ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ ตัวแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51

Assembler for Microcontroller MCS-51

ชื่อนักศึกษา

1. นายกิตติคุณ	อินอ่อน	รหัสประจำตัว	43035326
2. นายอิทธิพล	ไชยถาวร	รหัสประจำตัว	43035363
3. นายนิติพงษ์	ไคขุนทด	รหัสประจำตัว	43035607

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์โกศล ตราชู

อาจารย์ที่ปรึกษาร่วม อาจารย์สุชิน อาจหาญ

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1. อาจารย์ปิยะ จิตธรรมมาภิรมย์	.....
2. อาจารย์โกศล ตราชู	.....
3. อาจารย์สุชิน อาจหาญ	.....
4. อาจารย์ไพบูลย์ พวงวงศ์ตระกูล	.....
5. อาจารย์พงษ์เกียรติ เชษฐพิทักษ์สกุล	.....

วัน/เดือน/ปีที่สอบ วันจันทร์ที่ 2 กันยายน พ.ศ. 2545 เวลา 17.30 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่ 30 เดือน พย. พ.ศ. 2546

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

ตัวแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51

ASSEMBLER FOR MICROCONTROLLER MCS-51

นายกิตติคุณ อินอ่อน
นายอิทธิพล ไชยถาวร
นายนิติพงษ์ ไกขุนทด

เลขหมู่.....
เลขทะเบียน 48329
วัน, เดือน, ปี 15 ต.ค. 2546

.b.....
.i.....

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ตัวแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51

Assembler for Microcontroller MCS-51

วัตถุประสงค์

1) เพื่อศึกษาหลักการของตัวแปลภาษา และไมโครคอนโทรลเลอร์ MCS-51

2) เพื่อออกแบบโปรแกรมที่ใช้ในการแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51

3) เพื่อเขียนโปรแกรมที่ใช้ในการแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51

4) เพื่อทดสอบโปรแกรมแปลภาษาที่ได้จากการออกแบบและเขียนขึ้น

5) เพื่อนำโปรแกรมที่ผ่านการทดสอบและแก้ไขเสร็จเรียบร้อยไปใช้งาน

ประโยชน์ที่คาดว่าจะได้รับ

1) รู้และเข้าใจหลักการทำงานของตัวแปลภาษา และการทำงานของไมโครคอนโทรลเลอร์ MCS-51

2) ได้โปรแกรมต้นแบบสำหรับโปรแกรมแปลภาษาที่จะนำไปพัฒนา

3) ได้โปรแกรมแปลภาษาที่สามารถนำไปใช้งานได้จริง

4) ได้ผลการทดสอบโปรแกรมแปลภาษา

5) สามารถนำเอาโปรแกรมแปลภาษาที่ได้ผ่านการทดสอบ และแก้ไขแล้วไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	ตัวแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51
นักศึกษา	นายกิตติคุณ อินอ่อน
	นายอิทธิพล ไชยถาวร
	นายนิติพงษ์ ไคขุนทด
อาจารย์ที่ปรึกษา	อาจารย์โกศล ตราชู
อาจารย์ที่ปรึกษาร่วม	อาจารย์สุชิน อาจหาญ
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2545

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เสนอตัวแปลภาษาแอสเซมบลีแบบ 2 รอบสำหรับไมโครคอนโทรลเลอร์ MCS-51 ถูกสร้างขึ้นโดยอาศัยหลักการของคอมไพเลอร์ ตัวแปลภาษานี้ให้ผลลัพธ์เป็น Intel Hex File และ Listing File จากการทดลองสามารถแปลชุดคำสั่งรวมทั้งคำสั่งเทียมได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Assembler for Microcontroller MCS-51
Students	Mr. Kitikun Inion
	Mr. Ittipon Chaitawon
	Mr. Nitipong Kaikuntod
Advisor	Mr. Koson Trachu
Co-Advisor	Mr. Suchin Adhan
Education Level	Bachelor of Science in Industrial Education
Program in	Electronics and Computer
Academic Year	2002

ABSTRACT

This thesis presents translate the assembly language program. Is two-pass process assembler for microcontroller MCS-51. Result of program can be divide two from is Intel Hex File and Listing File. Result can translate has the pseudo operation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงไปได้ดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่ม และความอนุเคราะห์ของอาจารย์โกศล ตราชู และอาจารย์สุชิน อาจหาญ รวมทั้งคณะอาจารย์ประจำภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้คำแนะนำ แนวความคิด ความรู้ต่างๆ แนวทางการแก้ไขปัญหา ในการจัดทำปริญญานิพนธ์

นอกจากนี้ทางคณะผู้จัดทำขอบคุณภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ได้กรุณาให้การสนับสนุนในการจัดทำปริญญานิพนธ์ครั้งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎี และหลักการ	3
2.1 กล่าวนำ	3
2.2 ความรู้ทั่วไปเกี่ยวกับตัวแปลภาษา	3
2.3 โครงสร้างของตัวแปลภาษา	4
2.4 การวิเคราะห์ภาษาต้นแบบ	5
2.4.1 หน้าที่ของการวิเคราะห์ภาษาต้นแบบ	6
2.4.2 แผนภาพการเปลี่ยนแปลง	6
2.4.3 การนำแผนภาพการเปลี่ยนแปลงมาใช้ในการวิเคราะห์ภาษาต้นแบบ	8
2.5 การตรวจสอบวากยสัมพันธ์	13
2.5.1 การอนุพัทธ์	14
2.5.2 พาซทรี	14
2.5.3 แฮนเดิล	17
2.5.4 ไวยากรณ์กำกวม	17
2.5.5 หน้าที่ของการตรวจสอบวากยสัมพันธ์	19
2.5.6 นอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์	20
2.5.7 แพนิกโมด	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.6 การตรวจสอบความหมาย	23
2.7 การปรับปรุงรหัส	24
2.8 การเปลี่ยนรหัส	29
2.9 การจัดตาราง	29
2.10 การตรวจจับและดำเนินการต่อเมื่อพบความผิดพลาด	29
2.11 โครงสร้าง Intel Hex File	30
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	31
3.1 กล่าวนำ	31
3.2 โปรแกรมแอสเซมเบลอร์	33
3.3 ไวยากรณ์ของตัวแปลภาษา	34
3.3.1 ขั้นตอนการวิเคราะห์ภาษาต้นแบบของตัวแปลภาษา	37
3.3.2 ขั้นตอนการตรวจสอบความหมายและการผลิตรหัสของตัวแปลภาษา	38
3.3.3 การตรวจสอบวากยสัมพันธ์ของตัวแปลภาษา	43
บทที่ 4 การทดลอง และผลการทดลอง	45
4.1 กล่าวนำ	45
4.2 การทดสอบชุดคำสั่ง และคำสั่งเทียม	45
4.2.1 ผลการทดสอบเมื่อเรียกใช้โปรแกรม SXA51	52
4.2.2 ผลการทดสอบเมื่อเรียกใช้โปรแกรม ECOM51	53
4.2.3 ผลการทดสอบพารามิเตอร์ -1	58
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา	60
5.1 บทสรุป	60
5.2 ปัญหาและแนวทางการแก้ไข	60
5.3 แนวทางการพัฒนา	61
ภาคผนวก ก ผังการทำงานและโปรแกรม	62
ภาคผนวก ข คู่มือการใช้งาน	134
บรรณานุกรม	139
ประวัติผู้แต่ง	140

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 การกำหนดให้โทเคนต่างๆ มีรหัส และค่า	11
ตารางที่ 2.2 รูปแบบของข้อมูลในระบบ Intel Hex Format	31
ตารางที่ 3.1 สัญลักษณ์นอนเทอร์มินอล และรหัส	43
ตารางที่ 3.1 (ต่อ) สัญลักษณ์นอนเทอร์มินอล และรหัส	44

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูป	หน้า
รูปที่ 2.1 ขั้นตอนต่างๆ ของตัวแปลภาษา	5
รูปที่ 2.2 ความสัมพันธ์ระหว่างการวิเคราะห์ภาษาต้นแบบกับการตรวจสอบวากยสัมพันธ์	6
รูปที่ 2.3 แผนภาพทรานซิชันของตัวชี้เฉพาะ	7
รูปที่ 2.4 รหัสจำลองของแผนภาพทรานซิชันรูปที่ 2.3	7
รูปที่ 2.5 แผนภาพการเปลี่ยนแปลงของตัวชี้เฉพาะ	8
รูปที่ 2.6 แผนภาพทรานซิชันของโทเคน > และ >=	8
รูปที่ 2.7 รหัสจำลองของแผนภาพทรานซิชันรูปที่ 2.6	9
รูปที่ 2.8 แผนภาพทรานซิชันของโทเคน begin end if then และelse	10
รูปที่ 2.9 แผนภาพทรานซิชันของโทเคนค่าคงที่	10
รูปที่ 2.10 แผนภาพทรานซิชันของโทเคน < <= = <> > และ >=	10
รูปที่ 2.11 แผนภาพทรานซิชันของตัวอย่างที่ 2.2	12
รูปที่ 2.12 แผนภาพทรานซิชันของโทเคน begin และตัวชี้เฉพาะ	12
รูปที่ 2.13 ลักษณะของทรี	15
รูปที่ 2.14 พาซทรีของประโยค (id + id)	16
รูปที่ 2.15 พาซทรีของประโยค id * < id + id >	16
รูปที่ 2.16 พาซทรีของกึ่งประโยค < expr > + < expr >	16
รูปที่ 2.17 พาซทรี 2 แบบของประโยค id + id * id ซึ่งผลิตโดยไวยากรณ์ที่ (2.3)	18
รูปที่ 2.18 ตำแหน่งของพาซเซอร์ในโครงสร้างของตัวแปลภาษา	19
รูปที่ 2.19 การกระจายคำของไวยากรณ์ที่ (2.7)	21
รูปที่ 2.20 องค์ประกอบต่างๆ ของนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์	22
รูปที่ 2.21 รหัสจำลองแสดงขั้นตอนการตรวจสอบของนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์	22
รูปที่ 2.22 ซินแทกทรีของตัวอย่างที่ 2.9	24
รูปที่ 2.23 สัญกรณ์ 3 สิ่งที่เป็นอันดับของตัวอย่างที่ 2.9	24
รูปที่ 2.24 รหัสระหว่างกลางที่ใช้แทนคำสั่ง x = y ; y = a * b * (b * a + x) ;	25
รูปที่ 2.25 รหัสระหว่างกลางที่ปรับปรุงแล้ว	26

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 2.26 รหัสระหว่างกลางที่ใช้แทนคำสั่ง x = 2 * 9 / 3 * y	26
รูปที่ 2.27 รหัสระหว่างกลางที่ปรับปรุงแล้ว	26
รูปที่ 3.1 ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา	34
รูปที่ 3.1 (ต่อ) ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา	35
รูปที่ 3.1 (ต่อ) ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา	36
รูปที่ 3.1 (ต่อ) ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา	37
รูปที่ 3.2 แผนภาพทรานซิชันเพื่อแยกโทเคนจากโปรแกรมต้นแบบภาษาแอสเซมบลี	38
รูปที่ 3.3 ไวยากรณ์ล่ามของตัวแปลภาษา	39
รูปที่ 3.3 (ต่อ) ไวยากรณ์ล่ามของตัวแปลภาษา	40
รูปที่ 3.3 (ต่อ) ไวยากรณ์ล่ามของตัวแปลภาษา	41
รูปที่ 3.3 (ต่อ) ไวยากรณ์ล่ามของตัวแปลภาษา	42
รูปที่ 4.1 โปรแกรมที่ใช้ในการทดสอบ	45
รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ	46
รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ	47
รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ	48
รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ	49
รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ	50
รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ	51
รูปที่ 4.2 ผลการทดสอบเมื่อเรียกใช้โปรแกรม SXA51	52
รูปที่ 4.3 ผลการทดสอบเมื่อเรียกใช้โปรแกรม ECOM51	53
รูปที่ 4.4 ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51	54
รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51	55
รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51	56
รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51	57
รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51	58
รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51	59
รูปที่ ก.1 ผังงานของโปรแกรมหลัก	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ ก.1 ผังงานของโปรแกรมหลัก	64
รูปที่ ก.2 ผังงานของโปรแกรม LEXER.C	65
รูปที่ ก.3 โปรแกรม MAIN.C	66
รูปที่ ก.4 โปรแกรม LEXER.C	63
รูปที่ ก.5 โปรแกรม EMITTER.C	74
รูปที่ ก.6 โปรแกรม SYMBOL.C	80
รูปที่ ก.7 โปรแกรม EC51.C	86
รูปที่ ก.8 โปรแกรม EC51.Y	112
รูปที่ ก.9 โปรแกรม Y_TAB.H	130
รูปที่ ก.10 โปรแกรม EC51.H	132

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันมีความต้องการแนวทางในการหาขั้นตอนและวิธีการสร้างตัวแปลภาษาแนวทางในการสร้างโปรแกรมสำหรับการแปลภาษาต้นแบบเพื่อนำไปประยุกต์ใช้ในไมโครคอนโทรลเลอร์ตระกูลต่างๆ ในที่นี้เราจะใช้ไมโครคอนโทรลเลอร์ MCS-51 เป็นตัวทดสอบ เนื่องจากการพัฒนาโปรแกรมบนไมโครคอนโทรลเลอร์มีข้อจำกัดบางประการ จึงต้องทำการพัฒนาโปรแกรมให้ใช้บนเครื่องคอมพิวเตอร์แทน

1.2 ขีดความสามารถของโครงงาน

โครงงานนี้มีขีดความสามารถดังต่อไปนี้

1) เป็นตัวแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51
2) เป็นตัวแปลภาษาจากภาษาต้นฉบับไปเป็นภาษาเครื่อง
3) สามารถทำงานได้เฉพาะบนเครื่องคอมพิวเตอร์เท่านั้น
4) ใช้ภาษาซีในการพัฒนาโปรแกรม

