



บทที่ 9

โปรแกรมส่วนบริการโปรแกรมย่อย

โปรแกรมส่วนบริการโปรแกรมย่อย หรือ SR (SERVICE ROUTINE) มีหน้าที่ควบคุมและให้บริการในการเรียกใช้โปรแกรมย่อย โดยภายในส่วนบริการโปรแกรมย่อย จะแบ่งเป็นส่วนย่อยอีก 3 ส่วน คือ

1. โปรแกรมตั้งจุดเริ่มต้นในการทำงาน
2. โปรแกรมที่ทำหน้าที่จัดการการเรียกใช้โปรแกรมย่อย
3. โปรแกรมย่อยสำหรับให้เรียกใช้

โปรแกรมที่ทำหน้าที่ตั้งจุดเริ่มต้นในการทำงานของส่วนบริการโปรแกรมย่อย จะมีการทำงานต่อจาก โปรแกรมที่ทำหน้าที่กำหนดค่าเริ่มต้นในการทำงานของส่วนควบคุมอินพุต เอ้าท์พุต งานที่ทำในส่วนย่อยนี้คือ ทำการขอหน่วยความจำ RAM ที่จำเป็นต้องใช้ในโปรแกรมย่อยต่าง ๆ จากโปรแกรมส่วนควบคุมอินพุต เอ้าท์พุต พร้อมทั้งกำหนดค่าเริ่มต้นให้กับหน่วยความจำเหล่านั้น หลังจากที่กำหนดจุดเริ่มต้นในการทำงานเสร็จแล้ว ก็จะออกจากโปรแกรมส่วนบริการโปรแกรมย่อย เข้าสู่ส่วนโปรแกรมหลัก

โปรแกรมที่ทำหน้าที่จัดการการเรียกใช้โปรแกรมย่อย มีลักษณะเป็นโปรแกรมเล็ก ๆ โปรแกรมหนึ่ง มีจุดเริ่มต้นของโปรแกรมอยู่ที่แอดเดรส 3 ใช้สำหรับช่วยเหลือในการเรียกใช้โปรแกรมย่อยทั้งที่อยู่ในส่วนบริการโปรแกรมย่อยเอง และรวมทั้งโปรแกรมย่อยที่อยู่ในส่วนควบคุมอินพุต เอ้าท์พุตด้วย วิธีในการเรียกใช้จะกล่าวถึงในหัวข้อต่อไป

โปรแกรมย่อยสำหรับเรียกใช้ เป็นโปรแกรมย่อยส่วนที่ไม่ขึ้นกับฮาร์ดแวร์ เนื่องจากโปรแกรมย่อยที่ขึ้นกับฮาร์ดแวร์ ถูกแยกออกไปกลายเป็นโปรแกรมส่วนควบคุมอินพุต เอ้าท์พุตต่างหากแล้ว รายละเอียดของโปรแกรมย่อยแต่ละตัวจะกล่าวในหัวข้อต่อไป

9.1 การใช้งานโปรแกรมจัดการโปรแกรมย่อย

โปรแกรมจัดการโปรแกรมย่อย (FN CALL) มีหน้าที่ควบคุมการใช้งานโปรแกรมย่อยทั้งหมดในส่วนบริการโปรแกรมย่อยนี้ และช่วยอำนวยความสะดวกในการเรียกใช้โปรแกรมย่อยที่อยู่ในโปรแกรมส่วนควบคุมอินพุท เอ้าท์พุท

การเรียกใช้โปรแกรมย่อยในโปรแกรมส่วนควบคุมอินพุท เอ้าท์พุท โดยตรงนั้น ทำโดยนำหมายเลขของโปรแกรมย่อยที่ต้องการเรียกใช้ มาคำนวณหาแอดเดรสที่ใช้เก็บตำแหน่งเริ่มต้นของโปรแกรมย่อยตามสูตร $100H + (\text{หมายเลขโปรแกรมย่อย} * 2)$ เมื่อได้แอดเดรสมาแล้ว ก็ทำการอ่านค่า 2 ไบท์ที่อยู่ติดกันมาจากแอดเดรสที่คำนวณได้นั้น ซึ่งค่าที่ได้มาจะเป็นตำแหน่งเริ่มต้นของโปรแกรมย่อยที่ต้องการ จากนั้นจึงทำการเรียกไปยังโปรแกรมย่อยนั้น แต่เนื่องจากตำแหน่งเริ่มต้นของโปรแกรมย่อยจะถูกอ่านเข้ามาในรีจิสเตอร์ ดังนั้นจึงใช้คำสั่ง CALL ตรง ๆ ไม่ได้ ทำให้การเรียกใช้ยุ่งยากมาก

