

บทที่ 3

การพัฒนาโปรแกรม



3.1 จุดประสงค์การพัฒนาโปรแกรม

3.1.1 โปรแกรมควบคุมการทำงานของพลอตเตอร์

โปรแกรมส่วนนี้ใช้ภาษาแอสเซมบลีของไมโครโปรเซสเซอร์เบอร์ Z80

ในการพัฒนา มีจุดประสงค์ดังนี้

- 1) พัฒนาคำสั่งใหม่ขึ้นใช้งาน
- 2) พัฒนาการส่งข่าวไปยังไมโครคอมพิวเตอร์เมื่อพลอตเตอร์ทำงานจนหมดชุดของคำสั่งชุดแรกแล้ว

งานจนหมดชุดของคำสั่งชุดแรกแล้ว

- 3) ขยายบัพเฟอร์แอเรีย จาก 80 ไบต์ เป็น 1.5 กิโลไบต์ เพื่อให้เก็บคำสั่งที่ส่งมาจากไมโครคอมพิวเตอร์ได้มากขึ้น

คำสั่งใหม่ที่พัฒนาขึ้นได้แก่

- 1) PA (a,b) เป็นคำสั่งให้เลื่อนปากกาเป็นเส้นตรงจากจุดเดิมในขณะนั้นไปยังจุดใด ๆ โดยจุดนั้นมีตำแหน่งห่างจากจุดเริ่มต้นไปทางขวาตามแนวแกน X เป็นระยะทาง a/100 นิ้ว และห่างจากจุดเริ่มต้นไปด้านบนตามแนวแกน Y เป็นระยะทาง b/100 นิ้ว

- 2) PR ([-]a, [-] b) เป็นคำสั่งให้เลื่อนปากกาเป็นเส้นตรงจากจุดเดิมในขณะนั้นไปยังจุดใด ๆ โดยจุดนั้นมีตำแหน่งห่างจากจุดเดิมไปทางขวา หรือทางซ้ายตามแนวแกน X เป็นระยะทาง a/100 นิ้ว และห่างจากจุดเริ่มต้นไปด้านบนหรือด้านล่างตามแนวแกน Y เป็นระยะทาง b/100 นิ้ว

a เป็นตัวเลข 4 หลัก ตั้งแต่ 0000_{10} ถึง 1500_{10}

b เป็นตัวเลข 4 หลัก ตั้งแต่ 0000_{10} ถึง 1000_{10}

3) CH_<character string>_ เป็นคำสั่งให้เขียนตัวอักษรในแนวนอน

4) CV_<character string>_ เป็นคำสั่งให้เขียนตัวอักษรในแนวตั้ง

ตัวอักษรได้แก่ ตัวเลข 0 ถึง 9 ตัวอักษรภาษาอังกฤษ A ถึง Z และ

ตัวอักษรพิเศษ

5) Sn เป็นคำสั่งกำหนดขนาดของตัวอักษร

n เป็นตัวเลข 1 ถึง 5

3.1.2 โปรแกรมย่อยภาษาเบสิก

โปรแกรมส่วนนี้เป็นโปรแกรมภาษาเบสิกในลักษณะของชุดของโปรแกรมย่อย

สำหรับใช้กับพลอตเตอร์ซีพี-01 มีจุดประสงค์ในการพัฒนาคือ

สร้างชุดของโปรแกรมย่อยภาษาเบสิกเพื่อให้โปรแกรมภาษาเบสิกเรียกใช้ในการ

สร้างรูปภาพ โดยมีหน้าที่หลักดังนี้

- 1) ควบคุมการรับส่งข้อมูล
- 2) พลอตข้อมูล
- 3) เขียนแกน
- 4) เขียนข้อความและตัวเลขประกอบรูปภาพ

โปรแกรมย่อยเหล่านี้ได้แก่

1) INIT เป็นโปรแกรมสั่งให้อุปกรณ์ฮาร์ดแวร์ของไมโครคอมพิวเตอร์เตรียมการรับส่งข้อมูลและสั่งให้พลอตเตอร์เตรียมพร้อมที่จะทำงาน

2) FACTOR เป็นโปรแกรมกำหนดค่าสูงสุดและต่ำสุดของค่า X และค่า Y ความยาวของทั้ง 2 แกน

3) AXIS เป็นโปรแกรมเขียนแกน X และแกน Y ชีตบนแกนตัวเลขบนแกน และข้อความบนรูปภาพ

4) LOCATE เป็นโปรแกรมยกปากกาไปวางจุดที่จะเริ่มลากเส้นกราฟ

5) PLOTR เป็นโปรแกรมลากเส้นจากจุดเดิมไปยังจุดสัมบูรณ์ (Absolute) ใด ๆ

6) PLOTR เป็นโปรแกรมลากเส้นจากจุดเดิมไปยังจุดสัมพัทธ์ (Relative) ใด ๆ

7) CHAR เป็นโปรแกรมเขียนตัวอักษร ณ ตำแหน่งใด ๆ ในแนวตั้ง หรือแนวนอน โดยเลือกได้ 5 ขนาด

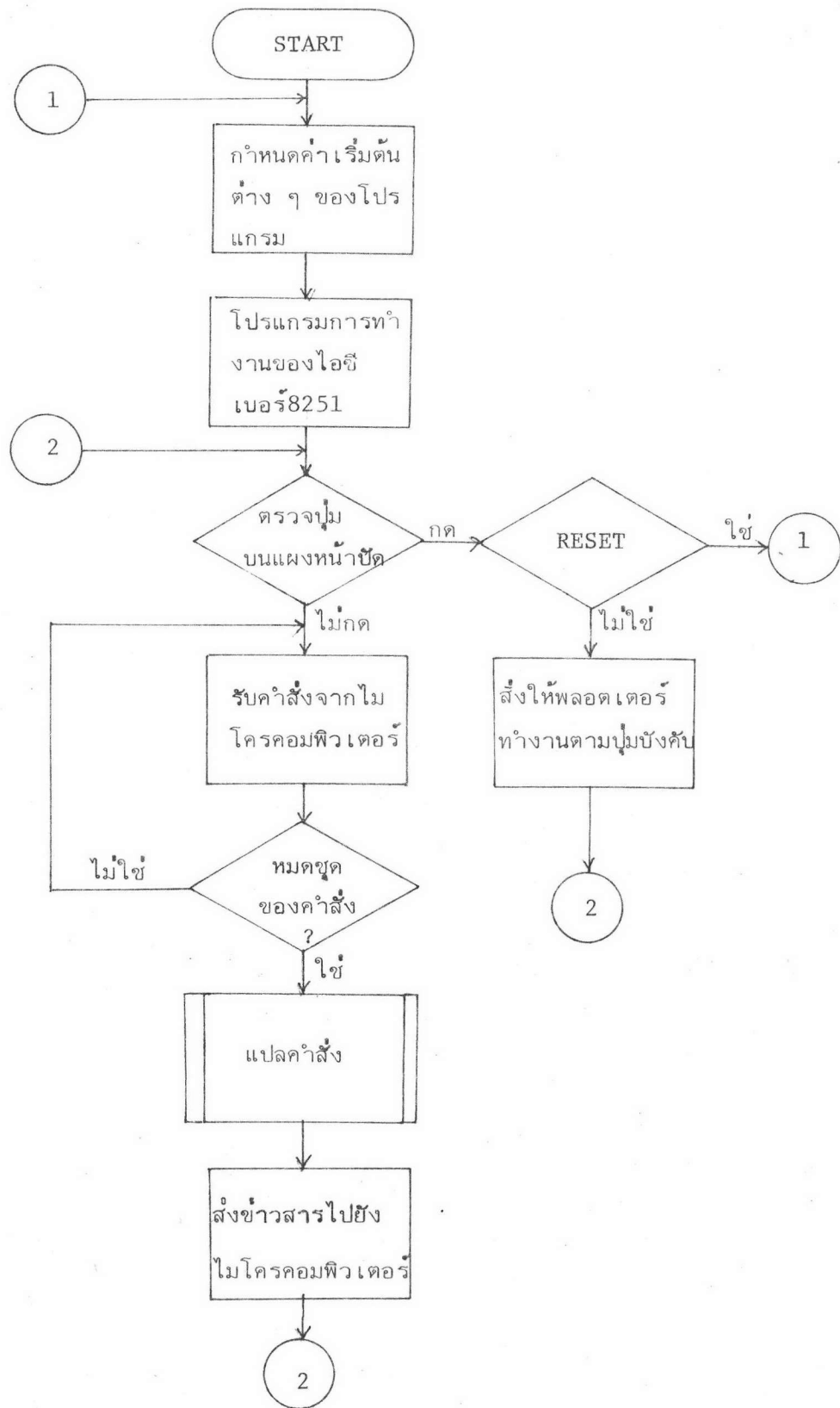
8) CIRCLE เป็นโปรแกรมเขียนวงกลม ณ ตำแหน่งใด ๆ