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อความสะดวกต่อการศึกษาและทำความเข้าใจในแต่ละบทจะประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 1 กล่าวถึงความเป็นมา และความสำคัญของปริญญานิพนธ์ ขีดความสามารถของโครงงาน และเนื้อหาโดยสังเขป

บทที่ 2 กล่าวถึงทฤษฎี และหลักการ ความรู้ทั่วไปเกี่ยวกับตัวแปลภาษา โครงสร้างของตัวแปลภาษา การวิเคราะห์ภาษาต้นแบบ การตรวจสอบวากยสัมพันธ์ การตรวจสอบความหมาย การปรับปรุงรหัส การเปลี่ยนรหัส การจัดตาราง การตรวจสอบและดำเนินการต่อเมื่อพบความผิดพลาด โครงสร้าง Intel Hex File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบ การสร้าง และการทำงาน ประกอบด้วยเนื้อหาในเรื่องของการออกแบบโปรแกรม เพื่อให้ผู้อ่านได้เข้าใจวิธีการในการออกแบบ และการสร้างเพื่อดำเนินการแก้ไขหรือการปรับปรุงโปรแกรมให้มีประสิทธิภาพเพิ่มขึ้นในโอกาสต่อไป

บทที่ 4 การทดลอง และผลการทดลอง กล่าวถึงขั้นตอนในการทำการทดลอง และผลที่ได้การทดลอบโปรแกรม โดยมีการเปรียบเทียบผลการทดลองที่ได้กับโปรแกรมที่มีใช้อยู่ในปัจจุบัน

บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไข และพัฒนา กล่าวถึงปัญหาที่เกิดขึ้นจากการทำงานของผู้ศึกษา และแนวทางในการแก้ไขที่ได้รับจากอาจารย์ที่ปรึกษาปริญญานิพนธ์ในด้านต่างๆ

ในภาคผนวกแสดงรายละเอียดของโปรแกรม และส่วนต่างๆ ที่ใช้จัดทำโครงงานดังนี้

ภาคผนวก ก ผังการทำงานและโปรแกรม

ภาคผนวก ข คู่มือการใช้งาน

บรรณานุกรม

ประวัติผู้แต่ง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี และหลักการ

2.1 กล่าวนำ

การเขียนโปรแกรมแปลภาษาของไมโครคอนโทรลเลอร์ MSC-51 นั้นเราใช้โปรแกรมภาษาซีในการเขียนโปรแกรมซึ่งเป็นโปรแกรมที่ทำงานภายใต้ระบบปฏิบัติการวินโดว์ดังนั้นเรายังต้องศึกษาเกี่ยวกับความรู้ทั่วไปเกี่ยวกับตัวแปลภาษา โครงสร้างของตัวแปลภาษา และขั้นตอนการทำงานต่างๆ ของตัวแปลภาษา โครงสร้าง Intel Hex File โครงสร้างของโปรแกรมภาษาแอสเซมบลี ซึ่งสามารถกล่าวโดยละเอียดได้ดังต่อไปนี้

2.2 ความรู้ทั่วไปเกี่ยวกับตัวแปลภาษา

เนื่องจากภาษาเครื่อง (Machine Language) เป็นภาษาที่เครื่องคอมพิวเตอร์เข้าใจ และสามารถติดต่อได้โดยตรงกับเครื่องซึ่งอยู่ในรูปของบิต (Bit) รีจิสเตอร์ (Register) และการทำงานพื้นฐาน (Primitive Machine Operation) ภาษาเครื่องเป็นลำดับของเลข 0 และ 1 เรียงต่อกัน ซึ่งถ้านำมาเขียนเป็นโปรแกรมจะเกิดความยุ่งยาก ไม่สะดวกมักจะเกิดปัญหา และข้อผิดพลาดเสมอ เนื่องจากคำสั่งต่างๆ เป็นรหัสตัวเลข ทำให้ยากต่อการปรับปรุงในการนำมาใช้ในภายหลัง จึงได้มีการพัฒนาภาษาสัญลักษณ์ (Symbolic Language) หรือภาษาแอสเซมบลี (Assembly Language) ขึ้นมา แต่อย่างไรก็ตามภาษาดังกล่าวยังคงใกล้เคียงกับภาษาเครื่อง แม้จะเปลี่ยนจากรหัสตัวเลขมาเป็นสัญลักษณ์แทน แต่คำสั่งต่างๆ ก็ยังเป็นการทำงานขั้นพื้นฐาน ในที่สุดจึงได้มีการพัฒนาเป็นภาษาระดับสูง โปรแกรมที่เขียนด้วยภาษาระดับสูง จะเขียนได้ง่ายกว่า มีความผิดพลาดน้อยกว่า และง่ายต่อการปรับปรุงแก้ไขในภายหลัง ภาษาในระยะหลังที่ไม่ใช่ภาษาเครื่องจึงต้องมีตัวแปล

2.2.1 ตัวแปล (Translator)

ตัวแปล หมายถึง โปรแกรมซึ่งทำหน้าที่แปลภาษาจากภาษาหนึ่งซึ่งเรียกว่า ภาษาต้นแบบ (Source Language) ไปเป็นอีกภาษาหนึ่งซึ่งเรียกว่า ภาษาเป้าหมาย (Object or Target Language) ตัวแปลที่ใช้ในการแปลภาษาระดับสูง ได้แก่ ฟอร์แทรน (Fortran) พีแอลวัน (PL/1) ปาสคาล (Pascal) หรือ โคบอล (Cobol) ไปเป็นภาษาเครื่องหรือในภาษาแอสเซมบลีจะเรียกว่า คอมไพเลอร์ (Compiler) ถ้าภาษาต้นแบบเป็นภาษาแอสเซมบลี และภาษาเป้าหมายเป็นภาษาเครื่อง ตัวแปลภาษานี้เรียกว่า แอสเซมเบอร์ (Assembler) ตัวแปลซึ่งแปลภาษาต้นแบบไปเป็นรหัสระหว่างกลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

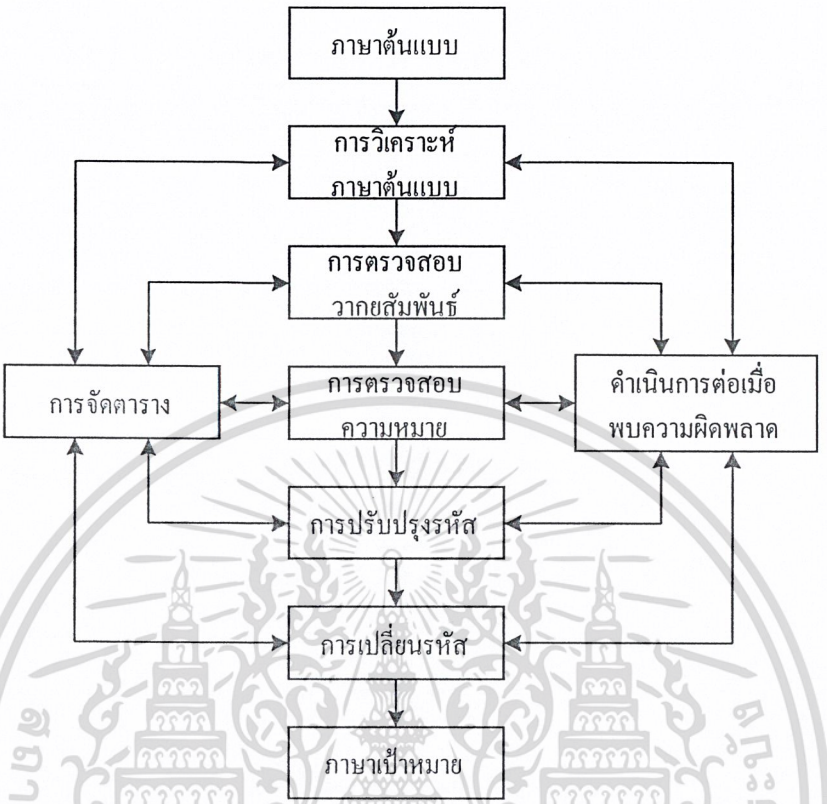
(Intermediate Code) แล้วปฏิบัติการ (Execute) กับรหัสระหว่างกลางนั้นโดยตรง เรียกว่า อินเตอร์พรีเตอร์ (Interpreter) มีภาษาต้นแบบบางภาษา เช่น คำสั่งที่ติดต่อกับระบบปฏิบัติการ (Operating System) เป็นรหัสระหว่างกลางโดยตัวมันเองไม่จำเป็นต้องแปลอินเตอร์พรีเตอร์จะตีความ (Interpret) และปฏิบัติการกับภาษาต้นแบบนั้นโดยตรง

ตัวแปลซึ่งแปลโปรแกรมที่เขียนด้วยภาษาระดับสูง ไปเป็นโปรแกรมภาษาระดับสูง เช่น ในภาษาฟอร์แทรนได้มีการสร้างคำสั่งแบบมีโครงสร้างเพิ่มเติมจากที่มีอยู่แล้ว ภาษาระดับสูงที่สร้างใหม่ขึ้นมานี้จะใช้ตัวแปลที่เรียกว่า พรีโพรเซสเซอร์ (Preprocessor) แปลไปเป็นภาษาฟอร์แทรน เป็นต้น

2.3 โครงสร้างของตัวแปลภาษา

โครงสร้างของตัวแปลภาษานั้นค่อนข้างซับซ้อนเพื่อให้ง่ายในการเขียนโปรแกรมจึงได้แบ่งออกเป็นขั้นตอนต่างๆ ดังรูปที่ 2.1 แต่ละขั้นตอนจะทำหน้าที่ต่างกัน เริ่มตั้งแต่ขั้นตอนแรกเรียกว่า การวิเคราะห์ภาษาต้นแบบ (Lexical Analyzer) ทำหน้าที่อ่านอักขระจากภาษาต้นแบบแล้วแยกอักขระเหล่านั้นออกเป็นกลุ่มๆ เรียกว่า เครื่องหมาย (Token) กลุ่มอักขระใดที่แยกออกเป็นเครื่องหมายก็ขึ้นอยู่กับหลักภาษาต้นแบบที่กำหนดไว้ ตัวอย่างเครื่องหมายของภาษาระดับสูงทั่วไปได้แก่ คำหลัก (Keyword) เช่น BEGIN IF END ชื่อตัวแปร (Variable Name) ค่าคงที่ (Constant) หรือตัวดำเนินการเลขคณิต (Arithmetic Operator) เช่น เครื่องหมาย + - * / ขั้นตอนที่สองเรียกว่า การตรวจสอบวากยสัมพันธ์ (Syntax Analyzer) ทำหน้าที่ตรวจสอบว่าเครื่องหมายที่ได้จากขั้นตอนที่หนึ่งนั้น ได้นำมาเขียนเรียงกันถูกต้องตามวากยสัมพันธ์ (Syntax) ของภาษาต้นแบบหรือไม่ เช่น เรียงกันเป็นนิพจน์ เป็นคำสั่ง IF เป็นคำสั่ง WHILE ถูกต้องหรือไม่ ขั้นที่สามเรียกว่าการตรวจสอบความหมาย (Semantic Analyzer) ทำหน้าที่ตรวจสอบความหมาย (Semantics) และตีความเพื่อผลิตรหัสระหว่างกลางที่ทำงานเหมือนภาษาต้นแบบ ในขั้นตอนที่สี่เรียกว่าการปรับปรุงรหัส (Code Optimizer) ทำหน้าที่ปรับปรุงรหัสระหว่างกลางที่ได้จากขั้นที่สามให้มีประสิทธิภาพมากขึ้น เช่น ขจัดการคำนวณที่ซับซ้อน และไม่มีความจำเป็นออกไป ขั้นตอนสุดท้ายเรียกว่าการเปลี่ยนรหัส (Code Generator) ทำหน้าที่เปลี่ยนรหัสระหว่างกลางให้เป็นภาษาเป้าหมาย นอกจากนี้ข้อมูลที่ได้จากขั้นตอนหนึ่งอาจจะนำไปใช้อีกขั้นตอนหนึ่งได้ ข้อมูลเหล่านี้จะถูกเก็บไว้ในตารางเพื่อใช้ร่วมกัน ตัวแปลภาษาจะต้องทำหน้าที่จัดตาราง (Table Manager) เหล่านี้ด้วย และเมื่อมีความผิดพลาดเกิดขึ้นในภาษาเป้าหมาย ตัวแปลภาษาจะต้องทำหน้าที่ตรวจจับ และดำเนินการต่อ (Error Detection and Recovery) ให้ได้โดยไม่หยุดการทำงานเมื่อพบความผิดครั้งแรก

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ขั้นตอนต่างๆ ของตัวแปลภาษา

ตัวแปลภาษาแต่ละตัวอาจจะมีขั้นตอนที่ไม่เหมือนกับที่ได้กล่าวมาแล้วทั้งหมด เช่น ในขั้นตอนที่สามอาจสร้างภาษาเป้าหมายโดยไม่ต้องสร้างรหัสระหว่างกลางเลยก็ได้ หรือในขั้นตอนการปรับปรุงรหัสระหว่างกลางอาจจะทำมากกว่าหนึ่งขั้นตอนก็ได้ คือ ทำการปรับปรุงรหัสระหว่างกลาง และปรับปรุงภาษาเป้าหมายก็ได้หรืออาจจะไม่มีขั้นตอนการปรับปรุงรหัสระหว่างกลางเลยก็ได้ เช่น ตัวแปลภาษาที่ใช้ในการเรียนการสอน ส่วนใหญ่จะเสียเวลาในการตรวจสอบหลักภาษามากกว่าที่จะนำภาษาเป้าหมายไปใช้งาน จึงไม่จำเป็นที่จะต้องทำขั้นตอนการปรับปรุงรหัสระหว่างกลาง

2.4 การวิเคราะห์ภาษาต้นแบบ (Lexical Analyzer)