หน้าที่อันหนึ่งของโปรแกรมจัดการโปรแกรมย่อยก็คือ เพื่อที่จะลดความยุ่งยากในการเรียกใช้โปรแกรมย่อยในโปรแกรมส่วนควบคุมอินพุท เอ้าท์พุท ที่กล่าวมานี้ โดยโปรแกรมจัดการโปรแกรมย่อยจะทำหน้าที่เรียกโปรแกรมเหล่านั้นให้ การเรียกใช้โปรแกรมย่อยเหล่านั้นโดยผ่านทางโปรแกรมจัดการโปรแกรมย่อย ทำได้โดย นำหมายเลขโปรแกรมย่อยใส่ในรีจิสเตอร์ A แล้วเรียกไปยังจุดเริ่มต้นของโปรแกรมจัดการโปรแกรมย่อยที่แอดเดรส 3 โดยใช้คำสั่งดังนี้

LD A, หมายเลขโปรแกรมย่อย

CALL 3

สำหรับการส่งพารามิเตอร์ไป และกลับ ขึ้นอยู่กับโปรแกรมย่อยแต่ละหมายเลข โดยถ้ามีการส่งพารามิเตอร์ไปยังโปรแกรมย่อย จะส่งไปทางรีจิสเตอร์ DE และถ้าโปรแกรมย่อยจะส่งค่ากลับมา จะส่งมาทางรีจิสเตอร์ HL

ข้อดีของการเรียกใช้โปรแกรมย่อยผ่านทางโปรแกรมจัดการโปรแกรมย่อย ก็คือผู้ที่ทำการพัฒนาโปรแกรมหลัก จำเป็นต้องรู้แต่เพียงว่า มีโปรแกรมย่อยอะไรให้เรียกใช้ได้บ้าง

แต่ละโปรแกรมย่อยทำหน้าที่อะไร รับ และส่งพารามิเตอร์อย่างไร และค่าแอดเดรส 3 ก็เพียงพอแล้ว ไม่จำเป็นต้องรู้รายละเอียดอื่น ๆ ในโปรแกรมส่วนควบคุมอินพุท เอาท์พุท และโปรแกรมส่วนบริการโปรแกรมย่อยเลย

หน้าที่ที่สองของโปรแกรมจัดการโปรแกรมย่อย คือทำหน้าที่ในการเรียกโปรแกรมย่อยที่อยู่ในโปรแกรมส่วนบริการโปรแกรมย่อย ซึ่งในกรณีโปรแกรมส่วนบริการโปรแกรมย่อยนี้ จะไม่มีการเก็บตำแหน่งเริ่มต้นของโปรแกรมย่อยไว้ในตาราง ดังนั้นการเข้าถึงโปรแกรมย่อยเหล่านี้มีวิธีเดียวคือ เรียกใช้ผ่านโปรแกรมจัดการโปรแกรมย่อย

การเรียกใช้โปรแกรมจัดการโปรแกรมย่อย ในกรณีนี้ใช้วิธีการเดียวกับกรณีก่อน คือ นำหมายเลขโปรแกรมย่อยใส่ในรีจิสเตอร์ A แล้วทำการเรียกไปยังแอดเดรส 3 โดยใช้คำสั่ง CALL 3 โดยหมายเลขของโปรแกรมย่อยในส่วนบริการโปรแกรมย่อยนี้จะเริ่มจากหมายเลข 80H หรือ 128 เป็นต้นไป

การทำงานของโปรแกรมจัดการโปรแกรมย่อย คือเมื่อถูกเรียกใช้จะทำการตรวจสอบค่าของรีจิสเตอร์ A ถ้าน้อยกว่า 80H จะทำการเรียกโปรแกรมย่อยในส่วนควบคุมอินพุท เอาท์พุทอีกทอดหนึ่ง โดยใช้ตารางที่แอดเดรส 100H แต่สำหรับกรณีที่ค่าของรีจิสเตอร์ A เท่ากับหรือมากกว่า 80H ก็จะทำให้การเรียกโปรแกรมย่อย ภายในส่วนบริการโปรแกรมย่อยเองมาทำงาน