9) QUIT เป็นโปรแกรมสั่งเลิกการทำงานของพลอตเตอร์

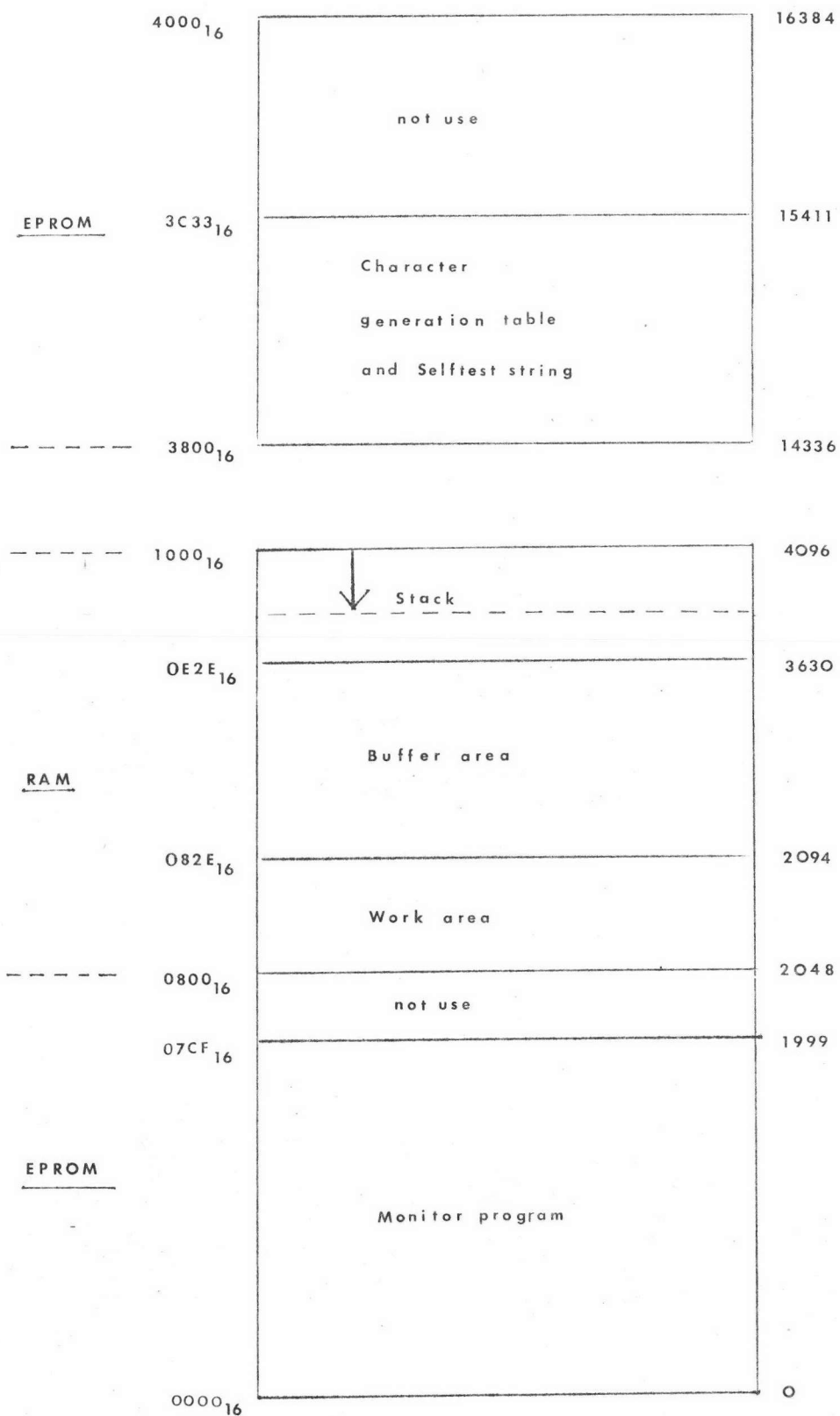
10) CPCOM เป็นโปรแกรมส่งคำสั่งที่ถูกต้องตามรูปแบบของพลอตเตอร์ซีพี-01

3.2 การพัฒนาโปรแกรมควบคุม

ผังการทำงานดังแสดงในรูป 3.1



รูปที่ 3.1 ผังการทำงานของโปรแกรมควบคุมฟลอปเตอร์ที่พัฒนาขึ้น



รูปที่ 3.2 แสดงการแบ่งหน่วยความจำของพลอตเตอร์ซีพี -01 ที่พัฒนาขึ้น

3.2.1 การทำงานของคำสั่งใหม่ที่พัฒนาขึ้นใช้งาน

1) คำสั่ง PA (Plot Absolute)

เมื่อโปรแกรมแปลคำสั่งได้รับคำสั่ง PA จะแปลงตัวเลขที่ตามหลังคำสั่ง ให้นำเป็นฐาน 16 ทหาระยะทางระหว่างตำแหน่งปัจจุบันกับตำแหน่งปลายทาง หากทิศทางของ จุดปลายทาง ทำการเชื่อมเส้นระหว่างจุดปัจจุบันและจุดปลายทาง การที่คำสั่งนี้ทำงานได้อย่างสมบูรณ์เนื่องจากการพัฒนาโปรแกรมการแปลคำสั่ง PA คือ

มีการเก็บตำแหน่งปัจจุบันของปากกาไว้ซึ่งเมื่อปากกาอยู่ที่จุดเริ่มต้นจะมีค่าเป็นศูนย์ เมื่อปากกาเคลื่อนที่ไปทางขวาหรือทางซ้าย 1 ล่เตป ตัวเก็บตำแหน่งของ x จะมีค่าเพิ่มขึ้นหรือลดลง 1 และเมื่อปากกาเคลื่อนที่ไปด้านบนหรือด้านล่าง 1 ล่เตป ตัวเก็บตำแหน่งของ y จะมีค่าเพิ่มขึ้นหรือลดลง 1 ไม่ว่าจะเคลื่อนที่ของปากกาจะเกิดขึ้นเพราะคำสั่งใด หรือเพราะการกดปุ่มบนแผงหน้าปัดก็ตามโปรแกรมจะนับจำนวนล่เตปของการเคลื่อนที่อยู่ตลอดเวลา ตัวเก็บตำแหน่งของ x มีค่าตั้งแต่ 0000_{10} ถึง 1500_{10} และตัวเก็บตำแหน่งของ y มีค่าตั้งแต่ 0000_{10} ถึง 1000_{10}

2) คำสั่ง PR (Plot Relative)

เมื่อโปรแกรมแปลคำสั่งได้รับคำสั่ง PR จะแปลงตัวเลขที่ตามหลังคำสั่ง ให้นำเป็นฐาน 16 แล้วนำไปบวกหรือลบกับตำแหน่งปัจจุบันของปากกา ผลลัพธ์ที่ได้จะเป็นตำแหน่งปลายทางที่ปากกาจะเคลื่อนที่ไป เมื่อได้ตำแหน่งปลายทางแล้ว โปรแกรมจะสั่งการทำงานให้แก่ โปรแกรม PA ฝั่งการทำงานของโปรแกรม PR ดังในรูปที่ 3.4

3) คำสั่ง Sn (Size)

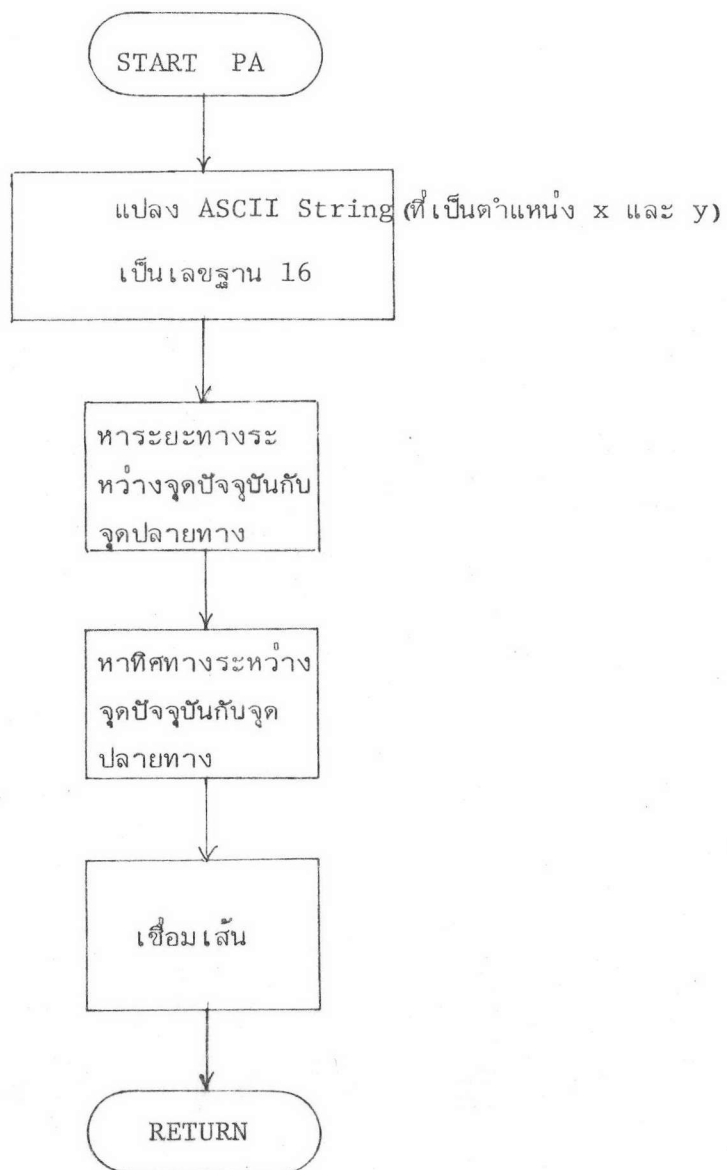
เมื่อโปรแกรมแปลคำสั่งได้รับคำสั่ง S จะเปลี่ยนแปลงค่าของตัวคูณขนาดตัวอักขระตามที่ได้รับคำสั่ง ดังนี้

