

회로도면과 IC Mask Layout을 위한 2차원

Graphic Editor

(SEAS (Symbolic Editing and Design Aid System))

趙 在 周*, 車 建 業*, 金 正 淳*

(Jae-Joo Jo, Gun-Up Cha and Jung-Soon Kim)

要 約

SEAS는 회로 도면과 IC mask layout 설계를 위한 2차원 graphic editor이다. 이 system은 범용 computer와 Tektronix 4110 계열 graphic terminal 상에서 수행된다. 이 system을 사용하여, 사용자는 회로도면 및 IC mask layout을 2단계 계층적으로 설계할 수 있다. 20개 이상의 built-in symbol들, 사용자가 만들어 줄 수 있는 internal symbol들, 60개의 macro와 16개의 function key 및 4단계의 on-screen menu operation을 제공하며, 도면 수정중에 nesting된 cell을 수정할 수 있다(edit-in-place). 또한 단일 element 및 element group의 reflection, rotation, move, copy뿐만 아니라, 다각형의 형태 변경등의 다양한 수정 기능도 제공한다. 수행 속도를 증대시키기 위하여 graphic terminal의 local memory를 사용하였으며, computer system간의 호환성을 주기 위하여 FORTRAN77 프로그래밍 언어로 coding하였다.

Abstract

This paper describes the SEAS which is a 2-dimensional graphic editor for schematic circuit and IC mask layout. This system runs on general purpose computer and Tektronix 4110 series graphic terminal. With this system, user can edit schematic circuits and/or IC mask layout with 2 level hierarchy. This system supports more than 20 kinds of built-in symbol, user definable internal symbols, 60 macros, 16 function keys, 4 level on-screen menu operation and edit-in-place of cell in main drawing. And it provides editing functions such as reflection, rotation, move and copy of one or a group of elements, and modification of polygona geometry etc. To improve the exccution speed, we used local memory of graphic terminal. For the portability of the program, the system is written in FORTRAN 77 programming language.

I. 서 론

기존의 turn-key 방식의 IC mask layout용 graphic editor는 Calma, Applicon등 다수의 system들이 상품화되어 판매되고 있고, 범용 computer에서 수행되는

graphic editor로는 Silvar-Lisco사의 SDS와 PRINCESS, Berkley 대학의 KIC 등이 상품화되어 있다. 그러나 이들 software들은 각각 IC mask layout전용, 혹은 schematic diagram 전용 등으로 분리되어 있고, 또 고가이며 특정 hardware에서만 수행이 가능하기 때문에, 소규모의 design center나 학교 등에서는 거의 사용할 수 없었다. 또한 source code가 제공되지 않기 때문에 사용자가 원하는 기능 추가등의 system

*正會員, (株)金星社 中央研究所
(Central Research Lab. GoldStar Company., Ltd.)
接受日字: 1986年 4月 1日

upgrade가 vender들에 의해서만 가능했다. 이러한 것 들 때문에 최근에 국내에서도 이에 대한 자체적인 연 구가 학교 또는 기업체에서 진행되고 있다. 당사에서 도 회로 및 IC mask layout겸용 graphic editor를 개 발하게 되었다.

범용 computer와 Tektronix 4110 계열 graphic terminal상에서 수행되는 graphic editor SEAS(Symbolic Editing and design Aid System)는 회로 및 IC mask layout 설계자를 위하여 설계되었다.

회로도를 설계하는 데는 특정 소자를 나타내거나 사 용자에 의해 특정 의미가 부여될 수 있는 기호(symbol) 및 직선 등이 필요하며, IC mask layout을 설계하는 데는 어떤 기하학적 영역을 정의할 수 있는 다각형이 필요하게 된다. 또한 회로도나 IC mask layout을 설 계하는 데는 특정한 도면을 빌어다 다른 도면에서 쓸 수 있는 것(cell reference)과 보조적인 것으로서 text를 빼 놓을 수 없다. SEAS에서는 상기한 symbol, cell, 다각형 도형들과 text 등을 element라고 부른다.

SEAS는 단일 element나 element group을 반전 (reflection), 회전(rotation), 이동(move), 복사(copy), 삭제(deletion)할 수 있는 기능 및 다각형 도형의 일 부분 변경, 가장 최근에 입력된 data의 삭제(undo), symbol에 자동적으로 번호를 부여하는 기능을 제공한다. 또한 20개 이상의 built-in symbol, 사용자가 지정 하는 symbol(internal symbol) 정의 기능, 수정중인 도면 내에서 nesting된 cell을 수정할 수 있는 edit-in-place라는 기능도 가지고 있다. 그리고 60개의 macro 들, 16개의 function key들, data의 입력과 수정을 용 이하게 하기 위한 다양한 status, 4 단계의 on-screen menu operation, viewable window 제어 및 Calma GDS II^[1]와의 graphic data interface, help utility 등 의 system utility를 가지고 있다.

