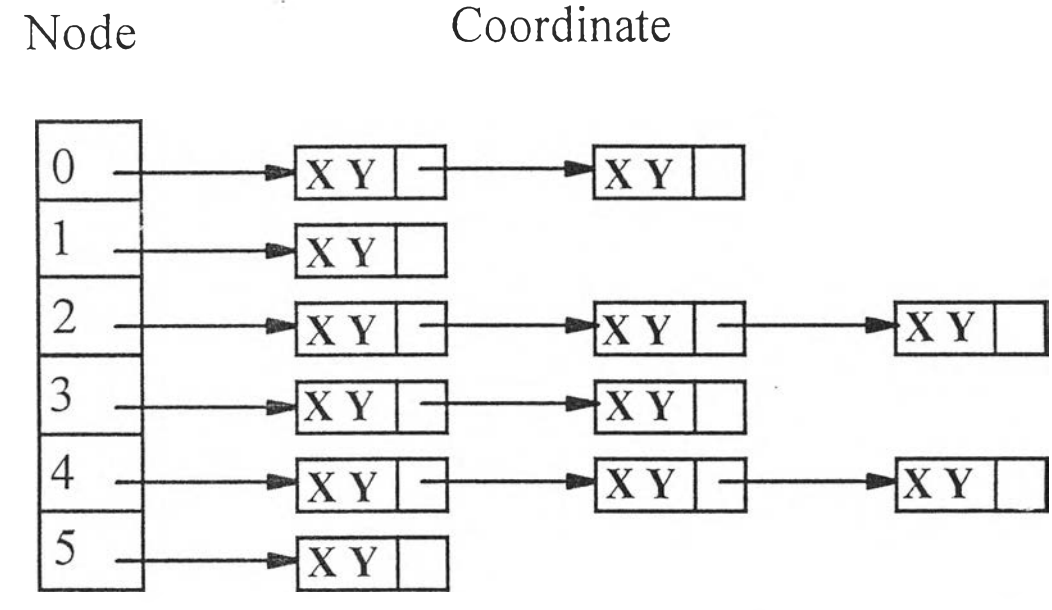
บทที่ 4

โครงสร้างข้อมูล และการติดต่อกับผู้ใช้

โครงสร้างข้อมูล

โครงสร้างข้อมูลในงานวิจัยนี้ ใช้ array of pointer ซึ่งชี้ไปยังโครงสร้างซึ่งเก็บค่าตำแหน่งบนจอภาพ เป็นตารางของโนดทำให้ทราบว่าพิกัดบนจอภาพที่จุดใดบ้างที่เป็นโนดหมายเลขเดียวกัน ดังรูปที่ 4.1



รูปที่ 4.1 แสดงโครงสร้าง linklist ของ node ที่ชี้ไปยังพิกัดที่เกี่ยวข้องกัน (Node Table)

โครงสร้างของ Cordinate และ Node ที่นิยามเป็นดังนี้

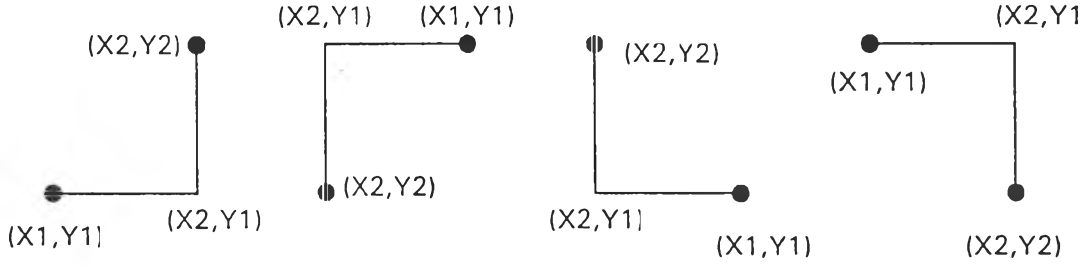
```
struct Cordinate {
  int x,y;
  Cordinate *CordinatePtr;
} *Node[NumNode];
```

