บทที่ 6

โปรแกรมต้นแบบสำหรับวิเคราะห์วงจรเชิงเส้นแบบท่อน

6.1 บทนำ

โปรแกรมต้นแบบสำหรับวิเคราะห์ผลตอบสนองเชิงเวลาของวงจรอิเล็กทรอนิกส์เชิงเส้นแบบท่อนทั่วไปที่ได้พัฒนาขึ้นนี้ ให้ชื่อว่า 'TAPE' ย่อมาจาก A Transient Analysis Program for Piecewise Linear Electronic Circuits 'TAPE' เขียนขึ้นโดยใช้ภาษา Applesoft Basic เพื่อให้ใช้กับไมโครคอมพิวเตอร์ Apple][หรือคอมพิวเตอร์รุ่นอื่นๆ ที่มีการใช้งานเหมือนกัน การที่เราเลือกใช้ภาษา Applesoft Basic ในการพัฒนาโปรแกรมก็เพราะเป็นภาษาที่ใช้ง่าย สะดวกต่อการตรวจสอบที่ผิด (debug) และมีโปรแกรมช่วยในการพัฒนา (developing software tool) มากพอควร เช่น APA, PLE, D.BUG เป็นต้น 'TAPE' จะมีขีดความสามารถและคุณสมบัติดังนี้

1) ใช้กับวงจรอิเล็กทรอนิกส์ที่ประกอบด้วย ตัวต้านทาน ตัวเหนี่ยวนำ ตัวเก็บประจุ แหล่งกำเนิดสัญญาน ไดโอด ทรานซิสเตอร์ ออปแอมป์ และ สวิตช์

2) สามารถคำนวณผลตอบสนองเชิงเวลา (transient analysis) ได้

3) สามารถควบคุมความผิดพลาดและปรับขนาดขั้นเวลาได้อัตโนมัติ

4) สำหรับเครื่อง Apple][ที่มีหน่วยความจำ 48K bytes 'TAPE' จะสามารถวิเคราะห์วงจรที่มีตัวแปรสูงสุดได้ถึง 28 ตัวแปร ซึ่งวงจรขนาดนี้จะมีประมาณ 27 โนด และอาจมีองค์ประกอบได้ถึง 100 ตัว

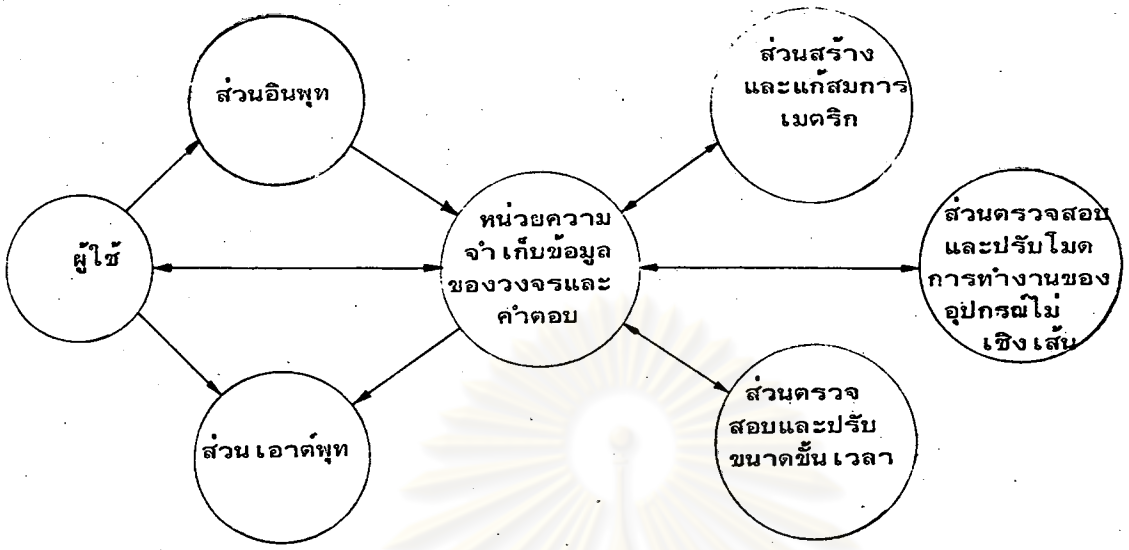
5) ใช้ง่าย แก้ไขข้อมูลได้ง่าย สามารถเก็บข้อมูลเกี่ยวกับวงจรไว้ในแผ่นจานแม่เหล็ก (Diskette) เพื่อเรียกออกมาวิเคราะห์หรือแก้ไขใหม่ภายหลังได้อีก

6) ผู้ใช้สามารถเปลี่ยนแปลงค่าพารามิเตอร์ของอุปกรณ์ไม่เชิงเส้นได้

บทนี้จะกล่าวถึงรายละเอียดของขีดความสามารถและคุณสมบัติทั้ง 6 ข้อของ 'TAPE' เราจะกล่าวรายละเอียดในบทนี้โดยจะมุ่งไปที่โครงสร้างและวิธีการใช้งานของโปรแกรมเพื่อให้เข้าใจวิธีการใช้งานได้อย่างถูกต้องและมีประสิทธิภาพ

6.2 โครงสร้างโปรแกรมและโมดการทำงานของ 'TAPE'

โครงสร้างของโปรแกรมแบ่งออกได้เป็นโปรแกรมย่อยหลักอยู่ 5 โปรแกรมคือ ส่วนอินพุท ส่วนสร้างและแก้สมการเมตริก ส่วนเอาต์พุท ส่วนตรวจสอบเพื่อปรับภาวะการทำงานของอุปกรณ์ไม่เชิงเส้น และส่วนตรวจสอบและปรับขนาดขั้นเวลาดังแสดงในรูปที่ 6.1 โปรแกรมย่อยหลักทั้ง 5 โปรแกรมนี้จะแบ่งการทำงานของ 'TAPE' ออกเป็น 2 โมด ในขณะหนึ่งจะมีชุดโปรแกรมย่อยเพียงโมดเดียวของ 'TAPE' เท่านั้นที่มีอยู่ในไมโครคอมพิวเตอร์ ส่วนอีกโมดหนึ่งจะอยู่ในแผ่นจานแม่เหล็กที่เก็บชุดโปรแกรมย่อยทั้งสองไว้ โมดทั้งสองนี้มีชื่อเรียกและมีหน้าที่โดยย่อดังนี้



รูปที่ 6.1 โครงสร้างหลักของโปรแกรม 'TAPE'

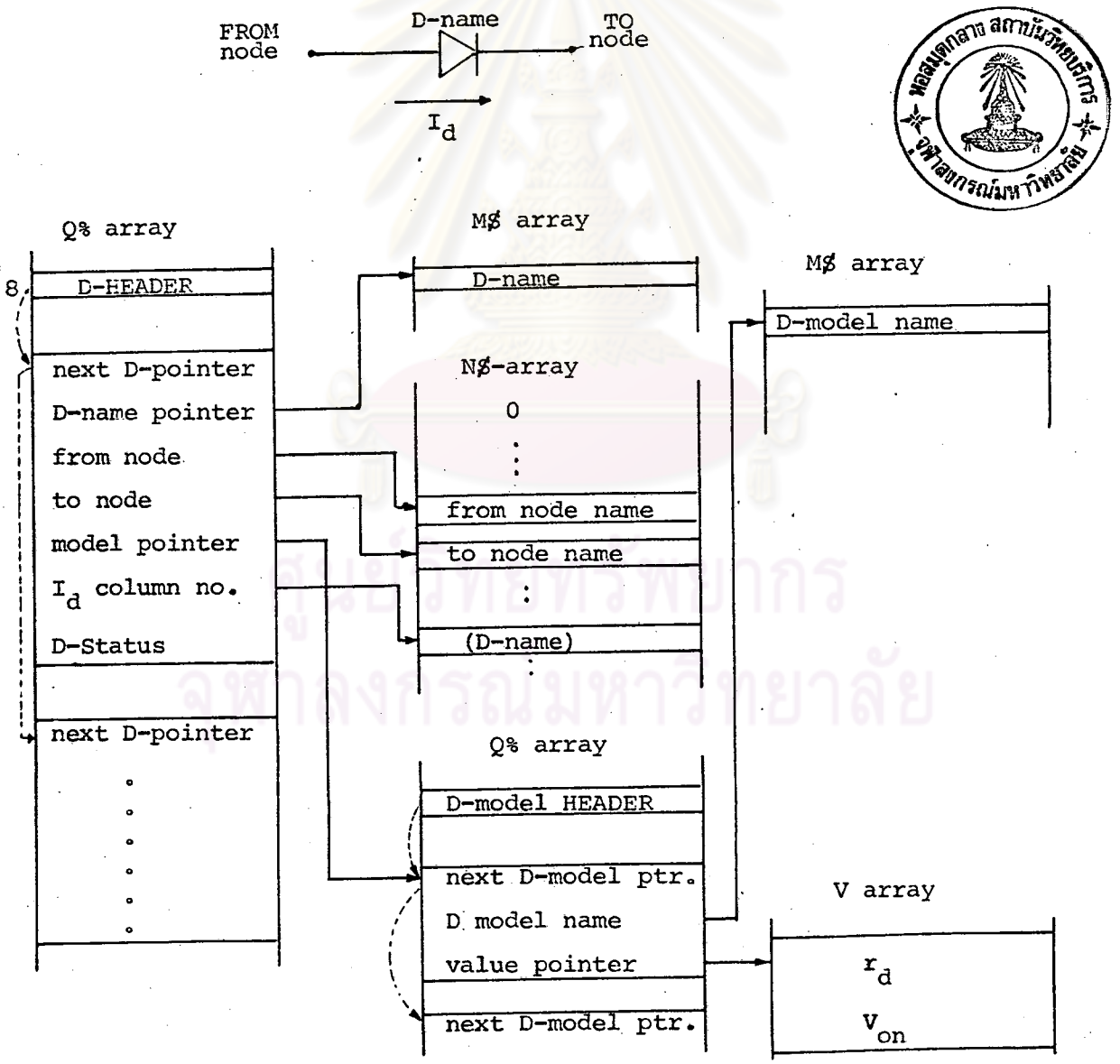
6.2.1 โมด INPUT เป็นโมดที่รับข้อมูลเกี่ยวกับวงจรจากผู้ใช้ ประกอบด้วยโปรแกรมส่วนอินพุทซึ่งทำหน้าที่ประมวลข้อมูลวงจรที่ผู้ใช้ป้อนเข้าไปและจัดระบบโครงสร้างในการเก็บข้อมูลแบบ linked list ซึ่งเป็นลักษณะเดียวกับที่ใช้ในโปรแกรม 'เล็ก 1.0[3]' ทำให้สามารถสร้าง ดัดแปลง แก้ไข และเพิ่มเติมวงจรได้ตามต้องการ อีกทั้งยังเก็บข้อมูลของวงจรไว้ในจานแม่เหล็กและเรียกออกมาใช้ได้โดยง่าย