SEAS는 VAX/VMS OS, 금성사 중앙연구소에서 개 발된 Tektronix terminal제어용 S/W인 UTCS^[2] 및 Calcomp 960, 965를 제어하는 Calcomp사의 HCBS를^[3] 이용하여 개발 되었으며, source code는 대부분의 computer system에서 이용 가능한 FORTRAN 77 프 로그램밍 언어로^[4] 되어있고, 자체 개발되었기 때문에 유지, 보수 및 system 확장성이 우수하다.

본 논문에서는 SEAS의 전반적인 기능과 구성, 그 리고 구현방법을 다루었다.

II. SEAS의 전반적인 구성

SEAS는 system 초기화, user interface, status 지 정 및 report, viewable screen window제어, data

update, macro와 function key제어, 그리고 system utility module로 구성되어 있으며, 이들 사이의 data 의 흐름과^[5] control의 흐름은 그림 1에서 보는 바와 같다.

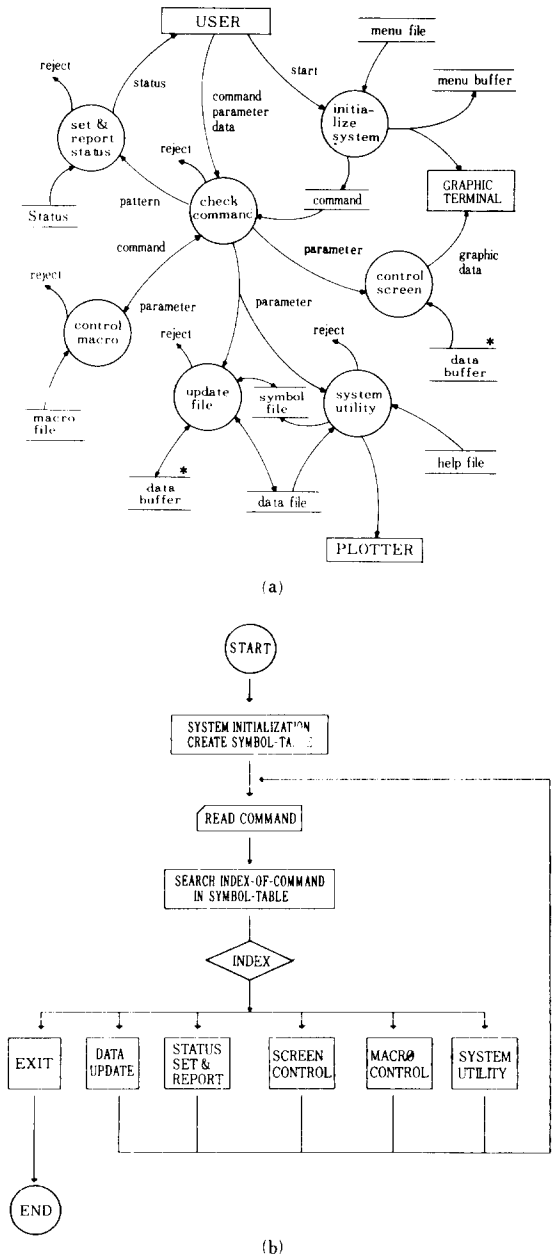


그림 1. SEAS 각 module사이의 관계
 (a) SEAS 각 module 사이의 data flow
 (b) SEAS 각 module 사이의 control flow
 Fig. 1. Relationship between SEAS modules.
 (a) Data flow between SEAS modules.
 (b) Control flow between SEAS modules.

Terminal을 초기화한 후에, system은 command와 parameter를 받아서 그 command가 갖는 기능을 수행하는 routine의 jump address를 가져온다(user interface module). Command에 따라서 system의 수행을 멈추거나 (EXIT), data를 update(data update module), macro와 function key 제어(macro와 function key제어 module), status를 지정하거나 report(status 지정 및 report module), viewable screen window의 제어(screen 제어 module)와 system utility의 제공(system utility module)을 위한 기능을 수행한다.

SEAS에서 사용되는 file은 도면 data를 저장하기 위한 도면 file, internal symbol을 저장하기 위한 symbol file과 macro를 저장하기 위한 macro file의 3 종류가 있다.

도면 file의 data는 2 단계 계층 구조를 가지며, 이것은 IC mask layout이나 회로 설계시, 계층적인 설계를 가능하게 하고 recursive call의 가능성을 배제할 수 있으며 memory space 및 수행 시간을 줄인다. 그림 2는 SEAS database의 계층구조를¹⁴⁾ 나타낸 것으로서, 어떤 한 도면은 기본적인 element, subcell(cell에 nesting된 도면)을 갖지 않는 cell과 사용자가 정의한 internal symbol을 nesting할 수 있음을 보여주고 있으며, 각 internal symbol은 recursive call이 안되는 범위에서 다른 internal symbol을 nesting할 수 있다. 따라서 cell nesting에 의한 display의 2단계 계층 구조와 internal symbol nesting에 의한 data의 다단계 계층구조를 실현한다. 그림 3은 system수행 시의 내부 buffer의 구성을 나타낸 것으로서, 도면의 검색 및 cell nesting의 효율적인 처리를 위한 계층적인 성격의 도면 level buffer의 구성(그림3(a),(b))과 다수의 primary search key를¹⁴⁾ 사용하여 효과적인 수정 작업을 수행하기 위한 element level buffer의 구성(그림 3 - (c), (d), (e))을 나타낸다.

