

## 論 文

## 연상 메모리의 자동설계에 관한 연구

正會員 金 鍾 善\* 正會員 白 寅 天\* 正會員 朴 魯 京\*\* 正會員 車 均 鉉\*

## A Study on the Automatic Design of Content Addressable Memory

Jong Seon KIM\*, In Cheon PAIK\*, Nho Kyung PARK\*\*,  
Kyun Hyeon TCHAHA\* *Regular Members*

**要 約** CAM은 RAM이나 PLA 처럼 규칙적인 구조를 갖고 있으므로 CAM 자동설계 프로그램을 제작하기 용이하다. 본 프로그램은 CIF 형태로 그 결과가 출력되고 수정 작업이나 결과를 화면에 보기위해 IBM/PC 상에서 디스플레이 프로그램을 개발하였다. CIF 파서는 YACC와 LEX로 제작하였고, 플로팅을 위해서는 ROLAND XY 플롯터를 사용하였다. 위의 과정을 하나의 메뉴안에서 실행에 따라 수행하도록 Full-Down 메뉴를 사용하여 종합하였다.

**ABSTRACT** Since CAM structure is regular structure as that of RAM or PLA, CAM generator program is easy to implement. This program outputs CAM layout in the form of CIF(Caltech Intermediate Format) and graphic display program is debugging or displaying CAM generator output, which are implemented on PC/AT with MS C(5.0) graphic library and C language. CIF parser is programmed with YACC(Yet Another Compiler Compiler) and LEX(Lexical Analyzer) in order to flat the CIF data. For the purpose of plotting, the layout output using ROLAND XY plotter is developed. By combining these program described above, from CIF generation to layout plotting can be executed on pull down menu according to user's option.

## I. 서 론

회로의 집적도가 증가함으로 회로의 설계시간을 단축시키고 높은 신뢰도를 갖는 회로를 설계하기 위해 컴퓨터를 이용한 회로설계가 필수적이 되었다.<sup>(1)</sup> 따라서 회로 설계상의 복잡성을 극복하

기 위한 설계방법의 하나로 모듈 컴파일러(module compiler) 또는 레이아웃 제너레이터(layout generator)에 대한 연구 및 개발이 진행되어 왔다.<sup>(2)</sup>

본 논문에서는 연상 메모리 소자가 PLA이나 RAM과 같이 규칙적인 구조를 가지고 있어, 설계시간을 단축하고 설계검증이 용이하며 사용자가 원하는 기능에 따라 입력된 비트수와 워드수에 따라 연상 메모리칩의 레이아웃이 자동으로 설계 되는 연상 메모리 제너레이터 프로그램을 개발하였다.<sup>(5)(6)</sup>

\* 高麗大學校 電子電算工學科  
Dept. of Electronic Eng., Korea University\*\* 湖西大學校 情報通信工學科  
Dept. of Information Telecommunication Eng.,  
Hoseo University  
論文番號 : 90-87 (接受1990. 8. 27)

또한 제너레이터 프로그램을 개발하는 과정에서 생성된 CIF 출력 데이터를 컴퓨터 화면에 디스플레이 시켜 봄으로써 출력 데이터의 이상 여부를 확인할 수 있는 프로그램을 MS C(5.0) 그래픽 라이브러리를 이용하여 개발하고 출력된 레이아웃 데이터 정보를 플롯터에 그리기 위한 프로그램도 개발하였다.<sup>10)</sup> 그리고 개발된 프로그램들은 풀다운(pull down) 메뉴 방식을 이용하여 하나의 소프트웨어 tool로써 만들어 CIF 데이터 생성에서부터 레이아웃 도면까지 설계자가 손쉽게 회로 설계를 할 수 있도록 하였다.<sup>12)</sup> 개발한 전체 프로그램의 블록도는 그림 1과 같다.

## II. 연상 메모리 제너레이터 프로그램

연상 메모리 구조가 입력된 비트수와 워드수에 따라 레이아웃 정보를 CIF 데이터로 출력시키는 제너레이터 프로그램을 개발하는데 있어서 레이아웃 설계는 인접 블록간의 피치(pitch)를 똑같이 하여 블록간의 배선이 필요없는 피치 배정

(pitch matching) 기법을 이용하여 블록들을 배치한 함으로써 전체 레이아웃이 구성되는 설계 방법을 적용하였으며 제너레이터 프로그램의 입력 가능한 최대 비트수와 워드수는 64비트와 256워드이다.<sup>12)23)</sup>

### 2.1 제너레이터 프로그램 기능

제너레이터 프로그램은 세가지 기능으로 다음과 같이 나눌 수 있으며,

- (1) Technology 파일을 읽어 들이는 기능
- (2) 레이아웃 정보를 CIF 데이터로 출력시키는 기능
- (3) Primitive 셀을 생성시키는 기능

각각의 기능에 대한 동작 내용은 다음과 같다.