โดย NumNode เป็นตัวแปลชนิดจำนวนเต็ม บอกจำนวนโนดสูงสุดที่โปรแกรมสามารถคำนวนได้ ในการค้นหาว่าพิกัดใดมีหมายเลขโนดเป็นเท่าไร โปรแกรมก็จะเข้าไปค้นหาในตารางโนดแล้วจึงส่งค่าหมายเลขโนดที่เกี่ยวข้องกับพิกัดที่ให้ไปกลับมาให้ ดังนี้

```
int GetNode(Cordinate *CordinatePtr)
{
  Cordinate *TempCordinatePtr;
  Boolean found;
  for (int i=0; (i<NumNode) && (Node[i]!=NULL); i++) {
    TempCordinatePtr=Node[i];
    found = false;
    while (!found) {
      if (CordinatePtr->x==TempCordinatePtr->x && CordinatePtr->y==TempCordinatePtr->y)
      {
        return (i);
      } else {
        if (TempCordinatePtr->CordinatePtr != NULL)
          TempCordinatePtr = TempCordinatePtr->CordinatePtr;
        else
          found = true;
      }
    }
  }
}
```

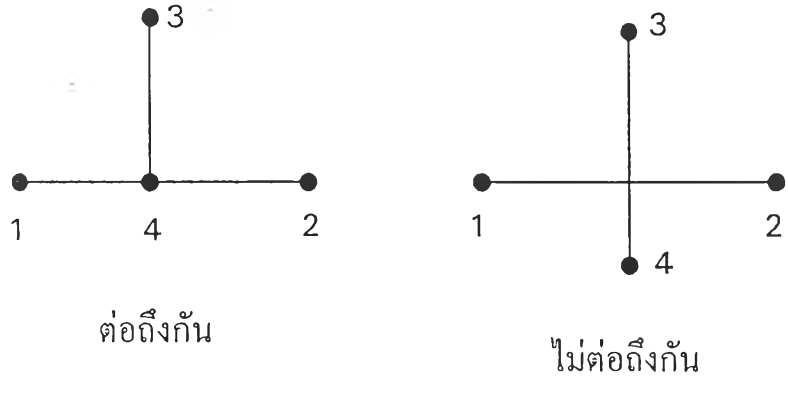
ฟังก์ชัน GetNode จะรับค่าพิกัดบนจอภาพเข้าไป เพื่อค้นหาว่าที่พิกัดนั้นมีหมายเลขโนดเป็นเท่าไร เพื่อนำไปใช้ประโยชน์ในขณะที่ผู้ใช้ให้แสดงหมายเลขโนด และจัดเก็บข้อมูลลงบนแผ่นจานแม่เหล็ก เพื่อวิเคราะห์วงจร ซึ่งผู้ใช้วาดบนจอภาพคอมพิวเตอร์

ในการสร้างตารางโนดนั้นเกี่ยวพันธ์กับการวาดสายไฟในวงจร เนื่องจากจุดที่สายไฟเชื่อมต่อถึงกันถือว่าเป็นโนดเดียวกัน ในการวาดสายไฟจะต้องมีข้อมูลจุดแรกในการเริ่มวาด (X1,Y1) และจุดที่สองเป็นสายไฟอีกปลายหนึ่ง (X2,Y2) ดังรูปที่ 4.2



รูปที่ 4.2 ภาพแสดงการวาดสายไฟ

ในการวาดสายไฟนี้จะลากจากจุด (X1, Y1) ไปยัง (X2, Y2) ตามแนวนอนก่อนแล้วจึงลากตามแนวตั้ง และการลากสายไฟนี้หากผู้ใช้ลากไปหยุดบนเส้นอื่นแสดงว่าสายไฟสองเส้นนั้นต่อถึงกัน หากลากข้ามเส้นอื่นแสดงว่าสองเส้นนั้นไม่ต่อถึงกัน ดังแสดงในรูปที่ 4.3