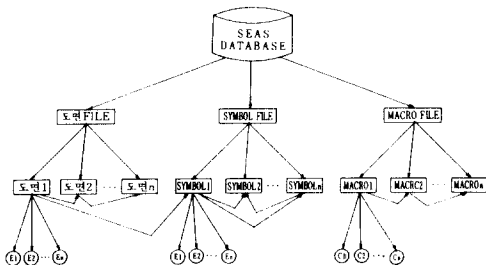


그림 2. SEAS database 구성
Fig. 2. SEAS database structure.

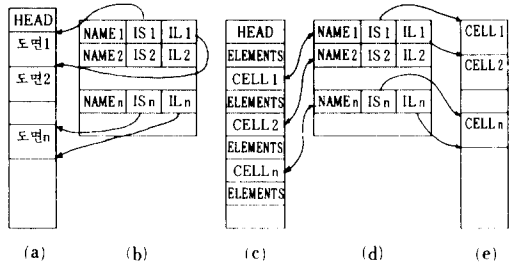


그림 3. 내부 buffer 구성
(a) 도면 file (b) 도면 list
(c) 도면 buffer (d) Cell list
(e) Cell buffer
Fig. 3. Internal buffer structure.
(a) Drawing file. (b) Drawing list.
(c) Drawing buffer. (d) Cell list.
(e) Cell buffer.

Internal symbol file과 macro file은 효율적인 처리를 위하여 direct access file로¹⁴⁾ 설치하며, 이들 data를 효과적으로 관리하며 요구된 것을 용이하게 찾기 위해서 symbol file에는 symbol list와 symbol index를 설치해 두어 사용하며, macro file에는 처음 3개의 record에 배정된 macro의 갯수와 각 macro의 위치를 저장해 둔다.

III. SEAS의 각 Module들

1. System 초기화 Module

System 초기화 module에서는 사용자가 system을 사용할 수 있도록 준비를 해 주어야 하는데, SEAS에서 해 주는 초기화는 command인식 routine의 구성, graphic terminal의 초기화, menu operation을¹⁷⁾ 위한 준비와 status 초기화가 있다.

Interactive system의 수행에 있어서 가장 필수적인 부분은 command 인식 routine으로서, 이 routine을 어떻게 구성하느냐에 따라 system의 처리속도와 편리성에 큰 영향을 미친다. 가장 널리 사용되는 command 인식 routine으로는 hashing에 의한 방법,¹⁸⁾ finite automata에 의한 방법,¹⁹⁾ binary search tree에 의한 방법¹⁸⁾등 다수가 있으나, SEAS에서는 binary search tree로 command 인식 routine을 구성하여 명령어의 변형, 삽입 및 삭제 등을 단지 buffer내의 명령어 및 traverse address를 변형시킴으로서 routine의 조직을 바꾸지 않고도 가능하게 하며, 또 빠르게 수행될 수 있도록 하였다. 그림 4는 binary search tree 구성의 일부를 나타낸 것이다.

Menu의 사용과 변형을 쉽게 하기 위해서 menu

jump address가 8일 경우는 built-in command가 아니며, 따라서 macro routine으로 jump한다. 그림 6은 user interface module의 control 흐름도이다.

User interface module을 위와 같이 설치할 경우 내부 routine이 다소 복잡해지는 단점은 있으나 사용자는 한꺼번에 command와 parameter들을 쉽게 입력할 수 있으며, 다수의 command를 하나로 묶어서 사용할 수 있는 장점이 있다.

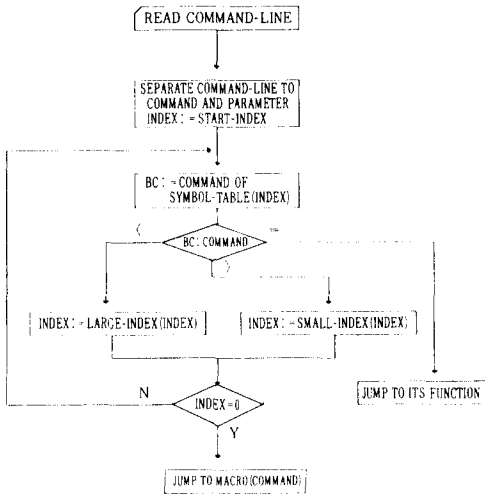


그림 6. User interface module의 control 흐름
Fig. 6. Control flow of user interface module.