첫째 : Technology 파일에는 설계 규칙을 기술하게 되는데 본 프로그램에서는 3um CMOS N well 공정이 Lambda 규칙으로 표기되어 있다.

둘째 : CIF 데이터 생성은 DEF START( ), DEF END( ), DRAW BOX( ), DRAW CELL( ) 등과 같은 함수로써 레이아웃 정보가 CIF 문법에 있는 명령어들로

pull down menu 구조

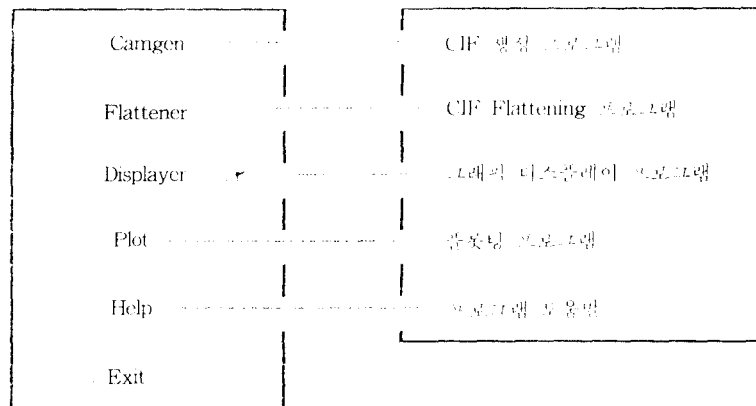


그림 1. 연상 메모리 자동설계를 위한 프로그램 블록도

변환되어 출력된다.

셋째 : Primitive 셀을 생성시키는 기능에서는 pMOS와 nMOS 그리고 4종류의 콘택(contact)등을 기본 셀로 생성시킨다.

위의 기능을 수행하는 제너레이터 프로그램의 흐름도는 그림 2와 같다.

### 2.2 제너레이터 프로그램 구조

실제한 연상 메모리칩의 구조를 모델로 삼아 각 블록의 최상위의 서브셀(subcell)들의 피치를 상하·좌우가 맞도록 레이아웃 실패를 하였으며 피치를 매칭시킨 이유는 다음과 같다.

- (1) 피치 매칭으로 인한 셀 레이아웃의 면적 증가가 배선 면적의 감소와 비교하면 제너레이터 프로그램이 만들수 있는 메모리 용량 한도내에서 배선의 복잡도가 증가할 경우 면적을 줄일 수 있는 효과가 있다.
- (2) 배선이 자동으로 수행되므로 배치만으로 전체 레이아웃이 완성되어 레이아웃 생성 속도가 빠르게 된다.

- (3) 워드 수나 비트 수가 증가하는 경우, 피치가 가장 큰 서브셀을 가진 블록이 상하·좌우로 길게 늘어나게 되어 칩이 rectangle 한 모양을 가질 수 없게 되나 피치 매칭은 이런 문제를 효율적으로 해결할 수 있다.

위의 장점을 가진 피치 매칭기법을 이용하여 사용자가 원하는 비트 수와 워드 수를 입력값으로 받아 연상 메모리칩을 생성시키는 제너레이터 프로그램을 개발하고 출력된 레이아웃은 그림 3과 같다. 전체적인 프로그램을 구성하는 함수들의 블록도는 그림 4이고 제너레이터를 구성하는 주요 함수들의 기능에 대한 설명은 표 1과 같다.<sup>10)</sup>

### III. CIF의 Primitive CIF 변환 프로그램

계층적인 구조를 갖는 CIF 데이터를 primitive CIF로 flatten 시키기 위해서 CIF 파서(parser)와 셀 라이브러리가 있어야 함으로 CIF 파서는 YACC(Yet Another Compiler Compiler)를 사용