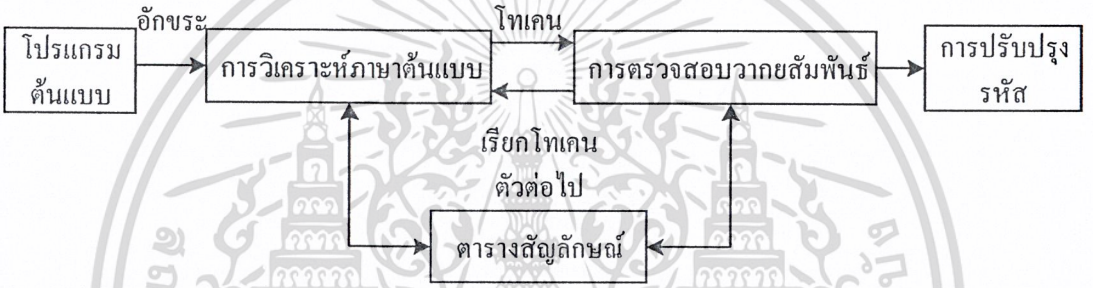
ขั้นตอนนี้ทำหน้าที่อ่านอักขระจากภาษาต้นแบบ แล้วทำการจัดการแยกอักขระเหล่านั้นออกเป็นเครื่องหมายตามลักษณะหน้าที่ และความหมาย เช่น เครื่องหมายคำหลัก เครื่องหมายชื่อตัวแปร เครื่องหมายค่าคงที่ วิธีการง่ายๆ ที่จะสร้างขั้นตอนการวิเคราะห์ภาษาต้นแบบ คือ การสร้างแผนภาพ (Diagram) แสดงโครงสร้างโทเคนของโปรแกรมต้นแบบจากแผนภาพต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แล้วเขียนโปรแกรม จากแผนภาพดังกล่าวเพื่อแยกอักขระหรือกลุ่มของอักขระในโปรแกรมต้นแบบออกเป็นโทเคน

2.4.1 หน้าที่ของการวิเคราะห์ภาษาต้นแบบ

การวิเคราะห์ภาษาต้นแบบนั้นจะเป็นขั้นตอนแรกของตัวแปลภาษาซึ่งจะหน้าที่หลัก คือ อ่านอักขระครั้งละตัวจากโปรแกรมต้นแบบ แล้วทำการส่งผลลัพธ์ที่เป็นเครื่องหมายให้กับขั้นตอนการตรวจสอบวากยสัมพันธ์ และเก็บค่าซึ่งจำเป็นต้องใช้ในขั้นตอนอื่นไว้ในตารางสัญลักษณ์ ความสัมพันธ์ระหว่างขั้นตอนทั้งสอง แสดงได้ดังรูปที่ 2.2



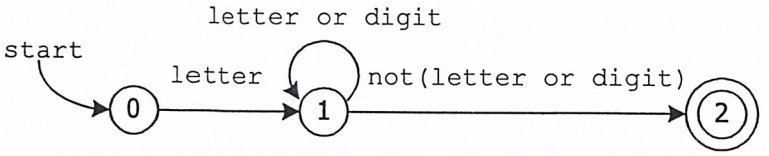
รูปที่ 2.2 ความสัมพันธ์ระหว่างการวิเคราะห์ภาษาต้นแบบกับการตรวจสอบวากยสัมพันธ์

หน้าที่อื่นๆ เช่น ขจัดคำอธิบาย (Comment) หรือช่องว่างขาว White Space สัญลักษณ์ที่ถือว่าเป็นช่องว่างขาวได้แก่ ช่องว่าง (Blank) แทป (Tab) และขึ้นบรรทัดใหม่ (Newline) ในโปรแกรมต้นแบบทิ้ง และเนื่องจากการวิเคราะห์ภาษาต้นแบบเป็นขั้นตอนที่รู้ว่าอ่านอักขระในโปรแกรมต้นแบบไปถึงบรรทัด (Line) ไหน สดมภ์ (Column) ที่เท่าใดในบรรทัดนั้นจึงจำเป็นต้องจดจำบรรทัด และสดมภ์ไว้ เพื่อให้ขั้นตอนอื่นๆ นำไปใช้บอกตำแหน่งความผิดพลาดในโปรแกรมต้นแบบ นอกจากนี้ยังต้องเก็บชื่อตัวแปร ชื่อการปฏิบัติ ชื่อฟังก์ชัน เลเบล หรือค่าคงที่ เป็นต้นไว้ในตารางสัญลักษณ์เพื่อใช้ร่วมกันกับขั้นตอนอื่นๆ ความผิดพลาดที่ตรวจจับได้ในขั้นตอนนี้ เช่น ผู้เขียนโปรแกรมเขียนเครื่องหมายผิด หรือมีอักขระที่ไม่อยู่ในชุดของอักขระของภาษาต้นแบบ

2.4.2 แผนภาพทรานซิชัน

แผนภาพทรานซิชัน (Transition Diagram) เป็นแผนภาพที่ประกอบด้วยวงกลมซึ่งเรียกว่า สถานะ (State) แต่ละสถานะจะเชื่อมโยงลูกศรเรียกว่า สัน (Edge) บนสันจะมีเลเบลเป็นอักขระใช้เป็นตัวกำหนดว่าจะไปสถานะใดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แผนภาพทรานซิชันของตัวชี้เฉพาะ

รูปที่ 2.3 แสดงแผนภาพทรานซิชันของตัวชี้เฉพาะ (Identifier) ซึ่งมีนิยามว่า จะต้องขึ้นต้นด้วยตัวอักษร (Letter) อย่างน้อยหนึ่งตัว และตามด้วยตัวอักษรหรือตัวเลข (Digit) กี่ตัวก็ได้ จากรูปอธิบายได้ดังนี้ สถานะเริ่มต้น (Start State) คือ สถานะ 0 ที่สถานะ 0 ถ้าอักขระที่อ่านเข้ามาเป็นตัวอักษรให้ไปยังสถานะ 1 ที่สถานะ 1 ถ้าอักขระที่อ่านเข้ามาเป็นอักษรหรือตัวเลขให้วนกลับมายังสถานะ 1 ใหม่ แต่ถ้าอักขระที่อ่านเข้ามาไม่เป็นตัวอักษรหรือตัวเลขให้ไปที่สถานะ 2 จะเห็นว่าที่สถานะ 2 มีวงกลมซ้อนกันสองวง แสดงให้ทราบว่าเป็นสถานะสุดท้าย บอกให้ทราบว่ากลุ่มของอักขระที่อ่านเข้ามาดังกล่าวเป็นโทเคนตัวชี้เฉพาะ

จากแผนภาพทรานซิชันในรูปที่ 2.3 นำมาเขียนเป็นรหัสจำลอง (Pseudo Code) ได้ดังรูปที่ 2.4

```
state 0 : C:=GETCHAR ( );
          if LETTER (C) then goto state1
          else FAIL ( );
state 1 : C:=GETCHAR ( );
          if LETTER (C) or DIGIT (C) then goto state1
          else goto state2;
state 2 : RETRACT;
          Return (id, INSTALL( ));
```

รูปที่ 2.4 รหัสจำลองของแผนภาพทรานซิชันรูปที่ 2.3

รหัสจำลองรูปที่ 2.4 มีฟังก์ชัน และการปฏิบัติดังนี้

GETCHAR ( ) เป็นฟังก์ชันที่ให้ค่าอักขระตัวต่อไป ซึ่งอ่านมาจากโปรแกรมต้นแบบ

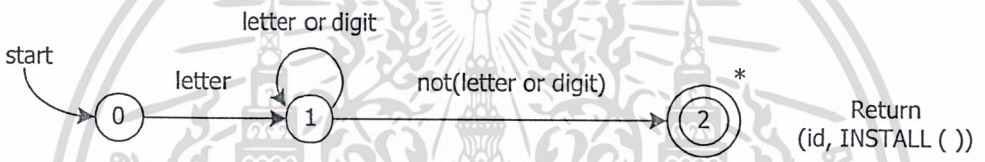
LETTER (C) เป็นฟังก์ชันที่ให้ค่าจริง ถ้าตัวแปร C เก็บค่าตัวอักษรมิฉะนั้นจะให้ค่าเท็จ

DIGIT (C) เป็นฟังก์ชันที่ให้ค่าจริง ถ้าตัวแปร C เก็บค่าตัวเลขมิฉะนั้นจะให้ค่าเท็จ

FAIL ( ) เป็นโพรซีเยอร์บอกข่าวสารความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

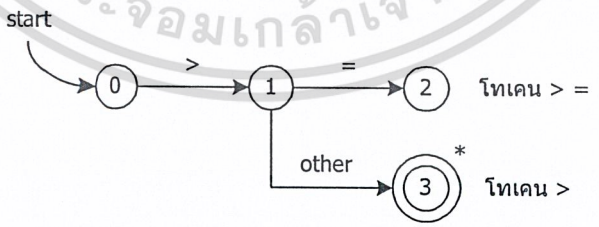
เนื่องจากอักขระที่อ่านก่อนที่จะมายังสถานะ 2 ไม่ใช่อักขระของตัวชี้เฉพาะดังนั้นที่สถานะ 2 จึงจำเป็นต้องเลื่อนอักขระในโปรแกรมต้นแบบกลับมาหนึ่งอักขระ ในกรณีเช่นนี้จะใช้คำว่า RETRACT ในรหัสจำลอง และในแผนภาพจะแทนด้วยเครื่องหมาย * ตรงสถานะดังกล่าว นอกจากนี้เรายังต้องเก็บตัวชี้เฉพาะไว้ในตารางสัญลักษณ์ด้วย ฉะนั้นที่สถานะ 2 นอกจากจะส่งรหัสของโทเคน (ในที่นี้ใช้ id แทนตัวชี้เฉพาะ) จะต้องส่งตำแหน่งของตัวชี้เฉพาะที่เก็บไว้ในตารางสัญลักษณ์ไปใช้ในการตรวจสอบวากยสัมพันธ์ด้วย ในรหัสจำลองใช้ฟังก์ชัน INSTALL( ) ซึ่งจะเป็นตัวที่ทำหน้าที่เก็บตัวชี้เฉพาะไว้ในตารางสัญลักษณ์ และให้ค่าตำแหน่งที่ตัวชี้เฉพาะนั้นเก็บในตารางสัญลักษณ์ จากรหัสจำลองเรานำมาปรับปรุงแผนภาพทรานซิชันใหม่เพื่อให้สอดคล้องกัน และเขียนรูปที่ 2.3 ใหม่ ได้ดังรูปที่ 2.5



รูปที่ 2.5 แผนภาพการเปลี่ยนแปลงของตัวชี้เฉพาะ

2.4.3 การนำแผนภาพทรานซิชันมาใช้ในการวิเคราะห์ภาษาต้นแบบ

ตัวอย่างที่ 2.1 สมมติโปรแกรมต้นแบบประกอบด้วยโทเคน > และ > = เท่านั้น สามารถเขียนแผนภาพทรานซิชันเพื่อแยกโทเคนดังกล่าวได้ ตามรูปที่ 2.6



รูปที่ 2.6 แผนภาพทรานซิชันของโทเคน > และ > =

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลเบล Other ในรูปที่ 2.6 ใช้ในกรณีต้องการบอกให้ทราบว่าเป็นอักขระอื่นๆ ที่ไม่ใช่ = ในกรณีที่สถานะหนึ่ง เช่น สถานะ s มีสันหลายๆ สันแยกออกไป เลเบล Other จะหมายถึงอักขระอื่นๆ ที่ไม่ใช่อักขระบนสันที่แยกออกมาจากสถานะ s จากรูปที่ 2.6 เขียนเป็นรหัสจำลองได้ตามรูปที่ 2.7

```
state 0: C:=GETCHAR( );
         if C= '>' then goto state1
         else FAIL ( );
state 1: C:=GETCHER( );
         if C= '=' then goto state2
         else goto state3;
state 2: Return (>=);
state 3: RETRACT;
         Return (>);
```

รูปที่ 2.7 รหัสจำลองของแผนภาพทรานซิชันรูปที่ 2.6

กรณีที่โปรแกรมต้นแบบเขียนดังนี้ > = > > = > > รหัสจำลองตามรูปที่ 2.7 จะแยกโทเคนได้ดังนี้ >= > >= > >

ตัวอย่างที่ 2.2 สมมติโปรแกรมต้นแบบประกอบด้วยโทเคน 4 ประเภทคือ

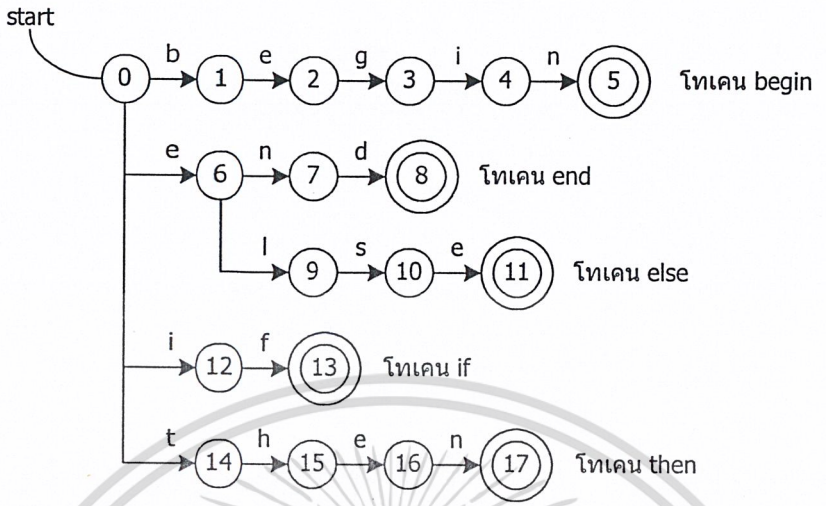
1) คำหลัก ได้แก่ โทเคน begin end if then และelse แสดงด้วยแผนภาพทรานซิชันได้ดังรูปที่ 2.8