3. Status지정 및 Report Module

Command나 parameter의 입력을 용이하도록 하기 위해서, 사용자가 다수의 status 및 element attribute를 지정하며, 지정된 status를 보고받을 수 있도록 하여 사용자가 data를 입력할 때마다 attribute를 지정하는 불편을 제거하였다. 표 1은 global data로 선언된 element attribute를 나열하였다.

또 각 buffer의 상태와 현재 수정중인 도면 이름 등을 global data로 처리하며, 각각의 도면 이름과 그도면이 차지하고 있는 buffer의 주소 등을 갖는 buffer를 따로 설치하여서 현재의 상황report 및도면 listing을 빠르게 해 주도록 하였다.

4. Screen 제어 Module

Screen window를 global data로 저장하며, data의 display 범위를 도면의 head에 저장해 두고 사용하여, terminal local memory의 data 저장부에 전 도면을 기록해서 사용함으로써 도면의 전체 또는 일부분의 dis-

표 1. 각 Element가 가지고 있는 성질들
Table 1. Element attribute.

| Element | |
|-------------|--|
| Common | Device code, layer code, reference coordinate |
| Passive E. | Device subtype, numbering, value |
| Active E. | Device subtype, numbering |
| Logic Gate | Device subtype, numbering, fanin or fanout number |
| I/O signal | Device subtype, numbering |
| Cell | Cell name, reflection axis, rotation angle, magnitude |
| Text string | Concatenation code, magnitude, font |
| Line type | Concatenation code, line style, input mode, point number, path width |
| Arc | Radius, start and terminate angle |
| Circle | Radius |

play 및 viewable screen window 이동 등을 전 data를 다시 그려주지 않고 local memory의 내용을 확대, 축소 또는 이동시켜서 빠르게 display시켜 주며, 그로 인해 host에서 local로, 혹은 그 역으로 전송되는 data의 양을 최소화하였다. 그러나 terminal local memory의 한계성으로 인해서 4000개 이상의 element를 갖는 도면에 대해서는 전 도면을 local memory에 저장할 수 없으므로 일부는 local memory에서, 일부는 host에서 처리될 수 있도록 하여서 대형 도면도 설계할 수 있도록 하였다. 그림 7은 screen제어 module의 data 흐름도이다.

5. Data Update Module

SEAS에서의 data update 기능은 도면 file을 제어하는 file 제어, data의 입력, 수정을 위한 data의 조건

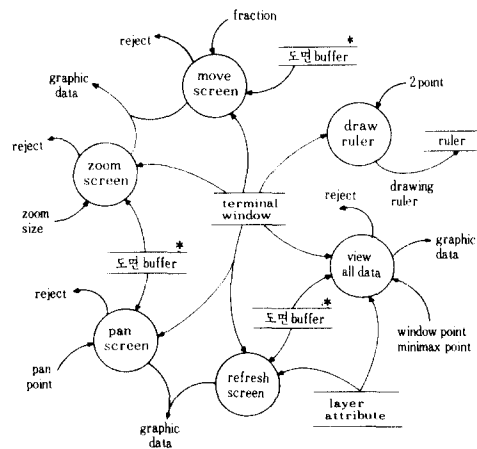


그림 7. Screen 제어 Module의 Data 흐름
Fig. 7. Data Flow of Screen Control Module

부 선택 및 조건부 변경, 수정중인 도면내에서의 nesting된 cell 수정, symbol의 자동적인 numbering등이 있다.

1) File 제어

Data가 입력될 때마다 disc에 저장하지 않고 내부 buffer에 넣음으로서 순간 정전이나 computer system의 down 등으로 인해 작업 결과가 무시되는 단점은 있으나, 부분적인 data저장 시간을 없애고 지금까지의 작업이 오류였을 경우, 그 작업 결과를 무시하고 전 version으로의 회복이 가능하다는 장점이 있으며, 앞에서 기술된 단점을 보완하기 위해서 system수행중에 사용자가 필요하다고 느낄 때마다 현 상태를 바꾸지 않고 disc에 저장할 수 있도록 하였다.

File을 open할때, 도면 list를 만들어 주고 이것을 도면 이름 별로 sort해 둬므로서 binary search에 의한 검색을 가능하게 하며, 도면의 삭제, 복사, 이름 변경등을 도면 list에서 처리하여 줌으로서 수정 작업에 필수적인 file 제어를 빠르게 처리하도록 하였다.

도면의 head에 element의 갯수, 도면 이름 및 도면의 최대와 최소 좌표를 가지고 있어서, 수정을 위한 도면의 open을 record의 검색이나 도면의 최대, 최소 좌표의 검색없이 진행함으로써 응답시간을 줄여 주며, record의 갯수 및 도면의 최대, 최소 좌표의 계산은 도면 수정의 종료에 부담시킴으로써 수정을 시작할 때 보다는 끝날 때 많은 시간을 소모하도록 하여서 사용자가 느끼는 지루함을 최소화하였다.