S1 หมายถึง ตัวอักขระมีขนาดเป็น 2 เท่า

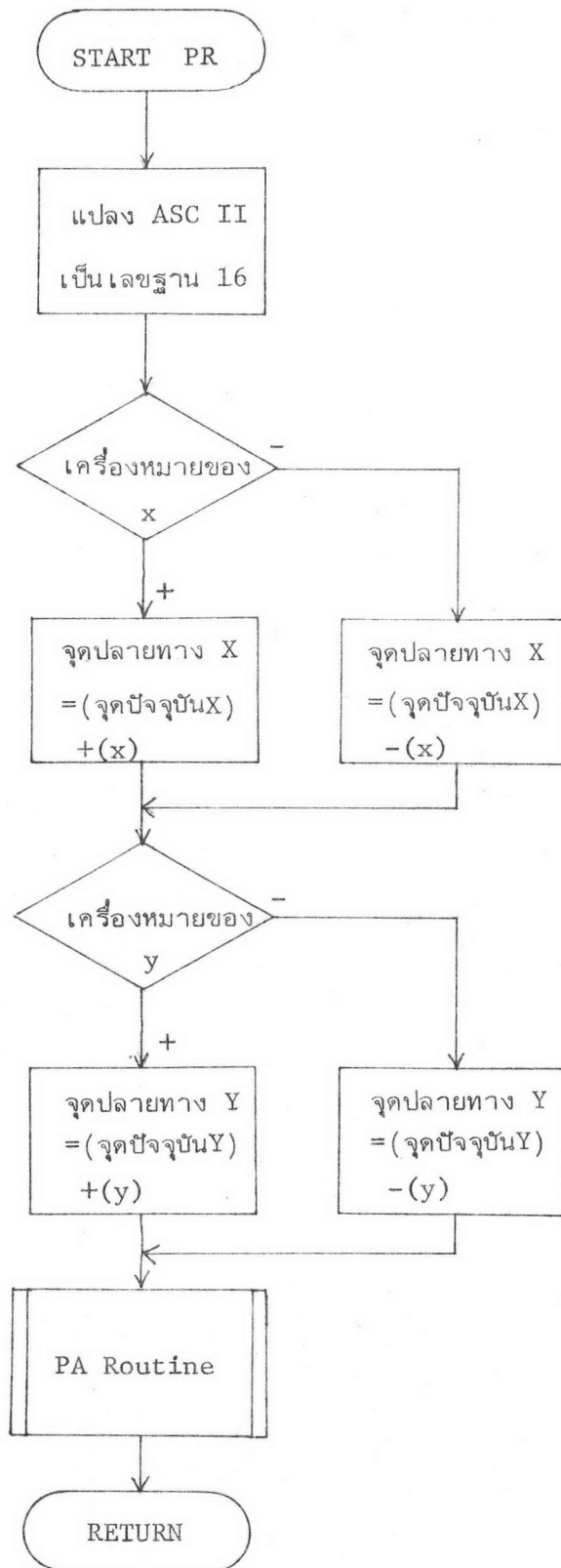
S2 หมายถึง ตัวอักขระมีขนาดเป็น 5 เท่า

S3 หมายถึง ตัวอักขระมีขนาดเป็น 10 เท่า

S4 หมายถึง ตัวอักขระมีขนาดเป็น 20 เท่า

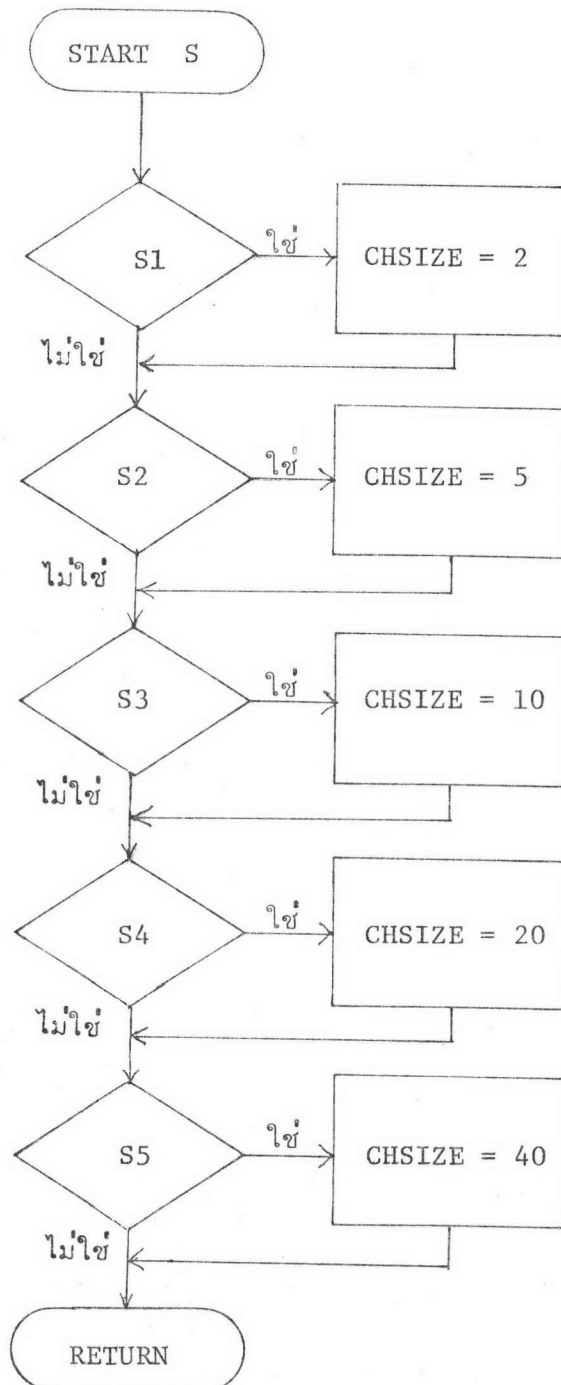


รูปที่ 3.3 ผังการทำงานของการแปลงคำสั่ง PA



รูปที่ 3.4 ผังการทำงานของ การแปลคำสั่ง PR([-] x, [-] y)

009827



รูปที่ 3.5 ผังการทำงานของกรการแปลคำสั่ง Sn

S5 หมายถึง ตัวอักษรที่ขนาดเป็น 40 เท่า

ตัวอักษรขนาด 1 เท่า จะมีขนาด .07 นิ้ว

4) คำสั่ง CH (Character Horizontal)

เมื่อโปรแกรมแปลคำสั่งได้รับคำสั่งนี้ จะเริ่มค้นหาตัวอักษรในตาราง

เมื่อพบตัวอักษรที่ตรงกับในคำสั่งแล้ว จะนำรหัสสำหรับการเขียนตัวอักษรนั้นมาถอดรหัสเพื่อ
สั่ง ปากกา เขียนตัวอักษรนั้น รหัส 1 ไบต์ สำหรับการลากเส้น 1 เส้น ความหมายของ
รหัสนี้มีดังนี้

บิตที่ 1-4 เป็นเลขฐานสองที่แสดงจำนวนลวดเตปที่จะลากปากกาไป

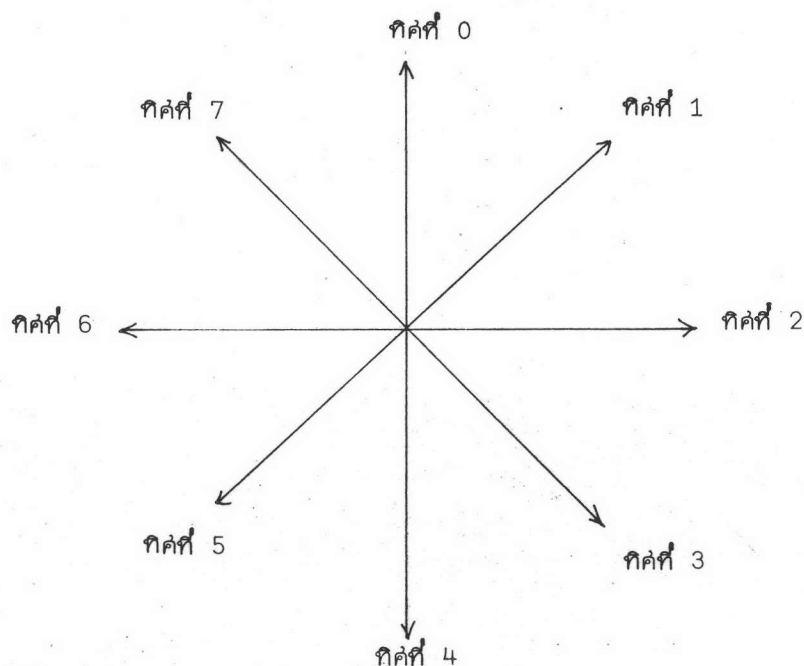
โดยมีค่าได้ตั้งแต่ 0000 ฐาน 2 ถึง 1111 ฐาน 2

หรือ 0 ถึง 15 ลวดเตป

บิตที่ 5-7 เป็นเลขฐานสองที่แสดงทิศทางของการลากปากกาไป

โดยมีค่าได้ตั้งแต่ 000 ฐาน 2 ถึง 111 ฐาน 2 หรือ

0 ถึง 7 ดังรูปที่ 3.6



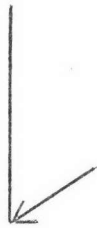
รูปที่ 3.6 แสดงทิศทางการเคลื่อนที่ของปากกาในการเขียนตัวอักษร