โปรแกรมย่อยของโปรแกรมส่วนบริการโปรแกรมย่อยแบ่งออกได้เป็น 2 กลุ่มคือ กลุ่มควบคุมการจองอุปกรณ์ที่เครื่องรับโทรทัศน์แต่ละเครื่องต้องผลิตกันใช้ และอีกกลุ่มหนึ่งทำหน้าที่ตรวจสอบเบอร์ของเครื่องรับโทรทัศน์ และรหัสคำสั่ง

9.2 โปรแกรมย่อยกลุ่มควบคุมการจองอุปกรณ์

โปรแกรมย่อยหมายเลข 128

ชื่อ
ขงจองช่องสัญญาณเสียง
(OPEN ICM)

ชื่อย่อที่ใช้ในโปรแกรม _OPEN_ICM

ค่าที่ต้องส่งตอนเรียก -
 ค่าที่ให้กลับมา รีจิสเตอร์ H : มีหรือไม่
 รีจิสเตอร์ L : หมายเลขช่องสัญญาณเสียง

ใช้สำหรับจองช่องสัญญาณเสียงที่ยังไม่มีการใช้งาน เมื่อเรียกใช้โปรแกรมย่อยนี้ จะให้ค่ากลับมาทางรีจิสเตอร์ H เป็น 0 หรือ 1 ถ้าให้ค่า 0 กลับมาหมายความว่าไม่มีช่องสัญญาณเสียงว่างให้ใช้ได้ ถ้าค่าที่ให้กลับมาเป็น 1 แสดงว่า สามารถจองได้ และหมายเลขของช่องสัญญาณเสียงที่จองได้นั้น จะส่งกลับมาทางรีจิสเตอร์ L

โปรแกรมย่อยหมายเลข 129
 ชื่อ คินช่องสัญญาณเสียง
 (CLOSE ICM)
 ชื่อย่อที่ใช้ในโปรแกรม _CLOSE_ICM
 ค่าที่ต้องส่งตอนเรียก รีจิสเตอร์ E : หมายเลขช่องสัญญาณเสียง
 ค่าที่ให้กลับมา -

ใช้สำหรับส่งช่องสัญญาณเสียงที่ใช้งานเสร็จแล้วคืน ในการเรียกใช้ต้องส่งค่าหมายเลขของช่องสัญญาณเสียงที่จะคืนไปทางรีจิสเตอร์ E เมื่อขอช่องสัญญาณเสียงมา เมื่อใช้เสร็จแล้วต้องส่งคืนเสมอ มิฉะนั้นช่องสัญญาณเสียงจะค่อย ๆ ลดลงจนหมดไปในที่สุด

โปรแกรมย่อยหมายเลข 134
 ชื่อ ขอบช่องตัวรับสัญญาณกดปุ่ม
 (OPEN DTMF)
 ชื่อย่อที่ใช้ในโปรแกรม _OPEN_REG
 ค่าที่ต้องส่งตอนเรียก -
 ค่าที่ให้กลับมา รีจิสเตอร์ H : มีหรือไม่
 รีจิสเตอร์ L : หมายเลขตัวรับสัญญาณกดปุ่ม

ใช้สำหรับจับตัวรับสัญญาณกดปุ่มที่ยังไม่มีการใช้งาน เมื่อเรียกใช้โปรแกรมย่อยนี้ จะให้ค่ากลับมาทางรีจิสเตอร์ H เป็น 0 หรือ 1 ถ้าให้ค่า 0 กลับมาหมายความว่าไม่มีตัวรับ

สัญญาณกดปุ่มว่างให้ใช้ได้ ถ้าค่าที่เด้งกลับมาเป็น 1 แสดงว่า สามารถจองได้ และหมายเลขของตัวรับสัญญาณกดปุ่มที่จองได้นั้น จะส่งกลับมาทางรีจิสเตอร์ L

โปรแกรมย่อยหมายเลข 135

ชื่อ คีนช่องตัวรับสัญญาณกดปุ่ม
(CLOSE DTMF)