2) Data의 선택

Data 선택의 일반적인 범주는 하나의 element를 선택하는 경우와 element의 group을 선택하는 경우의 2가지가 있다. 그러나 일반적인 도면에서는 선택을 원치 않는 data가 곳곳에 산재해 있다. 따라서 선택하지 않을 data의 구별과 IC 개발용 mask를 구별 제작하기 위해 layer의 개념이 필요하게 된다. SEAS에서는 64개의 layer를 사용하며, 이들 각각의 layer에 대해 attribute를 부여하여 선택가능 layer와 그렇지 않은 layer를 구별하여 사용한다.

하나의 element를 선택할 경우는 먼저 terminal의 local function으로 element를 선택하고, 만일 local function으로 element가 선택되지 않았을 경우에는 모든 symbol들을 그들의 type에 따라 직선으로 단순화시켜서, 입력된 기준점까지의 최소 거리를 갖는 직선을 선택한 후, 그것을 원래의 type으로 복귀시킨다. 그림 8은 element 선택을 위한 도면의 단순화를 나타낸다.

그러나 element group의 선택일 경우는 40점 이하로 주어지는 다각형 내에 있는 element중 선택가능

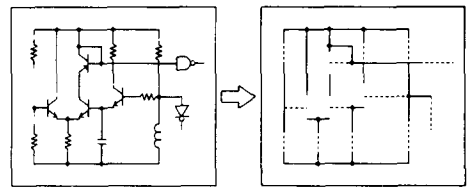


그림 8. Data 선택을 위한 도면의 단순화
Fig. 8. Drawing simplification for data selection.

layer를 갖는 element만을 group 수정 buffer에 넣는다. 내부 element 인식 algorithm은 angle합에 의한 인식과 교차점 수에 의한 인식의 2가지 경우가 있으나,¹¹⁾ 본 system에서는 전자를 사용하여 내부수행 step수를 줄였다. 그림 9는 내부 element인식 algorithm을 그림으로 표현한 것이다.

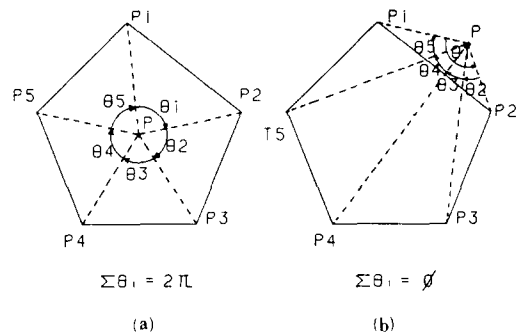


그림 9. 내각의 합에 의한 내부 element 인식 algorithm

- (a) P가 내부에 있는 경우
- (b) P가 외부에 있는 경우

Fig. 9. Containment test by sum of angle.
(a) P is inside. (b) P is outside.

3) Data의 변형

Host computer에 부담을 주지 않고 빠른 속도로 data의 변형을 진행하기 위해서 변형될 data를 선택할 때, terminal local memory의 한 부분에 선택된 data를 저장해 두고 data의 변형 시마다 해당부분의 local memory의 data attribute를 바꾸어 display시킨다. 그러나 다각형 형태의 도형의 형태 변형과 같이 한 element의 일부분만 변형시키는 것은 그 element에서 입력된 점에 가장 가까운 직선을 선택한 후에 그 직선을 변형시키고 인접한 직선들을 변형된 직선에 연결한다. 그림10은 수정 기능의 예를 보인 것이다.