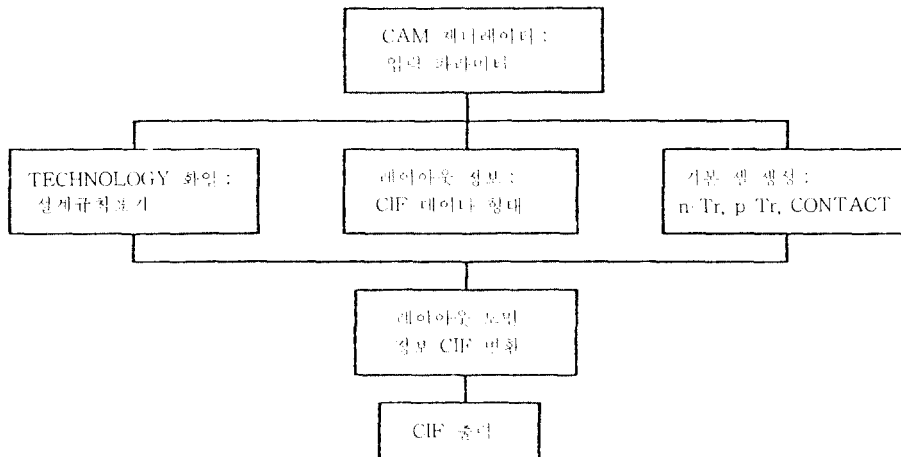


그림 2. 연상 메모리 제너레이터 프로그램 흐름도

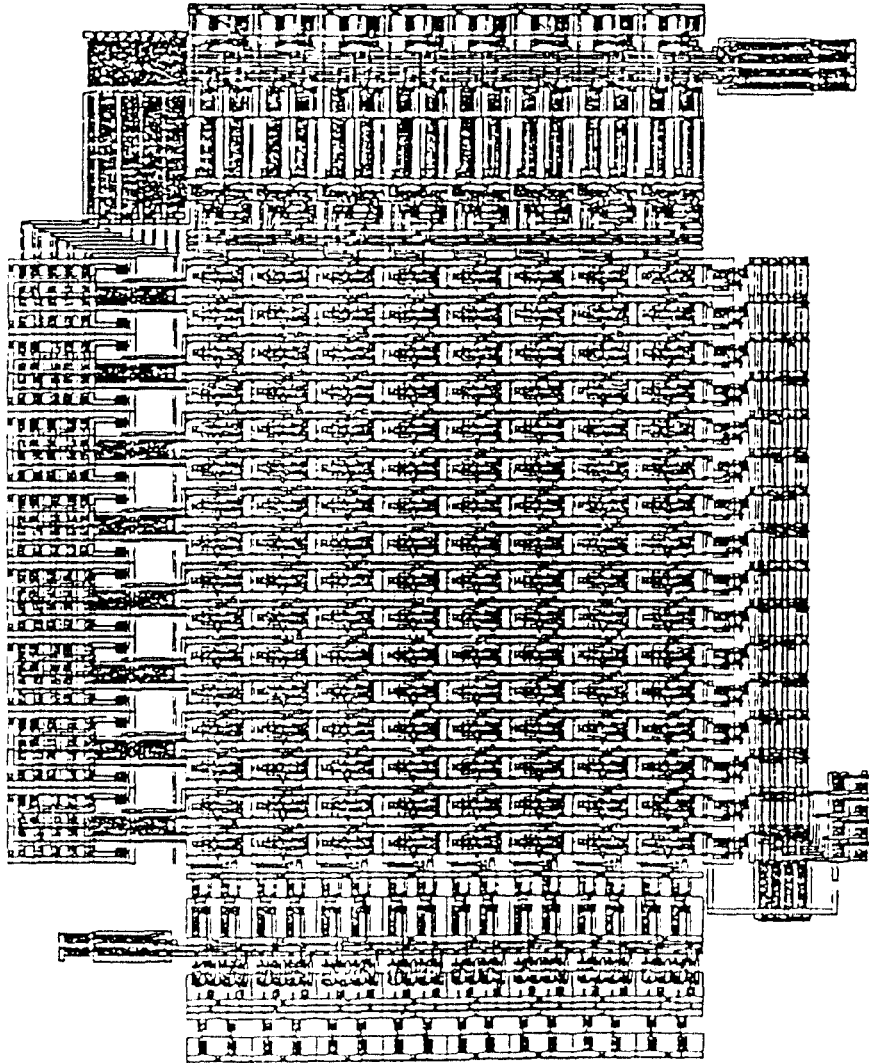


그림 3. CAM 메모리이다메지 생성의 구조 (8 bit × 16 word)