6.2.2 โมด SIM เป็นโมดที่ทำการวิเคราะห์วงจร ประกอบด้วยโปรแกรมส่วนสร้างและแก้สมการเมตริกของวงจร ส่วนตรวจสอบเพื่อปรับภาวะการทำงานของอุปกรณ์ไม่เชิงเส้น ส่วนตรวจสอบและปรับขนาดขั้นเวลาและส่วนเอาต์พุท ในโมดนี้โปรแกรมจะรับคำสั่งในด้านที่เกี่ยวกับการวิเคราะห์วงจร เช่น กำหนดขนาดขั้นเวลา h ขนาด LTE ช่วงเวลาที่ต้องการวิเคราะห์ เวลาที่เริ่มทำการวิเคราะห์ และ พิมพ์ผลการวิเคราะห์เป็นต้น ข้อมูลของวงจรที่ใช้วิเคราะห์จะรับมาจากโมด INPUT โดยการเก็บพื้นที่ส่วนบนของหน่วยความจำ RAM เป็นที่เก็บข้อมูลสำหรับการเชื่อมต่อระหว่างโมดทั้งสอง และในบางกรณีก็สามารถรับข้อมูลของวงจรมาจากจานแม่เหล็กได้ ในระหว่างการวิเคราะห์วงจร SIM จะไม่ติดต่อกับผู้ใช้จนกว่าจะวิเคราะห์เสร็จและเข้าสู่โปรแกรมส่วนเอาต์พุทแล้วเท่านั้น

รายละเอียดของโปรแกรมย่อยหลักต่างๆ และการใช้งานในแต่ละโมดจะได้กล่าวถึงในตอนต่อๆ ไป

6.3 ข้อมูลทางเทคนิคและการใช้งานในโมด INPUT

ในโมด INPUT โปรแกรมจะทำหน้าที่รับข้อมูลเกี่ยวกับวงจรและคำสั่งอื่นๆ ในลักษณะโต้ตอบ (interactive) กับผู้ใช้ ข้อมูลที่ผู้ใช้ป้อนเข้าไปจะถูกประมวลผลแล้วเก็บไว้ใน array สี่ชุดด้วยกันคือ array Q% array V array N$ และ array M$ Q% เป็น array หลักในการค้นหาข้อมูลที่เกี่ยวกับองค์ประกอบแต่ละตัว โครงสร้างข้อมูลใน Q% จะเป็นแบบ linked list คือ เก็บตัวชี้ (pointer) เพื่อไปค้นหาชื่อของอุปกรณ์ใน array M$ ค่าของอุปกรณ์ใน array V ชื่อโนดที่อุปกรณ์ต่ออยู่ใน array N$ และยังชี้บอกตำแหน่งของอุปกรณ์ชนิดเดียวกันที่อยู่ในตำแหน่งถัดไปด้วย นอกจากนี้ข้อมูลที่จำเป็นบางอย่างเช่น ตำแหน่งของตัวแปรกระแสและ ภาวะการทำงานของอุปกรณ์ไม่เชิงเส้นก็ยังเก็บไว้ใน array Q% เช่นกัน ดังตัวอย่างแสดงในรูปที่ 6.2



รูปที่ 6.2 แสดงการเก็บข้อมูลของไดโอด

อุปกรณ์ทุกตัวจะมี HEADER ของตัวมันเองอยู่ส่วนบนของ Q% array ดังเช่นในรูปที่ 6.2 HEADER ของไดโอดอยู่ที่ตำแหน่ง Q% (8) ซึ่งข้อมูลใน HEADER นี้จะชี้ไปที่ตำแหน่งชุดข้อมูลของไดโอด และส่วนบนสุดของชุดข้อมูลแต่ละชุดก็จะมีตัวชี้ (pointer) บอกตำแหน่งชุดข้อมูลของอุปกรณ์ชนิดเดียวกันที่อยู่ถัดไป การบอกตำแหน่งชุดข้อมูลของอุปกรณ์ชนิดเดียวกันเป็นทอดๆ นี้ ทำให้เราสามารถค้นหา แก้ไข เพิ่มเติมข้อมูลของอุปกรณ์ต่างๆ ได้โดยง่ายไม่ต้องพะวงถึงการจัดกลุ่มอุปกรณ์ และยังทำให้การแสดงผล (display) ข้อมูลของวงจรเป็นหมวดหมู่ของแต่ละอุปกรณ์อย่างมีระเบียบด้วย

ชื่อของโนดที่อุปกรณ์ต่ออยู่ในวงจรที่ใช้ป้อนข้อมูลเข้าไปจะถูกตรวจสอบกับชื่อของโนดที่มีอยู่แล้วใน N$ array ถ้าชื่อโนดนั้นมีอยู่แล้ว ตำแหน่งของมันใน N$ array ก็จะถูกเก็บไว้ใน Q% array แต่ถ้าชื่อโนดเป็นโนดใหม่ N$ array ก็จะเพิ่มขนาดขึ้นอีกหนึ่งและจะเก็บชื่อโนดใหม่นั้นไว้ ส่วนตำแหน่งของมันก็จะถูกเก็บไว้ใน Q% array เช่นเดิม ด้วยวิธีการเช่นนี้ถ้าพิจารณาให้ดีจะเห็นว่า ความจริงแล้วตำแหน่งของชื่อโนดที่เก็บไว้ใน Q% array ก็จะเป็นตำแหน่ง row และ column ที่จะโหลดข้อมูลของตราประจำอุปกรณ์ในการสร้างสมการเมตริกของวงจร และขนาดของ N$ array ก็เท่ากับขนาด (dimension) ของสมการเมตริกวงจรนั่นเอง ยกตัวอย่างเช่น ถ้า N$ array มีขนาด 1x5 เมตริกจัตุรัสของสมการวงจรจะมีขนาด 5x5

สำหรับอุปกรณ์บางอย่างเช่น ตัวเหนี่ยวนำ ออปแอมป์ ไดโอด สวิตช์ และแหล่งจ่ายแรงดันที่ต้องมีตัวแปรกระแสของตัวมันเองเป็นตัวแปรของสมการวงจรเพิ่มขึ้นมานั้น โปรแกรม 'TAPE' จะจัดการเพิ่มขนาดของ N$ array และกำหนดชื่อของตัวแปรกระแสเป็นชื่อเดียวกับองค์ประกอบโดยอัตโนมัติซึ่งจะทำในโมด SIM นั่นคือจะทำหลังจากที่ผู้ใช้ป้อนข้อมูลของวงจรเสร็จและเริ่มทำการวิเคราะห์แล้ว ตามตัวอย่างในรูป 6.2 จะเห็นว่าตำแหน่ง Id column no. ใน Q% array จะชี้ไปยังชื่อของไดโอด (D-name) ใน N$ array ที่สร้างขึ้นโดยโปรแกรมโมด SIM เพื่อกำหนดตัวแปรกระแสไดโอดและตำแหน่งสำหรับโหลดข้อมูลขององค์ประกอบในสมการเมตริกของวงจร

ออปแอมป์ ไดโอด และทรานซิสเตอร์ เป็นอุปกรณ์ที่ต้องมีค่าพารามิเตอร์ประจำตัวและผู้ใช้สามารถกำหนดค่าพารามิเตอร์เหล่านี้ได้ ยกตัวอย่างเช่นไดโอดในรูป 6.2 เมื่อผู้ใช้ใส่ข้อมูลของไดโอดในด้านของการต่อไดโอดในวงจร คือ D-name, from node, to node และกำหนดชื่อของเบอร์ไดโอดเช่น 1N4001 เป็นต้น โปรแกรมก็จะทำการค้นหาชื่อเบอร์นั้นใน M$ array ถ้าไดโอดเบอร์นั้นมีอยู่แล้ว model pointer ใน Q% array ก็จะชี้ไปยังชุดข้อมูลของเบอร์ค่าสมบัติของไดโอดใน array เดียวกัน แต่ถ้าเป็นเบอร์ใหม่โปรแกรมจะเตรียมที่ใน Q% array ขึ้นมาไว้โดยอัตโนมัติเพื่อให้ผู้ใช้ใส่ค่าพารามิเตอร์ในภายหลัง

อุปกรณ์ไม่เชิงเส้นที่มีการทำงานในหลายๆ ภาวะ รหัสของภาวะการทำงานในภาวะปัจจุบันก็จะเก็บไว้ใน Q% array ด้วยเช่นในตัวอย่างของไดโอดนั้น ถ้าตำแหน่ง D-STATUS เป็น 1 หมายถึงไดโอดกำลังอยู่ในภาวะนำกระแส และถ้า D-STATUS เป็น 0 ไดโอดจะอยู่ในภาวะไม่นำกระแส เป็นต้น

ค่าพารามิเตอร์ของอุปกรณ์ชนิดต่างๆ เช่น ความเหนี่ยวนำ (inductance) ความต้านทาน (resistance) และอื่นๆ ตลอดจนค่าในอดีตของตัวแปรสถานะเพื่อใช้สำหรับการคำนวณหาคำตอบและ LTE ของวงจรที่เวลาปัจจุบันจะถูกเก็บไว้ใน V array และตำแหน่งใน V array จะถูกชี้โดย Value pointer ใน Q% array ยกตัวอย่างในรูปที่ 6.2 ในชุดข้อมูลของพารามิเตอร์ของไดโอด (Diode Model) จะมี Value pointer เป็นตัวชี้ไปยังค่าสมบัติ คือ ความต้านทานขณะนำกระแส (r_d) และ แรงดันนำกระแส (V_{on}) ใน V array

ที่กล่าวมานี้เป็นการอธิบายลักษณะการเก็บและการค้นหาข้อมูลของวงจรอย่างสังเขป ส่วนการใช้งานในโมด INPUT นั้นง่าย ผู้ใช้สามารถทำความเข้าใจได้อย่างรวดเร็วและไม่ต้องพะวงถึงการเก็บข้อมูลที่กล่าวมา เพราะโปรแกรมจะทำหน้าที่ป้อนคำถามผู้ใช้เพียงแต่เลือกหรือใส่ข้อมูลที่ต้องการลงไปเท่านั้น ในโมด INPUT มีคำสั่งให้เลือกอยู่ 7 คำสั่งด้วยกัน ดังแสดงในรูป 6.3 ผู้ใช้เพียงแต่กดปุ่มตัวเลขเพื่อเลือกคำสั่งที่ต้องการใช้ซึ่งเราจะได้อธิบายความหมายและการใช้งานของคำสั่งทั้ง 7 ในหัวข้อต่อๆ ไป

```
PWR ELECTRONIC PROG MODE
        <1> EDIT
        <2> SIM
        <3> LOAD
        <4> SAVE
        <5> CHANGE
        <6> DISPLAY
        <7> QUIT

INPUT MODE NUMBER YOU WANT:
```

รูปที่ 6.3 การแสดงผลบนจอภาพในโมด INPUT

6.4 คำสั่ง EDIT

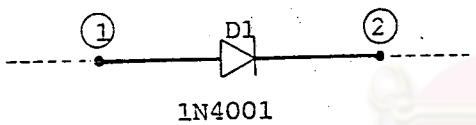
คำสั่งนี้ใช้สั่งให้โปรแกรมรับข้อมูลของวงจรจากผู้ใช้เข้ามาประมวลผลและเก็บไว้ใน array ทั้งสี่ที่ได้กล่าวมาแล้ว ในหัวข้อนี้เราจะแสดงการป้อนข้อมูลของอุปกรณ์แต่ละชนิดควบคู่ไปกับลักษณะการเก็บข้อมูลใน array ทั้งสี่ด้วย เพื่อให้เกิดความเข้าใจได้ดีขึ้น เมื่อผู้ใช้เลือกคำสั่ง EDIT โดยการกดเลข 1 แล้วโปรแกรมจะเข้าสู่เมนูย่อยดังในรูปที่ 6.4 เพื่อให้ใส่อักษรย่อของอุปกรณ์ที่ต้องการลงไป

PUT R, L, C, Q, O, SW, V, I, QM, OM, D, DM, OR END:

R = RESISITOR	QM = MODEL TRANSISTOR
C = CAPACITOR	OM = MODEL OPAMP
L = INDUCTOR	D = DIODE
Q = TRANSISTOR	SW = CONTROLLED SWITCH
O = OPAMP	DM = MODEL DIODE
V = VOLTAGE SOURCE	END = OUT OF EDIT MODE
I = CURRENT SOURCE	