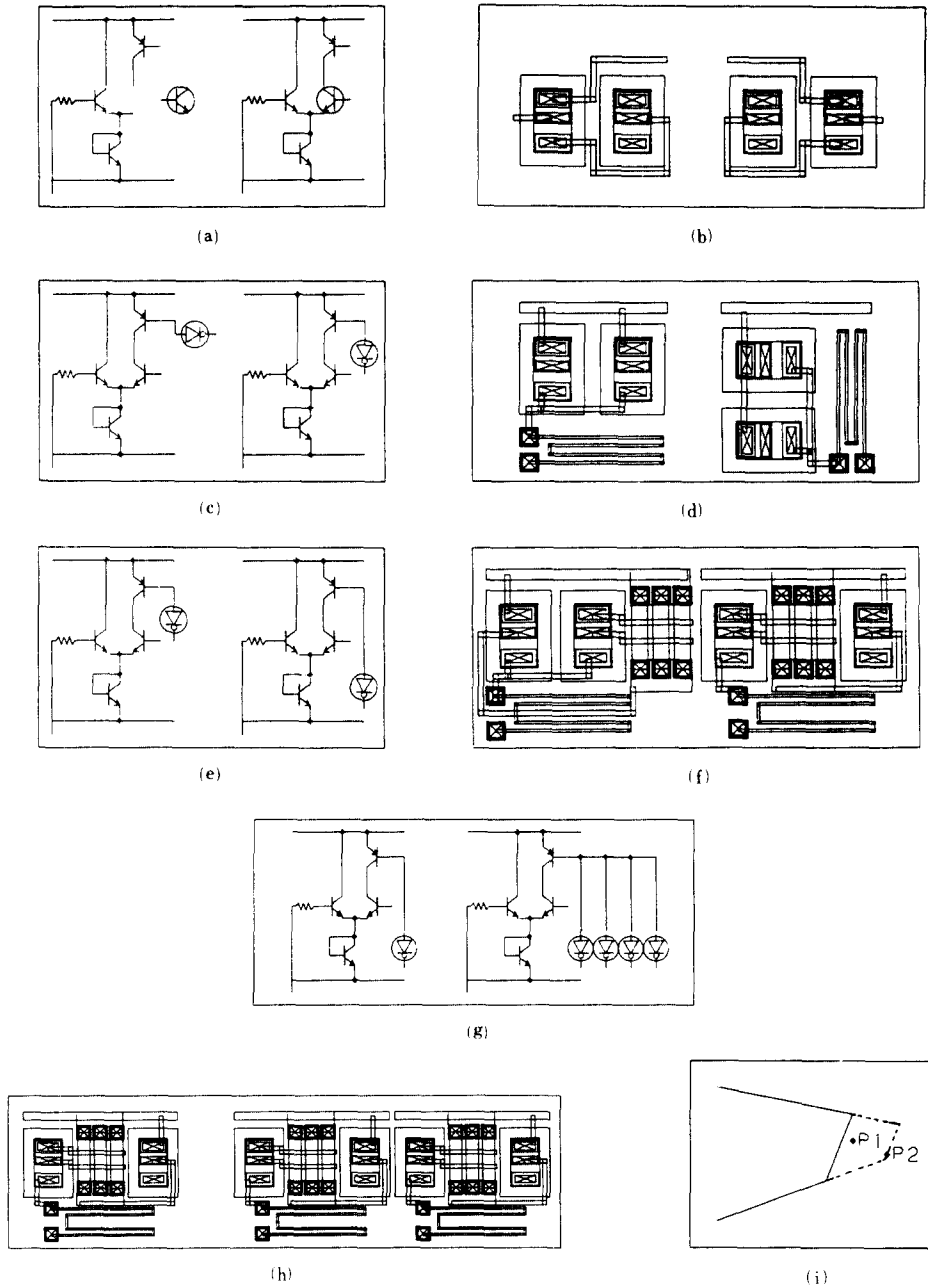


그림10. 수정 기능의 예

- (a) 단일 element 반전 (b) Element group 반전
- (c) 단일 element 회전 (d) Element group 회전
- (e) 단일 element 이동 (f) Element group 이동
- (g) 단일 element 복사 (h) Element group 복사 (i) 다각형 도형의 형태 변형

Fig. 10. Example of editing function.

- (a) Single element reflection. (b) Element group reflection.
- (c) Single element rotation. (d) Element group rotation.
- (e) Single element move. (f) Element group move.
- (g) Single element copy. (h) Element group copy. (i) Polygonal geometry stretch.

도면을 수정하는 도중에 cell을 새로 만들거나 nesting된 cell을 열어서 수정하는 일을 수정 buffer에서 수행함으로써, 도면을 열고 닫는 시간과 수고를 없애 주며, 수정된 cell이 nesting된 모든 곳을 찾아서 변형하는 것을 cell list에서 수행함으로써 전 data를 검색하는 작업을 없애고 빠르게 진행될 수 있도록 하였다.

4) Symbol의 자동적인 Numbering

회로 설계에 있어서, 일단 회로의 골격을 만든 후에 각 symbol의 number와 value를 나중에 결정하는 경우가 많이 있다. 이러한 경우 설계된 회로의 일부나 전부 혹은 개개 symbol의 number를 부여할 때가 있다. 따라서 사용자가 각 symbol마다 매번 number를 입력하는 것을 피하고 system내에서 빠르게 진행되도록 하기 위해서 자동적인 numbering 기능이 필요하게 된다.

각 symbol element별로 일시적인 buffer를 설치하고 좌표를 key로 하여 sort해 둬서 symbol element 검색을 binary search 기법으로 진행될 수 있도록 한다. Number의 자동적인 발생은 주어진 numbering pattern을 분석하여 identifier와 value 부분으로 분리한 후 value를 증가시켜 다시 identifier와 결합하는 방법으로 이루어진다. 다음은 자동적인 numbering 처리 algorithm을 나타낸다.

```

procedure AUTO-NUMBER
read symbol-name
create symbol-buffer
sort symbol-buffer with coordinate by
    HEAPSORT
read pattern
separate pattern to identifier and number-string
convert number-string to number
number := number - 1
repeat
    increment number
    convert number to number-string
    append number-string to identifier
    get record
    insert numbering to record
    put record
until end-of-symbol-buffer do
end AUTO-NUMBER
    
```

일단 symbol에 number가 지정되면 value의 자동적인 지정도 가능하며 따라서 simulation S/W의 입력 file을 value 지정용 file로 사용함으로써 value 지정의

순서를 단순하게 하며 추가의 지정 시간을 줄일 수 있다.