รูปที่ 4.3 แสดงสายไฟ 1-2 และ 3-4 ที่ต่อถึงกัน และต่อไม่ถึงกัน

ขบวนการในการตรวจสอบว่าจุด (X, Y) ใดๆ จะอยู่บนสายไฟเส้นใดเส้นหนึ่งในวงจร หรือไม่ ทำได้โดยตรวจสอบว่าจุด (X, Y) นั้นตรงตามเงื่อนไขด้านล่างนี้หรือไม่

$$X = X2 \text{ และ } Y2 \leq Y \leq Y1$$

$$Y = Y1 \text{ และ } X1 \leq X \leq X2$$

หรือ

$$X = X2 \text{ และ } Y1 \leq Y \leq Y2$$

$$Y = Y1 \text{ และ } X2 \leq X \leq X1$$

หรือ

$$X = X2 \text{ และ } Y2 \leq Y \leq Y1$$

$$Y = Y1 \text{ และ } X2 \leq X \leq X1$$

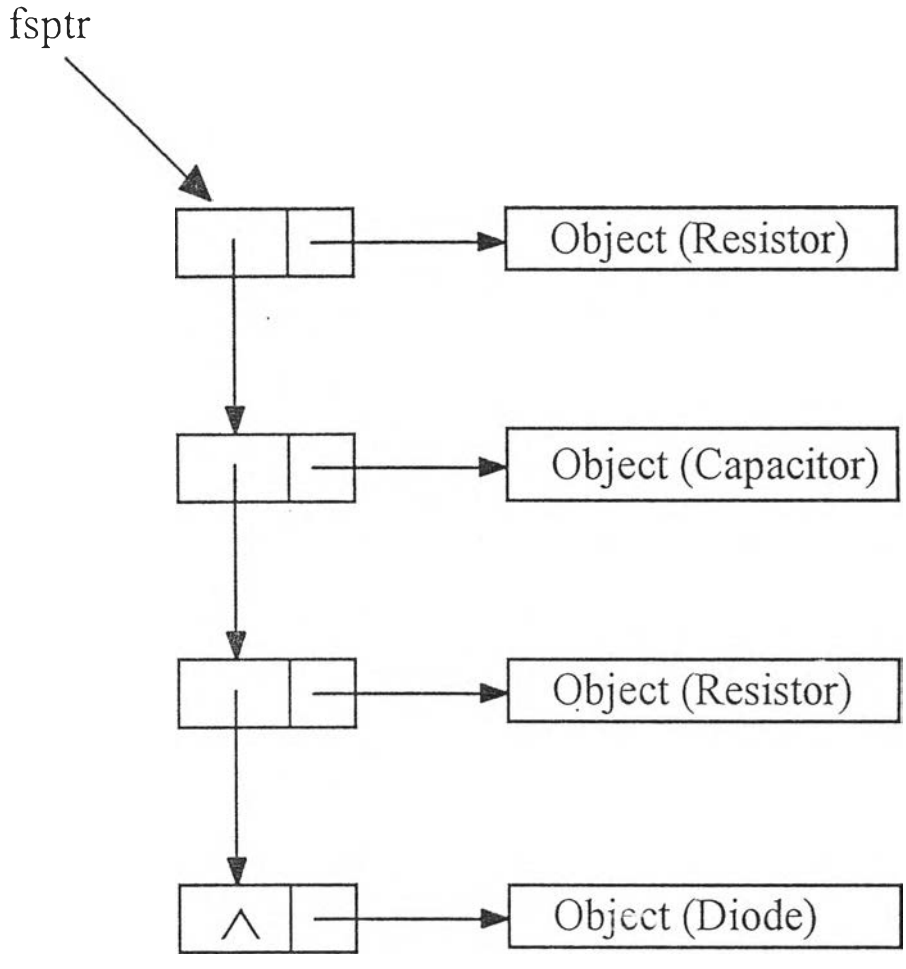
หรือ

$$X = X2 \text{ และ } Y1 \leq Y \leq Y2$$

$$Y = Y1 \text{ และ } X1 \leq X \leq X2$$

ถ้าจุด (X, Y) เป็นไปตามเงื่อนไขนี้ แสดงว่าจุด (X, Y) นั้นอยู่บนเส้น (X1, Y1), (X2, Y2) ส่วนของโปรแกรมที่ใช้ในการตรวจสอบเงื่อนไขนี้ เมื่อพบว่าสายไฟสองเส้นนั้นต่อถึงกัน โปรแกรมก็จะต้องจัดการรวมจุดทั้ง 4 จุดของสายไฟสองเส้นนั้นเป็นโนดหมายเลขเดียวกัน