รูปที่ 6.4 ข้อมูลบนจอภาพเมื่อเข้าสู่โมด EDIT

ถ้าต้องการป้อนข้อมูลของไดโอดในรูปที่ 6.5 (ก) ก็ให้กดตัว "D" จอภาพจะแสดงผลในรูปที่ 6.5 (ข) ผู้ใช้จะต้องใส่ข้อมูลลงไป (ตัวที่ขีดเส้นใต้) ให้ครบเสร็จแล้วโปรแกรมจะจัดข้อมูลใน array ทั้งสี่ดังในรูปที่ 6.2 และก็กลับมาถามผู้ใช้ใหม่ดังในรูปที่ 6.4 เมนูคำสั่งนี้จะสิ้นสุดลงและโปรแกรมจะกลับไปพิมพ์เมนูของโมด INPUT ก็ต่อเมื่อผู้ใช้ป้อนคำสั่ง "END"




```
EDIT DIODE

1    D-NAME:D1)
2    FR NODE:1)
3    TO NODE:2)
4    D-MODEL:1N4001)
5    D STATUS :1)
```

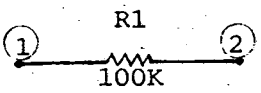
STATUS : 0 = OFF 1 = ON

(ก) ไดโอด D1 อยู่ในโมดนำกระแส (ข) การป้อนข้อมูลของไดโอด

รูปที่ 6.5 การป้อนข้อมูลของไดโอด

สำหรับการป้อนข้อมูลของอุปกรณ์อย่างอื่นๆ เราจะให้เป็นรูปการแสดงผลบนจอภาพและการจัดข้อมูลใน array เท่านั้น จะไม่ขอกล่าวในรายละเอียดเนื่องจากวิธีการคิดและประมวลผลนั้นเหมือนๆ กับที่ได้กล่าวมาแล้วในตัวอย่างของไดโอด

๖.๔.๑ ตัวต้านทาน

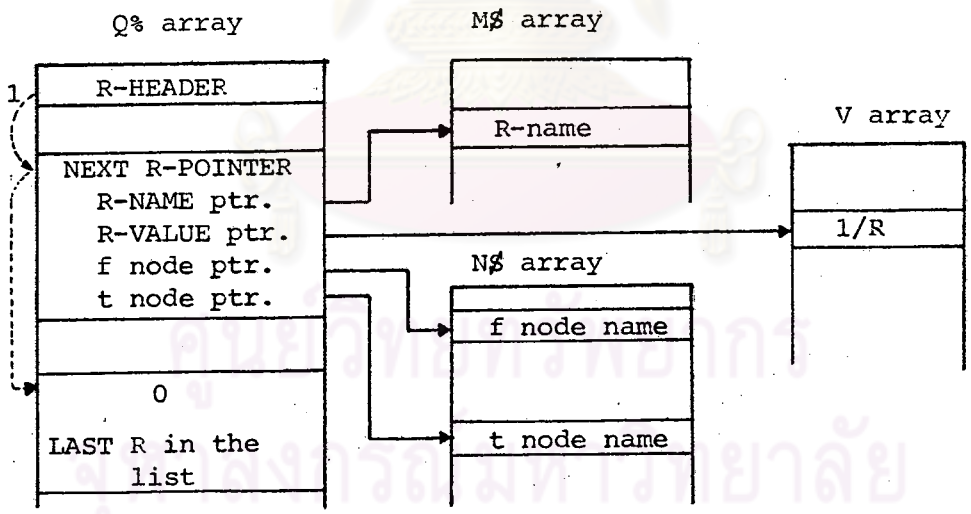


EDIT RESISTOR

1 R-NAME:: R1
2 R-VALUE: 100K
3 FR-NODE: 2
4 TO-NODE: 1

หมายเหตุ ในช่อง R-VALUE: K = 10^{3}
MEG = 10^{6}
U = 10^{-6}
P = 10^{-12}
M = 10^{-3}

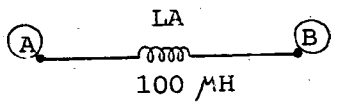
(ก)



(ข)

รูปที่ ๖.๖ การใส่ข้อมูลของความต้านทาน (ก) การแสดงผลบนจอภาพ
(ข) การจัดข้อมูลใน array

6.4.2 ตัวเหนี่ยวนำ

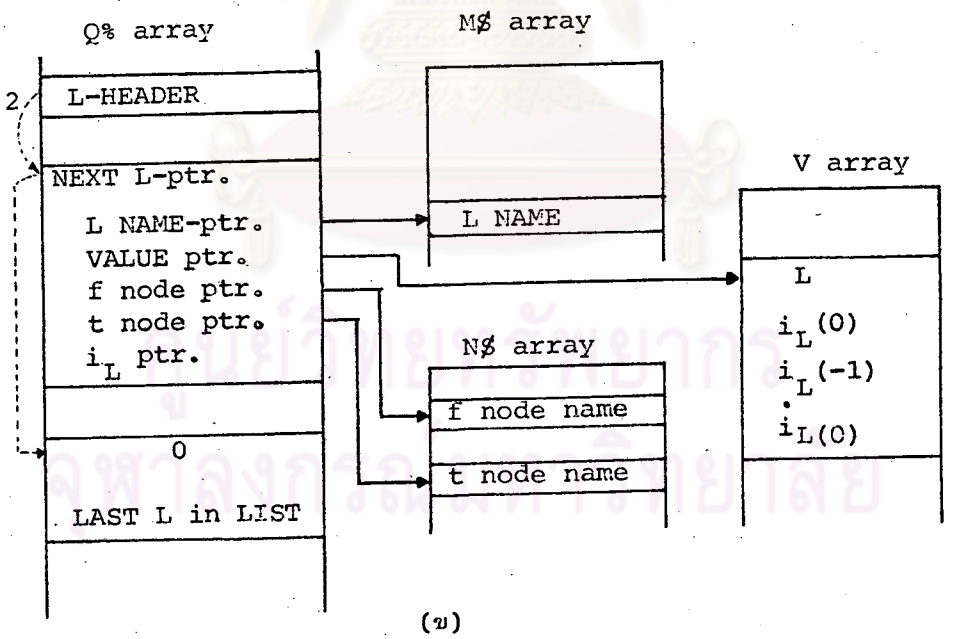


EDIT INDUCTOR

1 L-NAME:: LA
2 L-VALUE: 100U
3 FR NODE: A
4 TO NODE: B
5 I PAST1:
6 I PAST2:
7 SLOPE I:

หมายเหตุ ในข้อ 5, 6 และ 7 เป็นการใส่ค่าเริ่มต้น (Initial condition) ของตัวเหนี่ยวนำ ถ้าผู้ใช้ไม่ทราบก็ไม่ต้องใส่อะไรเลย

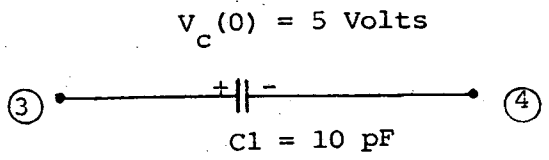
(ก)



(ข)

รูปที่ 6.7 การใส่ข้อมูลของตัวเหนี่ยวนำ

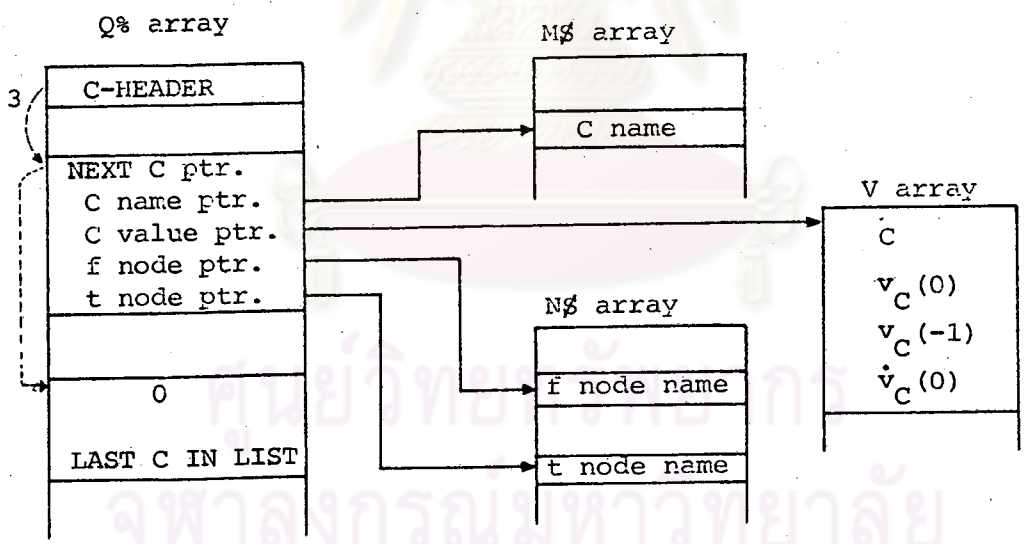
6.4.3 ตัวเก็บประจุ



EDIT CAPACITOR

1 C-NAME:: C1 ↵
2 C-VALUE: 10P ↵
3 FR NODE: 3 ↵
4 TO NODE: 4 ↵
5 V PAST1: 5 ↵
6 V PAST2: ↵
7 SLOPE V: ↵

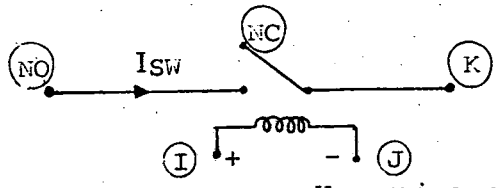
(ก)



(ข)

รูปที่ 6.8 การใส่ข้อมูลของตัวเก็บประจุ

6.4.4 สวิตช์



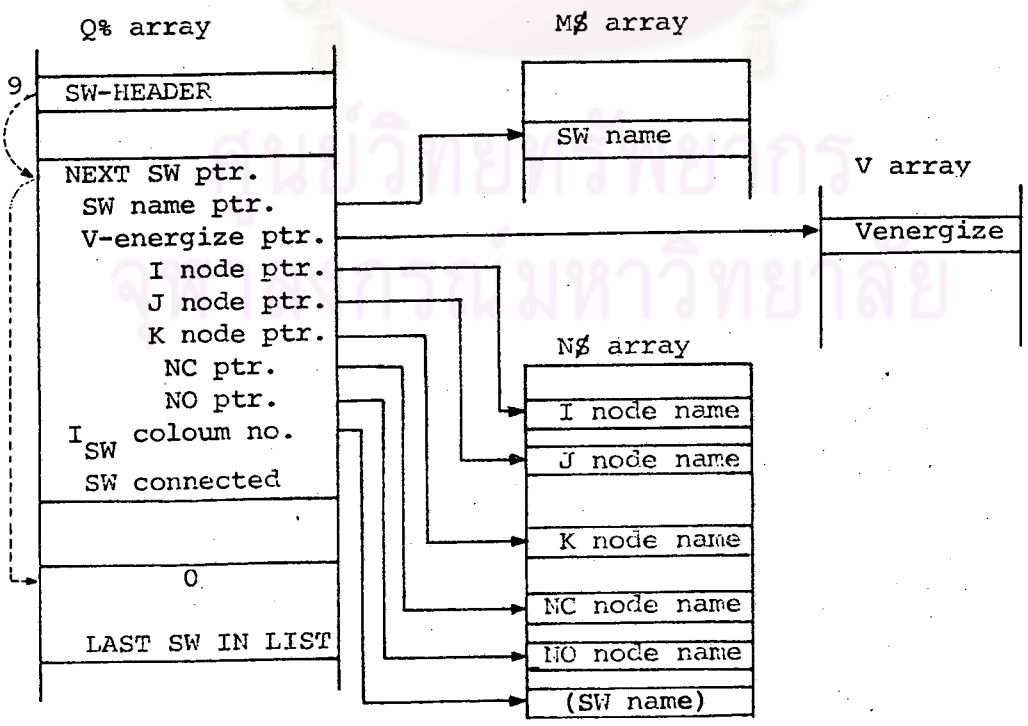
VEN = Venergize = 10 Volts

EDIT CONTROLLED SWITCH

1 SW-NAME: SW2
2 V-SWITCH: 10
3 I-NODE: I
4 J-NODE: J
5 K-NODE: K
6 NC-NODE: NC
7 NO-NODE: NO
8 INITIAL: NC

หมายเหตุ ในข้อ 6 และ 7 ถ้าโนด NC หรือ NO เป็นโนดลอยไว้เฉยๆ ให้ใส่คำว่า "NC" เพื่อบอกให้โปรแกรมรับรู้ ส่วนข้อ 8 เป็นการกำหนดภาวะเริ่มต้นของสวิตช์ว่าต่อกับโนดใดอยู่