ชื่อย่อที่ใช้ในโปรแกรม _CLOSE_REG

ค่าที่ต้องส่งตอนเรียก รีจิสเตอร์ E : หมายเลขตัวรับสัญญาณกดปุ่ม

ค่าที่เด้งกลับมา -

ใช้สำหรับส่งตัวรับสัญญาณกดปุ่มที่ใช้งานเสร็จแล้วคืน ในการเรียกใช้ต้องส่งค่าหมายเลขของตัวรับสัญญาณกดปุ่มที่จะคืนไปทางรีจิสเตอร์ E เมื่อขอตัวรับสัญญาณกดปุ่มมา เมื่อใช้เสร็จแล้วต้องส่งคืนเสมอ มิฉะนั้นตัวรับสัญญาณกดปุ่มจะค่อย ๆ ลดลงจนหมดไปในที่สุด

9.3 โปรแกรมย่อยกลุ่มตรวจสอบเบอร์ และรหัสคำสั่ง

โปรแกรมย่อยกลุ่มนี้ จะทำหน้าที่ในการตรวจสอบตัวเลขซึ่งส่งมาให้ในโครงสร้างของ STRING ว่าตรงกับเบอร์ของเครื่องรับโทรศัพท์ หรือรหัสคำสั่งใดหรือไม่ ในการตรวจสอบจะได้ผลออกมา 4 กรณี จะยกตัวอย่างเบอร์ของเครื่องรับโทรศัพท์ เพื่อประกอบการบรรยาย ดังนี้

โทรศัพท์เครื่องที่	เบอร์
0	20
1	21
2	3
3	45
4	1234
5	191

กรณีที่ 1 ตัวเลขไม่ตรงกับเบอร์โทรศัพท์ และ ไม่มีโอกาสจะตรง แม้ว่าจะมีตัวเลข

ตามมาอีก เช่น เลข 9 23 1235 เป็นต้น

กรณีที่ 2 ตัวเลขตรงกับเบอร์โทรศัพท์พอดี เช่น 20 1234 เป็นต้น

กรณีที่ 3 ตัวเลขตรงกับเบอร์โทรศัพท์ แต่เกิน เช่น 200 191919 31 เป็นต้น

กรณีที่ 4 ตัวเลขยังไม่พอที่จะตัดสินใจ เช่น 2 123 1 เป็นต้น เนื่องจากจำนวนหลักยังไม่พอ จะต้องดูหลักต่อ ๆ ไปอีก เช่น กรณีของเลข 2 ต้องดูหลักต่อไปด้วย ถ้าหลักต่อไปเป็น 0 หรือ 1 ก็จะตรงกับเบอร์โทรศัพท์ ถ้าเป็นเลขอื่นก็จะไม่ตรง

โปรแกรมย่อยหมายเลข 130

ชื่อ

ตรวจเบอร์โทรศัพท์

(MATCH STRING TO LC NAME)