6 Macro 및 Function key 제어 Module

같은 순서로 여러 번 반복되는 command들을 하나의 이름이나 key로 지정하여 사용할 수 있으면, 사용자는 일련의 command들을 계속 반복해서 typing하는 번거로움을 피할 수 있다. 위와 같은 번거로움을 피하기 위해 일련의 command들을 하나의 이름에 배정하는 macro 정의와 지정된 key에 배정하는 function key 지정이 사용된다.

Macro의 설치 및 수행 방법은 먼저 각 macro에 20개의 record를 할당하고 macro file의 처음 3개의 record에 이미 정의된 macro의 hashing key를 기록해 두어서 macro의 정의 여부와 정의된 macro의 command list를 편리하게 처리하며, macro 정의의 collision 여부를 인식하기 위해서 각 macro의 첫번째 record에 macro name과 지정된 command 수를 기록해 두며, 또 반복 수행이 필요한 macro는 두번째 record에 반복횟수를 저장해 둔다. Macro의 recursive call을 방지하기 위해서 macro에 command를 지정할 때, 정의 중인 macro name과 비교하여서 같으면 지정하지 않고 경고를 보내준다. 그림11은 hashing key 계산 방법의 control 흐름도이다.

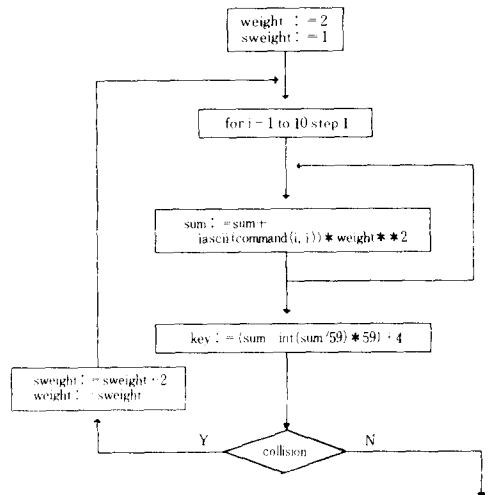


그림11. Hashing key 발생의 control 흐름
Fig. 11. Control flow for hashing key generation.

Function key 지정 및 수행은 각 function key별로 buffer를 설치하고 각 key에 고유한 ID를 부여하여서 사용자가 function key를 typing했을 때, 그 key의 ID

를 받아들여 key별 buffer의 해당되는 부분에 command를 기록하거나 buffer의 해당되는 부분으로부터 기록된 command를 읽어서 수행한다.

7. System Utility Module

SEAS에서 설계된 data와 graphic editor로서 가장 널리 사용되고 있는 Calma GDSⅡ와의 graphic data interface를 제공하며, 사용자에게 internal symbol을 정의하여 회로 도면과 그것의 상징적인 symbol을 함께 사용할 수 있도록 하여서 data의 다 단계 계층 구조를 실현시키며 user friendliness를 개선한다. 또 part list 추출, help utility 및 형식별 도면 출력이 있으나 여기서는 graphic data interface, internal symbol 정의와 도면 출력에 대해서 소개한다.

1) Calma GDSⅡ와의 Graphic data interface

Calma GDSⅡ의 stream format으로부터 SEAS data로의 변환은 1:1로 진행된다. SEAS에서는 2 level 이상의 subcell nesting과 stretchable cell을 허용하고 있지 않기 때문에 이들은 1:1 변환 이상의 처리가 이루어져야 한다. 따라서 2 level 이상의 nesting을 갖는 cell은 풀어져서 1 level cell로 변환되며, stretchable cell은 풀어져서 개개의 element상태로 변환된다.

2 level cell의 분해는 도면에서 subcell의 nesting을 나타내는 record를 제거하고 도면의 최 하단부에 subcell data를 첨가시키는 방법으로 진행된다. Multi-level cell의 분해는 2 level cell 분해 기능의 recursive call에 의한 방법으로 진행된다.

SEAS data에서 Calma GDSⅡ의 stream format으로의 변환도 역시 1:1로 진행된다. 그러나 사용자가 정의한 internal symbol들은 그 자신의 schematic circuit의 이름과 구별 되어야 하므로 symbol name에 "\$SYM"이라는 extension을 첨가한 이름의 cell로 정의되어 변환되며, built-in symbol은 새로운 cell로 정의되어 변환되고, 나머지 element는 1:1로 변환된다.

VAX11/780의 숫자 표현 기법과 Calma GDSⅡ stream format의 숫자 표현 기법이 다르기 때문에 변환시, 이들 숫자 표현은 반드시 변환되어야 한다.

2) Internal symbol 정의기능

다 차원의 data 처리 및 display를 허용하고 data 처리속도를 증가시키기 위해서 nesting된 cell을 내부 도면이 아닌 상징적인 internal symbol로 대치시켜서 작업을 수행하도록 하였다.