การเขียน

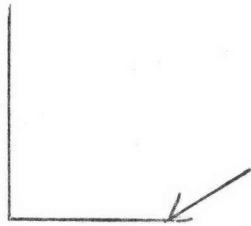
รหัส



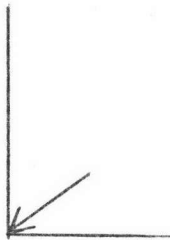
07



C7



A6

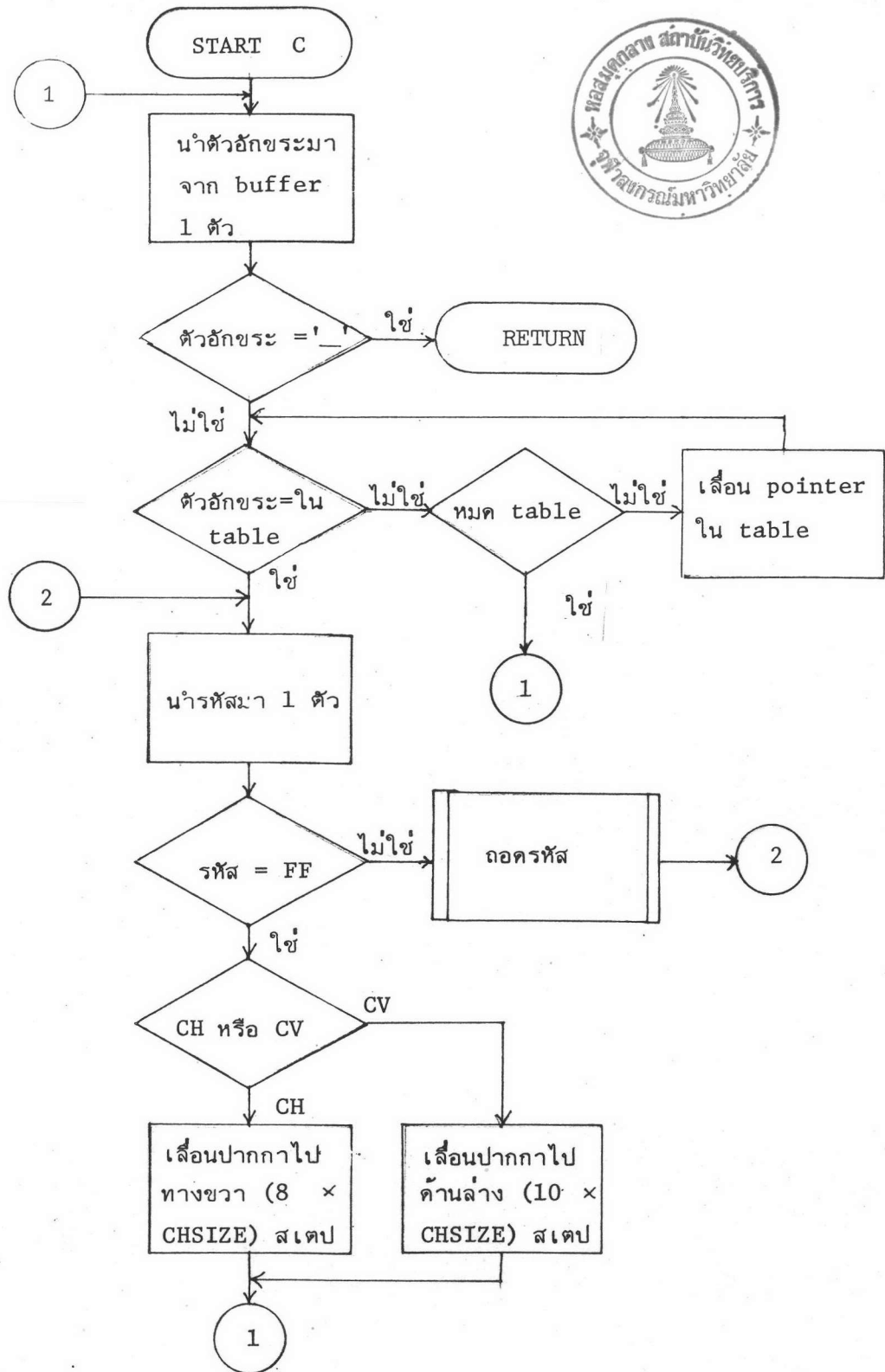


66

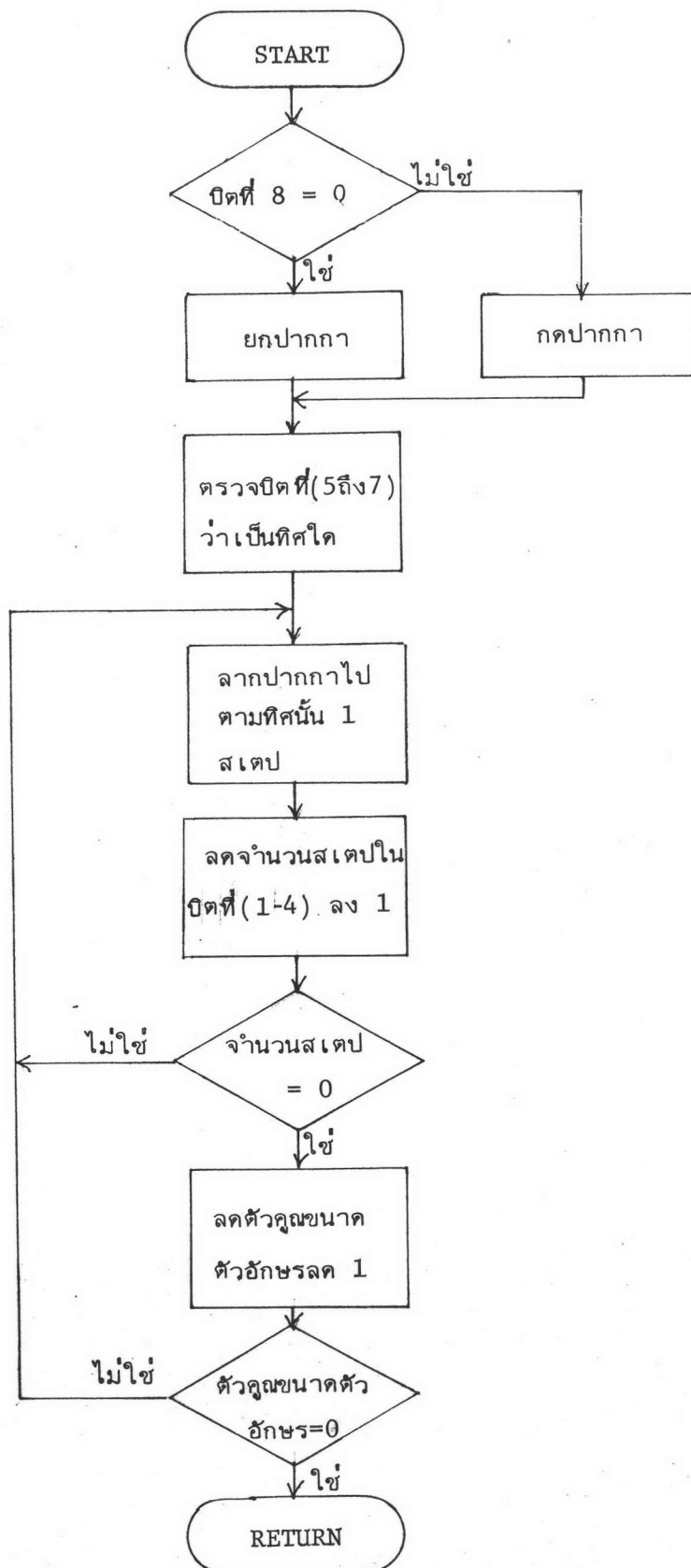


FF

รูปที่ 3.7 แสดงการเขียนอักษรตัว L



รูปที่ 3.8 ผังการทำงานของการแปลคำสี่ง CH และ CV



รูปที่ 3.9 ผังการทำงานของกรอटरหลักการเขียนตัวอักษร

บิตที่ 8 แสดงการกดหรือยกปากกา 0 แสดงว่ายกปากกาก่อนลาก
เส้น

1 แสดงว่ากดปากกาก่อนลากเส้น

เช่น รหัส A6 ฐาน 16 หรือ 1010 0110 ฐาน 2 มีความหมายว่า
ให้กดปากกาลากเส้นไปทางทิศที่ 2 (ทางขวา) เป็นระยะทาง 6 ลิตเป

รหัส 1 ไบต์ ที่มีค่า FF ฐาน 16 หรือ 1111 1111 ฐาน 2 หมายถึง
ถึงเส้นริ้วเส้นการเขียนตัวอักษรตัวนั้น อักษรตัว H ในคำสั่ง CH จะบอกให้เลื่อนปากกาไปทาง
ขวา 8 ลิตเป เพื่อเตรียมเขียนอักษรตัวต่อไป

ตัวอย่างเช่น รหัสสำหรับการเขียนอักษรตัว L คือ 07 C7 A6 66

FF

07 หมายความว่า ยกปากกาเลื่อนไปทิศที่ 0 (ด้านบน) เป็นระยะทาง 7 ลิตเป

C7 หมายความว่า กดปากกาเลื่อนไปทิศที่ 4 (ด้านล่าง) เป็นระยะทาง 7 ลิตเป

A6 หมายความว่า กดปากกาเลื่อนไปทิศที่ 2 (ทางขวา) เป็นระยะทาง 6 ลิตเป

66 หมายความว่า ยกปากกาเลื่อนไปทิศที่ 6 (ทางซ้าย) เป็นระยะทาง 6 ลิตเป

FF หมายความว่า เส้นริ้วเส้นการเขียนอักษรตัว L

เครื่องหมายขีดเส้นใต้ (Underscore) เป็นเครื่องหมายบอกการเริ่มต้นและสิ้นสุด
ชุดของตัวอักษร (Character String) ที่ต้องการเขียน

5) คำสั่ง CV (Character Vertical)

การแปลคำสั่ง CV มีขั้นตอนและวิธีการเช่นเดียวกับคำสั่ง CH แต่ต่างกัน
ที่ว่า เมื่อเส้นริ้วเส้นการเขียนตัวอักษรตัวหนึ่งแล้ว ตัว V ในคำสั่ง CV จะบอกให้ปากกาเลื่อน
ไปด้านล่าง 10 ลิตเป เพื่อเตรียมเขียนอักษรตัวต่อไป

3.2.2 การส่งข่าวสารไปยังไมโครคอมพิวเตอร์

เมื่อโปรแกรมควบคุมทำการแปลคำสั่งในซอฟต์แวร์แอสเซมบลีจะมีโปรแกรมส่วน
หนึ่งทำหน้าที่ส่งข่าวสารไปยังไมโครคอมพิวเตอร์ ข่าวสารนี้เป็นรหัสความยาว 1 ไบต์ คือ
02₁₆ ซึ่งเป็นรหัสที่มีความหมายว่า พลอตเตอร์แปลคำสั่งชุดเต็มเสร็จแล้วอยู่ในสถานะที่รอรับ
คำสั่งชุดใหม่ เต็มเมื่อไม่มีโปรแกรมส่วนนี้ ระหว่างที่พลอตเตอร์กำลังทำงานลากเส้นต่าง ๆ อยู่

ถ้ามีคำสั่งส่งมาจากไมโครคอมพิวเตอร์อีกคำสั่งเหล่านั้นจะสูญหายไปบางส่วน หรืออาจสูญหายไปทั้งหมดทำให้พลอตเตอร์ทำงานผิดพลาด ดังนั้นโปรแกรมล้นนี้จะทำให้ขั้นตอนการรับคำสั่งจากไมโครคอมพิวเตอร์ และการแปลคำสั่ง เป็นไปโดยถูกต้องสมบูรณ์เป็นไปตามสังหระเวลาที่เหมาะสม

3.2.3 การขยายบัฟเฟอร์แอเรีย

เติมบัฟเฟอร์แอเรีย มีขนาด 80 ไบต์ ซึ่งอาจไม่เพียงพอต่อการใช้งาน บัฟเฟอร์แอเรียซึ่งถูกย้ายจากแอดเดรสที่ 0800₁₆ ถึง 084F₁₆ ไปอยู่ที่แอดเดรส 082E₁₆ ถึง 0E2D₁₆ มีขนาดเพิ่มขึ้นเป็น 1.5 กิโลไบต์ ทำให้มีเนื้อที่เพียงพอในการเก็บคำสั่งได้มากขึ้น อย่างไรก็ตามขนาดของบัฟเฟอร์นี้เป็นขนาดสูงสุดที่ใช้งานได้ ผู้ใช้สามารถเลือกขนาดของบัฟเฟอร์ได้ตามต้องการ

3.3 การพัฒนาโปรแกรมย่อยภาษาเบสิก

3.3.1 การใช้โปรแกรมย่อยในภาษาเบสิก

ผู้ใช้ภาษาเบสิกสามารถเก็บโปรแกรมย่อยภาษาเบสิกไว้ในแผ่นบันทึกได้ เมื่อมีความจำเป็นต้องใช้โปรแกรมใดจึงใช้คำสั่งนำโปรแกรมนั้นลงสู่หน่วยความจำหลักได้ ใช้คำสั่งดังนี้

```
MERGE <filename>
```

โดยทั่วไปเมื่อโปรแกรมย่อยถูกนำลงสู่หน่วยความจำหลักแล้วจะรวมเป็นโปรแกรมเดียวกันกับโปรแกรมหลักโดยเรียงตามลำดับของเลขกำกับคำสั่ง (Line number) ดังนั้นโปรแกรมย่อยจะถูกกำหนดให้มีเลขกำกับคำสั่งสูง เพราะเมื่อรวมกับโปรแกรมหลักแล้ว โปรแกรมย่อยจะอยู่ส่วนท้ายและไม่ปะปนกับตัวโปรแกรมหลัก

การเรียกโปรแกรมย่อยทำโดยใช้คำสั่ง

```
GOSUB <line number>
```

การกลับจากโปรแกรมย่อยทำโดยใช้คำสั่ง

```
RETURN
```

```

ตัวอย่างการเรียกโปรแกรมย่อย

10  REM MAIN PROGRAM
20  PRINT "MAIN PROGRAM"
30  GOSUB 70 'BRANCH TO A SUBROUTINE
40  PRINT "END OF MAIN PROGRAM"
50  END
60  REM END OF MAIN PROGRAM
70  REM THIS IS A SUBROUTINE
80      PRINT "*****"
90      RETURN
100 REM END OF SUBROUTINE