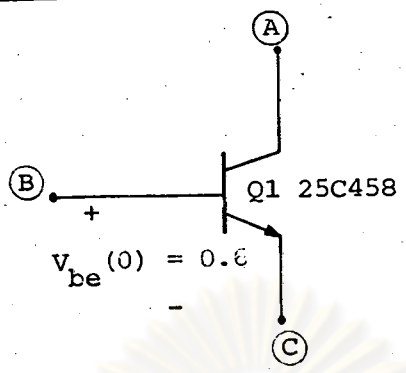
(ก)



(ข)

รูปที่ 6.9 การใส่ข้อมูลของสวิตช์

6.4.5 ทรานซิสเตอร์



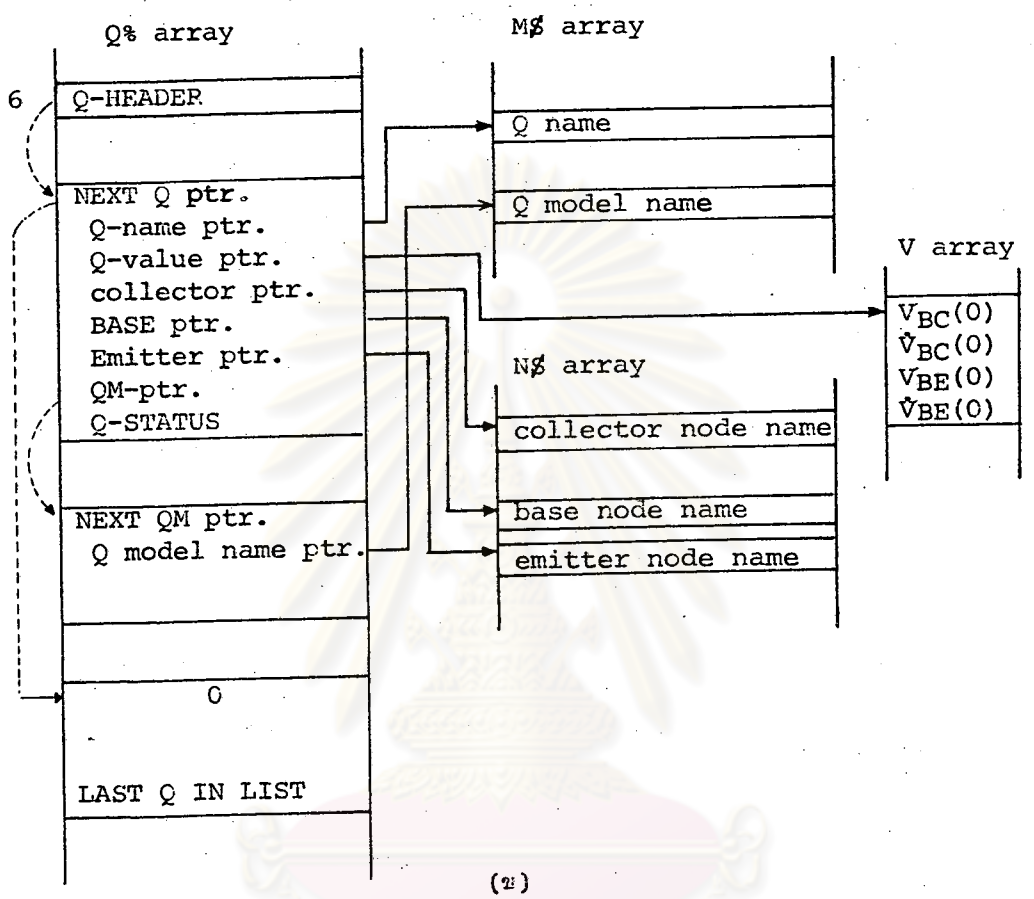
EDIT TRANSISTOR

1 Q-NAME:: Q1
2 C-NODE:: A
3 B-NODE:: B
4 E-NODE:: C
5 Q-MODEL: 25C458
6 Vbc PAST:
7 Vbe PAST: 0.6
8 Q-STATUS: 1

STATUS : 0=CUT OFF 1=ACTIVE 2=SATURATE 3=REV ACT

หมายเหตุ ในข้อ 8 ; Q-STATUS คือภาวะหรือโมด
การทำงานเริ่มต้นของทรานซิสเตอร์

(ก)



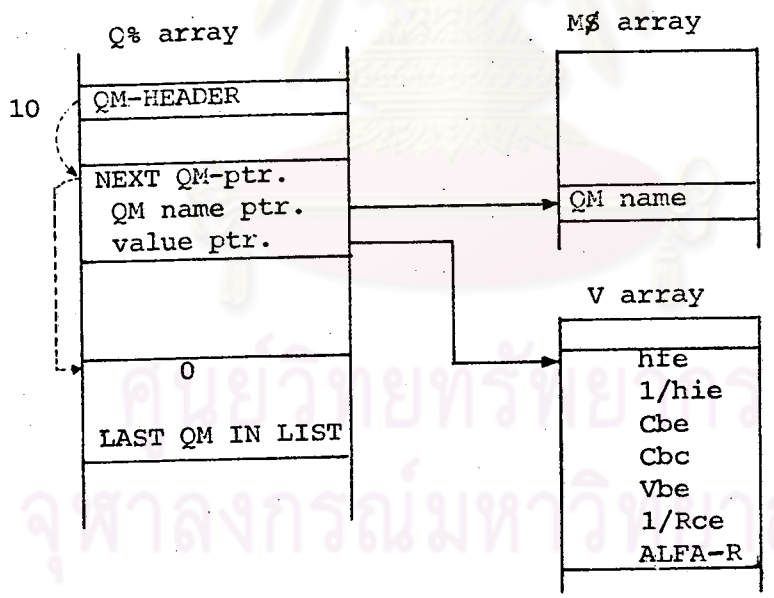
รูปที่ 6.10 การใส่ข้อมูลของทรานซิสเตอร์

6.4.6 ค่าพารามิเตอร์ประจำเบอร์ทรานซิสเตอร์ (Q-model)

```
EDIT MODEL TRANSISTOR

1   MODEL NAME          : 25C458
2   HFE VALUE( 100.00 ): 400
3   1/hie VAL( 1E-3   ): 1E-3
4   Cbe VALUE( 0.00   ):
5   Cbc VALUE( 0.00   ):
6   Vbe VALUE( 0.6    ): 0.6
7   1/RCE VAL( 0.00   ): 1E-9
8   ALFA R VAL(0.5    ): 0.4
```

หมายเหตุ ตัวเลขในเครื่องหมาย "( )" คือ ค่าพารามิเตอร์โดยทั่วไปของทรานซิสเตอร์ ส่วนในข้อ 6 ถ้าใส่ V_{be} เป็นเลขบวกจะหมายถึงทรานซิสเตอร์แบบ NPN ถ้า V_{be} เป็นเลขลบจะหมายถึงทรานซิสเตอร์แบบ PNP

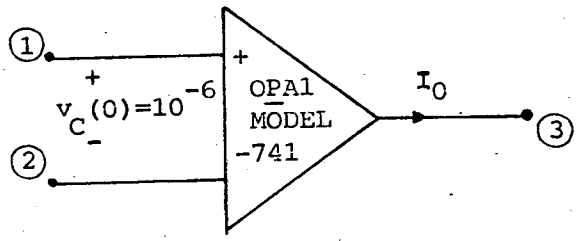
(ก)



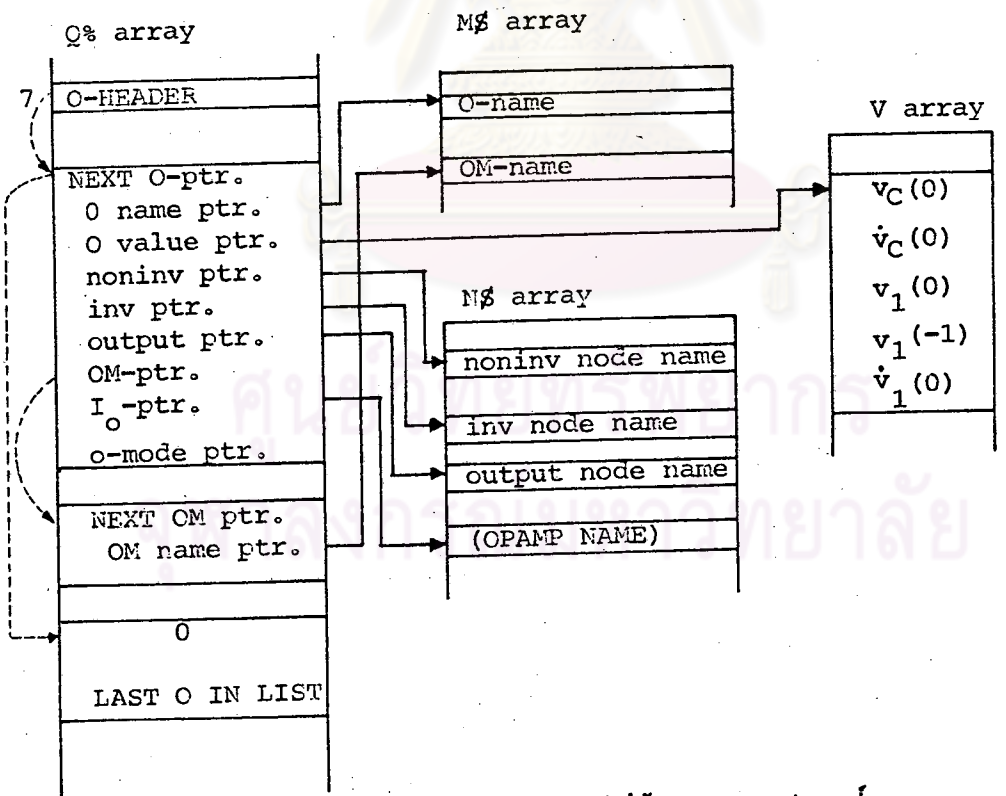
(ข)

รูปที่ 6.11 การใส่ค่าพารามิเตอร์ของทรานซิสเตอร์

6.4.7 ออปแอมป์




```
EDIT OPAMP

1   OPAMP NAME: OPA1
2  NONINV NODE: 1
3   INVER NODE: 2
4   OUTP NODE : 3
5   MODEL-TYPE: 741
6   VC PAST#1 : 1E-6
7   VO PAST#1 :
8   VO PAST#2 :
9   OPAMP STATUS1

STATUS : 0=ACT   1=POS SAT   2=NEG SAT
```