ในขณะสร้างตารางของโนด เพื่อใช้เป็นข้อมูลในการจัดเก็บไฟล์ ซึ่งในการจัดเก็บไฟล์นี้ต้องจัดเก็บในรูปที่โปรแกรมวิเคราะห์วงจรไฟฟ้าสามารถอ่านไฟล์ที่จัดเก็บนี้ขึ้นมา แล้วไม่มีข้อมูลสูญหายไป และเข้าใจว่าอุปกรณ์แต่ละอุปกรณ์มีการเชื่อมต่อสัมพันธ์กันอย่างไร ทำให้สามารถทำการวิเคราะห์วงจรจากไฟล์ที่จัดเก็บนี้ได้ทันที

สำหรับโครงสร้างในการเก็บข้อมูลของอุปกรณ์ต่างๆ ที่ผู้ใช้เรียกขึ้นมาใช้วาดรูปวงจรอิเล็กทรอนิกส์ เพื่อทำการวิเคราะห์นั้นเป็น linklist ของตัวชี้ ซึ่งชี้ไปยังวัตถุ ซึ่งแสดงโครงสร้างของ linklist ดังรูปที่ 4.4 ในโครงสร้างของ linklist ที่แสดงนี้สามารถนำอุปกรณ์ต่างๆ ไปเก็บไว้ใน linklist เดียวกันได้



รูปที่ 4.4 แสดงโครงสร้างของ linklist ของอุปกรณ์

โครงสร้างของ linklist ของอุปกรณ์ต่างๆ ที่นิยามเป็นดังนี้

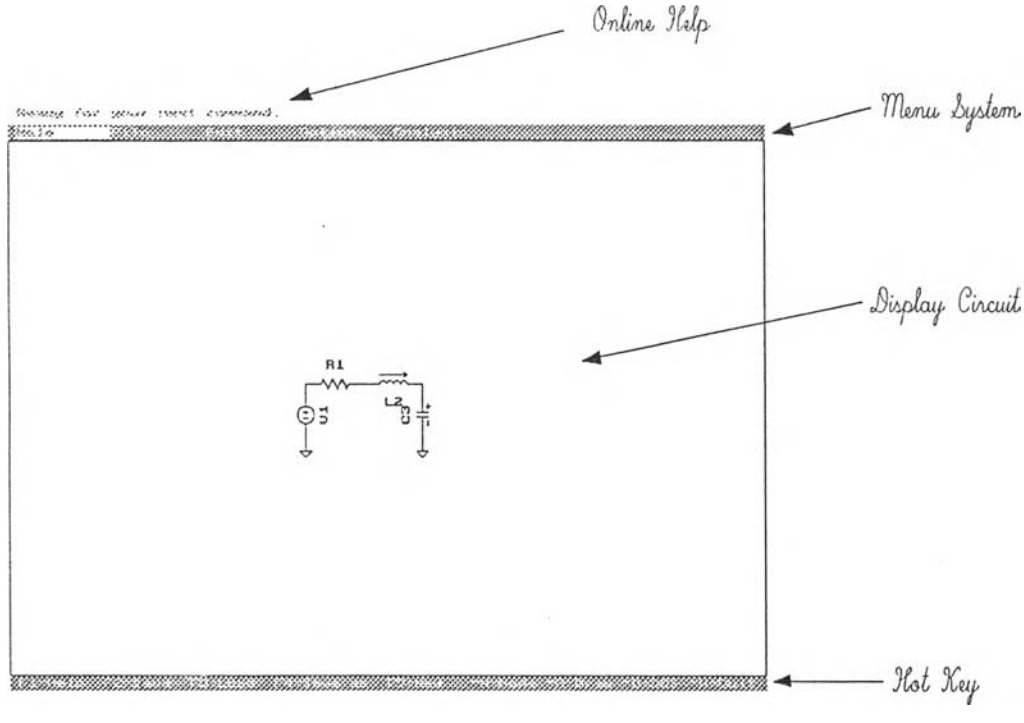
```
struct element {
    Emessage *Ptr;
    element *next;
} *sptr;
element *fsptr;
```