ชื่อย่อที่ใช้ในโปรแกรม _MATCH_NAME

ค่าที่ต้องส่งคอนเรียก รีจิสเตอร์ DE : แอคเตอเรส

ค่าที่ให้กลับมา รีจิสเตอร์ H : ผลการเปรียบเทียบ

รีจิสเตอร์ L : หมายเลขโทรศัพท์

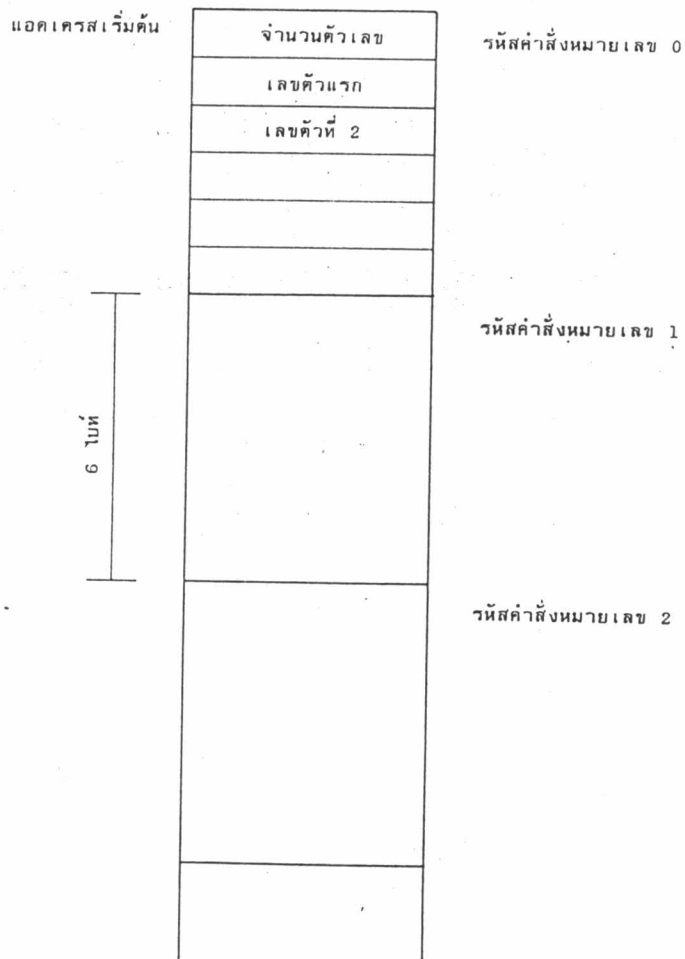
ใช้สำหรับเปรียบเทียบตัวเลขกับเบอร์ของเครื่องรับโทรศัพท์ทุกเครื่อง ในการเรียกใช้งาน จะนำตัวเลขที่จะเปรียบเทียบเก็บไว้ในหน่วยความจำในโครงสร้างแบบ STRING แล้วส่งแอดเดอเรสเริ่มต้นของ STRING นี้ไปให้โปรแกรมย่อย โดยผ่านทางรีจิสเตอร์ DE โปรแกรมย่อยจะให้ผลการเปรียบเทียบกลับมาทางรีจิสเตอร์ H เป็นค่า 0 ถึง 3 ซึ่งมีความหมายดังนี้

รีจิสเตอร์ H	ผลการเปรียบเทียบ
0	ไม่ตรง
1	ยังตัดสินใจไม่ได้
2	ตรง
3	ตรง แต่เลขที่ส่งไปยาวเกิน

ในกรณีที่ค่าที่ให้กลับมาในรีจิสเตอร์ H เป็น 2 ตรงกับเบอร์ของเครื่องรับโทรศัพท์เครื่องใดเครื่องหนึ่ง หรือ 3 ตรงเหมือนกัน แต่เลขยาวเกินนั้น โปรแกรมย่อยจะให้หมายเลขของเครื่องรับโทรศัพท์ที่มีเบอร์ตรงกันนั้นกลับมาทางรีจิสเตอร์ L ด้วย

สำหรับการตรวจสอบรหัสคำสั่ง ก็มีลักษณะการทำงานทำนองเดียวกับการตรวจเบอร์โทรศัพท์ แต่เนื่องจาก รหัสคำสั่งนั้นขึ้นอยู่กับส่วนโปรแกรมหลักแตกต่างจากเบอร์ของเครื่องรับโทรศัพท์ที่กำหนดโดยส่วนควบคุมอินพุท เอ้าทพุท ซึ่งสามารถถามผ่านโปรแกรมย่อยหมายเลข 4 และ 7 ได้ว่ามีเครื่องรับโทรศัพท์กี่เครื่อง และแต่ละเครื่องมีเบอร์เป็นอย่างไร

ดังนั้น ถ้าโปรแกรมหลักต้องการให้ส่วนบริการโปรแกรมย่อยทำหน้าที่ตรวจเทียบตัวเลขกับรหัสให้ ในตอนเริ่มต้นจะต้องบอกมายังส่วนบริการโปรแกรมย่อย ว่ามีรหัสคำสั่งอยู่ที่คำสั่ง และแต่ละคำสั่งคือตัวเลขอะไร โดยกำหนดให้ส่งมาในรูปแบบของตาราง ดังรูปที่ 9.1



รูปที่ 9.1 แสดง รูปแบบของตารางคำสั่ง

รูปแบบของตารางคำสั่งคือ จะเว้นที่ไว้ให้ 6 ไบท์สำหรับแต่ละคำสั่ง การเก็บคำสั่ง จะเก็บในโครงสร้าง STRING ซึ่งสามารถเก็บตัวเลขได้ความยาวสูงสุด 5 หลัก (เก็บไม่ครบก็ได้) ลำดับการเก็บนั้น 6 ไบท์แรกสำหรับสำหรับคำสั่งหมายเลข 0 ตามด้วยคำสั่งหมายเลข 1 2 3 ไปเรื่อย ๆ การส่งตารางคำสั่งให้กับโปรแกรมส่วนบริการโปรแกรมย่อย ทำโดยส่งค่า จำนวนคำสั่ง และแอดเดรสเริ่มต้นของตารางผ่านทางโปรแกรมย่อยต่อไปนี้

