

EDAS_P 시스템에서 Graphic Tool開發 (SYMED_P, SCHED_P)

柳 瑛 昱

〈要 約〉

EDAS_P 시스템을 이루고 있는 8개의 tool 중 symbol의 모양을 정의하는 SYMED_P, 전자회로도를 입력시키는 SCHED_P, 그리고 논리 검증결과를 그림으로 표현하는 WAVGR_P는 graphics 기술을 이용하여 정보를 취급한다. 본고에서는 그림으로 정보를 표현하는데 필요한 기술을 발췌적으로 선별하여 소개하였으며, 그림정보를 표현하는 형식 및 정보의 종류를 설명한다.

I. 서 론

EDAS_P(Electronic Design Automation System_Personal)는 전자회로를 설계하기 위

한 개인용 CAE 시스템으로서 전자회로도를 입력시키는 과정으로부터 netlist의 추출, 논리분석, 파형분석의 전과정을 컴퓨터는 이용하여 수행할 수 있도록 개발된 시스템이다.

이 시스템 내에는 3개의 graphic tool이 있는데 symbol의 모양을 정의하기 위한 symbolic editor인 SYMED_P, schematic을 정의하는 schematic editor인 SCHED_P, 그리고 simulation의 결과를 파형으로 표현하는 waveform grapher인 WAVGR_P이다. 이 tool들은 정보를 그림으로 표현하고 mouse와 cursor를 이용해 데이터 입력, 명령의 선택 등을 행할 수 있는 대화형 graphic tool이다.

본 논문은 이 세계의 tool중 SYMED_P와 SCHED_P의 데이터 구조로부터 pop-up menu의 구현, 좌표계변환, main program의 구성 등에 대해 논술한다.

II. 본 론

1. 2차원 Graphics에서의 Transformation

2차원에서 위치 데이터로 이루어진 도형에 변형을 가해 move, rotate, mirror 등의 조작을 해야할 경우가 생기는데 이때 위치 데이터를 어떻게 바꾸어 주어야 하는가 하는 문제는 2×2 matrix를 (x, y) 좌표값에 곱해 줌으로써 쉽게 해답을 얻을 수 있다. 여러 개의 변형을 연속적으로 가해주어야 하는 경우도 변형에 해당하는 matrix를 연속적으로 곱해 주면 된다.

Identity Scale Counterclockwise rotation

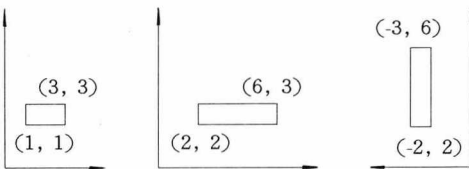
$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}$$

Mirror X Mirror Y Translation(move)

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

예를 들어 (1, 2)와 (2, 3)을 대각점으로 하는 box를 x방향으로 2배, y방향으로 1배의 scaling을 가한 후 반시계방향으로 90° 회전시킨다고 하면 아래와 같이 변형에 해당하는 transformation matrix를 차례로 곱해주면 된다. 그 결과가 <그림 1>에 나타나 있다.

$$(x, y) \cdot \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos 90 & \sin 90 \\ -\sin 90 & \cos 90 \end{bmatrix}$$