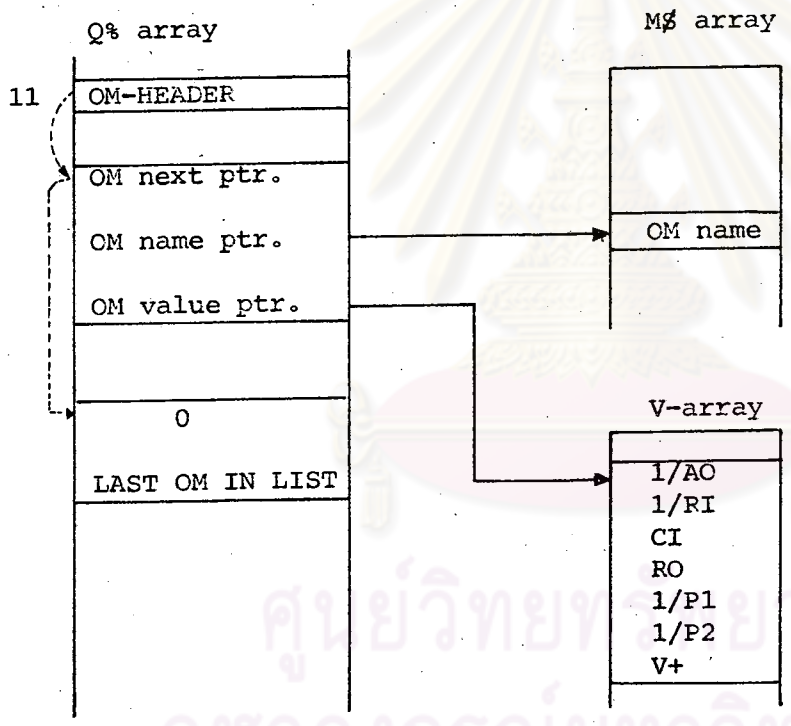
รูปที่ 6.12 การใส่ข้อมูลของออปแอมป์

6.4.8 ค่าพารามิเตอร์ของออปแอมป์ (Model Opamp)

```
EDIT MODEL OPERATIONAL AMPLIFIER

1   MODEL NAME      ): 741
2   AO-VALUE (1E+10): 1E+5
3   RI-VALUE (1E+10): 10K
4   CI-VALUE ( 0.00 ):
5   RO-VALUE ( 0.0  ): 10
6   POLE FREQ NR 1  ): 10
7   POLE FREQ NR 2  ): 1MEG
8   V-LIMITER (100  ): 15
```

หมายเหตุ ในข้อ 8 V-LIMITER คือแรงดันเอาต์พุตในภาวะอิ่มตัว

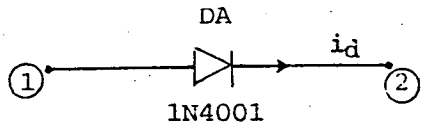
(ก)



(ข)

รูปที่ 6.13 การใส่ค่าพารามิเตอร์ของออปแอมป์

6.4.9 ไดโอด

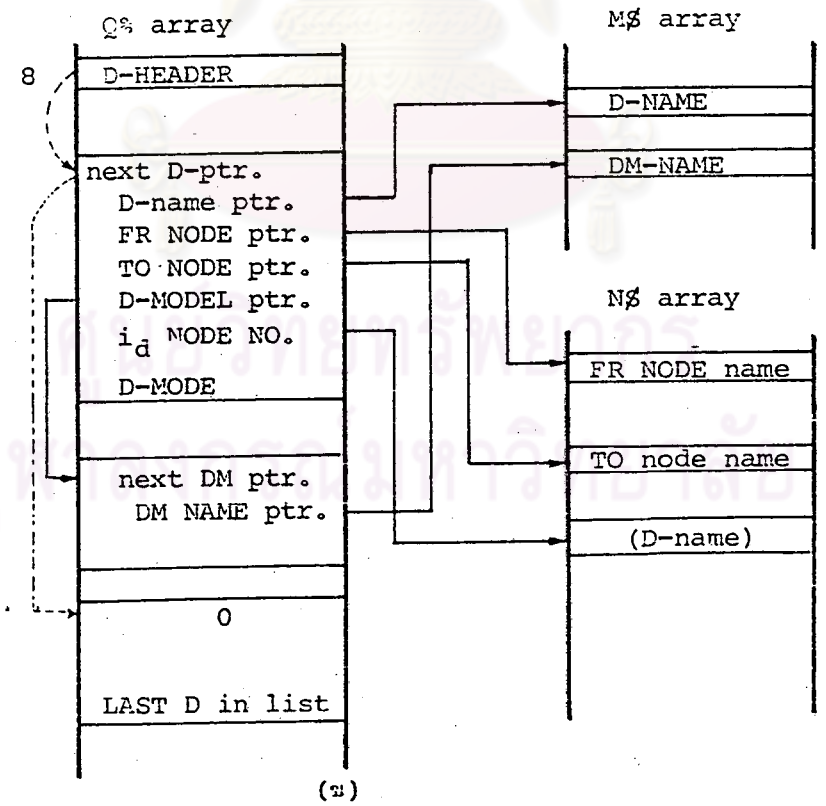


EDIT DIODE

1 D-NAME : DA
2 FR NODE: 1
3 TO NODE: 2
4 D-MODEL: 1N4001
5 D-STATUS: 1

STATUS : 0=D OFF 1=D ON

(ก)



(ข)

รูปที่ 6.14 การใส่ข้อมูลของไดโอด

6.4.10 ค่าพารามิเตอร์ของไดโอด (Model Diode)

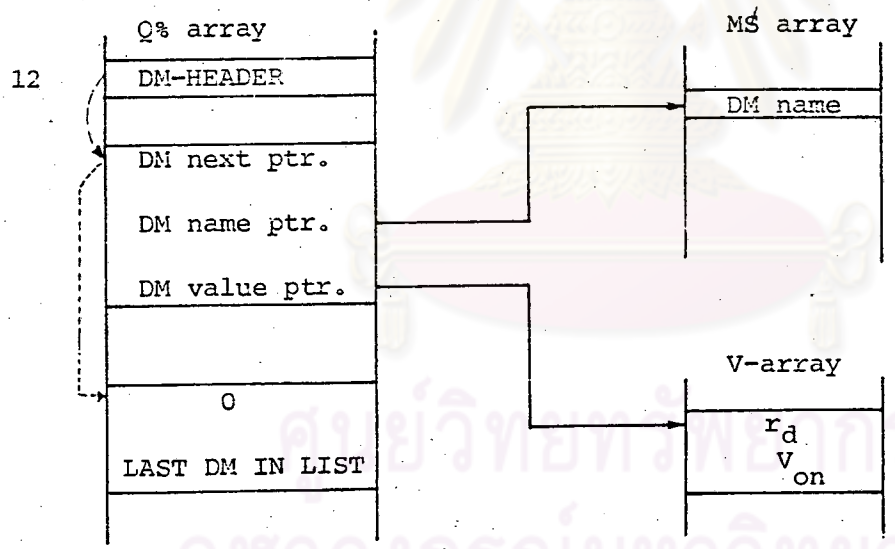
EDIT D-MODEL

1 DM-NAME :1N4001

2 RD-VALUE :10

3 V_{on}-VAL :0.6

(ก)



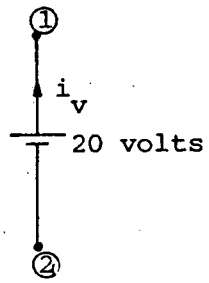
(ข)

รูปที่ 6.15 การใส่ค่าพารามิเตอร์ของไดโอด

6.4.11 แหล่งกำเนิดแรงดัน

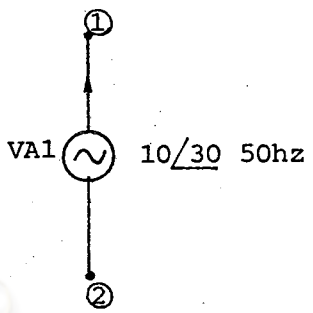
แหล่งจ่ายแรงดันเรามีให้ผู้ใช้เลือก 3 ชนิดคือ แหล่งจ่ายไฟตรง (DC) แหล่งจ่ายไฟสลับ (AC) และแหล่งจ่ายแรงดันเชิงเส้นแบบท่อน (PWL) ซึ่งลักษณะการใส่ข้อมูลแสดงไว้ในรูป 6.16 (ก) 6.16 (ข) และ 6.16 (ค) ตามลำดับ ส่วนการเก็บข้อมูลใน array ของแหล่งกำเนิดแรงดันทั้ง 3 ชนิดแสดงไว้ในรูปที่ 6.16 (ง)

```
EDIT VOLTAGE SOURCE

 TYPE OF VOLTAGE SOURCE>: DC>
 V-SOURCE NAME: V1>
 FR NODE: 1>
 TO NODE: 2>
:::DC-VOLTAGE SOURCE:::
 DC-VALUE: 20>
```



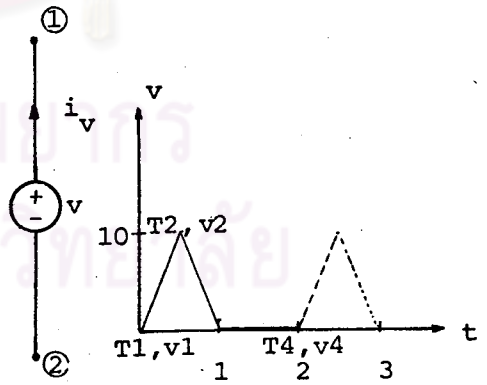
(ก)

```
EDIT VOLTAGE SOURCE

 TYPE OF VOLTAGE SOURCE>: AC>
 V-SOURCE NAME: VA1>
 FR NODE: 1>
 TO NODE: 2>
:::AC-VOLTAGE SOURCE:::
MAGNITUDE: 10>      FREQUENCY: 50>
PHASE    : 30>
```