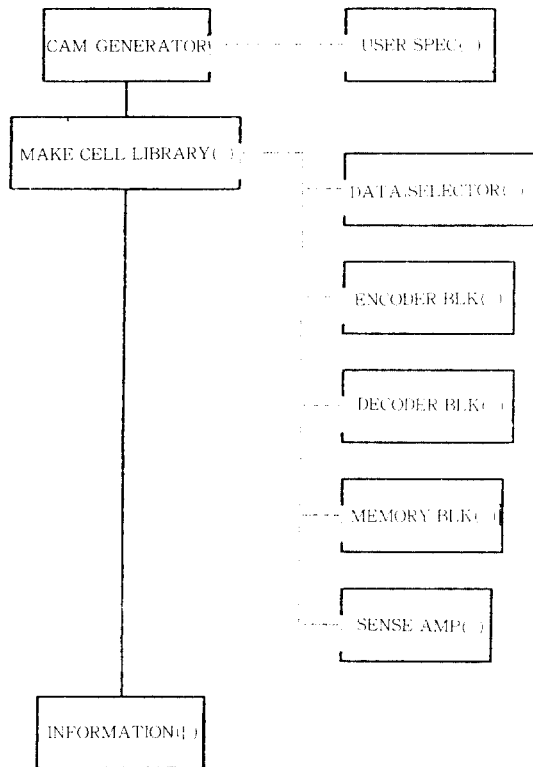


그림 4. 연상 메모리 제너레이터를 구성하는 함수 블록도

하여 만들었으며 셀 라이브러리는 C 언어의 함수들의 집합으로 사상(mapping) 시키 구성하였다.<sup>(8)(9)</sup>

### 3.1 CIF 화일 구조

CIF는 집적회로를 설계할때 필요한 도형정보들을 표현하는 수단으로서 비데오 디스플레이, 플롯터, PG(pattern generation)장치등의 표준 입력 형태로써 사용되고 있으며 다른 심볼릭 레이아웃 언어나 설계 프로그램에 의하여 생성되어지는 중간 형태의 코드로서, 데이터가 계층적인 구조를 갖고 있으므로 집적회로 설계와 검증 및 수정이 편리한 코드이다.

### 3.2 CIF 파서(parser)와 C 코드와의 연결

표 1. 주요 함수명 및 기능

함수명	함수별 기능 설명
USER SPEC( )	입력 파라미터를 처리하는 함수
MAKE-CELL LIBRARY( )	CIF 문법으로 변환하는 함수
DATA-SELECTOR( )	데이터 셀렉터 블록을 생성하는 함수
MEMORY BLK( )	메모리 셀 어레이 블록을 생성하는 함수
DECODER BLK( )	어드레스 디코더 블록을 생성하는 함수
ENCODER BLK( )	어드레스 인코더 블록을 생성하는 함수
SENSE-AMP( )	감지 증폭기 블록을 생성하는 함수
INFORMATION( )	집계한 연상메모리칩의 블록 크기와 전체 칩 크기등을 출력시키는 함수

Niklaus Wirth가 제안한 CIF 문법을 YACC 로써 기술하고, 이를 뒷바처 주기 위해서 어후 분석기인 LEX를 사용하였으며 MS-DOS 상에서 수행하기 위해 함수 yywrap( )를 첨가시켰다. CIF 변환 프로그램의 입력으로 CIF 데이터를 받아 들어 CIF 명령어에 맞게 C 언어로 변환하며 셀을 호출하거나 그림을 그리는 부분은 함수 main( )에 등록하고 셀들은 정의된 심볼 이름에 따라서 각각의 함수로 만든다. CIF 명령어를 처리하는데 있어서 도형은 BOX 만으로 정의하였으며 기타 명령어들의 C 언어 변환은 표 2와 같다.

그리고, CIF 명령어의 C 언어 변환중에서 회전변환인 경우는 다음과 같으며, 회전 변환 후 포인트 벡터는 벡터 X 매트릭스로 표시된다.

$$X' = \cos \phi \cdot X - \sin \phi \cdot Y$$

$$Y' = \sin \phi \cdot X + \cos \phi \cdot Y$$

단, X'Y' : 회전후 좌표

X Y : 회전전 좌표

$\phi$  : 회전각

표 2. C 변수 함수명 및 기능

CIF 명령어	C 변환 함수명 및 기능
BOX	draw-box(layer,length,width,cx,cy,dx,y) layer는 각 layer의 화인포인트이고, 도형의 정보를 받아 transformation 정보와 통합하여 새로운 box를 layer 화일에 출력한다.
CALL	define start command 규칙이 감촉되었을때 셀 번호를 받아 셀 함수로 만들고 셀의 정보를 계속해서 기술해 나간다.
TRANSFORMANTION	trans( ), mirrorx( ), mirrory( ), rotate( ) 는 CIF의 transformation, mirror, rotate 명령어가 감촉될 때 위의 함수들이 생성되고, 좌표들의 변환은 매트릭스 곱셈함수 mtxXmtx ( )에 의해 이루어진다.

$$R a b \rightarrow T_n = \begin{pmatrix} a/c & b/c & 0 \\ -b/c & a/c & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad a,b: X,Y \text{ 좌표값} \\ c = (a^2 + b^2)^{1/2}$$

위의 절차를 수행하는 함수를 rotate(a,b,tmtx)라 할때 회전 변환 과정은 다음과 같다.

```
copymtx(stdmtx, tmtx); /* Copy unit matrix */
rotate(a,b,tmtx); /* Rotation */
mtxXmtx(tmtx, oldmtx, newmtx); /* Final matrix produce */
CELL 100(newmtx); /* Cell drawing */
```

예로 회전 변환 (R a b) 경우, 회전 변환 매트릭스 T<sub>n</sub>은 아래와 같다.