2) ตัวชี้เฉพาะแสดงด้วยแผนภาพทรานซิชันได้ดังรูปที่ 2.5

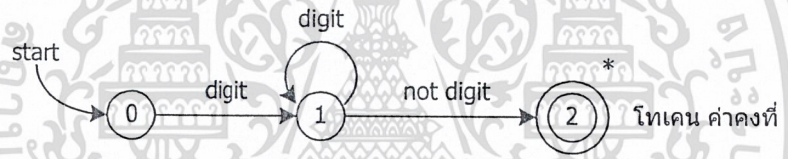
3) ค่าคงที่แสดงด้วยแผนภาพทรานซิชันได้ดังรูปที่ 2.9

4) ตัวดำเนินการสัมพันธ์ (Relational Operator) ได้แก่ < <= = < > > และ >= แสดงด้วยแผนภาพทรานซิชันได้ดังรูปที่ 2.10

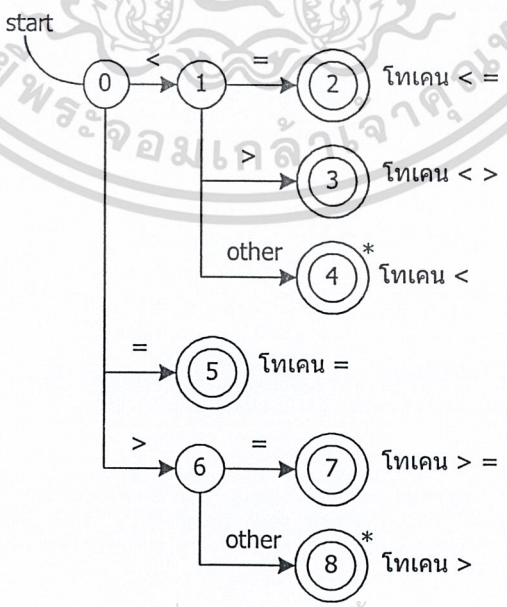
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แผนภาพทรานซิชันของโทเคน begin end if then และ else



รูปที่ 2.9 แผนภาพทรานซิชันของโทเคนค่าคงที่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10 แผนภาพทรานซิชันของโทเคน < <= = <> > และ >=

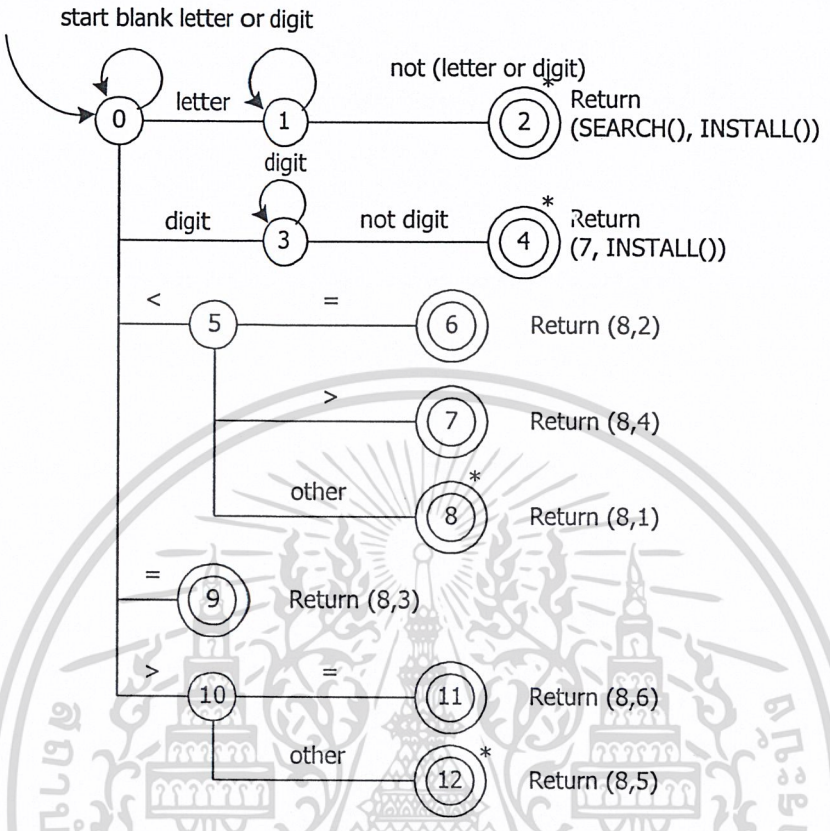
เราสามารถเขียนแผนภาพทรานซิชันรวมเพื่อแยกโทเคนทั้ง 4 ประเภท และขจัดช่องว่างในโปรแกรมต้นแบบทิ้งได้ดังรูปที่ 2.11 โดยกำหนดให้โทเคนต่างๆ มีรหัส และค่าดังนี้

ตารางที่ 2.1 การกำหนดให้โทเคนต่างๆ มีรหัส และค่า

โทเคน	รหัส	ค่า
Begin	1	-
End	2	-
If	3	-
Then	4	-
Else	5	-
ตัวชี้เฉพาะ	6	ตำแหน่งของตัวชี้เฉพาะในตารางสัญลักษณ์
ค่าคงที่	7	ตำแหน่งของค่าคงที่ในตารางสัญลักษณ์
<	8	1
<=	8	2
=	8	3
<>	8	4
>	8	5
>=	8	6

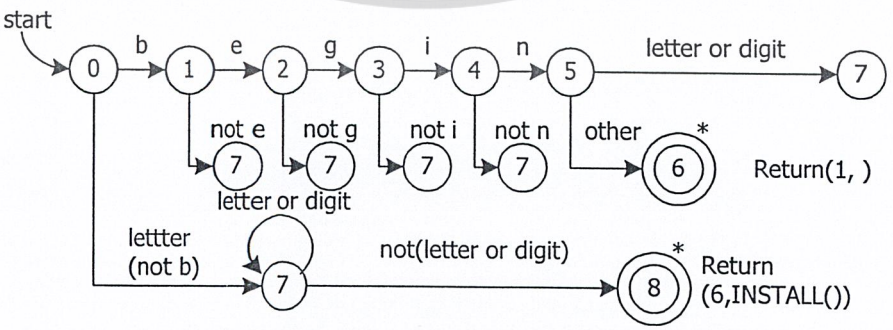
แผนภาพทรานซิชันของคำหลักในรูปที่ 2.8 สามารถตัดทิ้งได้เนื่องจากคำหลักเป็นลำดับของตัวอักษร ซึ่งเป็นกรณีพิเศษของตัวชี้เฉพาะจึงสามารถเขียนแผนภาพทรานซิชันเฉพาะโครงสร้างของตัวชี้เฉพาะก็พอ และเพื่อให้สามารถแยกคำหลักออกจากตัวชี้เฉพาะ ที่สถานะสุดท้ายของตัวชี้เฉพาะคือ สถานะ 2 ให้นำลำดับของตัวอักษรที่ได้ไปค้นในตาราง ซึ่งบรรจุเฉพาะโทเคนและรหัสของโทเคนประเภทคำหลัก ถ้าพบแสดงว่าเป็นคำหลักให้ส่งรหัสของโทเคนที่พบ แต่ถ้าไม่พบแสดงว่าเป็นโทเคนตัวชี้เฉพาะให้นำไปเก็บในตารางสัญลักษณ์พร้อมทั้งส่งรหัสของโทเคนตัวชี้เฉพาะ และตำแหน่งของตัวชี้เฉพาะในตารางสัญลักษณ์ กรณีที่กล่าวมานี้จะใช้ฟังก์ชัน SEARCH ( ) สำหรับตรวจค้นในตารางคำหลัก ซึ่งถ้าตรวจพบจะให้รหัสของโทเคนที่เป็นคำหลัก แต่ถ้าไม่พบจะให้รหัสของโทเคนเป็นตัวชี้เฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แผนภาพทรานซิชันของตัวอย่างที่ 2.2

วิธีการเขียนแผนภาพคำหลักกับแผนภาพตัวชี้เฉพาะเป็นแผนภาพเดียวกันจะช่วยลดจำนวนสถานะลงได้อย่างมาก รูปที่ 2.12 เปรียบเทียบให้เห็นจำนวนสถานะที่เพิ่มมากขึ้น ถ้าเขียนแผนภาพทรานซิชันแยกระหว่างคำหลัก (ในภาพแสดงเฉพาะคำหลัก begin) กับตัวชี้เฉพาะ



รูปที่ 2.12 แผนภาพทรานซิชันของโทเคน begin และตัวชี้เฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างข้างต้นแสดงให้เห็นวิธีการสร้างขั้นตอนการวิเคราะห์ภาษาต้นแบบโดยการเขียนแผนภาพทรานซิชันรวมกับทุกๆ โทเคนในโปรแกรมต้นแบบโครงสร้างของโทเคนขึ้นอยู่กับภาษาต้นแบบ และไวยากรณ์คอนเทกซ์ฟรีที่อธิบายภาษาต้นแบบ

2.5 การตรวจสอบวากยสัมพันธ์ (Syntax Analyzer)

วากยสัมพันธ์ของภาษาระดับสูงนั้นสามารถอธิบายได้ด้วยไวยากรณ์คอนเทกซ์ฟรีเป็นส่วนใหญ่ ดังนั้นหน้าที่หลักของการตรวจสอบวากยสัมพันธ์ คือ ตรวจสอบวากยสัมพันธ์ของภาษาส่วนที่อธิบายได้ด้วยไวยากรณ์คอนเทกซ์ฟรี ซึ่งจะมีขั้นตอน และวิธีในการตรวจสอบไวยากรณ์คอนเทกซ์ฟรีได้ด้วยกันหลายวิธี แต่ในที่นี้จะกล่าวถึงเฉพาะนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์ (Nonrecursive Predictive Parser) เท่านั้น ไวยากรณ์คอนเทกซ์ฟรีที่ (2.1) เป็นตัวอย่างการอธิบายวากยสัมพันธ์ของภาษา ซึ่งประกอบด้วย 2 ประโยค คือ "แดงกินกล้วย" และ "ดำกินกล้วย"

$$
\begin{aligned}
&\langle \text{ประโยค} \rangle \rightarrow \langle \text{ประธาน} \rangle \ \langle \text{กริยา} \rangle \ \langle \text{กรรม} \rangle \\
&\langle \text{ประธาน} \rangle \rightarrow \text{แดง} \mid \text{ดำ} \\
&\langle \text{กริยา} \rangle \rightarrow \text{กิน} \\
&\langle \text{กรรม} \rangle \rightarrow \text{กล้วย}
\end{aligned}
\tag{2.1}
$$

ไวยากรณ์ที่ (2.1) เขียนเฉพาะโครงสร้างประโยคเท่านั้น สัญลักษณ์ทางซ้ายของโครงสร้างประโยคเป็นสัญลักษณ์นอนเทอร์มินอล โดยจะเขียนสัญลักษณ์ดังกล่าวเริ่มต้นด้วยเครื่องหมาย < ตามด้วยอักขระใดๆ และปิดท้ายด้วยเครื่องหมาย > ส่วนสัญลักษณ์อื่นๆ เป็นสัญลักษณ์เทอร์มินอล สัญลักษณ์นอนเทอร์มินอลด้านซ้ายของโครงสร้างประโยคแรกเป็นสัญลักษณ์เริ่มต้น ไวยากรณ์ที่ (2.1) มีสัญลักษณ์นอนเทอร์มินอล 4 ตัวคือ < ประโยค > < ประธาน > < กริยา > และ < กรรม > มีสัญลักษณ์เทอร์มินอล 4 ตัวคือ แดง ดำ กิน และ กล้วย โดยที่ < ประโยค > เป็นสัญลักษณ์เริ่มต้น

อีกตัวอย่างหนึ่งถ้าต้องการอธิบายวากยสัมพันธ์ของภาษาง่ายๆ โดยภาษาดังกล่าวขึ้นต้นด้วยคำว่า begin ตามด้วยคำสั่งอย่างน้อยหนึ่งคำสั่ง ถ้าเขียนมากกว่าหนึ่งคำสั่งให้คั่นด้วยเครื่องหมาย ; และจบด้วยคำว่า end สมมติว่าภาษานี้มีคำสั่งเดียวคือ คำสั่ง stop สามารถเขียนไวยากรณ์คอนเทกซ์ฟรีเพื่ออธิบายภาษานี้ ได้ดังนี้

```
<program>   → begin <stmt-list> end
<stmt-list> → <stmt>
            | <stmt> ; <stmt-list>
<stmt>      → stop
```
(2.2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1 การอนุพัทธ์

การตรวจสอบสตริงใดๆ ว่าเป็นประโยคของภาษาที่ผลิตโดยไวยากรณ์คอนเทอร์ฟรีใดหรือไม่ ถ้าเป็นประโยคจะต้องพิสูจน์ให้ได้ว่า สตริงนั้นอนุพัทธ์จากสัญลักษณ์เริ่มต้น มิฉะนั้นสตริงนั้นก็ไม่เป็นประโยค พิจารณาไวยากรณ์คอนเทกซ์ฟรีต่อไปนี้

```
(1) <expr>  ⟶  <expr> + <expr>
(2)            ¦  <expr> * <expr>
(3)            ¦  ( <expr> )
(4)            ¦  id
```
(2.3)