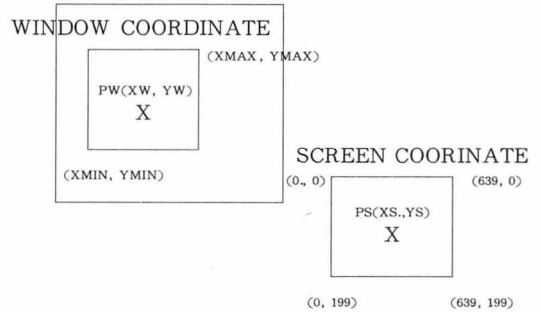
<그림 1> Transformation의 예

2. 좌표계 변환

Graphic tool은 데이터를 그림으로 표현하

로 사용하는 좌표계에 따라 좌표값 변환을 해야 한다. 실제로 데이터가 표현되는 좌표계는 2차원 직각 좌표계로써 world coordinate라 부른다. 이 정보를 컴퓨터 화면에 나타내고자 할 때 두개의 좌표계를 고려해야 하는데, 하나는 화면이 갖고 있는 좌표계로써 해상도에 제한되는 영역이고 또 하나는 window coordinate다. Window란 world coordinate의 일부 네모난 영역으로써 그 크기와 위치에 따라 그림의 모양이 달리 표현된다. <그림 2>에 EDAS_P 시스템에 사용된 좌표계간의 관계를 나타냈다.

WORLD COORINATE



<그림 2> 좌표계간의 관계

Ps는 screen 좌표계에 따라 표현된 좌표값(Xs, Ys)이고 Pw는 Ps에 대응되는 world 좌표계로 표현된 좌표값(Xw, Yw)라고 할 때 이들 두 점간의 좌표값 변환은 다음의 식을 사용한다.

$$X_s = 639 \times \frac{X_w - X_{min}}{X_{max} - X_{min}}$$

$$Y_s = 199 \times \left(1 - \frac{Y_w - Y_{min}}{Y_{max} - Y_{min}} \right)$$

$$X_w = \frac{X_s \times (X_{max} - X_{min})}{639} + X_{min}$$

$$Y_w = \frac{(199 - Y_s) \times (Y_{max} - Y_{min})}{199} + Y_{min}$$

3. Mouse 와의 Interface

Mouse는 컴퓨터와 RS-232 serial port에 연결되어 사용하는데 mouse를 소프트웨어와 연관 지어 이용하려면 cursor의 모양, 이동, 나타남 그리고 mouse가 갖고 있는 button의 상태 등을 받아들이는 기능이 있어야만 한다. 다행히 mouse 자체가 그러한 기능을 갖고 있어 프로그래머는 4개의 정수 변수의 값만 잘 지정해 주고 subroutine을 부르면 필요한 정보가 4개의 변수에 입력되어 되돌아 온다.

CALL MOUSE(M1, M2, M3, M4)

M1 변수값에 따라 mouse의 기능이 다른데 요약하면 아래와 같다.

- 0 : Reset mouse parameter
- 1 : Show cursor
- 2 : Hide cursor
- 3 : Get mouse position and button status
- 4 : Set mouse cursor position
- 5 : Get button press information
- 6 : Get button release information
- 7 : Set minimum and maximum horizontal position
- 8 : Set Minimum and maximum vertical position
- 9 : Set graphics cursor block
- 10 : Set text cursor
- 11 : Read mouse motion counter
- 12 : Set user-defined subroutine input mask
- 13 : Light pen emulation mode on
- 14 : Light pen emulation mode off
- 15 : Set Mickey/Pixel Ratio

4. Pop-up Menu의 구현

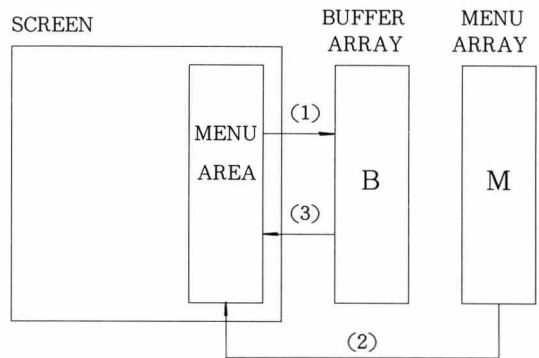
Pop-up이란 화면 일부에 어떤 모양이 붙속 튀어나온다는 의미로서 pop-up menu라 함은 명령이 종류별로 요약된 menu판이 화면의 그림위로 튀어나와 명령을 선택하도록 한다는 것

이다. Graphic 화면은 bit-map 형식으로서 화면상의 한 point가 컴퓨터 main memory의 어떤 번지 데이터의 한 bit(color graphic은 하나 이상의 bit)에 해당하는 것이며 화면에 그림을 그린다는 것은 해당 메모리에 정보를 입력시키는 것이다. 이와같이 어떤 그림에 해당하는 데이터를 bit map data라고 할 때 정수값을 저장하는 array나 bit map data를 저장하는 array나 정보의 표현 수단만 다를 뿐이지 같은 종류이다.