Sweep *SweepPtr;

เมื่อผู้ใช้เรียกอุปกรณ์ใดขึ้นมา เพื่อต่อวงจรอุปกรณ์นั้นก็จะถูกเพิ่มลงไปใน linklist ของอุปกรณ์ สำหรับวิธีการวิเคราะห์วงจร หรือ Sweep นั้นมีตัวที่ไปยังวัตถุนี้เพียงตัวเดียวไม่ได้เป็น linklist เนื่องจากในวงจรแต่ละครั้งจะวิเคราะห์ได้วิธีใดวิธีหนึ่งเท่านั้น

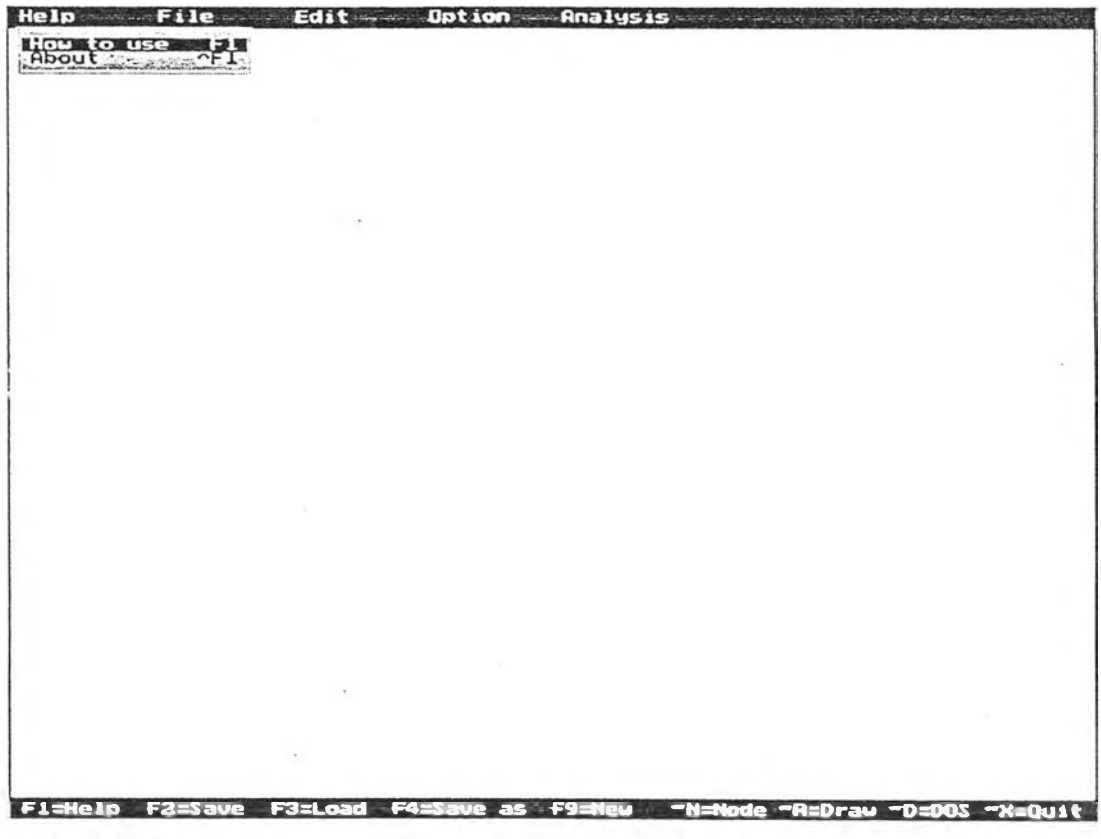
<u>การติดต่อกับผู้ใช้</u>

จอภาพ การออกแบบส่วนต่างๆ ของจอภาพในการติดต่อกับผู้ใช้ได้ทำการออกแบบโครงสร้างของจอภาพเป็นดังรูปที่ 4.5 โดยนำระบบเมนูแบบดึงลงมาใช้ เพื่อให้มีส่วนแสดงภาพวงจรได้เต็มที่ และง่ายต่อการใช้งาน เนื่องจากผู้ใช้ส่วนใหญ่จะเคยผ่านการใช้งานระบบเมนูเช่นนี้มาก่อนแล้ว นอกจากนั้นยังมี Online Help เพื่อให้คำแนะนำแก่ผู้ใช้โปรแกรมตลอดเวลาที่มีปัญหา



รูปที่ 4.5 แสดงรูปแบบหน้าจอ

ระบบเมนู ระบบเมนูได้ถูกออกแบบให้เป็นแบบเมนูดึงลง ในขณะที่เมนูไม่มีการแสดงผู้ใช้สามารถใช้ Hot Key ทุก Key ที่อยู่ในเมนูใช้งานคำสั่งของเมนูนั้นได้ ซึ่งในกรณีนี้ใช้สำหรับผู้ใช้งานโปรแกรมได้คล่องแล้ว แต่ในกรณีที่เมนูใดเมนูหนึ่งแสดงอยู่บนจอภาพผู้ใช้สามารถใช้คำสั่งต่างๆ ได้ โดยเลื่อนแถบสว่างไปที่คำสั่งนั้น แล้วกด Enter หรือกดตัวอักษรตัวใหญ่ของคำสั่งในเมนูนั้น