ไวยากรณ์ที่ (2.3) อธิบายนิพจน์เลขคณิต ดังนี้

1) id เป็นนิพจน์เลขคณิต

2) นิพจน์เลขคณิต + (บวก) นิพจน์เลขคณิต เป็นนิพจน์เลขคณิต

3) นิพจน์เลขคณิต * (คูณ) นิพจน์เลขคณิต เป็นนิพจน์เลขคณิต

4) นิพจน์เลขคณิตเขียนอยู่ภายในเครื่องหมาย ( ) เป็นนิพจน์เลขคณิต

ตัวอย่างที่ 2.3 สตริง (id+id) เป็นประโยคของภาษาที่ผลิตโดยไวยากรณ์ที่ (2.3) เพราะ

```
<expr>  ⇒ ( <expr> )                  ใช้โปรดักชัน (3)
        ⇒ ( <expr> + <expr> )          ใช้โปรดักชัน (1)
        ⇒ ( id + <expr> )              ใช้โปรดักชัน (4)
        ⇒ ( id + id )                  ใช้โปรดักชัน (4)
```

นั่นคือ $\langle expr \rangle \Rightarrow^{+} (id + id)$

ตัวอย่างที่ 2.4 สตริง + id * id ไม่เป็นประโยคของภาษาที่ผลิตโดยไวยากรณ์ที่ (2.3) เพราะไม่สามารถทำให้ < expr > อนุพัทธ์ + id * id

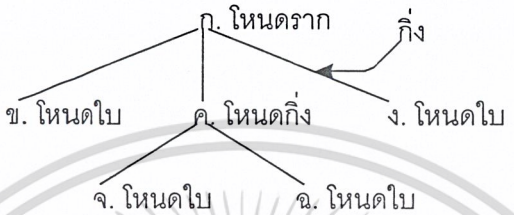
การอนุพัทธ์จะเริ่มที่สัญลักษณ์เริ่มต้น แล้วเลือกโปรดักชันใดโปรดักชันหนึ่งเพียงโปรดักชันเดียวทำการอนุพัทธ์ทันที จะได้สตริงซึ่งเป็นกึ่งประโยค (หรือประโยค) จากกึ่งประโยคที่ได้ทำการอนุพัทธ์ทันที เช่นนี้ไปเรื่อยๆ จนกว่าจะได้ประโยคซึ่งตรงกับสตริงที่ต้องการจะทำการตรวจสอบ

2.5.2 พาซทรี (Parse Tree)

ประโยคหรือกึ่งประโยคที่ได้จากการทำอนุพัทธ์ สามารถแสดงได้ด้วยทรี (Tree) และจะเรียกทรีประเภทนี้ว่า พาซทรี (Parse Tree) ลักษณะของทรีเหมือนกับต้นไม้ที่มีรากอยู่บนสุด และกิ่งก้านสาขาเจริญงอกงามลงมาข้างล่าง ดังแสดงตามรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรีจะประกอบด้วยโหนด (Node) และกิ่ง (Branch) ตรงส่วนรากเรียกว่า โหนดราก (Root Node) จากโหนดรากจะมีกิ่งแตกลงมาด้านล่าง ถ้าตอนปลายกิ่งไม่มีกิ่งแตกลงมาด้านล่างอีก เรียกว่า โหนดใบ (Leave Node) แต่ถ้ามีกิ่งแตกลงมาด้านล่างอีกจะเรียกว่า โหนดกิ่ง (Branch Node)

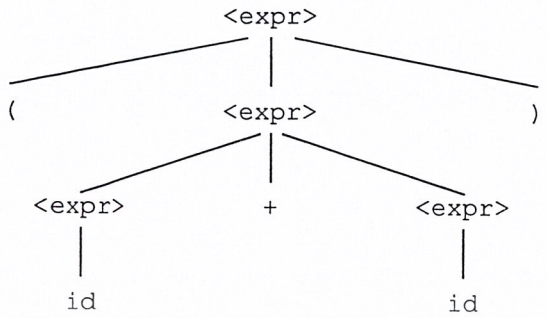


รูปที่ 2.13 ลักษณะของทรี

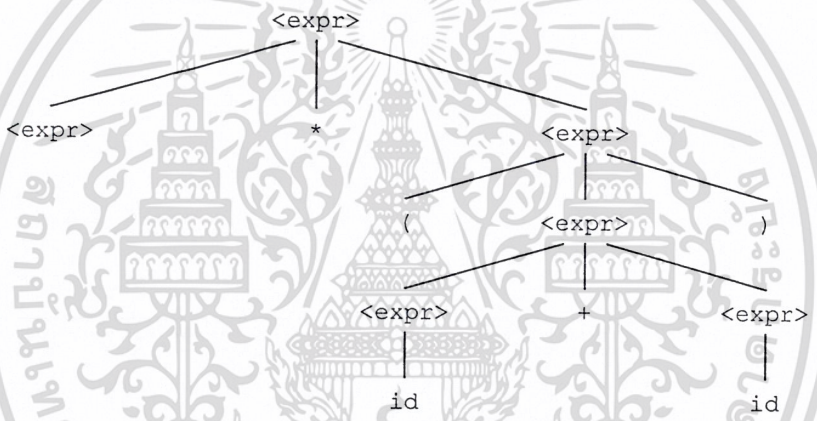
โหนดที่มีกิ่งแตกลงมาเรียกว่า โหนดแม่ (Parent Node) ส่วนโหนดที่อยู่ตอนปลายกิ่งของโหนดแม่เรียกว่า โหนดลูก (Child Node) เช่น รูปที่ 2.13 โหนด ก. เป็นโหนดแม่ของโหนด ข. ค. และ ง. หรือโหนด ค. เป็นโหนดแม่ของโหนด จ. และ ฉ. ในทางกลับกันโหนด ข. ค. และ ง. เป็นโหนดลูกของโหนด ก. หรือโหนด จ. และ ฉ. เป็นโหนดลูกของโหนด ค.

ทรีที่เป็นพาซทรีจะมีลักษณะดังนี้ ตรงโหนดรากจะเป็นสัญลักษณ์เริ่มต้นโหนดกิ่งจะเป็นสัญลักษณ์นอนเทอร์มินอล ส่วนโหนดใบอาจจะเป็นสัญลักษณ์นอนเทอร์มินอลหรือสัญลักษณ์เทอร์มินอล ถ้าทุกๆ โหนดใบเป็นสัญลักษณ์เทอร์มินอล พาซทรีนั้นจะแสดงประโยค แต่ถ้ามีโหนดใบโหนดใดโหนดหนึ่งเป็นสัญลักษณ์นอนเทอร์มินอล พาซทรีนั้นก็จะแสดงกึ่งประโยค ความสัมพันธ์ของโหนดแม่ และโหนดลูกจะแทนการทำอนุพัทธ์ทันที โปรดักชันที่เลือกมาอนุพัทธ์คือ โปรดักชันที่มีสัญลักษณ์นอนเทอร์มินอลซ้ายมือเป็นโหนดลแม่ และสัญลักษณ์ของสตริงขวามือทั้งหมดเป็นโหนดลูก โดยเรียงสัญลักษณ์จากซ้ายไปขวาตามลำดับ สัญลักษณ์ตรงโหนดใบทั้งหมดของพาซทรีเรียงจากซ้ายไปขวาเรียกว่า ยีล์ด (Yield) หรือฟรอนเตียร์ (Frontier) ซึ่งจะแทนกึ่งประโยค หรือประโยค ตัวอย่างจากไวยากรณ์ที่ (2.3) การอนุพัทธ์ของประโยค (id + id) แสดงได้ด้วยพาซทรีดังรูปที่ 2.14 การอนุพัทธ์ของประโยค id * (id + id) แสดงได้ด้วยพาซทรีดังรูปที่ 2.15 และการอนุพัทธ์กึ่งประโยค < expr > + < expr > แสดงได้ด้วยพาซทรีดังรูปที่ 2.16

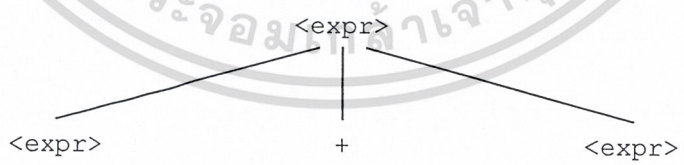
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 พาซทรีของประโยค (id + id)



รูปที่ 2.15 พาซทรีของประโยค id * < id + id >



รูปที่ 2.16 พาซทรีของกึ่งประโยค < expr > + < expr >

ข้อสังเกต พาซทรีไม่สามารถแสดงลำดับของการทำอนุพัทธ์ทันที

วิธีสร้างพาซทรีทำได้ 2 ลักษณะคือ

1) การเริ่มสร้างจากฟรอนเตียร์ขึ้นไปยังสัญลักษณ์เริ่มต้น เรียกว่า การสร้างจากล่างขึ้นบน (Bottom-Up Construction)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2) การเริ่มสร้างจากสัญลักษณ์เริ่มต้นลงไปยังฟรอนเตียร์ เรียกว่า การสร้างจากบนลงล่าง (Top-Down Construction)

2.5.3 แฮนเดิล

พิจารณาการอนุพัทธ์ขวาที่สุดต่อไปนี้ $S \underset{r}{\Rightarrow}_m uAw \underset{r}{\Rightarrow}_m uvw$

เมื่อ S เป็นลัญลักษณ์เริ่มต้น

A เป็นสัญลักษณ์นอนเทอร์มินอล

w เป็นสตริงบนคลีนโคสเชอของเซตเทอร์มินอลอัลฟาเบต

u และ v เป็นสตริงบนคลีนโคสเชอของเซตเทอร์มินอลอัลฟาเบตยูเนียนกับเซตนอนเทอร์มินอลอัลฟาเบต

ตัวอย่างที่ 2.5 จากไวยากรณ์ที่ (2.4) ต่อไปนี้

```
<S> → a <B> c <A> e
<A> → d                    (2.4)
<B> → <B> b|b
```

แฮนเดิลของกึ่งประโยคขวา abbcde คือ โปรดักชัน $< B > \longrightarrow b$ และตำแหน่งที่ 2 เพราะ $< S > \underset{r}{\Rightarrow}_m a < B > c < A > e \underset{r}{\Rightarrow}_m a < B > cde \underset{r}{\Rightarrow}_m a < B > bcde \underset{r}{\Rightarrow}_m abbcde$ นั่นคือ $< S > \underset{r}{\Rightarrow}_m a < B > bcde \underset{r}{\Rightarrow}_m abbcde$ ในกรณีนี้จะบอกสั้นๆ ว่า b เป็นกึ่งประโยคขวา abbcde ไม่ได้ เพราะไม่ทราบว่าเป็น b ตำแหน่งที่ 2 หรือ 3 แต่สามารถบอกสั้นๆ ได้ว่า $< B > b$ เป็นแฮนเดิลของกึ่งประโยคขวา $a < B > bcde$ เพราะมีโปรดักชัน $< B > \longrightarrow < B > b$ และ $< B > b$ ที่ตำแหน่ง 2 เท่านั้น

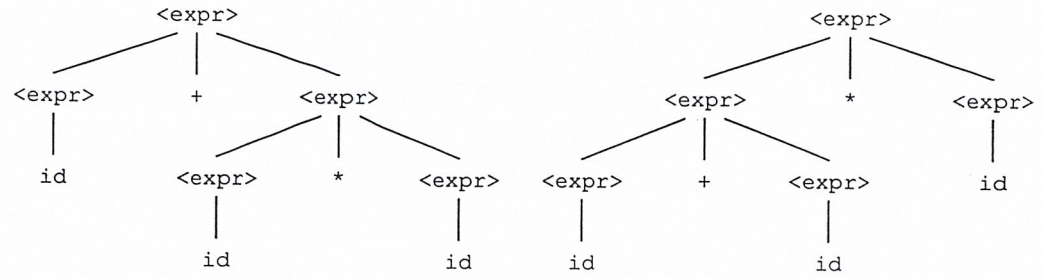
2.5.4 ไวยากรณ์กำกวม

ไวยากรณ์คอนเทกซ์ฟรีที่ผลิตภาษาซึ่งประโยคใดประโยคหนึ่งสามารถสร้างพาซทรีได้มากกว่าหนึ่งแบบเรียกไวยากรณ์คอนเทกซ์ฟรีนั้นว่า ไวยากรณ์กำกวม (Ambiguous Grammars)

ตัวอย่างที่ 2.6 ไวยากรณ์ที่ (2.3) เป็นไวยากรณ์กำกวม เพราะประโยค id + id * id สามารถสร้างพาซทรีได้มากกว่าหนึ่งแบบ ดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

48329



รูปที่ 2.17 พาซทรี 2 แบบของประโยค id + id * id ซึ่งผลิตโดยไวยากรณ์ที่ (2.3)

อีกวิธีหนึ่งที่ใช้ตรวจสอบว่าไวยากรณ์คอนเทกซ์ฟรีเป็นไวยากรณ์กำกวมหรือไม่ คือ ให้พิจารณาจากกึ่งประโยค ซึ่งถ้าหากว่ากึ่งประโยคใดกึ่งประโยคหนึ่งของภาษาที่ผลิตโดยไวยากรณ์คอนเทกซ์ฟรี มีแฮนเดิลมากกว่าหนึ่งแฮนเดิล แสดงว่าไวยากรณ์คอนเทกซ์ฟรีนั้นเป็นไวยากรณ์กำกวม

ตัวอย่างที่ 2.7 ไวยากรณ์ที่ (2.5) เป็นไวยากรณ์กำกวม

```
<if-stmt>  → if <expr> then <stmts> else <stmts>
           | if <expr> then <stmts>
<expr>     → a                                        (2.5)
<stmts>    → <if-stmt>|b
```

เพราะกึ่งประโยคขวา if < exor > then if < expr > then < stmts > else b มีแฮนเดิลมากกว่าหนึ่ง แฮนเดิล ดังนี้

```
1) <if-stmt>   r⇒m   if <expr> then <stmts> else <stmts>
               r⇒m   if <expr> then <stmts> else <stmts> b
               r⇒m   if <expr> then <if-stmts> else b
               r⇒m   if <expr> then if  <expr> then
                     <stmts> else b

2) <if-stmt>   r⇒m   if <expr> then <stmts>
               r⇒m   if <expr> then <if-stmts>
               r⇒m   if <expr> then if <expr> then <stmts>
                     else <stmp>
               r⇒m   if <expr> then if  <expr> then <stmts>
                     else b
```