Pop-up menu를 만들려면 먼저 일정한 크기의 menu를 화면에 그린 후 그 bit map 정보를 어떤 array에 (M이라고 부르자) 저장한다. Graphic tool 수행시 menu를 나타내려면 세 단계를 거쳐야 한다.

- Menu가 놓일 영역에 해당하는 bit map data를 buffer array에 (B라고 부르자) 임시저장한다.
- Menu array M에 저장된 bit map data를 screen 상에 menu가 놓일 영역에 배치한다.
- 사용자가 명령을 선택하면 buffer array B에 저장되었던 bit map data를 menu가 놓였던 위치에 배치해 원래의 그림으로 복귀한다.

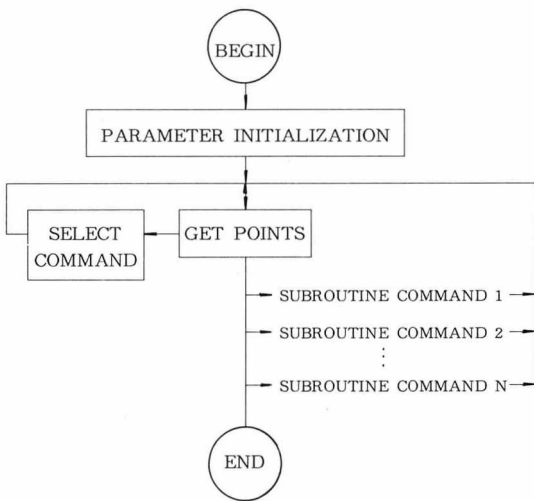
이런 과정이 신속히 일어나면 사용자의 눈에는 menu가 갑자기 발생하였다 없어지는 것으로 나타나게 된다.



<그림 3> Pop-up Menu의 구현

5. Main Program의 구성

Graphic tool은 여러개의 명령을 갖고 있어 각 명령마다 subroutine이 따로 개발되어 있어 정보를 지우거나 고치거나 더하고, 정보의 변형을 화면에 그림으로 표현하는 역할을 한다. <그림 4>에 표현된 main의 흐름도를 보면 맨 처음 프로그램 수행에 필요한 변수의 초기치를 입력시키고, 해당명령이 필요로 하는 좌표를 입력받는다. 이때 사용자가 pop-up menu를 이용해 명령을 바꾸면 그 명령이 필요로 하는 좌표를 새로이 입력받는다. 필요한 좌표가 모두 입력되면 수행명령에 해당하는 subroutine에서 명령을 수행한 후 다시 좌표를 입력받는 과정을 반복한다.



<그림 4> Main의 흐름도

6. SYMED_P에서 도형들의 Data Format

Symbolic editor인 SYMED_P에서 사용되는 도형은 connector, text, line, circle, box, arc의 6가지가 있다. 각 도형은 종류에 따라 필요로 하는 데이터의 종류와 형태가 다르므로 이에 관한 정보를 포함해야 하며, 도형선택을 도형이

차지하고 있는 영역으로부터 알아내므로 도형의 bounding box 정보도 포함해야 한다. 또 한가지 주목할 사항은 도형의 좌표값을 절대좌표값과 상대좌표값을 적절히 배열함으로써 도형의 수정 및 변형이 쉽도록 데이터의 구조를 설계하였다.