หรือจะใช้ Hot Key ที่อยู่ด้านหลังคำสั่งนั้น ในที่เมนูแสดงอยู่นี้ Hot Key ที่ใช้ได้จะต้องเป็น Hot Key ที่แสดงอยู่บนจอภาพเท่านั้น เช่นตามรูปที่ 4.6 โปรแกรมจะรับ Hot Key คือ H, F1, A, ^F1, F2, F4, F9, ~N, ~R, ~D, และ ~X เท่านั้น



รูปที่ 4.6 แสดงสภาวะที่เมนูรับ Hot Key เฉพาะที่เฉพาะที่แสดงบนจอภาพ

หน้าต่างรับข้อมูล (Dialog Box) เป็นส่วนของจอภาพในการรับข้อมูลเข้าจะปรากฏเมื่อผู้ใช้ดึงอุปกรณ์ต่างๆ ออกมา เพื่อต่อวงจรหน้าต่างรับข้อมูลนี้จะถามค่าต่างๆ ที่จำเป็นสำหรับอุปกรณ์นั้น ถ้าผู้ใช้เรียกอุปกรณ์ ซึ่งมีอยู่บนจอภาพ เพื่อมาแก้ไขค่าโปรแกรมจะนำค่าเก่าก่อนการแก้ไขมาแสดง ถ้าเป็นอุปกรณ์ใหม่ไม่มีค่าเก่ามาแสดงโปรแกรมจะนำค่า ซึ่งโปรแกรมกำหนดไว้มาแสดง (defult value) โดยมีเงื่อนไขว่าถ้าบรรทัดแรกของหน้าต่างรับข้อมูลไม่มีข้อมูลเข้าแสดงว่ายกเลิกการดึงอุปกรณ์นั้นออกมาใช้งาน และถ้าบรรทัดที่สองไม่มีข้อมูลเข้าแสดงว่าผู้ใช้ต้องการลบอุปกรณ์ที่ปรากฏชื่อในบรรทัดแรกออกจากวงจร และเมื่อผู้ใช้กลอกข้อมูลเสร็จเรียบร้อยหน้าต่างรับข้อมูลก็จะปิดไป และให้ผู้ใช้วางอุปกรณ์ที่เลือกมา หรือย้ายตำแหน่งอุปกรณ์ ซึ่งผู้ใช้ระบุชื่อไว้ในหน้าต่างรับข้อมูล