จากข้อที่ 1 แฮนเดิลคือ จัดลำดับ if < expr > then < stmts > และจากข้อที่ 2 แฮนเดิลคือ จัดลำดับ b

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

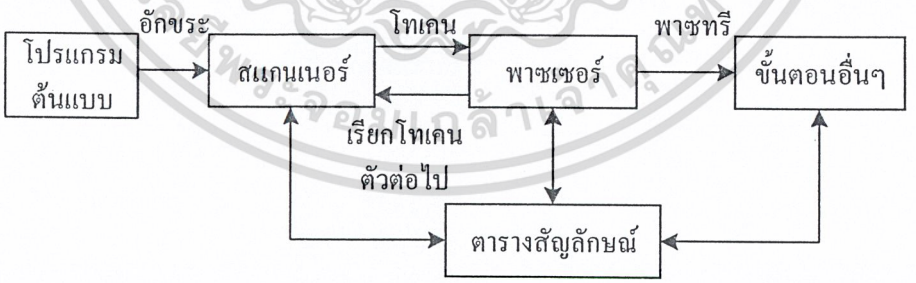
ภาษาที่ผลิตโดยไวยากรณ์กำกวมไม่ได้หมายความว่าภาษานั้นกำกวมด้วย เพราะภาษาหนึ่งๆ อาจจะผลิตโดยไวยากรณ์ได้มากกว่าหนึ่งไวยากรณ์

ตัวอย่างที่ 2.8 ไวยากรณ์ที่ (2.6) ไม่เป็นไวยากรณ์กำกวมและผลิตภาษาเดียวกันกับไวยากรณ์ที่ (2.3)

```
<expr>      →    <expr> + <term> | <term>
<term>      →    <term> * <factor> | <factor>          (2.6)
<factor>    →    ( <expr> ) | id
```

2.5.5 หน้าที่ของการตรวจสอบวากยสัมพันธ์

การตรวจสอบวากยสัมพันธ์รับโทเคนซึ่งก็คือ สัญลักษณ์เทอร์มินอลนั้นเอง จากการวิเคราะห์ภาษาต้นแบบแล้วนำมาตรวจสอบว่าโทเคนเหล่านั้น เรียงกันเป็นประโยคของภาษาที่ผลิตโดยไวยากรณ์ที่ใช้อธิบายภาษานั้นหรือไม่ ถ้าเป็นประโยคการตรวจสอบวากยสัมพันธ์จะสร้างพาซทรีของประโยคนั้น การสร้างพาซทรีอาจจะแสดงโดยชัดแจ้ง (Explicit) เช่น ในรูปลิงลิสต์ (Linked List) หรือโดยปริยาย (Implicit) คือ ไม่ได้เก็บอยู่ในลักษณะโครงสร้างข้อมูลแบบใดเลย แต่จะปรากฏในทางตรรกะ ขณะที่โปรแกรมส่วนที่เป็นการตรวจสอบวากยสัมพันธ์ทำงาน โปรแกรมส่วนที่เป็นการตรวจสอบวากยสัมพันธ์มักจะเรียกอีกอย่างหนึ่งว่า พาซเซอร์ (Parser) และโปรแกรมส่วนที่เป็นการวิเคราะห์ภาษาต้นแบบมักจะเรียกอีกอย่างหนึ่งว่า สแกนเนอร์ (Scanner) ซึ่งรูปที่ 2.18 แสดงความสัมพันธ์ดังกล่าวข้างต้น



รูปที่ 2.18 ตำแหน่งของพาซเซอร์ในโครงสร้างของตัวแปลภาษา

กรณีที่โทเคนไม่เรียงกันเป็นประโยค พาซเซอร์จะต้องสามารถตรวจจับความผิดพลาด และดำเนินการต่อให้ได้ เพื่อที่จะได้ตรวจจับความผิดพลาดครั้งต่อไปที่อาจจะเกิดขึ้นได้อีก ตัวแปลภาษาที่เป็นที่ยอมรับจะไม่หยุดการตรวจสอบเมื่อพบความผิดพลาดครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการตรวจสอบวากยสัมพันธ์ ส่วนที่อธิบายได้ด้วยไวยากรณ์คอนเทกซ์ฟรีอาจจะกระทำได้ในสองลักษณะ เช่นเดียวกับลักษณะการสร้างพาซทรี คือ ทำจากบนลงล่างเรียกว่า พาซเซอร์จากบนลงล่าง (Top-Down Parser) หรือทำจากล่างขึ้นบนเรียกว่า พาซเซอร์จากล่างขึ้นบน (Bottom-Up Parser)

2.5.6 นอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์

นอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์เป็นพาซเซอร์จากบนลงล่าง และจะทำการอนุพัทธ์ซ้ายที่สุด ลักษณะการตรวจสอบใช้วิธีดูจากโทเคนซึ่งได้จากสแกนเนอร์เป็นตัวทำนายเพื่อตัดสินใจเลือกโปรดักชันที่จะมาอนุพัทธ์ทันที ซึ่งสามารถพิจารณาจากไวยากรณ์ที่ (2.7) ต่อไปนี้

```
1) <S> →  a <A>
2)        | bd <B>
3) <A> →  a                                    (2.7)
4)        | c
5) <B> →  b
```

สมมติว่าต้องการตรวจสอบว่า ac เป็นประโยคของภาษาที่อธิบายด้วยไวยากรณ์ที่ (2.7) หรือไม่ สามารถอธิบายเป็นขั้นตอนง่ายๆ ได้ดังนี้

ขั้นตอนที่ 1 พาซทรีเริ่มต้นที่สัญลักษณ์เริ่มต้น

ขั้นตอนที่ 2 พาซเซอร์เรียกโทเคนตัวต่อไปจากสแกนเนอร์คือ A เนื่องจาก < S > เป็นสัญลักษณ์นอนเทอร์มินอลซ้ายที่สุด ดังนั้นพิจารณาโปรดักชันที่มี < S > อยู่ด้านซ้ายมีเพียง 2 โปรดักชัน คือ (1) และ (2) เหตุผลที่เราเลือกโปรดักชัน (1) เพราะสัญลักษณ์ตัวแรกของสตริงขวามือของโปรดักชัน (1) ตรงกับโทเคน a ทำอนุพัทธ์ทันทีได้กึ่งประโยค a < A >

ขั้นตอนที่ 3 นอนเทอร์มินอลซ้ายที่สุดของกึ่งประโยค a < A > คือ < A > พาซเซอร์เรียกโทเคนตัวต่อไปจากสแกนเนอร์คือ c ในทำนองเดียวกันกับขั้นตอนที่ 2 ต้องเลือกโปรดักชัน (3) มาทำอนุพัทธ์ทันทีได้ประโยค ac แสดงว่า ac เป็นประโยค

ขั้นตอนข้างต้นสามารถนำโครงสร้างข้อมูลแบบการเรียงทับซ้อน (Stack) มาใช้ได้เป็นอย่างดีในที่นี้เขียนสัญลักษณ์ในการเรียงทับซ้อนอยู่บนบรรทัดเดียวกัน โดยสัญลักษณ์ที่อยู่ทางขวาสุดของการเรียงทับซ้อนเป็นสัญลักษณ์บนสุดของการเรียงทับซ้อน (Top of Stack Symbol) สัญลักษณ์ซ้ายสุดของโทเคนที่เหลือเป็นโทเคนตัวต่อไปที่จะนำมาพิจารณา ส่วนเครื่องหมาย $ ใช้แทนซ้ายสุดของการเรียงทับซ้อน (Bottom of Stack) และแสดงการสิ้นสุดของโทเคน (End of Token)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะสังเกตเห็นได้ว่าสัญลักษณ์บนสุดของการเรียงทับซ้อน อาจจะเป็นสัญลักษณ์นอนเทอร์มินอลหรือสัญลักษณ์เทอร์มินอลอย่างใดอย่างหนึ่งเท่านั้น กรณีที่เป็นสัญลักษณ์นอนเทอร์มินอลจะต้องเลือกมาทำอนุพัทธ์ทันที โดยการป๊อป (Pop) สัญลักษณ์นอนเทอร์มินอลบนสุดของการเรียงทับซ้อนออกจากการเรียงทับซ้อน แล้วพุช (Push) สัญลักษณ์ทั้งหมดของสตริงขวามือของโปรดักชันที่เลือกมาทำอนุพัทธ์ทันทีลงในการเรียงทับซ้อน โดยพุชสัญลักษณ์ทางขวาสุดของสตริงขวามือลงไปก่อน และให้สัญลักษณ์ทางซ้ายสุดของสตริงขวามือเป็นสัญลักษณ์บนสุดของการเรียงทับซ้อน ส่วนกรณีที่สัญลักษณ์บนสุดของการเรียงทับซ้อนเป็นสัญลักษณ์เทอร์มินอล ให้เทียบกับโทเคนที่นำมาพิจารณา ถ้าเป็นกรณีเดียวกันให้ป๊อปสัญลักษณ์เทอร์มินอลบนสุดของการเรียงทับซ้อนทิ้ง แล้วเลือกโทเคนตัวต่อไปจากสแกนเนอร์ พิจารณาเช่นนี้ไปเรื่อยๆ จนกว่าจะยอมรับว่าโทเคนทั้งหมดเรียงกันเป็นประโยค หรือตรวจพบความผิดพลาด กรณียอมรับจะเกิดขึ้นเมื่อบนการเรียงทับซ้อนไม่มีสัญลักษณ์ใดๆ เลยคือมีเฉพาะเครื่องหมาย $ และไม่มีโทเคนเหลืออยู่อีกเลยคือ มีเฉพาะเครื่องหมาย $ เช่นกัน ส่วนในกรณีตรวจจับพบความผิดพลาดจะเกิดขึ้นในสามกรณี คือ ไม่สามารถเลือกโปรดักชันใดมาทำอนุพัทธ์ทันที สัญลักษณ์เทอร์มินอลบนสุดของการเรียงทับซ้อนไม่ตรงกับโทเคนที่นำมาพิจารณา และการเรียงทับซ้อนว่าเปล่าแต่ยังมีโทเคนเหลืออยู่

การเลือกโปรดักชันที่จะมาทำอนุพัทธ์ทันที หรือการตรวจพบความผิดพลาดในกรณีที่ไม่สามารถหาโปรดักชันที่จะมาทำอนุพัทธ์ทันที ทำได้โดยสร้างตารางพาร์ซิง (Parsing Pable) ไวยากรณ์ที่ (2.7) สามารถสร้างตารางพาร์ซิงได้ดังรูปที่ 2.19

	a	b	c	d	$
<S>	<S> → a<A>	<S> → bd<B>			
<A>	<A> → a		<A> → c		
<B>		<B> → b			

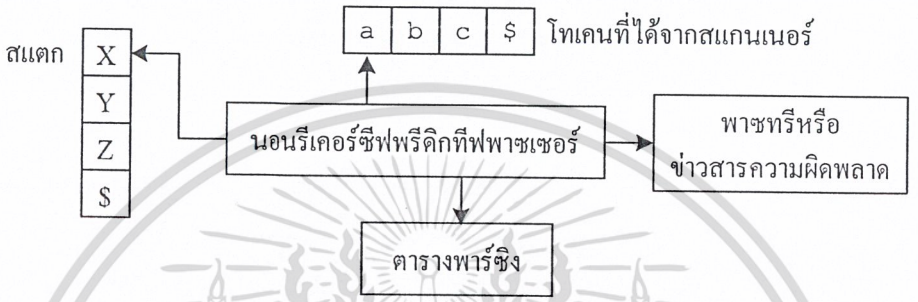
รูปที่ 2.19 การกระจายคำของไวยากรณ์ที่ (2.7)

ตารางพาร์ซิงเป็นตารางสองมิติ มีจำนวนแถวเท่ากับจำนวนสัญลักษณ์นอนเทอร์มินอลทั้งหมดของไวยากรณ์ที่นำมาสร้างตาราง และมีจำนวนสดมภ์เท่ากับจำนวนสัญลักษณ์เทอร์มินอลทั้งหมดของไวยากรณ์ที่นำมาสร้างตารางบวกหนึ่ง (รวมเครื่องหมาย $) แต่ละแถวจะมีสัญลักษณ์นอนเทอร์มินอลกำกับ และแต่ละสดมภ์จะมีโทเคน และเครื่องหมายแสดงการสิ้นสุดของโทเคนกำกับ สัญลักษณ์นอนเทอร์มินอลบนสุดของการเรียงทับซ้อนกับโทเคนที่นำมาพิจารณาใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนุพัทธ์ทันทีหรือตรวจจับความผิดพลาดนั้นเอง ช่องว่างที่มีโปรดักชันอยู่เป็นโปรดักชันที่เลือกมาทำอนุพัทธ์ทันที

สรุปแล้วนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์มีองค์ประกอบต่างๆ ดังแสดงในรูปที่ 2.20 และเขียนขั้นตอนในการตรวจสอบด้วยรหัสจำลองได้ดังรูปที่ 2.21



รูปที่ 2.20 องค์ประกอบต่างๆ ของนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์

```
PUSH (S); SCANNER (a);
repeat
  TOP(X);
  if NONTERMINAL(X) then
    if PRODUCTION(X,a,i) then
     begin
      POP(); PUSH-RHS(i);
     end
    else
     ERROR();
  else
    if X=a then
     begin
      POP(); SCANNER(a);
     end
    else
     ERROR();
until (X='$');
if (a='$') then ACCEPT()
else ERROR();
```

รูปที่ 2.21 รหัสจำลองแสดงขั้นตอนการตรวจสอบของนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.7 แพนิกโมด