하나의 회로 도면이 만들어지면 그 회로 도면의 I/O port의 수가 일치하는 동일한 이름의 상응하는 internal symbol이 자동으로 만들어지며, 사용자는 I/O port

수는 바꿀 수는 없고 단지 그 모양만을 변화시켜서 사용함으로써 도면과 internal symbol의 상호 연관관계를 해치지 않고도 user friendliness를 개선할 수 있다.

또 built-in symbol의 한정으로 인해 새로운 symbol을 정의할 필요가 있으면, 사용자는 system 설계자에게 요청하지 않고도 자신의 symbol을 정의해서 사용할 수 있으며, 특정 기능을 갖는 block의 도면을 만들어 두지 않고 internal symbol을 먼저 정의하여 그것을 cell과 같이 nesting시켜 사용하고, 추후에 그 부분의 도면을 만들어 대치시키는 것을 허용함으로써 사용자는 확정되지 않은 block을 black box로 처리하여 top-down 방식의 설계도 진행시킬 수 있도록 하였다.

3) 도면 출력

입력된 도면을 출력시키기 위해서는 사용자는 도면 출력을 위한 parameter를 입력시켜서 그 parameter가 도면 출력 routine의 내부에서 처리될 수 있도록 해야 한다. 이 때 입력될 parameter로는 도면 file 이름, 도면 이름, plot layer 및 layer group, 각 layer의 pen, plot window 및 출력 도면형식 등이 있다.

출력 도면 file을 disc에서 읽어서 도면 별로 저장하면서 도면 list를 만들고 출력하고자 하는 도면을 출력용 buffer에 옮겨 주어서 출력 준비를 완료한다. 하나의 text string으로 입력된 plot layer 및 layer group을 분석하여서 각 layer의 group 수를 인식하고 각 layer에 대한 pen 번호도 저장한 다음 출력 scale만큼 도면에 크기를 주고 layer group의 수만큼 도면 출력 routine을 반복시킨다.

도면출력 routine에서는 하나의 element를 일시 출력 buffer에 가져온 후 현 layer group에 element의 layer가 해당하는가를 조사하고, 해당되면 plot window 내에 있는지를 검사하여 window 내에 있으면 출력하고, 걸쳐져 있으면 window로 잘라서 출력한다.

현 layer group의 출력이 끝나면, 사용자에게 출력 sheet나 pen을 교체할 기회를 주며, 출력 routine을 다시 진행시키도록하여 한번의 출력으로 다양한 색깔로 그려진 출력 도면이나 layer별로 그려진 다수의 출력 도면을 얻을 수 있도록 하였다.

IV. 결 론

Turn-key로 입수된 system에 비하여, SEAS는 자체적으로 개발되었기 때문에 유지, 보수 및 system 확장성이 좋으며, 다양한 수정 기능과 internal symbol의 정의로 다 단계 계층적인 설계 및 symbol을 광범위하게 확장할 수 있어서 설계 과정의 단순화에 기여하며, display의 2 단계 계층 구조와 terminal local

memory의 채택으로 screen refresh를 고속으로 처리 하여서 전체적인 설계 능력을 향상시킨다.

그러나 schematic editor로서의 기능만으로는 단지 회로 도면의 문서화와 단순화에는 도움을 주나, 도면의 효용성 검토는 기대할 수 없다. 따라서 이 system으로 설계된 회로 도면으로부터 netlist를 추출하고, 이를 바탕으로 하여서 simulation software에의 입력 file의 구성과 IC mask layout으로부터 추출된 netlist와 비교하는 network consistency check를 위한 연구 및 layout 설계를 편리하게 하기 위한 다양한 기능의 개발이 추진되어야 한다.

參 考 文 獻

- [1] "GDSII Reference Manual", Calma Co., pp. 4-28~4-30, 10-83~10-85, 1983.
- [2] "UTCS User Manual", Computer S/W Dev. Team, Central Research Lab., GoldStar Co., Ltd., 1985.
- [3] "HCBS User Manual", California Computer Products, Inc. 1983.
- [4] "VAX-11 Fortran Language Reference Manual", Digital Equipment Co., pp. 7-5~7-39, 1982.
- [5] Tom D. Marco, *Structured Analysis and System Specification*. Yourdon Inc., pp. 59-80, 1978.
- [6] C.J. Date, *An Introduction to Database Systems*. Addison-Wesley Publishing Co., pp. 64-70, 1977.
- [7] "GDSII Menu Manual", Calma Company, 1983.
- [8] Horowitz, Sahni, *Fundamentals of Computer Algorithms*. Computer Science press Inc., pp. 53-61, pp. 82-93, pp. 100-107, 1978.
- [9] Julius T. Tou, *Applied Automata Theory*. Prentice-Hall, pp. 56-59, 1968.
- [10] "Tak. 4110 Series Command Reference", Tektronix, Inc., pp. 7-12~7-384, 1983.
- [11] Won Lyang Chung, *Interactive Graphic Computer Aided Design*. KAIST S/W Development Center, pp. 432-433, 1981.