```

3.3.2 การใช้ภาษาเบสิกในการอินเทอร์เฟส (Interface)

ในการทำโปรแกรมเพื่อการอินเทอร์เฟสด้วยภาษาเบสิกนั้น มีคำสั่งที่

สำคัญ 3 คำสั่งคือ

INP(I) ให้ค่า 1 ไบต์ ที่อ่านจาก พอร์ต I

OUT I, J ส่งค่า 1 ไบต์ ออกทาง พอร์ต I โดย J เป็นข้อมูลที่ถูกลำ

WAIT <port number>, I หยุดการทำงานของโปรแกรม ขณะที่
 ตรวจสอบสถานะ (Status) ของอินพุทพอร์ต โปรแกรมจะหยุดการทำงานจนกว่าข้อมูล 1 ไบต์
 ที่อ่านจากพอร์ต จะมีลักษณะของบิตแพทเทิร์น (Bit pattern) ตามที่ระบุไว้ด้วย I

การส่งข้อมูล 1 ไบต์ ออกทางพอร์ต 22₁₆ ของเครื่องไมโครคอมพิวเตอร์ IMSAI

มีขั้นตอนดังนี้

1) กำหนดหน้าที่การทำงานของไอซีเบอร์ 8251

ใช้คำสั่งดังนี้

```
OUT &H23,&H40
```



```
OUT    &H23, &HCF
```

```
OUT    &H23, &H27
```

2) ตรวจสอบสถานะของไอซีเบอร์ 8251 ว่าพร้อมที่จะส่งข้อมูลด้วยการตรวจสอบลูปบิตที่ 1 ของข้อมูล 1 ไบต์ จากพอร์ท 23_{16} ว่าเป็น 1 ใช่คำสั่ง ดังนี้

```
WAIT   &H23, &H01
```

3) ส่งข้อมูลที่ต้องการส่งออกจากพอร์ท 22_{16} ใช่คำสั่ง ดังนี้

```
OUT    &H22, J
```

J เป็นตัวเลข 0 ถึง 255 หรือ 0 ถึง FF_{16} เช่นถ้าต้องการส่งตัวอักษร A ซึ่งมี ASCII เป็น 41_{16} จะใช่คำสั่ง

```
OUT    &H22, &H41
```

ส่วนการรับข้อมูลที่ส่งมาอย่างไรไมโครคอมพิวเตอร์ทำได้โดยตรวจสอบไอซีเบอร์ 8251 ว่ามีข้อมูลส่งมาด้วยการตรวจสอบลูปบิตที่ 2 ของข้อมูล 1 ไบต์ จากพอร์ท 23_{16} ว่าเป็น 1 ใช่คำสั่งดังนี้

```
WAIT   &H23, &H02
```

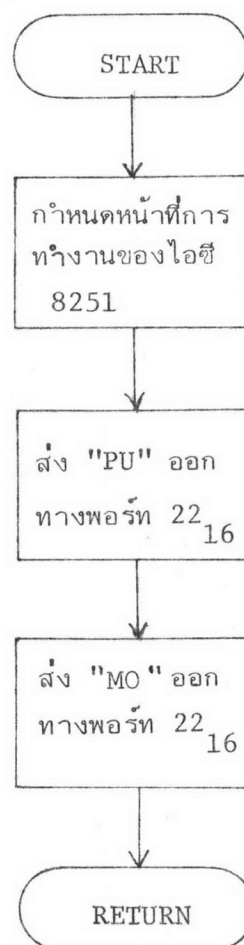
และรับข้อมูลโดยใช่คำสั่ง

```
CHECK.P = INP (&H22)
```

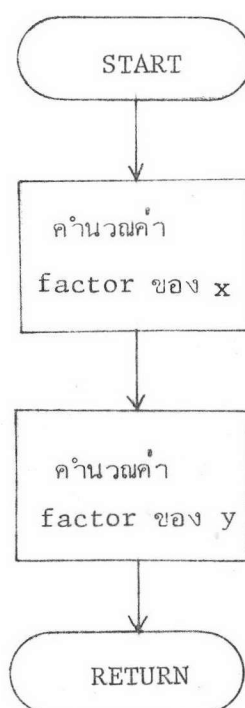
ข้อมูลจะถูกรับและนำไปเก็บไว้ที่ตัวแปรชื่อ CHECK.P

3.3.3 การทำงานของโปรแกรมย่อยภาษาเบสิก

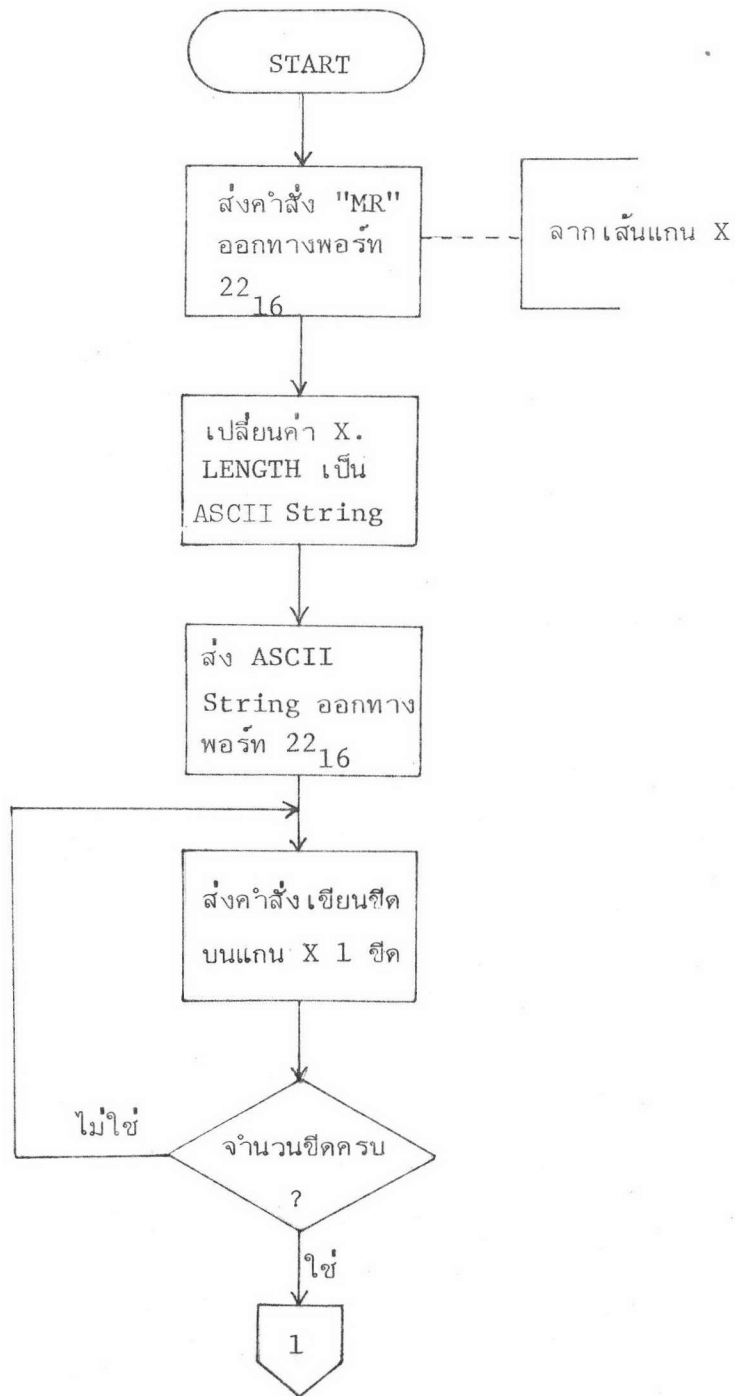
การทำงานของโปรแกรมย่อยภาษาเบสิกแต่ละโปรแกรมมีผังการทำงานดังต่อไปนี้



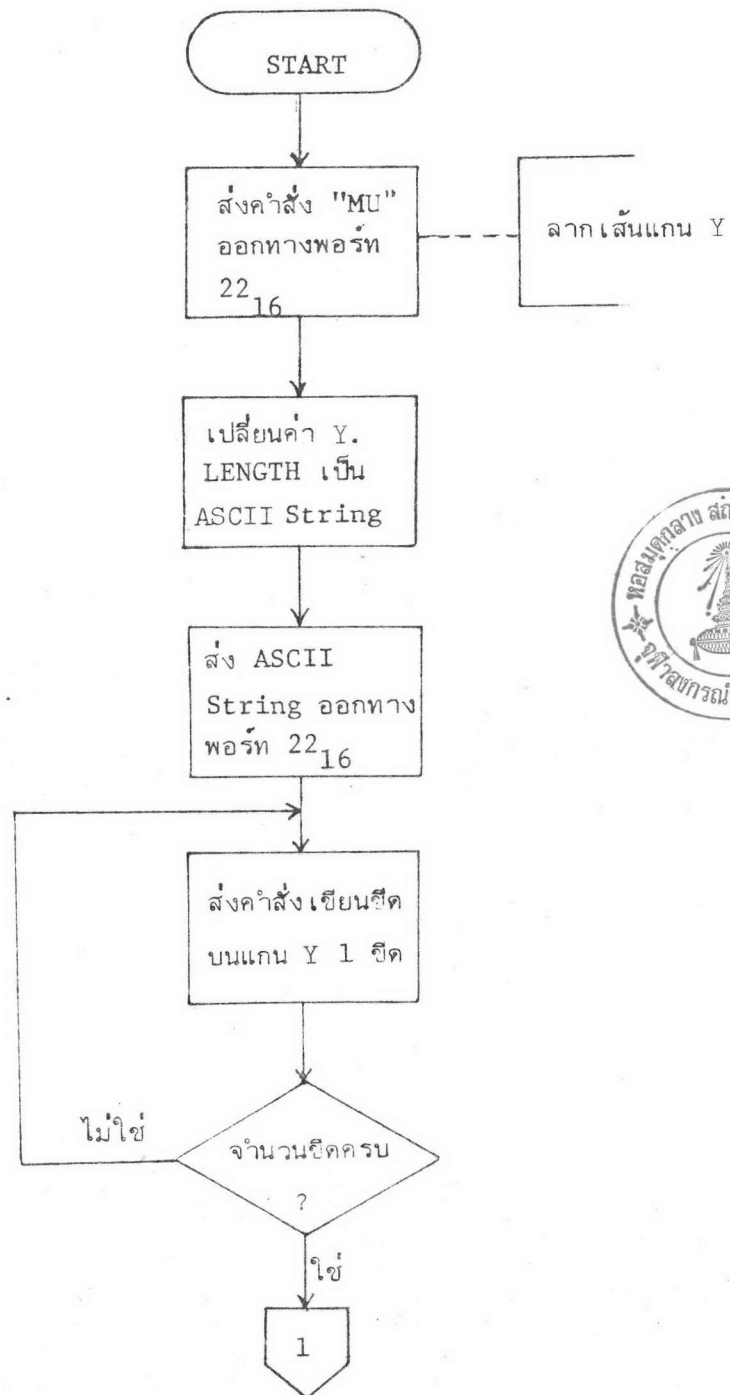
รูปที่ 3.10 ผังการทำงานของโปรแกรมย่อย INIT