วิธีดำเนินการต่อเมื่อพบความผิดพลาดในขั้นตอนการตรวจสอบวากยสัมพันธ์มีหลายวิธี ในที่นี้จะกล่าวถึงเฉพาะวิธีแพนิกโมด (Panic Mode) พาซเซอร์ที่ใช้วิธีนี้ เมื่อตรวจพบความผิดพลาด พาซเซอร์จะละทิ้งโทเคนจากโปรแกรมต้นแบบ และสัญลักษณ์บนการเรียงทับซ้อน จนกว่า โทเคนโปรแกรมต้นแบบ และสัญลักษณ์บนการเรียงทับซ้อนจะมีความสัมพันธ์กันพอที่จะทำให้พาซเซอร์ทำงานต่อไปได้ ปกติแล้วจะใช้ชุดของสัญลักษณ์ซินโครไนซ์ (Synchronize Symbol) เป็นตัวกำหนด สัญลักษณ์ซินโครไนซ์ก็คือ สัญลักษณ์เทอร์มินอลนั่นเอง ปกติมักจะใช้สัญลักษณ์สิ้นสุดประโยค เช่น ; หรือ end เป็นสัญลักษณ์ซินโครไนซ์ วิธีดำเนินการต่อทำโดยการอ่านโทเคนจากโปรแกรมต้นแบบทิ้ง จนกว่าจะพบโทเคนที่ตรงกับสัญลักษณ์ซินโครไนซ์จากนั้นจึงป๊อปสัญลักษณ์บนการเรียงทับซ้อนของพาซเซอร์ทิ้งจนกว่าจะพบสัญลักษณ์ที่จะทำให้พาซเซอร์ทำงานต่อไปได้ เช่น คำสั่งกำหนดค่า a : = b + c d [6] * e; เขียนตัวดำเนินการเลขคณิตตกไประหว่าง c กับ d วิธีแพนิกโมดจะละทิ้ง ‘d ’ ‘[’ ‘6’ และ ‘]’ จนพบ ‘*’ เมื่อ ‘*’ เป็นสัญลักษณ์ซินโครไนซ์จากนั้นจึงไปตรวจสอบสัญลักษณ์บนสุดของการเรียงทับซ้อน และป๊อปทิ้งจนกว่าจะเป็นสัญลักษณ์ ‘*’ หรือสัญลักษณ์นอนเทอร์มินอลที่สามารถแทนที่ได้โดยการคำนวณนิพจน์ย่อยเป็นต้น ข้อดีของวิธีแพนิกโมด คือ เป็นวิธีที่ง่าย และไม่เกิดปัญหาวนไม่สิ้นสุด เพราะไม่มีการแทรกสัญลักษณ์ใดๆ มีแต่การละทิ้งเท่านั้น ส่วนข้อเสียก็คือ อาจจะไม่สามารถตรวจพบความผิดพลาดจากสัญลักษณ์ที่ทิ้งไปแล้ว

2.6 การตรวจสอบความหมาย (Semantic Analyzer)

เป็นขั้นตอนในการตรวจสอบความหมายของภาษา และสร้างรหัสของชุดของคำสั่งที่ผ่านการตรวจสอบความถูกต้องของโครงสร้างแล้วให้อยู่ในรูปแบบ หรือขั้นตอนการทำงานที่ใกล้เคียงกับภาษาเครื่อง รหัสดังกล่าวเรียกว่ารหัสระหว่างกลาง ตัวอย่างรหัสระหว่างกลาง เช่น สัญกรณ์โพริช (Polish Notation) สัญกรณ์ n สิ่งที่เป็นอันดับ (N-Tuple Notation) ซินแทกซ์ทรี (Syntex Tree)

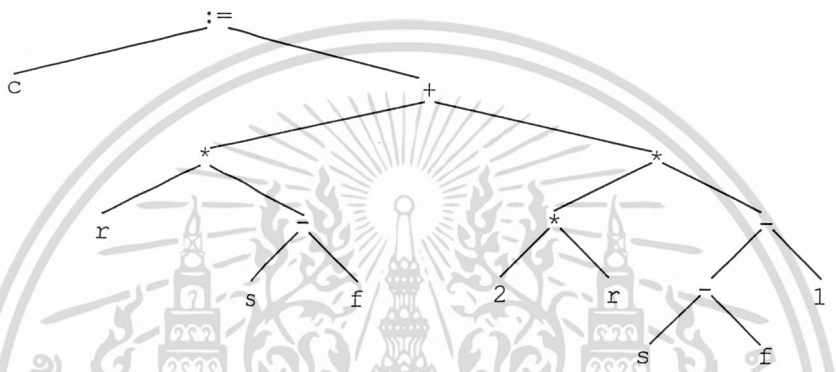
ตัวอย่างที่ 2.9 คำสั่งกำหนดค่า (Assignment Statement) ในภาษาปาสคาลต่อไปนี้

```
c : = r * (s - f) + 2 * r * (s - f - 1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนด้วยซินแทกซ์ทรีได้ดังรูปที่ 2.22 และแทนด้วยสัญกรณ์ 3 สิ่งที่เป็นอันดับได้ดังรูปที่ 2.23

การใช้รหัสระหว่างกลางมีข้อดี คือ ทำให้ง่ายต่อการปรับปรุงให้มีประสิทธิภาพในการทำงานมากขึ้น และง่ายต่อการแปลงเป็นภาษาเครื่องต่างๆ ซึ่งไม่เหมือนกัน อย่างไรก็ตามขั้นตอนนี้อาจจะรวมกับขั้นตอนการเปลี่ยนรหัส แล้วสร้างภาษาแอสเซมบลีหรือภาษาเครื่องเลยก็ได้



รูปที่ 2.22 ซินแทกทรีของตัวอย่างที่ 2.9

(1)	-	s	f
(2)	*	r	(1)
(3)	*	2	r
(4)	-	s	f
(5)	-	(4)	1
(6)	*	(3)	(5)
(7)	+	(2)	(6)
(8)	:=	c	(7)

รูปที่ 2.23 สัญกรณ์ 3 สิ่งที่เป็นอันดับของตัวอย่างที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การปรับปรุงรหัส (Code Optimizer)

เป็นขั้นตอนที่ทำการแก้ไขปรับปรุงรหัสระหว่างกลาง จากขั้นตอนการตรวจสอบความหมาย การเปลี่ยนรหัสให้มีประสิทธิภาพในการทำงานมากขึ้นซึ่งตัวแปลภาษาบางตัวอาจจะไม่มีขั้นตอนนี้ก็ได้ หรืออาจมีขั้นนี้มากกว่าหนึ่งขั้นตอนก็ได้ การแก้ไขปรับปรุงรหัสระหว่างกลางให้มีประสิทธิภาพในการทำงานมากขึ้นกระทำได้ดังนี้

1) ขจัดนิพจน์คณิตศาสตร์ย่อยที่เกิดซ้ำกัน เช่น คำสั่งในภาษาต้นแบบต่อไปนี้

```
x = y ;
y = a * b * (b * a + x) ;
```

ถ้าแทนด้วยรหัสระหว่างกลางจะได้ดังนี้

(1)	:=	x	y
(2)	*	a	b
(3)	*	b	a
(4)	+	(3)	x
(5)	*	(2)	(4)
(6)	:=	y	(5)

รูปที่ 2.24 รหัสระหว่างกลางที่ใช้แทนคำสั่ง x = y ; y = a * b * (b * a + x) ;

เราสามารถปรับปรุงรหัสระหว่างกลางได้ เพราะนิพจน์ย่อย a*b มีค่าเท่ากับนิพจน์ย่อย b*a จึงไม่จำเป็นคำนวณนิพจน์ย่อย b*a ซ้ำอีก รหัสระหว่างกลางบรรทัดที่ (3) จึงตัดทิ้งได้ แล้วใช้ผลที่ได้จากบรรทัดที่ (2) แทน ดังนั้นหลังจากการปรับปรุงแล้วจะได้รหัสระหว่างกลางที่ปรับปรุงแล้วดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(1)	:=	x	y
(2)	*	a	b
(3)	+	(2)	x
(4)	*	(2)	(3)
(5)	:=	y	(4)

รูปที่ 2.25 รหัสระหว่างกลางที่ปรับปรุงแล้ว

2.7.1 คำนวณนิพจน์เลขคณิตในช่วงการแปล

เช่น คำสั่งในภาษาต้นแบบดังต่อไปนี้

```
x = 2 * 9 / 3 * y
```

ถ้ารหัสระหว่างกลางที่ใช้แทนคำสั่งข้างต้นนี้ คือ

(1)	*	2	9
(2)	/	(1)	3
(3)	*	(2)	y
(4)	:=	x	(3)

รูปที่ 2.26 รหัสระหว่างกลางที่ใช้แทนคำสั่ง x = 2 * 9 / 3 * y

เราสามารถปรับปรุงรหัสระหว่างกลางนี้ได้ ทั้งนี้เนื่องจากนิพจน์ย่อย 2*9/3 ทุกเทอมเป็นค่าคงที่ จึงสามารถคำนวณในช่วงการแปลได้ ผลลัพธ์จากนิพจน์ข้างต้นมีค่าเท่ากับ 6 ดังนั้นรหัสระหว่างกลางที่ปรับปรุงแล้วจะเป็นดังนี้

(1)	*	6	y
(2)	:=	x	(1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.27 รหัสระหว่างกลางที่ปรับปรุงแล้ว

ถ้าต้องการปรับปรุงชุดคำสั่งภาษาแอสเซมบลี ต่อไปนี้

```
LOAD        B
MUL         C
STORE       T1
LOAD        T1
ADD         D
STORE       T2
LOAD        A
ADD         T2
STORE       T3
LOAD        T3
STORE       X
```

เราสามารถปรับปรุงชุดคำสั่งเหล่านี้ได้ เพราะชุดคำสั่ง

```
STORE       T1
LOAD        T1
```

สามารถตัดทิ้งได้ เนื่องจากในกรณีนี้ T1 ไม่มีความจำเป็นต้องใช้ และเมื่อพิจารณาชุดคำสั่งต่อไปนี้

```
STORE       T2
LOAD        A
ADD         T2
```

เราสามารถสลับคำสั่งได้ เพราะ T2+A มีค่าเท่ากับ A+T2 จึงสามารถเขียนชุดคำสั่งนี้ใหม่ได้ดังนี้

```
STORE       T2
LOAD        T2
ADD         A
```

เมื่อสลับชุดคำสั่งแล้ว ชุดคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STORE       T2
LOAD        T2
```

สามารถตัดทิ้งได้ ดังที่กล่าวมาแล้วข้างต้น ดังนั้นหลังจากที่ขจัดคำสั่งที่ไม่จำเป็นออกไปแล้ว จะได้ภาษาแอสเซมบลีที่ปรับปรุงแล้วดังนี้

```
LOAD        B
MUL         C
ADD         D
ADD         A
STORE       X
```

การแก้ไขปรับปรุงภาษาเครื่องเพื่อให้มีประสิทธิภาพในการทำงานมากขึ้น ทำได้ดังนี้

1) การเพิ่มค่าในหน่วยความจำที่ต้องการอีก 1 แทนที่จะเก็บเป็นค่าคงที่ 1 เพื่อนำไปบวกกับค่าในหน่วยความจำที่ต้องการด้วยคำสั่งบวก ก็เปลี่ยนไปใช้คำสั่งเพิ่มค่าที่ละ 1 แทน เป็นต้น

2) เลือกใช้คำสั่งที่ทำงานเร็วกว่า เช่น แทนที่จะคูณด้วย 2 ก็ใช้คำสั่งในการเลื่อนบิตมาทางซ้ายแทน เป็นต้น

ตัวอย่างการแก้ไขปรับปรุงในลักษณะอื่นๆ มีดังนี้

1) คำนวณนิพจน์ตรรกศาสตร์บางคำ เช่น คำสั่ง if x or y or z then write ('o.k.'); อาจจะไม่จำเป็นคำนวณค่า x, y และ z ทุกค่า เพราะกรณีที่ x เป็นจริง แสดงว่านิพจน์ตรรกศาสตร์นี้เป็นจริง ไม่ต้องคำนวณค่า y และ z อีกแล้ว