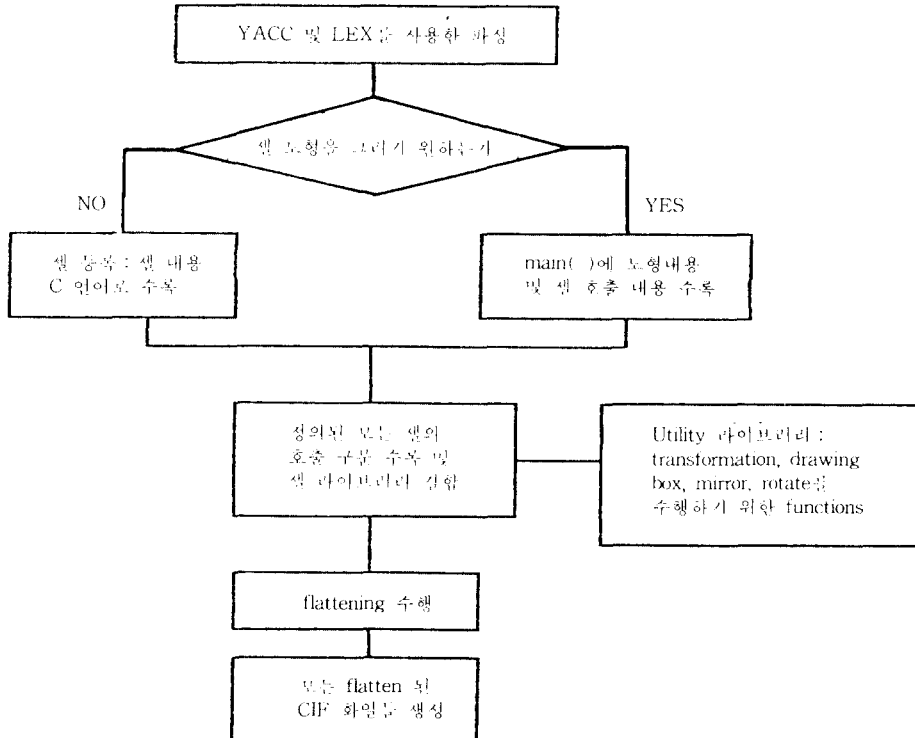


그림 5. Flattener 프로그램 흐름도

CIF flattener에서는 CIF에 등록되어 있는 모든 셀들을 각 화일에 저장하여 그래픽 디스플레이 프로그램과 플롯팅 프로그램의 입력 데이터로 사용한다. CIF flattener 프로그램의 전체적인 흐름도는 그림 5와 같다.

#### IV. 그래픽 디스플레이 프로그램

현재 이용되고 있는 CAD 장비에서 널리 사용되는 도형 정보 형태의 하나로 CIF 형태를 들수 있으며 CIF를 출력하는 프로그램들중에는 CAM 제너레이터, PLA 제너레이터 그리고 ROM 제너레이터등이 있다.

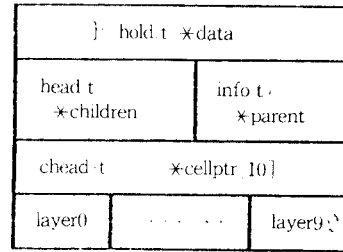
위의 제너레이터 프로그램을 개발하는 과정에서 출력 데이터를 컴퓨터 화면에 디스플레이 시켜본다든지 또는 레이아웃 정보가 정확한지를 빠른 시간내에 셀 별, 블록 별 그리고 전체 레이아웃 도면을 확인하기 위한 프로그램이 필요하게 되어 IBM / PC AT에서 MS-C(5.0)에서 제공하는 그래픽 라이브러리를 이용하여 C 언어로 그래픽 디스플레이 프로그램을 개발하였다.

##### 4.1 그래픽 디스플레이의 화면 메뉴 구조

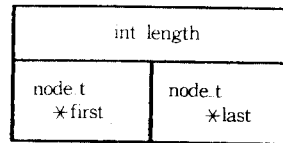
IBM PC / AT의 비데오 모드가 640×350 pixels 로 되어있어 행과 열의 크기가 약 1.8 : 1로써 레이아웃 도면이 일대일 대칭되어 컴퓨터 화면에 디스플레이 되지 않아 활용 화면 크기를 일대일로 대칭되게 조정하였다. 화면메뉴의 구성은 화일 선택 메뉴와 Zoom 선택 메뉴로 되어 있다.