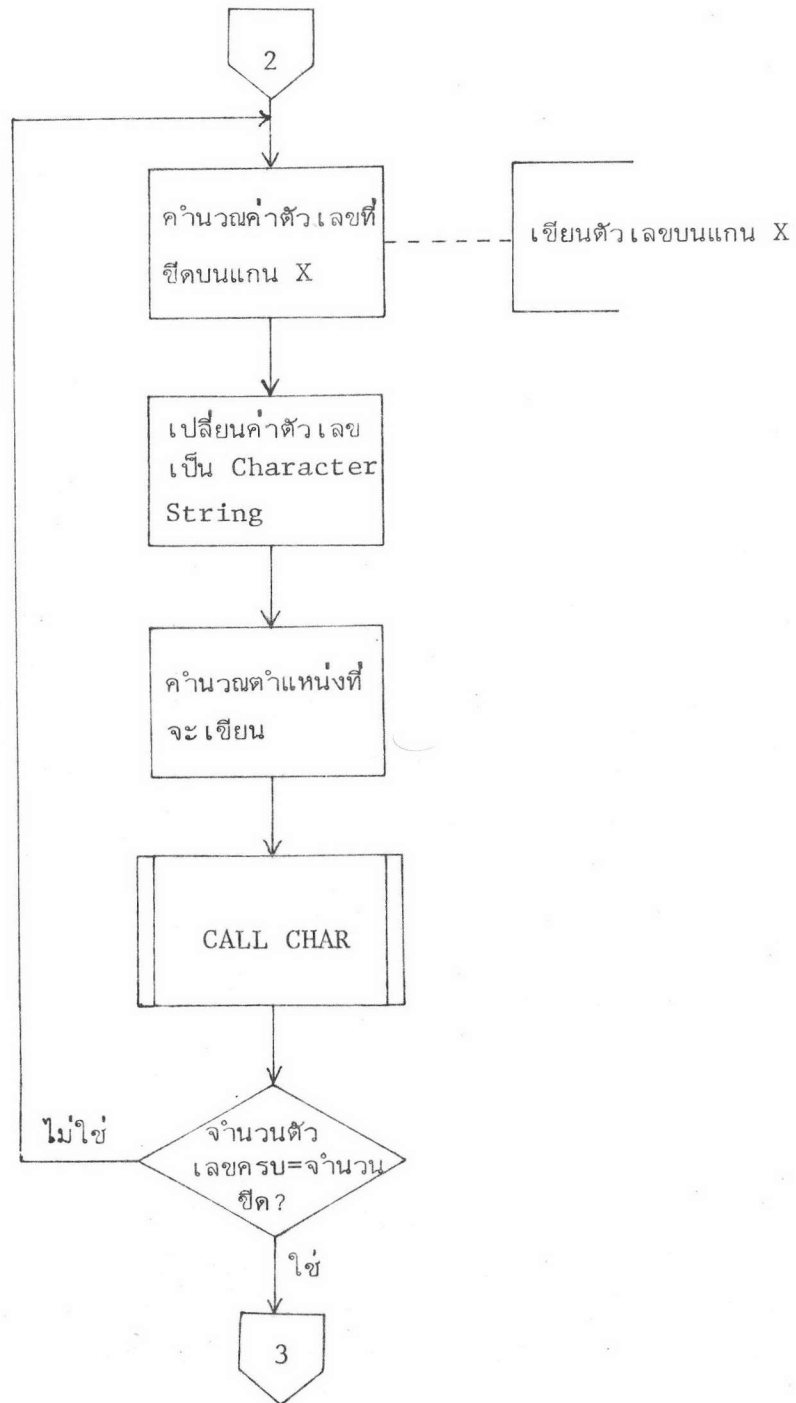
รูปที่ 3.11 แผนผังการทำงานของโปรแกรมย่อย FACTOR



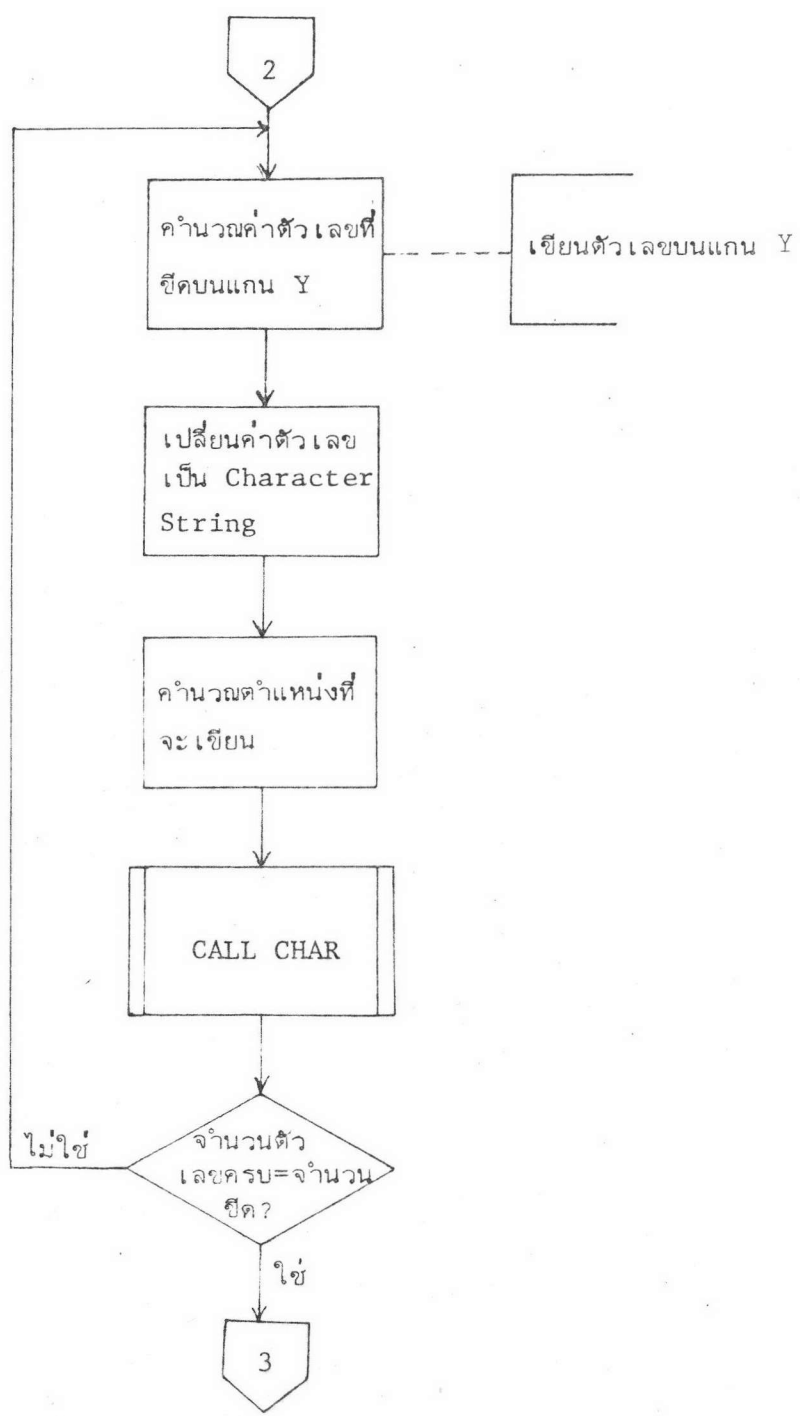
รูปที่ 3.12 ผังการทำงานของโปรแกรมย่อย AXIS



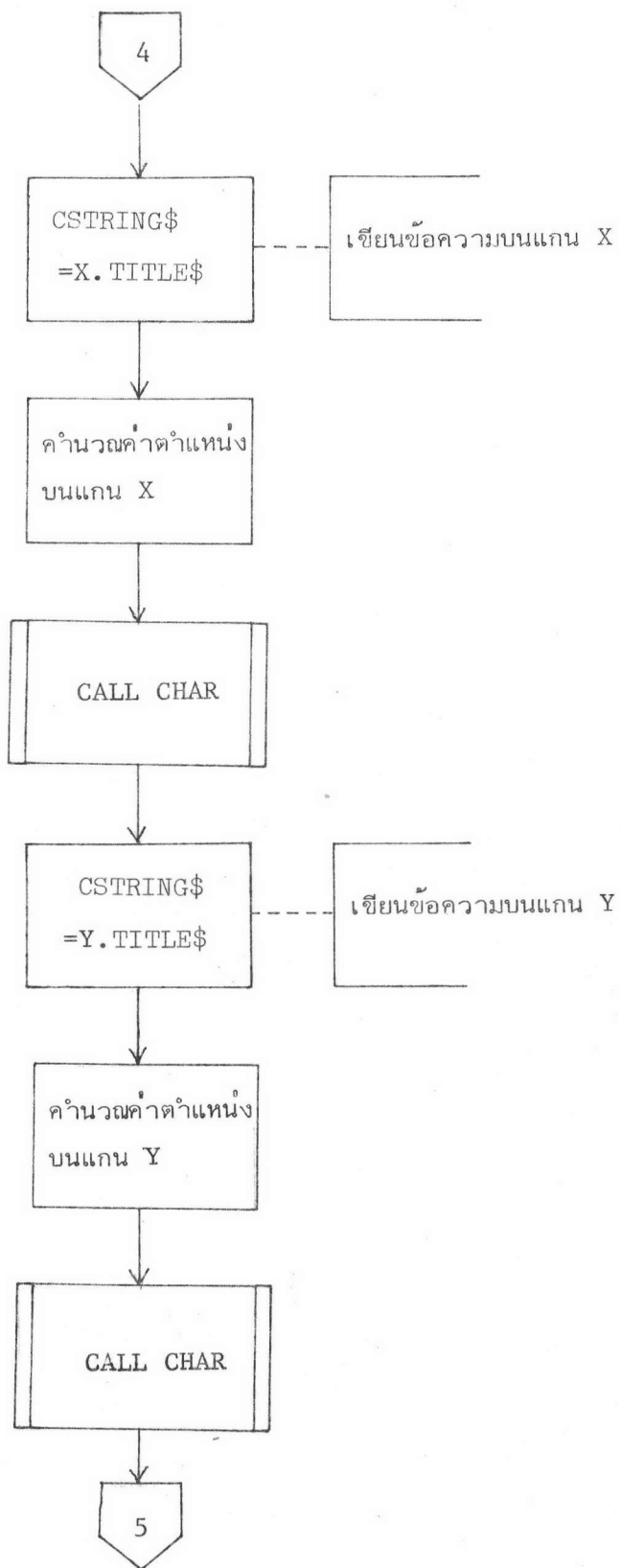
รูปที่ 3.13 ผังการทำงานของโปรแกรมย่อย AXIS (ต่อ)