โปรแกรมย่อยหมายเลข 131

ชื่อ	บอกแอดเดรสเริ่มต้นของตารางรหัสคำสั่ง (SET COMMAND TABLE START ADDRESS)
ชื่อย่อที่ใช้ในโปรแกรม	_COMM_START
ค่าที่ต้องส่งตอนเรียก	รีจิสเตอร์ DE : แอดเดรส
ค่าที่ให้กลับมา	-

ใช้สำหรับบอกโปรแกรมส่วนบริการโปรแกรมย่อยว่า จุดเริ่มต้นของตารางรหัสคำสั่ง อยู่ที่แอดเดรสไหน การเรียกโปรแกรมย่อยนี้ จะต้องส่งแอดเดรสเริ่มต้นของตารางรหัสคำสั่ง ไปทางรีจิสเตอร์ DE

โปรแกรมย่อยหมายเลข 132

ชื่อ	บอกจำนวนคำสั่ง (NUMBER OF COMMAND)
ชื่อย่อที่ใช้ในโปรแกรม	_COMM_NUMBER
ค่าที่ต้องส่งตอนเรียก	รีจิสเตอร์ E : จำนวน
ค่าที่ให้กลับมา	-

ใช้สำหรับบอกโปรแกรมส่วนบริการโปรแกรมย่อยว่า รหัสคำสั่งนั้นมีทั้งหมดกี่คำสั่ง โดยการเรียกโปรแกรมย่อยนี้ พร้อมกับค่าจำนวนของคำสั่ง ไปทางรีจิสเตอร์ E

การบอกแอดเดรสเริ่มต้นของตารางรหัสคำสั่ง และจำนวนรหัสคำสั่ง ทำเพียงครั้งเดียวก็พอ ไม่จำเป็นต้องบอกใหม่ทุกครั้งที่ทำการตรวจสอบ สำหรับตารางรหัสคำสั่งนั้น จะอยู่ใน ROM หรือ RAM ก็ได้ แต่หลังจากบอกไปแล้ว ห้ามทำการเปลี่ยนแปลงค่าในตาราง

เพราะโปรแกรมส่วนบริการโปรแกรมย่อยจะจำแอดเดรสเริ่มต้น และขนาดของตารางไว้เท่านั้น ไม่ได้จำเอาค่าในตารางไปด้วย

โปรแกรมย่อยหมายเลข 133

ชื่อ	เทียบรหัสคำสั่ง
	(MATCH STRING COMMAND)
ชื่อย่อที่ใช้ในโปรแกรม	_MATCH_COMM
ค่าที่ต้องส่งตอนเรียก	รีจิสเตอร์ DE : แอดเดรส
ค่าที่ให้กลับมา	รีจิสเตอร์ H : ผลการเปรียบเทียบ
	รีจิสเตอร์ L : หมายเลขคำสั่ง

ใช้สำหรับเปรียบเทียบตัวเลข กับตารางคำสั่ง ในการเรียกใช้งาน จะนำตัวเลขที่จะเปรียบเทียบเก็บไว้ในหน่วยความจำในโครงสร้างแบบ STRING แล้วส่งแอดเดรสเริ่มต้นของ STRING นี้ไปให้โปรแกรมย่อย โดยผ่านทางรีจิสเตอร์ DE โปรแกรมย่อยจะให้ผลการเปรียบเทียบกลับมาทางรีจิสเตอร์ H เป็นค่า 0 ถึง 3 ซึ่งมีความหมายดังนี้

รีจิสเตอร์ H	ผลการเปรียบเทียบ
0	ไม่ตรง
1	ยังตัดสินใจไม่ได้
2	ตรง
3	ตรง แต่เลขที่ส่งไปยาวเกิน

ในกรณีที่ค่าที่ให้กลับมาในรีจิสเตอร์ H เป็น 2 ตรงกับรหัสของคำสั่งคำสั่งใดคำสั่งหนึ่ง หรือ 3 ตรงเหมือนกัน แต่เลขยาวเกินนั้น โปรแกรมย่อยจะให้หมายเลขของคำสั่งที่มีรหัสตรงกับตัวเลขที่ส่งไปนั้นกลับมาทางรีจิสเตอร์ L ด้วย