##### 4.2 그래픽 디스플레이 프로그램의 구조

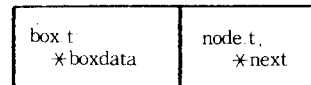
그래픽 디스플레이 프로그램 사용 방법은 컴퓨터 화면상에 디스플레이된 main 메뉴에서 User 가 등록된 화일 목록을 보고자할 때에는 알파벳 "F(f)"를 입력하면 화일 목록이 화면에 디스플레이된다. 이때 선택할 셀 이름을 입력시키면 셀이름에 따른 도형 정보에 관한 포인터를 해쉬 테이블(hashtable)에서 찾아 return 시키고 없으면 셀이름의 화일로부터 CIF 데이터를 받아



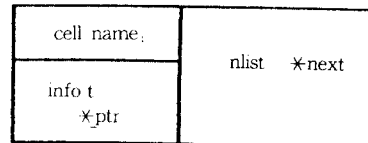
(a) 셀을 위한 node의 데이터 구조 (info-t)



(b) list를 위한 head의 데이터 구조 (head-t)



(c) box의 list를 위한 데이터 구조 (node-t)



(d) 해쉬 테이블을 위한 데이터 구조

그림 6. 셀과 도형 정보를 위한 각 node의 데이터 구조

그림 6과 같은 데이터 구조로 정보를 등록하며 그 포인터를 해쉬 테이블에 등록하고 호출했던 곳으로 포인터를 리턴시켜 CIF 데이터를 디스플레이 시킨다. 여기서 CIF 데이터의 파싱 루틴 (parsing routine)은 CIF flattener의 파서를 사용하고 MS C(5.0) 그래픽의 제한성때문에 각 layer 별로 linked list 구조에 도형 정보를 구축하였다. 또한 그래픽 디스플레이 프로그램을 구성하고 있는 함수들의 관계는 그림 7과 같다.

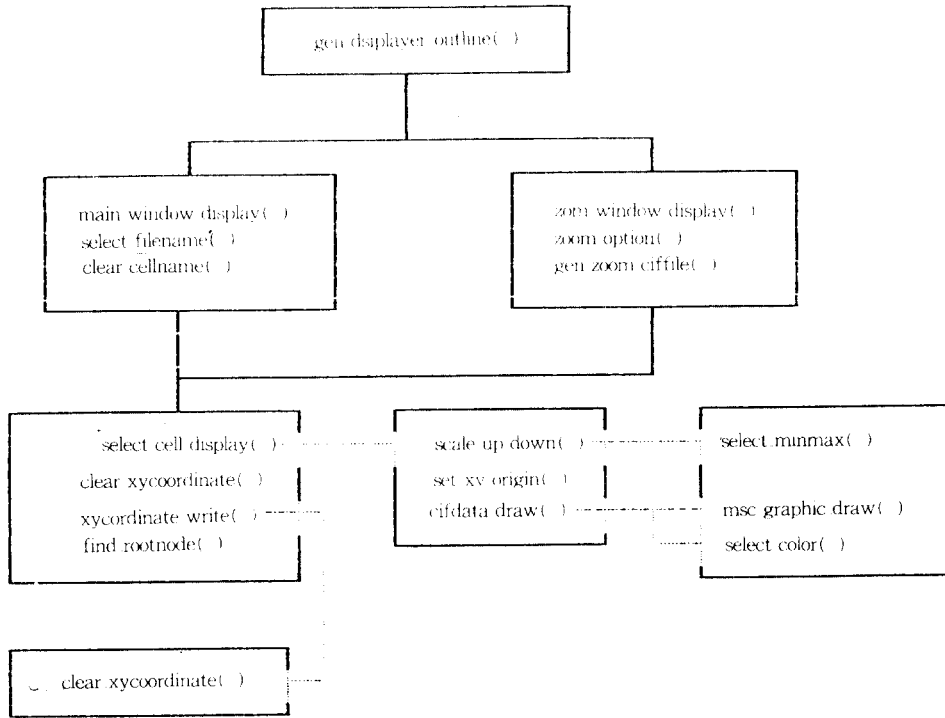


그림 7. 그래픽 디스플레이 프로그램 함수들의 상관 관계

### V. CIF의 DXY 명령어 변환 프로그램

CAM 제너레이터의 CIF 출력 데이터를 플롯터에 그리기 위해서 널리 사용되고 있는 ROL AND X-Y 플롯터를 구동시키는 DXY 명령어 변환프로그램을 개발하였다.