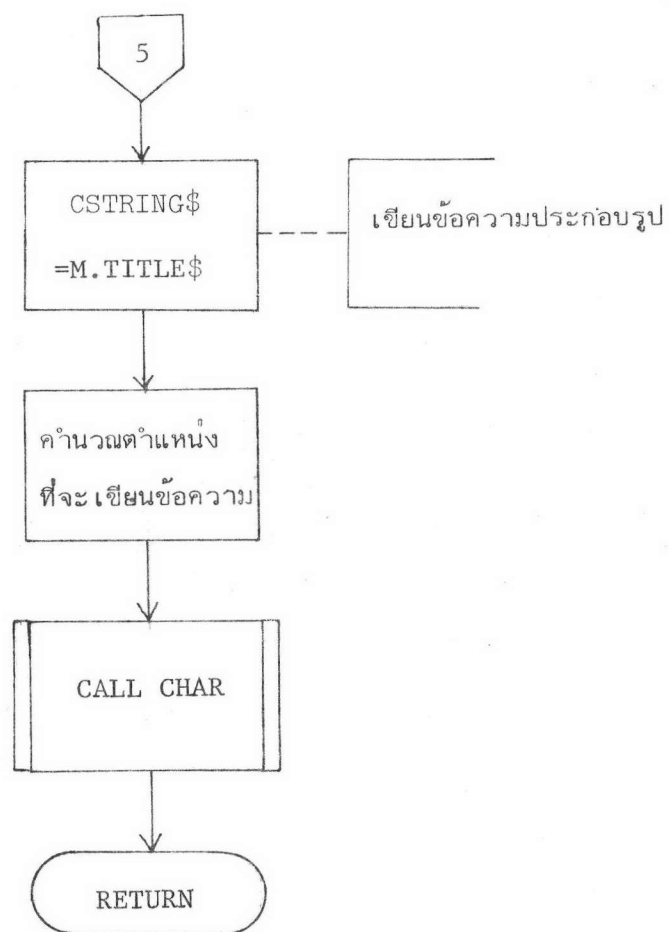
รูปที่ 3.14 ผังการทำงานของโปรแกรมย่อย AXIS (ต่อ)



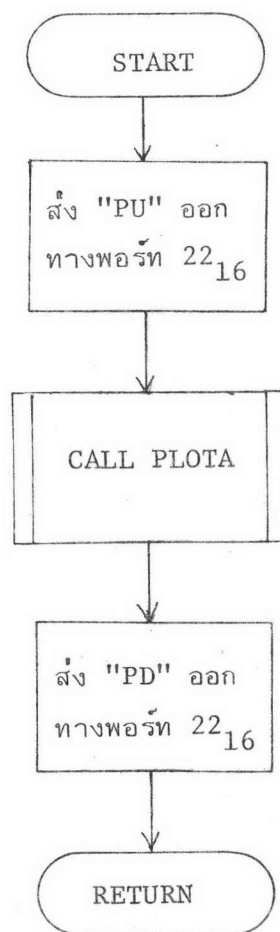
รูปที่ 3.15 ผังการทำงานของโปรแกรมย่อย AXIS (ต่อ)



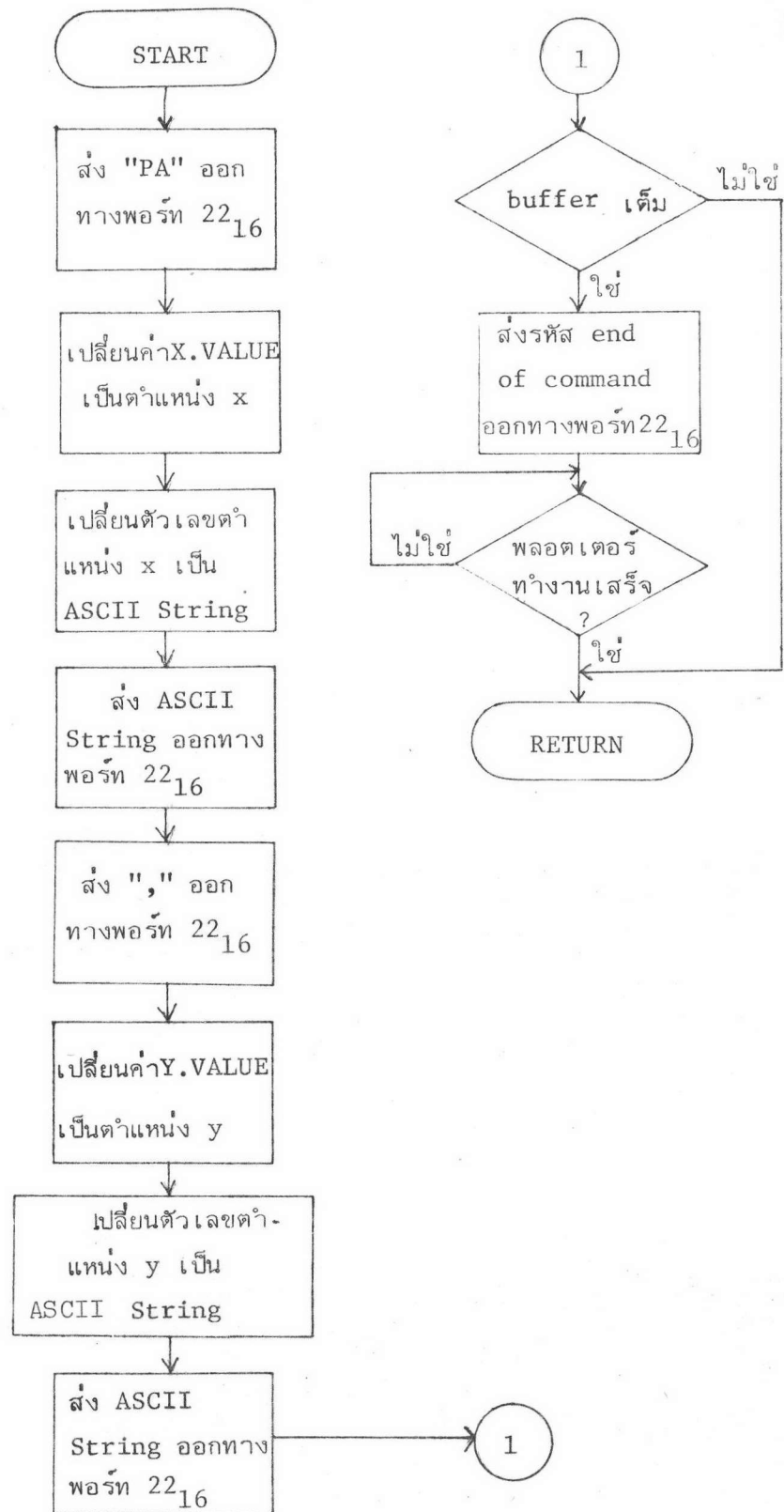
รูปที่ 3.16 ฝั่งการทำงานของโปรแกรมย่อย AXIS (ต่อ)



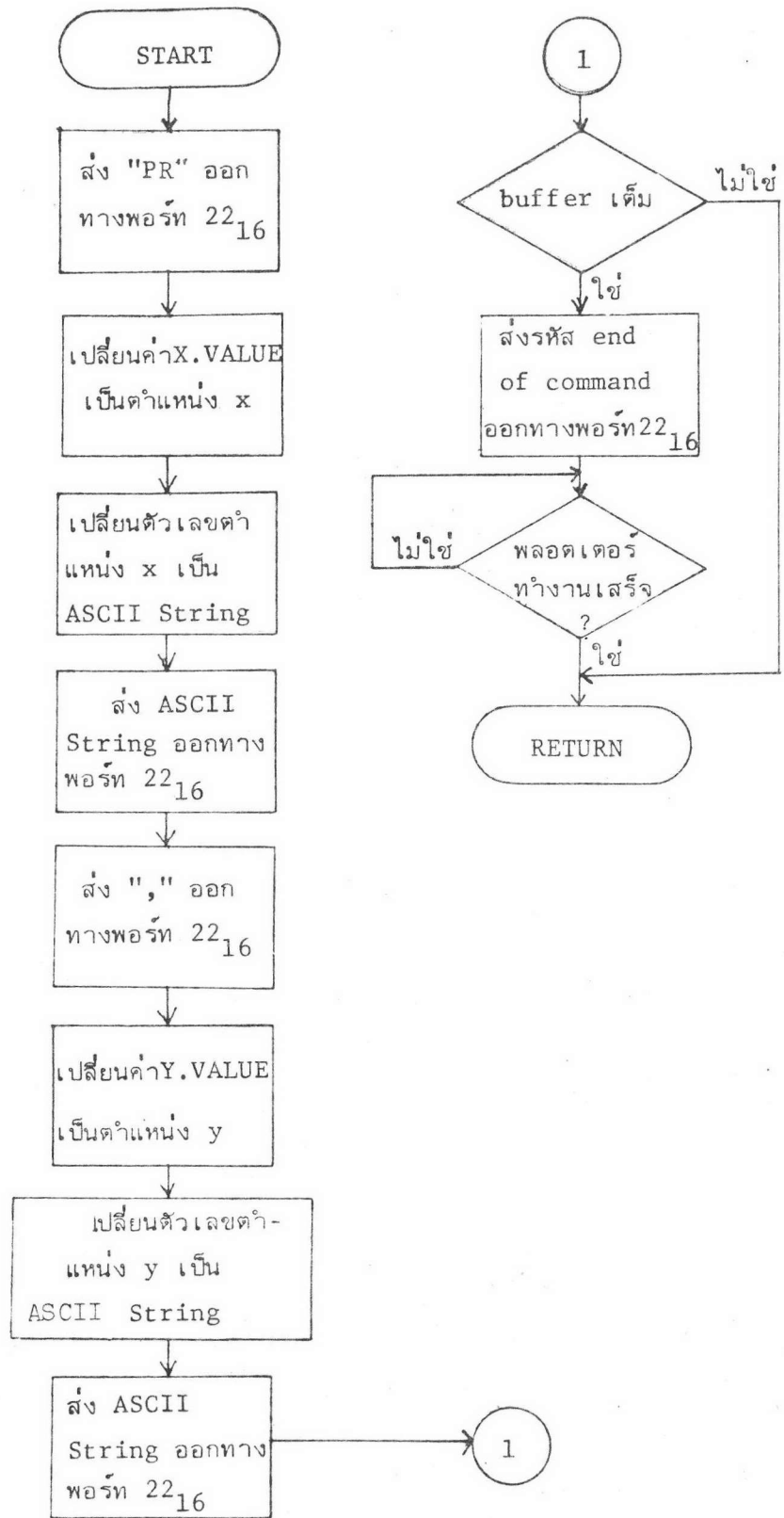
รูปที่ 3.17 ผังการทำงานของโปรแกรมย่อย AXIS (ต่อ)