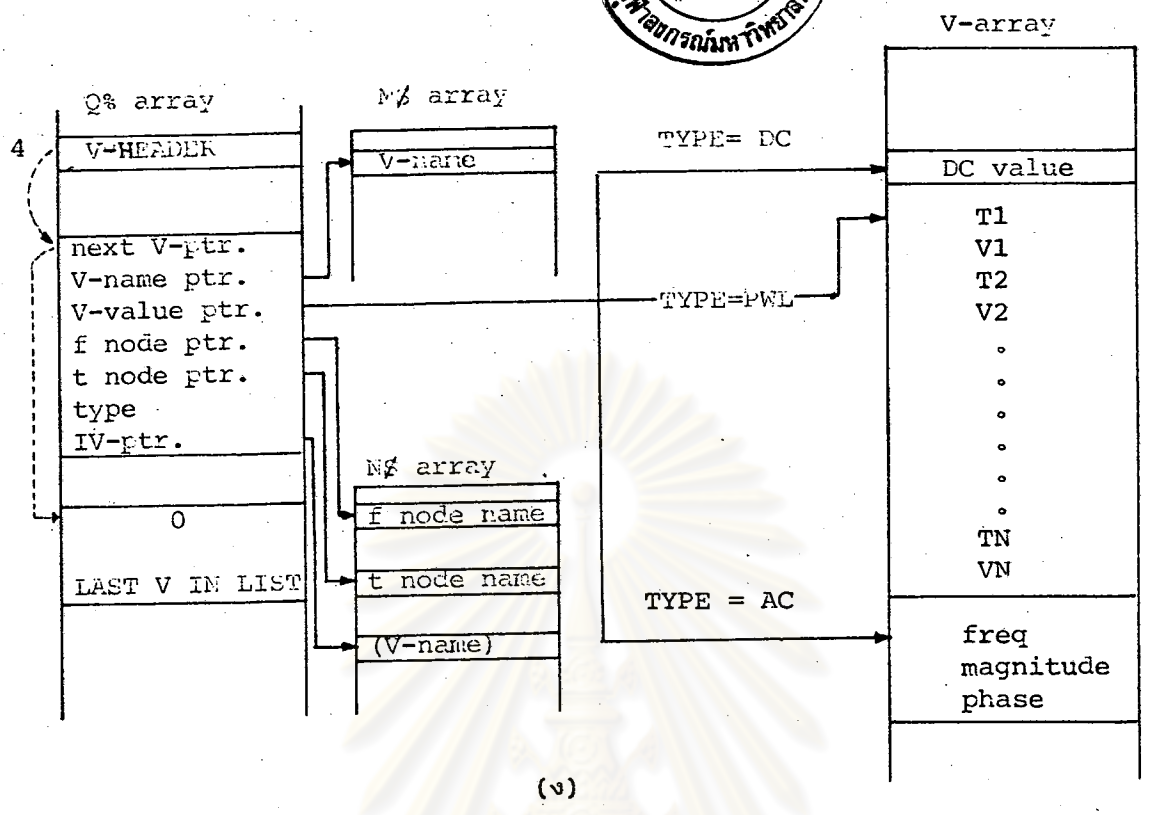
(ข)

```
EDIT VOLTAGE SOURCE

 TYPE OF VOLTAGE SOURCE>: PWL>
  V-SOURCE NAME: VP1>
  FR NODE: 1>
  TO NODE: 0>
  PUT NUMBER OF VOLTAGE POINT : 4
:::PUT VOLTAGE:::::::AT TIME T::::
  AT TIME: 0>          V-VALUE: 0>
  AT TIME: 0.5>        V-VALUE: 10>
  AT TIME: 1>          V-VALUE: 0>
  AT TIME: 2>          V-VALUE: 0>
```



หมายเหตุ รูปสัญญาณของแหล่งกำเนิดแรงดันจะทวนตัวมันเองไปเรื่อยๆ ถ้าเวลาที่คำนวณมากกว่าเวลาสูงสุดที่ผู้ใช้ใส่ข้อมูลให้

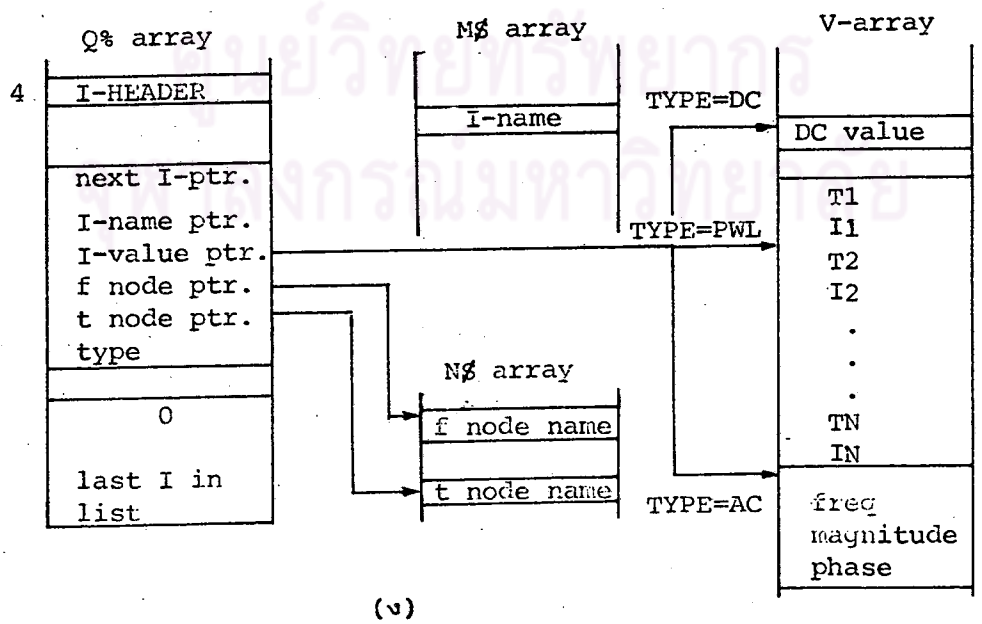
(ค)



รูปที่ 6.16 การใส่ข้อมูลของแหล่งกำเนิดแรงดัน

6.4.12 แหล่งกำเนิดกระแส

การใส่ข้อมูลของแหล่งกำเนิดกระแสมีลักษณะเหมือนการใส่ข้อมูลของแหล่งกำเนิดแรงดันทุกประการ ดังนั้นจึงแสดงเฉพาะการจัดข้อมูลใน array ของแหล่งกำเนิดกระแสดังในรูป 6.17



รูปที่ 6.17 การจัดข้อมูลของแหล่งกำเนิดกระแส

6.5 คำสั่ง SIM LOAD SAVE CHANGE DISPLAY และ QUIT

6.5.1 คำสั่ง SIM

ผู้ใช้จะเลือกคำสั่งนี้ก็ต่อเมื่อได้ใส่ข้อมูลของวงจรเสร็จสิ้นแล้วและต้องการจะวิเคราะห์วงจร ข้อมูลของวงจรจะถูกโหลดเก็บไว้ในไฟล์ข้อมูลชื่อ "SF" บนจานแม่เหล็กก่อน และก็จะเริ่มโหลดโปรแกรมในส่วน "SIM" เข้ามาแทนและทำการ RUN ซึ่งรายละเอียดจะกล่าวไว้ในหัวข้อต่อไป

6.5.2 คำสั่ง LOAD

คำสั่งนี้เป็นการโหลดข้อมูลของวงจรที่อยู่บนจานแม่เหล็กเข้ามาเก็บไว้ใน array ทั้งสี่ ผู้ใช้เพียงแต่ใส่ชื่อไฟล์ข้อมูลที่ต้องการโหลดเข้ามาเท่านั้น

6.5.3 คำสั่ง SAVE

คำสั่งนี้ทำหน้าที่เก็บข้อมูลของวงจรลงในจานแม่เหล็กโดยผู้ใช้จะต้องกำหนดชื่อไฟล์ข้อมูลวงจรเสียก่อน เพื่อจะได้เรียกออกมาดูได้ภายหลัง

6.5.4 คำสั่ง CHANGE

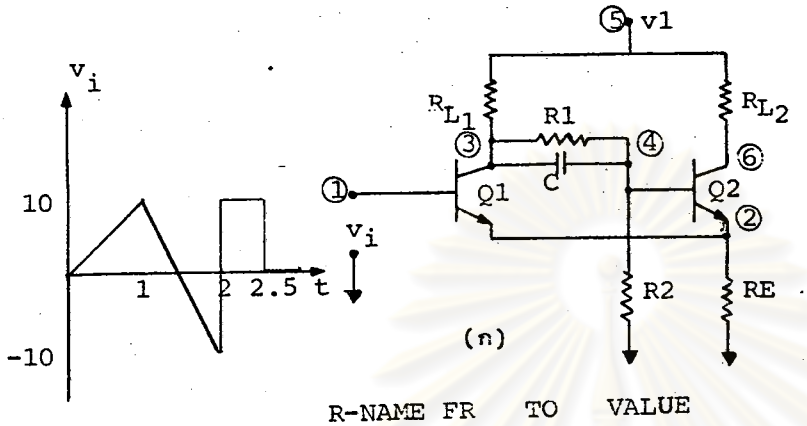
เราจะใช้คำสั่ง CHANGE เมื่อต้องการแก้ไขเปลี่ยนแปลงข้อมูลวงจรโดยโปรแกรมจะถามถึงชนิดและชื่อของอุปกรณ์ที่เราต้องการเปลี่ยนแปลงค่าดังเมนูที่แสดงในรูปที่ 6.18 และก็จะให้ข้อมูลของอุปกรณ์ที่เราต้องการเหมือนในหัวข้อ 6.5 เพื่อให้ผู้ใช้ใส่ค่าใหม่เข้าไป

```
CHANGE COMPONENT DATA
  1  CHANGE R
  2  CHANGE L
  3  CHANGE C
  4  CHANGE Q
  5  CHANGE O
  6  CHANGE Q-MODEL
  7  CHANGE O-MODEL
  8  CHANGE V-SOURCE
  9  CHANGE I-SOURCE
  A  CHANGE DIODE
  B  CHANGE SWITCH
  C  CHANGE D-MODEL
  D  OUT OF CHANGE MODE


PUT MODE NUMBER YOU WANT:

PUT COMPOMENT'S NAME:
```

รูปที่ 6.18 เมนูของคำสั่ง CHANGE

6.5.5 คำสั่ง DISPLAY

DISPLAY เป็นคำสั่งที่แสดงผลข้อมูลของวงจรที่อยู่ใน array ทั้งสี่ การแสดงข้อมูลนั้นจะเรียงชนิดของอุปกรณ์เป็นอย่างๆ ไป ยกตัวอย่างวงจรในรูป 6.19 (ก) ซึ่งแสดงผลข้อมูลเป็นดังรูป 6.19 (ข)



(ก)

```
R-NAME FR   TO   VALUE
RL1    3    5    4700.00001
RL2    5    6    3900.00001
R1     3    4    22000
R2     4    0    22000
RE     2    0    2200
C-NAME FR   TO   C-VALUE       V PREV1
C      3    4    1E-06         10.7583
Q-NAME C   B  E   MODEL VBE PREV VCB PREV
Q1     3   1  2   2SC   0         -3.3672
Q2     6   4  2   2SC   0         .71961
MODEL:HFE:HIE:CBE:CBC:VBE:RCE:ALFA-R
2SC 100 1000 1E-10 0 .7 9.99999999E+19  .5
V-NAME FR  TO    # VOLTAGE TYPE
V1     5   0          DC
 DC-VALUE=12
V-NAME FR  TO    # VOLTAGE TYPE
V2     1   0          PWL
AT TIME:0          V-VALUE:0
AT TIME:1          V-VALUE:10
AT TIME:2          V-VALUE:-10
AT TIME:2          V-VALUE:10
AT TIME:2.5        V-VALUE:10
AT TIME:2.5        V-VALUE:0
AT TIME:3          V-VALUE:0
PRESS ANY KEY TO  CONTINUE:
```

(ข)