#### 5.1 DXY 명령어 구조

DXY 명령어는 소문자나 대문자 모두를 입력으로 사용할 수 있으며 그래픽 구조를 제어하는 DXY 명령어의 파라미터는 한 문자 또는 두 문자로 구성되어 있다. 표 3은 사용한 DXY 명령어 종류와 기능에 대해서 기술하였다.

#### 5.2 CIF의 DXY 명령어 변환 프로그램의 구성

CIF의 DXY 명령어 변환 프로그램 동작은

표 3. DXY 명령어의 형태와 기능

명령어	형태	기능
"D" Draw	D x1,y1,...xn,yn	절대 좌표계를 이용하여 선을 그리기 기능
"M" Move	M x,Y	절대 좌표계 내에서 펜을 이동하는 기능
"I" Relative Draw	I x1,y1,...xn,yn	상대 좌표계에서 선을 이동하는 기능
"P" Pmt	P c1,c2,...cn	ASCII 코드 문자와 심볼을 그리기 기능
"J" Pen Change	Jn	지정한 펜을 선택하는 기능

플롯팅 크기를 결정하는 크기 인수를 파라미터로 입력받고 그리고자하는 셀의 포인터를 해쉬 테이블에서 찾아 그 포인터로부터 각 layer 별로 리스트를 운행하여 순차적으로 CIF 명령어를



표 4. CIF 명령어와 DXY 명령어의 변환 관계

CIF 명령어	DXY 명령어
Layer specification	"J" Pen change
Box length, width, center, direction	"M" Move "I" Relative Draw "B" Line Scale "L" Line Type "D" Draw
Comments arbitrary text	"M" Move "P" Print "S" Alpha Scale
End Marker	"H" Home

DXY 명령어로 변환한다. 표 4은 CIF 명령어와 DXY 명령어 사이의 변환 관계를 나타내고 있으며, 그림 8은 CIF의 DXY 명령어 변환 프로그램의 흐름도를 보여주고 있다.