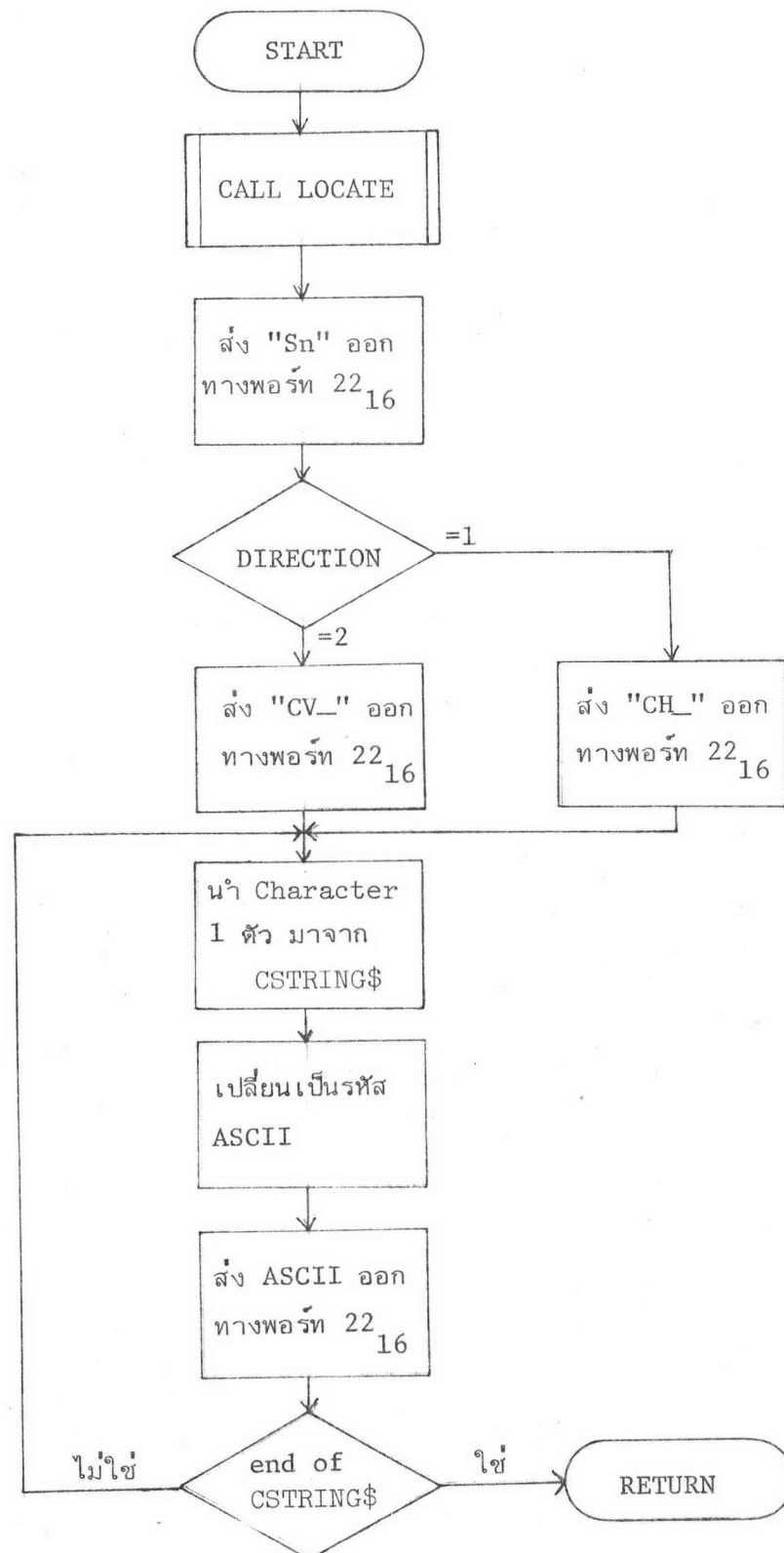
รูปที่ 3.18 ผังการทำงานของโปรแกรมย่อย LOCATE



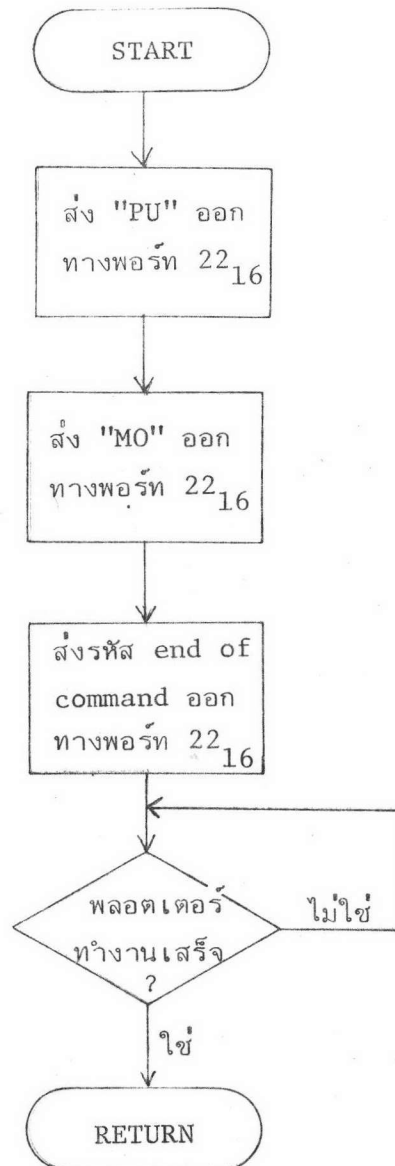
รูปที่ 3.19 ผังการทำงานของโปรแกรมย่อย PLOTA



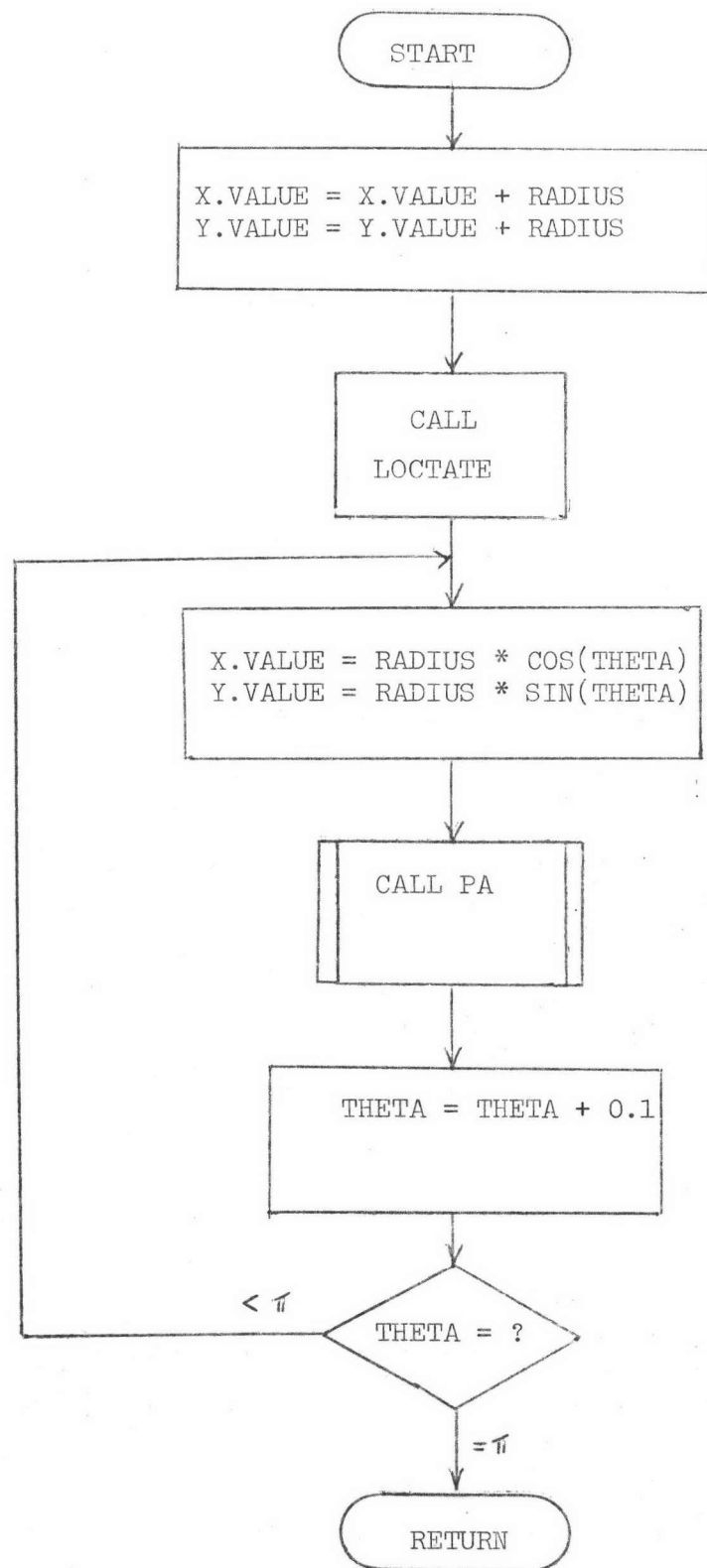
รูปที่ 3.20 ผังการทำงานของโปรแกรมย่อย PLOTR



รูปที่ 3.21 ผังการทำงานของโปรแกรมย่อย CHAR



รูปที่ 3.22 แผนผังการทำงานของโปรแกรมย่อย QUIT



รูปที่ 3.2 3 ผังการทำงานของโปรแกรมย่อย CIRCLE