program마다 요구되는 입출력 file의 format이 고유하므로, 이들을 서로 접속시키기 위한 interface routine이 필요하게 되는데, CIF를 file format으로 채택 할 경우 이러한 interface routine의 coding 작업이 매우 수월해짐에 착안하여, 이 논문에서는 CIF를 layout data base 언어로 하여 PLA의 설계작업을 논리함수로 부터 layout까지 자동화하는 과정을 수행 하였다. CIF

는 lay out은 물론 야로부터 design rule check 혹은 회로 추출 작업까지 수행하기 위한 file을 만들기 편리한 언어로 생각되고 있다.

본 연구에서는 CIF file로부터 graphic file을 만들기 위한 CIFPLOT과 이로부터 graphic terminal과 plotter의 그림을 얻기 위한 TEK PLOT, CALPLOT등의 graphic package를 개발하여, 간단한 PLA layout을 그려 보았다. 기본 cell의 설계는 NMOS인 경우에 참고문헌[2]의 설계를 주로 인용하였으나, CMOS인 경우를 포함한 다른 제작기술에 대하여도 마찬가지로 기본 cell만 CIF로 표현하고 PLACAD와 같은 editing routine에 의해 규칙성이 있는 layout을 갖는 chip을 설계하는데 매우 유용하게 쓰일 것으로 보인다. 또한, CIF를 근거로 하여 circuit extractor, design rule checker, sticks editor 등의 설계 프로그램이 작성되는 것이 매우 바람직 할 것으로 생각된다. Caltech에서 발표한 CIF는 floating point, parameterized file등을 취급 할 수 없게 되어 있으므로, 이러한 점들을 보완하도록 program 개발작업이 진행되는 것이 좋을 것으로 생각 된다.

## 參 考 文 獻

- [1] R. Ayres, *VLSI Silicon Compilation and the Art of Automatic Microchip Design*. Prentice-Hall, Inc., 1983.
- [2] Mead & Conway, *Introduction to VLSI Systems*. Addison-Wesley, 1980.
- [3] Tektronix PLOT 10, Tektronix Inc., 1977.
- [4] *Programming Calcomp Electromechanical Plotters*. California Computer Products, Inc.