### VI. 결 론

컴퓨터를 이용하여 연상 메모리 칩을 자동으로 설계하기 위한 프로그램들을 하나의 소프트웨어

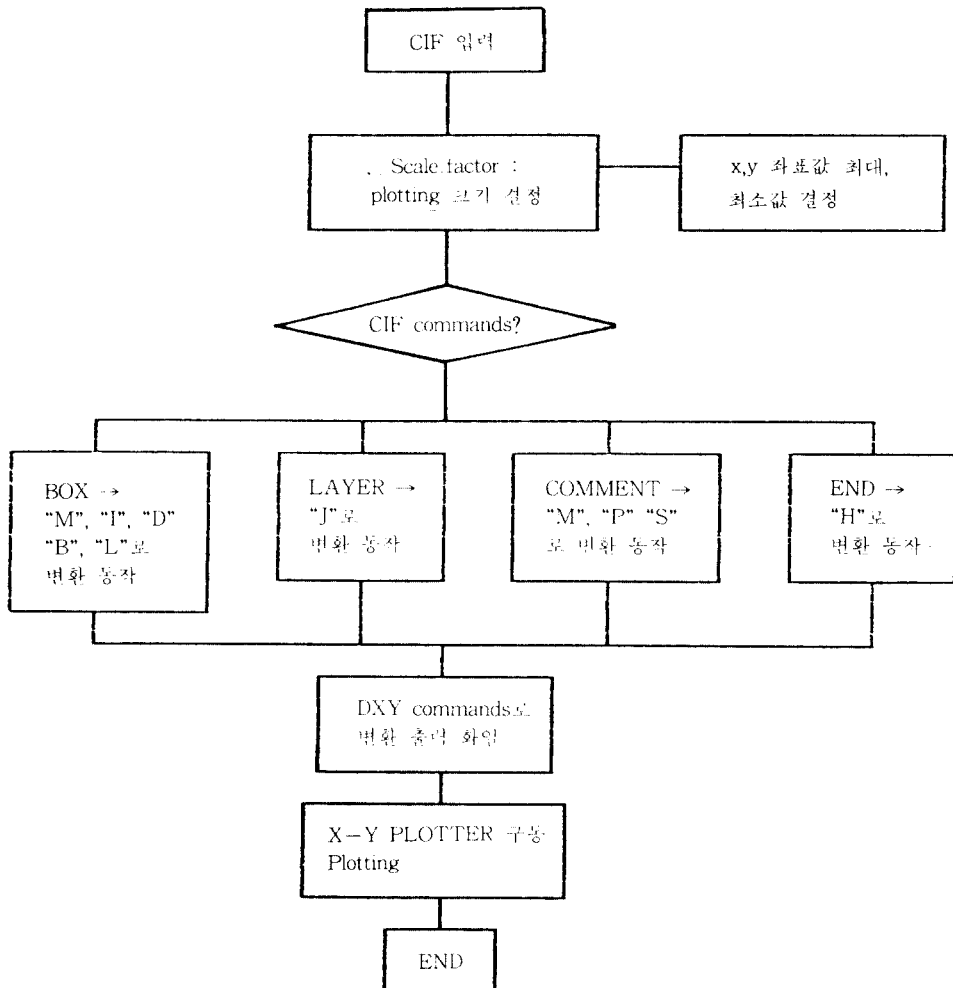


그림 8. CIF의 DXY 명령어 변환 프로그램의 흐름도

tool로 개발했다. 회로의 집적도가 증가함에 따라 회로 설계 시간을 단축하고 회로의 신뢰도를 높이기 위한 방법으로 컴퓨터를 이용한 회로 설계 방식이 보편화되고 있는 현실에서 PC / AT를 이용하여 연상 메모리의 레이아웃 정보가 CIF를 자동 생성되는 프로그램, CIF 데이터를 primitive CIF로 flatten하는 CIF 파서 프로그램, 화면에 설계한 디바이스의 레이아웃 도면을 디스플레이하는 프로그램, 그리고 디바이스 레이아웃 도면의 크기를 조절하여 플롯터에 플롯팅하는 프로그램등 일련의 프로그램들을 PC 상에서 개발하였다.

앞으로 개발된 프로그램을 연상 메모리 소자에 대한 전용 모듈 컴파일러로 확장하는 연구가 기대된다.

### 참 고 문 헌

1. Hiroshi Kactota, "An 8 kbit content Addressable and Reentrant memory" IEEE J. of solid state circuits, vol.SC-20, No.5, pp.951-957, October 1985.
2. Tactanobu Nikaido "A 1k bits Associative Memory LSI", Proceeding of the 11th Conference on Solid State Devices, Japanese Journal of Applied Physics, pp.51-54, vol 22(1983) Supplement 22 I.

3. Takeshi Ogura, "A 20-kbits Associative Memory LSI for Artificial Intelligence Machines", IEEE J. of solid-state circuits, vol. 24, No. 4, pp 1014-1020 August 1989.
4. T.Kohonen, "Content-Addressable Memories", Berlin, Heidelberg, New York : Springer-verlag 1980.
5. 오창준외, "C-言語를 이용한 CMOS ROM 제너레이터의開發", 87 特定 研究 結果 發表會 論文集, pp 99-102, 1988年3月
6. 車均鉉, 朴魯京外 "CMOS PLA 自動 Layout Program 開發에 關한 研究" 大韓 電子 工學會, pp 887-895, 87 24-5 22
7. Alfred V. Aho & Jeffrey D.Ullman, "Principle of Compiler Design", Addison Welsley, Chap 4,5 1977.
8. Xenix Software Development System, "Software Development Guide", Zeus Computer.
9. Programmer's Manual For UNIX System III, vol 2 B, October, section 21, 1981.
10. Steven M. Rubin, "Computer Aids for VLSI Design", Addison Wesley, 1987.
11. 車均鉉, 朴魯京外, "Content Addressable Memory Generator", ICVC 89, Seoul, pp. 348-351, October.
12. Stephen M. Trimberger, "An Introduction to CAD for VLSI", Kluwer Academic, Chap 6 7. 1987.



金 鍾 善 (Jong Seon KIM) 正會員  
1964年 8月10日生  
1988年 2月 : 고려대학교 전자전산공학과 졸업  
1990年 2月 : 고려대학교 대학원 전자공학과 졸업  
1990年 1月 : 삼성전자(주) 부진연구소  
現在 : 상호전라성제 2팀 연구원



白 寅 天 (In Cheon PAIK) 正會員  
1963年 1月14日生  
1985年 : 高麗大學校 電子工學科 卒業 (學士)  
1987年 : 高麗大學校 大學院 電子工學科 卒業 (碩士)  
1987年 ~ 現在 : 高麗大學校 大學院 電子工學科 博士課程



朴 晝 京(Nho Kyeong PARK) 正會員  
1958年 1月 8日生  
1984年 2月：高麗大學校 電子電算工學  
科 卒業  
1986年 2月：高麗大學校 工學碩士學位  
取得  
1989年 2月：高麗大學校 博士課程 修了  
※主關心分野는 ISDN 분야, 回路 및 시  
스템 設計等임.



車 均 鉉(Kyun Hyon TCHAH) 正會員  
1939年 3月 26日生  
1965年：서울大學校 工學士  
1967年：美國伊利노이大學校 工學碩士  
學位 取得  
1976年：서울大學校 工學博士學位 取得  
1987年～現在：高麗大學校 電子電算工  
學科 教授  
※主關心分野는 CAD 및 通信시스템等.