รูปที่ 6.19 ลักษณะการแสดงผลข้อมูลของวงจร

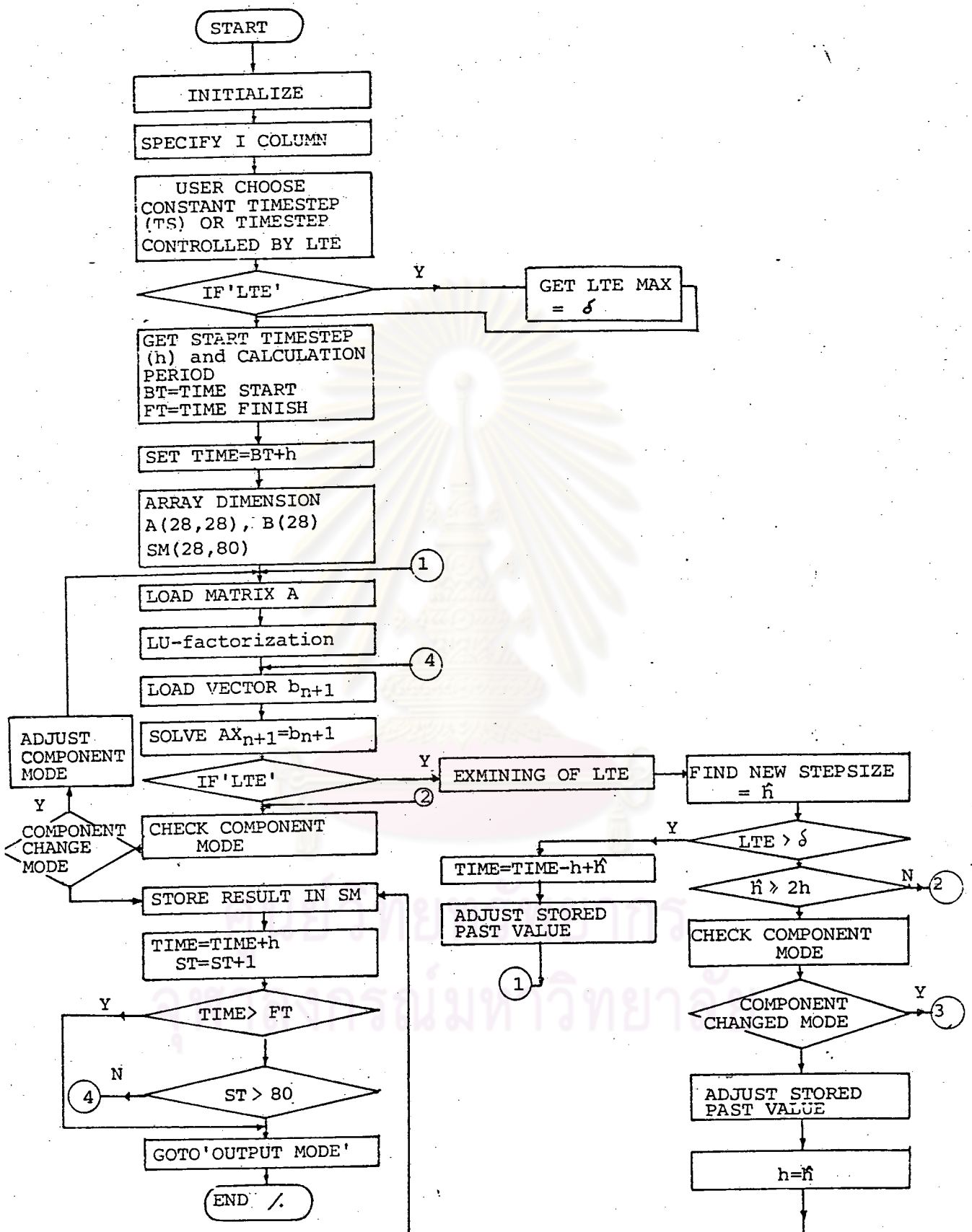
6.6 ข้อมูลทางเทคนิคและการใช้งานในโมด SIM

ในโมดนี้เราสามารถแบ่งโปรแกรมออกเป็น 2 ส่วน คือ ส่วนวิเคราะห์วงจรและส่วนเอาต์พุท ส่วนวิเคราะห์วงจรจะประกอบด้วย ส่วนสร้างและแก้สมการเมตริก ส่วนตรวจสอบและปรับขนาดขั้นเวลา ส่วนตรวจสอบและปรับภาวะการทำงานของอุปกรณ์ไม่เชิงเส้น โปรแกรมจะรับข้อมูลของวงจรจากโมด INPUT ได้ 2 ทาง คือ รับจากหน่วยความจำส่วนบนที่โปรแกรมได้กันพื้นที่สำหรับเก็บค่าต่างๆ ไว้แล้ว หรือรับผ่านทางจานแม่เหล็ก (จะรับข้อมูลผ่านจานแม่เหล็กเมื่อผู้ใช้ทำการคำนวณวงจรซ้ำในช่วงเวลาเดิมเท่านั้น) ส่วนข้อมูลในด้านที่เกี่ยวกับการวิเคราะห์วงจรเช่น ขนาดขั้นเวลา h และช่วงเวลาที่ต้องการวิเคราะห์นั้น ผู้ใช้จะต้องใส่ข้อมูลให้โปรแกรมในส่วนวิเคราะห์วงจรนี้ หลังจากได้ข้อมูลต่างๆ ครบแล้วโปรแกรมก็จะเริ่มทำการวิเคราะห์และเมื่อเสร็จสิ้นการคำนวณโปรแกรมจะเข้าสู่ส่วนเอาต์พุทซึ่งประกอบด้วยโปรแกรมส่วนเอาต์พุทเพื่อให้ผู้ใช้เลือกคำสั่งที่ต้องการจะทำต่อ เช่น แสดงผลคำตอบ ทำการพิมพ์คำตอบ ทำการคำนวณซ้ำ หรืออื่นๆ ตามต้องการ

6.6.1 โปรแกรมส่วนวิเคราะห์วงจร

เราอาจกล่าวได้ว่าส่วนวิเคราะห์วงจรนั้นเป็นหัวใจของโปรแกรม 'TAPE' เนื่องจากขั้นตอนการวิเคราะห์วงจร และเทคนิคต่างๆ ที่ได้กล่าวมาตั้งแต่บทที่ 1 ถึง 5 นั้นได้ถูกนำมาใช้ในโปรแกรมส่วนนี้ทั้งหมดซึ่งสามารถพิจารณาได้จากโฟลว์ชาร์ต (Flowchart) ของโปรแกรมส่วนวิเคราะห์วงจรในรูปที่ 6.20 และรายละเอียดของโฟลว์ชาร์ตซึ่งจะได้อธิบายโดยอ้างถึงเนื้อหา และสมการในบทที่ผ่านๆ มา ดังต่อไปนี้

เมื่อเข้าสู่ส่วนวิเคราะห์วงจร โปรแกรมจะแสดงข้อมูลดังในรูป 6.21 เพื่อให้ผู้ใช้เลือกว่าจะทำการวิเคราะห์แบบให้ขั้นเวลาคงที่ (TS) หรือให้ขั้นเวลาเปลี่ยนตามค่าของ LTE ซึ่งต้องกำหนดค่า LTE MAX (δ) กำหนดขั้นเวลาเริ่มต้น (h) และสุดท้ายเป็นการกำหนดช่วงเวลาที่ต้องการวิเคราะห์ (BT, FT) หลังจากนั้นโปรแกรมก็จะกำหนดขนาดของเมตริกที่จะใช้คือ A(28,28), B(28), และ SM(28,80) และเริ่มการวิเคราะห์ที่จุดเวลา BT+h โดยการสร้างสมการเมตริกจตุรัสของวงจร (เมตริก A) ด้วยวิธีโมดิฟายด์โดนัลโดยอาศัยตราประจำอุปกรณ์ตามที่กล่าวถึงในหัวข้อที่ 2.3, 3.8, 4.3 - 4.6 โครงสร้างการโหลดข้อมูลของเมตริก A แสดงไว้ในรูป 6.22 จากนั้นก็ทำการแปลงเมตริก A ให้อยู่ในรูปของเมตริก LxU ด้วยวิธีแยกตัวประกอบแอล-ยู เก็บไว้ในเมตริก A เดิม เวคเตอร์ค่าคงที่ b_{n+1} จะถูกสร้างขึ้นภายหลังและเก็บไว้ในเมตริก B ซึ่งโครงสร้างการโหลดข้อมูลในเมตริกของตัวแปรขาเข้าและค่าในอดีตต่างๆ แสดงไว้ในรูป 6.23 หลังจากนั้นโปรแกรมจะทำการแก้สมการ Ax=B หาคำตอบออกมา ถ้าผู้ใช้เลือกให้มีการปรับขนาดขั้นเวลาตามค่าของ LTE คำตอบของสมการวงจรที่ได้จะถูกนำไปคำนวณหาค่า LTE ของอุปกรณ์พลวัต แต่ละตัวและดำเนินการไปตามขั้นตอนในหัวข้อที่ 5.3 แต่ถ้าผู้ใช้กำหนดให้ขั้นเวลาคงที่ (TS) โปรแกรมก็จะข้ามส่วนตรวจสอบและปรับขนาดขั้นเวลานี้ไปทำการวิเคราะห์ในส่วนตรวจสอบและปรับภาวะการทำงานของอุปกรณ์ไม่เชิงเส้นต่อไป ถ้าอุปกรณ์ไม่เชิงเส้นเปลี่ยนภาวะการทำงานโปรแกรมส่วนนี้จะทำการเปลี่ยนภาวะของอุปกรณ์ที่เก็บไว้ใน Q% array เพื่อให้การโหลดข้อมูลของอุปกรณ์ในการสร้างเมตริก A ถูกต้องตามภาวะการทำงานจริงและทำการสร้างและแก้สมการวงจรซ้ำที่จุดเวลาเดิม



รูปที่ 6.20 ไฟลว์ชาร์ตการทำงานในโมด'SIM'

SIMULATED PROG MODE:

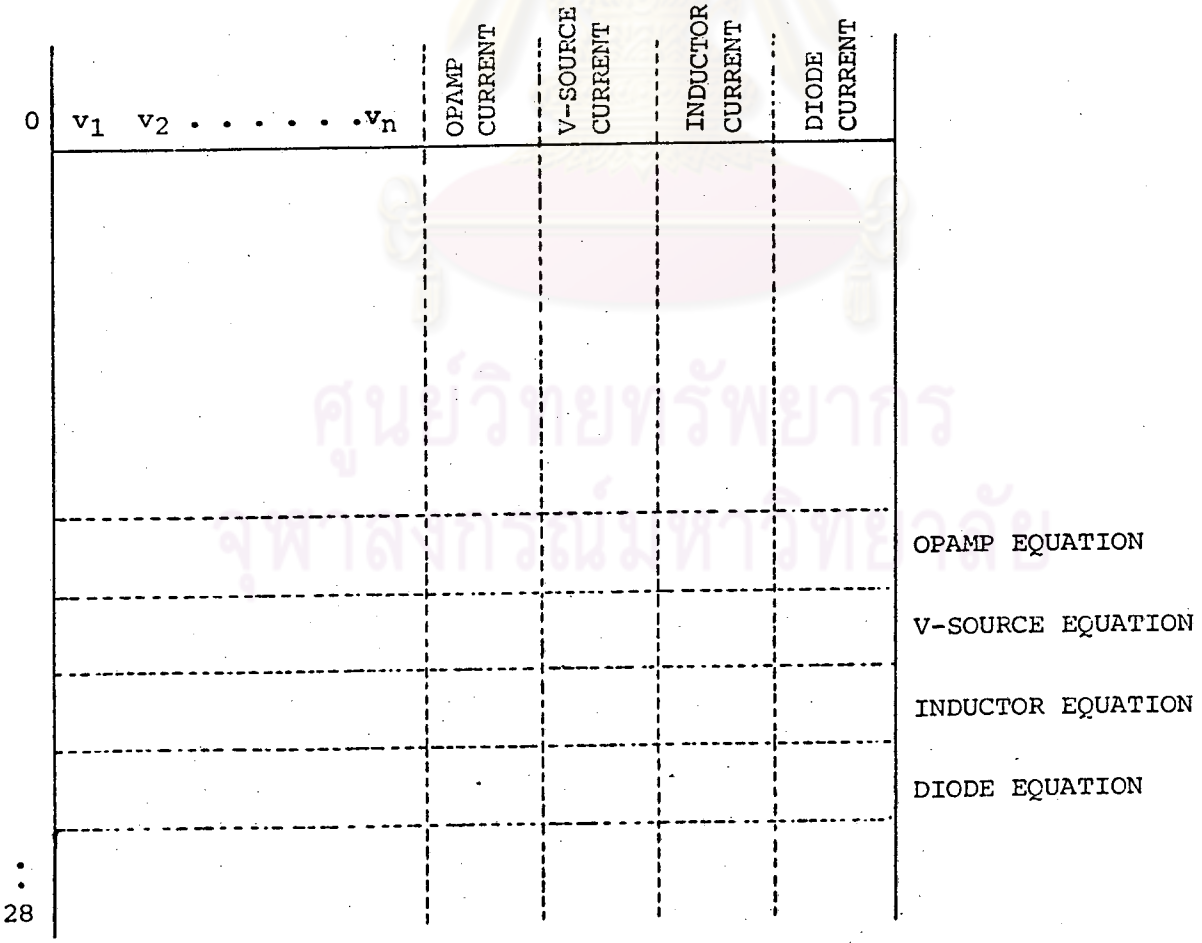
PUT LOCAL TRUNCATION AVAILABLE[LTE] OR TIMESTEP[TS]:___