가. Array ID(*, 5)

이 array는 각 도형에 관한 정보중 도형의 종류, 도형이 차지하는 영역 그리고 도형의 좌표가 저장된 영역을 알리는 pointer로 이루어진다.

[icon code, pointer to PD(*, 1), X11, Y11, Xur, Yur]

- icon code : 도형의 종류를 구별하는 code

- 1 : line 4 : arc
- 2 : circle 5 : connector
- 3 : box 6 : text

- pointer to PD(*, 1) : 도형의 좌표값등이 저장된 array PD(*, 1)의 시작행을 알리는 pointer

- X11, Y11, Xur, Yur : 도형이 차지하고 있는 영역의 좌하점 좌표(X11, Y11) 및 우상단 좌표(Xur, Yur). 이 좌표는 도형의 원점(첫번째로 입력된 위치 좌표)에 대한 상대좌표값이다.

나. Array PD(*, 1)

각 도형이 갖고 있는 정보를 저장하는 array로서 도형을 그리기 위해 입력된 첫번째 좌표가 그 도형의 원점으로서 절대좌표값이고 나머지 좌표는 도형의 원점에 대한 상대좌표값이다.

- line 과 arc Xf , Yf(Absolute)
- n , o
- X1 , Y1(Relative)
- . .
- . .
- Xn , Yn(Relative)

- circle Xcen , Ycen(Absolute)
Radius , 0
- box X11 , Y11(Absolute)
Xur , Yur(Relative)
- connector Xcen , Ycen(Absolute)
pointer to , rotation(0-3)
string array
- text Xcen , Ycen(Absolute)
pointer to , rotation(0-3)
string array
Scalex , Scaley

- X mirror : symbol의 x mirror를 지시
(0-1)
- Y mirror : symbol의 y mirror를 지시
(0-1)
- Symbol identity : symbol의 고유이름
양의정수 : 시스템이 준 고유번호
음의정수 : 사용자가 준 symbol의 이름.
절대값이 string array로
의 pointer

Array SYM(*, 6)는 각 symbol의 종류에 따라 하나씩의 entry를 갖는 array로써 아래와 같은 형식이다.

7. SCHED_P에서 도형들의 Data Format

Schematic editor인 SCHED_P에서 사용되는 array는 symbol에 관한 정보를 저장하는 array SB(*, 6), SYM(*, 6), ID(*, 1), PD(*, 1)과 interactive 하게 연결상태가 검사되는 수평 wire의 정보를 저장하는 array HL(*, 3), 수직 wire의 정보를 저장하는 array VL(*, 3), node point의 정보를 저장하는 array CRS(*, 2)가 있다.

가. Symbol 정보를 저장하는 Array

Array SB(*, 6)는 배치된 symbol 각각에 대해 하나씩의 entry를 갖는 array로써 각 symbol 마다의 고유한 값을 저장한다. 저장된 정보의 종류와 형식은 아래와 같다.

[pointer to SYM(*, 6), Xref, Yref, rotation, xmirror, ymirror, symbol identity]

- pointer to SYM(*, 6) : symbol의 종류를 구별하기 위한 array SYM(*, 6)행번호를 지시
- Xref, Yref : symbol의 중심점 좌표 (Absolute)
- Rotation : symbol의 rotation을 지시 (0-3)

[pointer to string array, start row : pointer to ID(*, 1), end row ; pointer to ID(*, 1), X11, Y11, Xur, Yur]

- pointer to string array : symbol의 file name을 알려준다.
- start row, end row : 해당 symbol의 도형 데이터를 저장한 array ID(*, 1)의 시작행과 끝행을 알려주는 pointer
- X11, Y11, Xur, Yur : symbol의 bounding box의 좌하단과 우상단 좌표값(Relative to symbol origin)

이밖에 array ID(*, 1)는 SYMED_P의 ID(*, 5)중 처음 두개의 값이고, PD(*, 1)은 동일하다. 단 모든 좌표값이 symbol origin에 대한 상대좌표값이다.