INPUT START TIMESTEP:___

INPUT BEGIN TIME:___

INPUT FINISH TIME:___

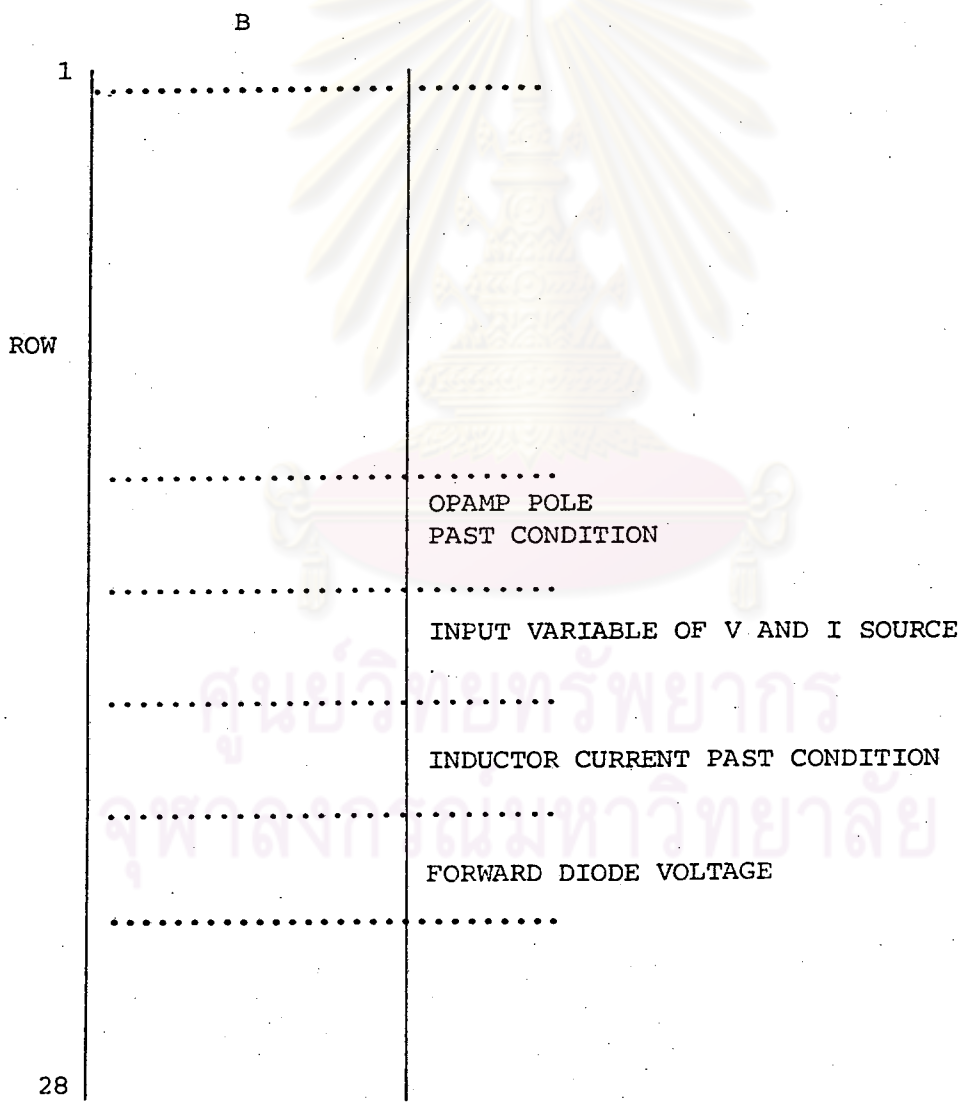
รูปที่ 6.21 การแสดงข้อมูลทางจอภาพเมื่อเข้าโมดวิเคราะห์วงจร



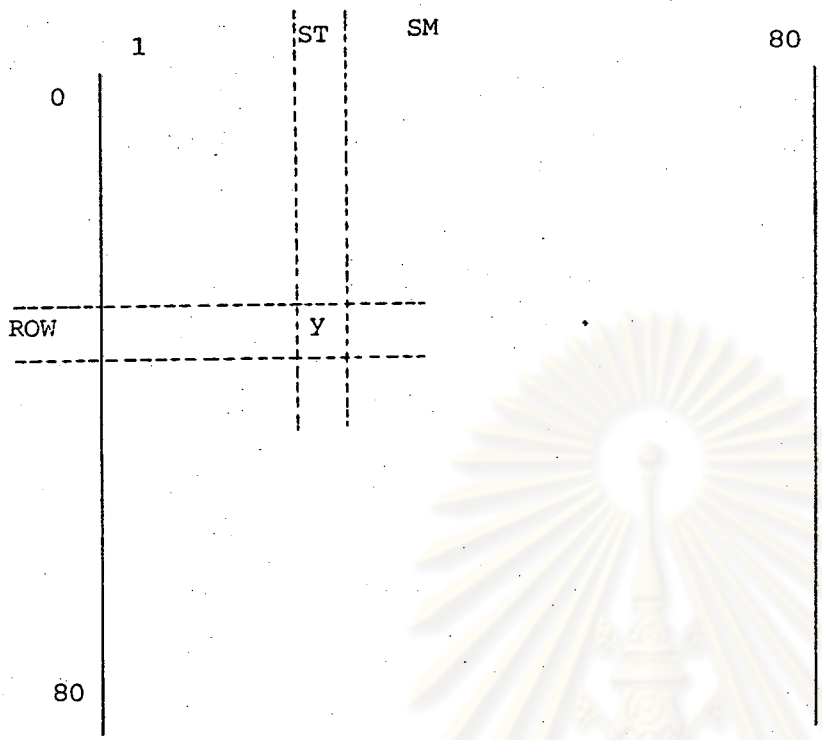
รูปที่ 6.22 โครงสร้างการโหลดข้อมูลในเมตริก A

ถ้าหากอุปกรณ์ไม่เชิงเส้นยังทำงานในภาวะเดิมคำตอบที่ได้ (แรงดันโนดทุกตัว กระแสของตัวเหนี่ยวนำ กระแสของสวิตช์ กระแสของออปแอมป์ และ กระแสของแหล่งกำเนิดแรงดัน) จะถูกเก็บไว้ในเมตริก SM ซึ่งเป็นเมตริกที่เก็บคำตอบของแต่ละจุดเวลาและมีโครงสร้างดังในรูปที่ 6.24 ขณะเดียวกันก็ทำการปรับค่าในอดีตของอุปกรณ์พลวัตทุกตัวด้วย จากนั้นโปรแกรมจะไปทำการคำนวณที่จุดเวลาถัดไปถ้าหากจุดเวลานั้นมีค่าไม่เกินเวลาที่ผู้ใช้ต้องการ [FT] หรือถ้าเมตริก SM ไม่เต็มเสียก่อน (นั่นคือ ผ่านจุดการคำนวณมาแล้ว 80 จุด)

เมื่อเสร็จสิ้นการคำนวณโปรแกรมส่วนเอาต์พุทจะรับหน้าที่ต่อโดยการแสดงข้อมูลทางจอภาพดังในรูปที่ 6.25 เพื่อให้ผู้ใช้เลือกคำสั่งที่ต้องการ ซึ่งมีอยู่ด้วยกันทั้งหมด 6 คำสั่ง ซึ่งเราจะอธิบายวิธีการใช้งานทั้ง 6 คำสั่งในหัวข้อต่อไป



รูปที่ 6.23 แสดงการโหลดข้อมูลลงในเมตริก B



หมายเหตุ:

ในแต่ละ column(1 ถึง 80) ของ SM จะเก็บค่าตัวแปรของสมการที่จุดเวลาหนึ่งๆ และที่ตำแหน่ง SM(0,ST) ของแต่ละ COLUMN จะเก็บค่าจุดเวลาที่ทำการวิเคราะห์ ตามรูป SM(B(ROW),ST)=y และ y= B (ROW)

รูปที่ 6.24 โครงสร้างการเก็บผลการคำนวณในเมตริก SM

SIMULATED PROG MODE

[1] LIST RESULT

[2] CONTINUE CALCULATION

[3] REPEAT CALCULATION

[4] GOTO INPUT PROGRAM

[5] PRINT RESULT

[6] END

รูปที่ 6.25 การแสดงผลข้อมูลทางจอภาพในโมดเอาต์พุท

๖.๖.๒ คำสั่งในส่วนเอาต์พุท

LIST RESULT และ PRINT RESULT เป็นคำสั่งแสดงคำตอบของสมการวงจรที่เก็บไว้ในเมตริก SM ซึ่งทั้งสองคำสั่งนี้มีวิธีใช้งานเหมือนกันต่างกันตรงที่ LIST RESULT จะไม่มีการพิมพ์ผลข้อมูลออกทางเครื่องพิมพ์เท่านั้น ผู้ใช้เพียงแต่กำหนดชื่อของโนดถ้าต้องการดูค่าแรงดันหรือกำหนดชื่อของไดโอด ออปแอมป์ แหล่งจ่ายแรงดัน ตัวเหนี่ยวนำและสวิตช์ถ้าต้องการดูค่ากระแสไหลผ่านอุปกรณ์เหล่านี้โปรแกรมก็จะพิมพ์ผลให้ดังตัวอย่างในรูปที่ ๖.๒๖ ซึ่งเป็นคำตอบของวงจรในรูปที่ ๖.๑๙ (ก)

รูปที่ ๖.๒๖ การแสดงผลทางจอภาพเมื่อ LIST RESULT และ PRINT RESULT

CONTINUE CALCULATION เมื่อผู้ใช้ต้องการคำนวณต่อจากจุดเวลาเดิมก็ให้ใช้คำสั่งนี้โดยกำหนดจุดเวลา (FT) ใหม่ตามต้องการ

REPEAT CALCULATION เมื่อผู้ใช้ต้องการทำการคำนวณซ้ำในช่วงเวลาเดิม เพื่อทดลองหรือเปรียบเทียบผลการคำนวณโดยการเปลี่ยนแปลงขนาดขั้นเวลาหรือ LTE นั้นก็ให้ใช้คำสั่งนี้ โมดนี้ก็จะทำการโหลดข้อมูลของวงจรจากจานแม่เหล็ก และภาวะเริ่มต้น (initial value) เข้ามาใหม่

GOTO INPUT PROGRAM คำสั่งนี้จะทำหน้าที่โหลดโปรแกรมโมด INPUT เข้ามาแทนที่โปรแกรมโมด SIM ใช้กรณีที่ผู้ใช้ต้องการกลับไปโมด INPUT เพื่อสร้างหรือแก้ไขวงจรใหม่

END ใช้เมื่อต้องการออกจาก 'TAPE'