나. Wire의 정보를 저장하는 Array

Wire는 수평선과 수직선으로 나누어지며 node와 node사이의 선분을 하나의 정보로 취급한다. Array도 수평선의 정보를 저장하는 HL(*, 3), 수직선의 정보를 저장하는 VL(*, 3), wire의 끝점인 node의 정보를 저장하는 CRS(*, 2)가 있어 이들 정보를 sorting하고 searching하면서 wire간의 연결이 자동으로 검사된다.

Array HL(*, 3)는 수평선의 정보를 아래와 같은 형식으로 저장한다.

[Xf, Xt, Y, wire-identity]

(Xf, Y) (Xt, Y) : 수평선의 왼쪽과 오른쪽 끝 점좌표

wire - identity : wire의 고유이름
 양의 정수 : 시스템이 준 고유번호
 음의 정수 : 사용자가 준 wire의 이름.
 절대값이 string array로
 의 pointer

Array VL(*, 3)은 수직선에 관한 정보를 아래의 형식으로 저장한다.

[Yf, Yt, X, wire-identity]

(X, Yf) (X, Yt) : 수직선의 아래쪽과 위쪽 끝 점좌표

wire - identity : wire의 고유이름
 양의 정수 : 시스템이 준 고유번호
 음의 정수 : 사용자가 준 wire의 이름.
 절대값이 string array로
 의 pointer

Array CRS(*, 2)는 각 node의 위치 및 wire의 연결상태를 알려준다.

[X, Y, status]

(X, Y) : node가 위치한 좌표값
 status : node에 연결된 wire의 상태를 알려준다. 수평선이 연결되면 2를 증가시키고 수직선이 연결되면 1을 증가시킨다.

III. 결 론

EDAS_P 시스템의 graphic tool은 순수 국내 기술로 개발되었는데 graphics 기술 및 데이터의 저장형식, 명령 수행방식 등 모든것을 정의하고 실현하는데 어려움이 많았고 또한 많은 수정을 거쳤다. 자료로 얻는 논문이나 관련 서류 또한 기능 소개 정도로 끝나 프로그램의 구조를 설계하는 데는 도움이 되지 못했고 유경험자의 자문을 구하는 것 또한 불가능했다. 이

런 연유로 어떤 부분은 복잡하고 비효율적인 면도 지적될 수 있다. 본고에서 논한 pop-up menu의 구현, SYMED_P의 data format과 SCHED_P의 data format 등을 차후 이 분야에 연구개발을 계속하는 데 참고가 되리라 기대한다.

〈參 考 文 獻〉

1. Steven Harrington, Computer Graphics - A Programming Approach, McGraw-Hill pp. 488, 1983.
2. Bruce A. Artwick, Applied Concepts in Microcomputer Graphics, Prentice Hall, pp. 374, 1984.
3. Roy E. Myers, Microcomputer Graphics for the IBM PC, Addison-Wesley Co., p. 268, 1984.
4. "The Human Factors of Computer Graphics Interaction Techniques," CG&A, pp. 13~48, Nov. 1984.
5. E. Freeman, "IBM PC-based Software for CAE and CAD," EDN, vol. 31, no. 19, pp. 162~180, Sept. 1986.
6. Technology to Watch "Turning a PC into a Silicon Compiler," Electronics, pp. 37~40, June 1986.
7. 유영욱의 44명, 설계 자동화 시스템 개발에 관한 연구, 1985년도 과학 기술처 특정 연구 사업, 최종 보고서, 한국전자통신연구소, 1986. 7.
8. EDAS_P 운용 기술서, 참고 문헌[7]의 별책 부록.
9. 박인학, 이철동, 유영욱, "전자회로 설계를 위한 개인용 CAE 시스템", 전자공학회 1986년도 CAD·반도체·재료 및 부품연구회 합동학술발표회 논문집, 11~14 1986. 5.