2) เคลื่อนย้ายคำสั่งที่ไม่มีผลในการทำงานซ้ำออกไป เช่น คำสั่งภาษาต้นแบบต่อไปนี้

```
for I :=1 to 10 do
  begin
for j:=1 to 20 do
  begin
       a:=10;
       b:=y[i+2];
       c:=c+10;
  end;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ว่าจะมีการคำนวณกี่รอบก็ตามค่าของ a จะยังคงเท่าเดิมเสมอ จึงควรย้ายคำสั่ง a : = 10; ออกจากคำสั่ง for นอกจากนี้คำสั่ง b : = y[i+2]; ไม่ว่า j จะมีค่าเท่าใดก็จะไม่มีผลกับคำสั่งนี้เลย จึงควรย้ายคำสั่งนี้ออกจากคำสั่ง for วงใน ไปไว้ในคำสั่ง for วงนอก

การปรับปรุงรหัสระหว่างกลางนี้เป็นการปรับปรุงที่ไม่ขึ้นอยู่กับเครื่องคอมพิวเตอร์ ส่วนการปรับปรุงภาษาแอสเซมบลี และภาษาเครื่องเป็นการปรับปรุงที่จะขึ้นอยู่กับเครื่องคอมพิวเตอร์

2.8 การสร้างรหัส (Code Generator)

กรณีที่ขั้นตอนการตรวจสอบความหมายสร้างรหัสระหว่างกลาง ขั้นตอนนี้จะมีการเปลี่ยนรหัสระหว่างกลาง ดังกล่าวให้เป็นภาษาแอสเซมบลีหรือภาษาเครื่อง ซึ่งจะขึ้นอยู่กับเครื่องที่ต้องการมีการกำหนดตำแหน่งของข้อมูลในหน่วยความจำ รวมถึงการกำหนดรีจิสเตอร์ที่จะใช้เป็นต้น ในขณะที่พาซเซอร์เลือกโปรดักชันมาทำอนุพัทธ์ทันที พาซเซอร์จะเข้าใจความหมายและผลิตภาษาเป้าหมายไปพร้อมๆ กัน การทำงานในลักษณะนี้เรียกว่า ซินแทกซ์โดยตรง (Syntax Directed) เพราะโปรดักชันของไวยากรณ์ในขั้นตอนซินแทกซ์อนาไลเซอร์จะถูกใช้ทั้งในการตรวจสอบวากยสัมพันธ์ของภาษา ความหมาย และตีความเพื่อผลิตภาษาเป้าหมายพร้อมๆ กัน การใช้สัญลักษณ์แอกชัน (Action Symbol) เพิ่มเข้าไปในโปรดักชัน และการเขียนแอกชันรูทีน (Action Routine) สำหรับสัญลักษณ์แอกชันแต่ละตัว

2.9 การจัดตาราง

หน้าที่ที่สำคัญอีกอย่างหนึ่งของตัวแปลภาษา คือ จะต้องบันทึกชื่อตัวแปร ชื่อโพรซีเยอร์ (Procedure Name) ชื่อฟังก์ชัน (Function Name) เลเบล (Label) หรือค่าคงที่ เป็นต้น ที่ปรากฏในโปรแกรมต้นแบบไว้ในตาราง พร้อมทั้งบันทึกรายละเอียดต่างๆ เช่น ชนิด (Type) เลขที่ (Address) ในกรณีเป็นโพซีเยอร์หรือฟังก์ชัน จะต้องบันทึกรายละเอียดเกี่ยวกับจำนวนพารามิเตอร์ (Parameter) ชนิดของพารามิเตอร์แต่ละตัว พารามิเตอร์แต่ละตัวมีการผ่านค่าแบบไหน เช่น โดยค่า (By Value) โดยที่อยู่ (By Reference) หรือโดยชื่อ (By Name) ขั้นตอนต่างๆ ที่กล่าวมาข้างต้นจะร่วมการใช้ตารางเหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 การตรวจจับและดำเนินการต่อเมื่อพบความผิดพลาด

ตัวแปลภาษาที่ถูกต้อง ถ้าโปรแกรมต้นแบบเขียนผิดหลักภาษาจะต้องบอกความผิดพลาดให้ผู้เขียนรู้ แต่ถ้าโปรแกรมต้นแบบเขียนถูกหลักภาษาจะต้องแปลเป็นโปรแกรมเป้าหมายที่ถูกต้อง ตัวแปลภาษาที่ดีจะต้องตรวจจับความผิดพลาดให้ได้มากที่สุด และใกล้เคียงกับความผิดพลาดที่เกิดขึ้นให้มากที่สุด นอกจากจะตรวจจับความผิดพลาดได้แล้วจะต้องสามารถดำเนินการต่อเพื่อที่จะให้สามารถตรวจจับความผิดพลาดที่อาจเกิดขึ้นอีกได้ โดยจะไม่หยุดการทำงานเมื่อพบความผิดพลาดครั้งแรก ตัวแปลภาษาบางตัวถึงกับเดาความผิดพลาดที่อาจจะเกิดขึ้น และจะพยายามแก้ไขความผิดพลาดให้ถูกต้อง ปัจจุบันยังไม่มีวิธีการใดที่จะสามารถแก้ไขความผิดพลาดให้ถูกต้องได้ จึงหันมาใช้วิธีการซ่อมแซมความผิดพลาดแทน ดังนั้นสามารถแบ่งลักษณะของตัวแปลภาษาที่เป็นที่ยอมรับและไม่เป็นที่ยอมรับได้ดังนี้

กรณีที่ไม่เป็นที่ยอมรับ

1) ความผิดพลาดที่เกิดจากการเขียนตัวแปลภาษาผิด เช่น ตัวแปลภาษาหยุดการทำงานโดยไม่ทราบสาเหตุ ตัวแปลภาษาทำงานวนไม่สิ้นสุดหรือตัวแปลภาษาผลิตภาษาเป้าหมายผิด เป็นต้น

2) ตัวแปลภาษาทำงานถูกต้องแต่จะหยุดทำงานเมื่อพบความผิดพลาดครั้งแรก

กรณีที่เป็นที่ยอมรับ

1) ตัวแปลภาษาจะบอกข่าวสารความผิดพลาด เมื่อตรวจพบความผิดพลาด และพยายามแก้ไขต่อไป เพื่อที่จะสามารถตรวจจับความผิดพลาดซึ่งอาจเกิดขึ้นในภายหลังอีก

2) ตัวแปลภาษาจะบอกข่าวสารความผิดพลาด เมื่อตรวจพบความผิดพลาด และพยายามซ่อมแซม โดยบอกให้ผู้ใช้รู้ถึงสิ่งที่ซ่อมแซมนั้น และผลิตภาษาเป้าหมายที่ถูกต้องถ้าสิ่งที่ซ่อมแซมนั้นเป็นที่ยอมรับของผู้ใช้

3) ตัวแปลภาษาจะบอกข่าวสารความผิดพลาด เมื่อตรวจพบความผิดพลาด และพยายามแก้ไขให้ถูกต้องตรงตามความต้องการของผู้ใช้ ซึ่งปัจจุบันยังเป็นไปไม่ได้

ตัวแปลภาษาที่ดีควรจะแสดงข่าวสารความผิดพลาดด้วยข้อความที่ปรากฏในโปรแกรมต้นแบบและไม่ควรใช้รหัสแต่ควรบอกเป็นข้อความ เช่น ไม่ควรใช้รหัส E001 แต่ควรใช้ข้อความว่า “missing right parenthesis” เป็นต้น นอกจากนี้ควรจะบอกให้ตรงเป้าหมายให้มากที่สุด เช่น กรณีที่ตัวแปร TOTAL ไม่ได้ประกาศไว้ในโพซีเยอร์ CALC แทนที่จะบอกว่า “missing declaration” แต่ควรจะบอกว่า “error: line 10 column 15 TOTAL not declared in Procedure CALC”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 โครงสร้าง Intel Hex File

อินเทลเฮกซ์ฟอร์แมต (Intel Hex Format) มีรูปแบบคล้ายกับแบบแอสกี้เฮกซ์ฟอร์แมตแต่ในรูปแบบของอินเทลนี้จะมีการอ้างแอดเดรส และตรวจสอบความผิดพลาดของข้อมูลนี้ด้วย ทั้งนี้เพื่อให้การบันทึกข้อมูลมีความถูกต้องน่าเชื่อถือมากขึ้น รูปแบบของข้อมูลแบบนี้มีลักษณะดังตารางที่ 2.2

ตารางที่ 2.2 รูปแบบของข้อมูลในระบบ Intel Hex Format

ชื่อสัญลักษณ์	จำนวนตัวอักษร	รายละเอียด
Record Mark	1	ในแต่ละบรรทัดของข้อมูลที่ต้องการบันทึกต้องเริ่มด้วยตัวโคลอน (:)
Record Iength	2	จำนวนข้อมูลที่ต้องการบันทึก
Address Field	2	แสดงแอดเดรสที่เก็บข้อมูลในไฟล์แรกปกติมีค่า 00
Record Type	2	ชนิดของข้อมูลที่ทำการบันทึกโดยถ้าเป็น ข้อมูล หมายถึง ข้อมูลของโปรแกรม หมายถึง จบแฟ้มข้อมูล หมายถึง แอดเดรสเพิ่มขึ้น หมายถึง แอดเดรสเริ่มต้น
Data Field	เปลี่ยนแปลง	จะขึ้นอยู่กับรหัสชนิดของข้อมูลที่ทำการบันทึก หมายถึง จำนวนข้อมูลที่ต้องการโปรแกรม หมายถึง ไม่ใช้งาน (ว่าง) หมายถึง เซกเมนต์สำหรับแอดเดรสที่มากกว่า 64 กิโลไบต์ ข้อมูลจะถูกเริ่มต้นเก็บที่เซกเมนต์ x 10h บวกกับค่าของพื้นที่แอดเดรสตัวอย่าง แอดเดรสเพิ่มเติมมีค่า F800h พื้นที่ของแอดเดรสเท่ากับ 1000h นั้นข้อมูลจะไปเริ่มต้นบันทึกที่ตำแหน่ง F900h
Checksum	2	การคำนวณค่า Checksum มีดังนี้ รวมค่าของข้อมูลทั้งหมดที่ทำการบันทึกจากนั้นทำทูคอมพลีเมนต์ ค่า Checksum คือ ไบต์ค่าของข้อมูลสุดท้ายหลังการทำทูคอมพลีเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 2.10 เพิ่มข้อมูลแบบอินเทลเฮกซ์ฟอร์แมต

: 03801300028223C3

: 20820000D28AE589440A54/FF589DZAF757F00757E00

: 257D00D28ED2AAE57FB403F1302AA9D

: 16822000020000C28E858D7D858B7E758D00758B00D

: 8E057F32C1

: 00000001FF

ตัวอย่างที่ 2.11 การคำนวณ Checksum สำหรับบรรทัดที่ 1

(1) รวมค่าของข้อมูลทั้ง 7 ค่าได้ 10 7h หรือ 100000111b

(2) ทำทูคอมพลีเมนต์ได้ 011111000 + 1 = 011111001b หรือ F9h

(3) ค่า Checksum เท่ากับ F9

ค่า Checksum นี้มีประโยชน์ในการตรวจสอบข้อมูลที่ทำการบันทึกว่าถูกต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

การเขียนโปรแกรมตัวแปลภาษาสำหรับไมโครคอนโทรลเลอร์ MCS-51 ในที่นี้จะเขียนโปรแกรมโดยใช้ภาษาซีในการออกแบบ และจะมีการเปรียบเทียบการทำงานกับโปรแกรมที่มีอยู่แล้ว คือ โปรแกรม SXA51 นำมาเปรียบเทียบผลการทดลอง ในการเขียนโปรแกรมจะใช้หลักการทำงานของตัวแปลภาษาในการสร้าง

3.2 โปรแกรมแอสเซมเบลอร์

โปรแกรมแอสเซมเบลอร์เป็นโปรแกรมที่ทำหน้าที่แปลภาษาแอสเซมบลีเป็นรหัสคำสั่ง (Opcode) เป็นโปรแกรมที่ทำหน้าที่แปลภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล MCS-51 โปรแกรมหนึ่ง

การเขียนโปรแกรมควบคุมการทำงาน จะเขียนซอร์สโปรแกรม (Source Program) ขึ้นก่อนด้วยโปรแกรมสร้างข้อความ เช่น ED หรือ Notepad ใน Sidekick แล้วเก็บไฟล์ที่มีนามสกุลเป็น ASM รูปแบบของโปรแกรมที่เขียนจะต้องเขียนตามหลักของภาษาแอสเซมบลี แล้วจึงนำโปรแกรมไปแปล เมื่อนำซอร์สโปรแกรมที่มีนามสกุลเป็น ASM มาทำการแปลด้วยโปรแกรมแปลภาษาแล้วจะได้ไฟล์ประเภท Hex File ที่มีชื่อเดียวกับซอร์สโปรแกรมแต่มีนามสกุล HEX เกิดขึ้น และถ้าเราใช้พารามิเตอร์จะได้ Listing File ที่มีนามสกุลเป็น LST หากไม่มีข้อผิดพลาดของการเขียนโปรแกรม จะได้ไฟล์ .HEX และ .LST เพิ่มขึ้นมา

1) Intel Hex File เป็นไฟล์ที่สร้างขึ้นจากตัวแปลโปรแกรมภาษาแอสเซมบลีต่างๆ รวมถึง SXA51 ด้วย โดยรูปแบบของไฟล์นี้จะเป็นไฟล์ประเภท ASCII และ Text ซึ่งสามารถพิมพ์ออกมาดูทางหน้าจอหรือพิมพ์ออกเครื่องพิมพ์ได้ โครงสร้างการเก็บข้อมูลในไฟล์นี้จะแบ่งออกเป็นเรคคอร์ด

2) Listing File เป็นไฟล์ที่สร้างขึ้นมาจากการแปลไฟล์ภาษาแอสเซมบลี ด้วยโปรแกรมแอสเซมเบลอร์ Listing File เป็นไฟล์ประเภท Text เราสามารถพิมพ์ออกหน้าจอภาพหรือพิมพ์ออกเครื่องพิมพ์ได้เช่นเดียวกับ Hex File ข้อมูลในไฟล์นี้จะประกอบด้วยตำแหน่งของหน่วยความจำ รหัสคำสั่งและคำสั่งที่เขียนอยู่ในซอร์สไฟล์รวมอยู่ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ไวยากรณ์ของตัวแปลภาษา

ไวยากรณ์ของตัวแปลภาษา ประกอบไปด้วยกฎต่างๆ ดังนี้

```
Program          :    linelist
Linelist         :    linelist    line
                      | line
line             :    undeFsym ' : 'linevest
                      | linerest
linerest         :    divective '\n'
                      | instr '\n'
                      | '\n'
divetives        :    D_ORG defexpr
                      | D_BYTE     blist
                      | D_WORD     wlist
                      | D_SKIP     defevexpr
                      | D_EQU      Expr
                      | D_FLAG     SYMBOL       Flag
                      | D_END
defexpr          :    expr
flag             :    flagv BITPOS
flag             :    SYMBOL
                      | VALUE
Undefsym         :    SYMBOL
Blist            :    blist ',' data 8
                      | blist ',' STRING
                      | data P
                      | STRING
Wlist            :    wlist ',' data 16
                      | data 16
expr             :    '*'
                      | '('expr')'
                      | '_'expr%prec '*'
                      | expr ' | '   expr
                      | expr ' & '   expr
                      | expr ' * '   expr
                      | expr ' / '   expr
                      | expr ' % '   expr
                      | expr ' _- ' expr
                      | expr ' + '   expr
                      | SYMBOL
                      | VALUE
Instr            :    NOP
                      | ACALL      addr 11
                      | AJMP       addr 11
```

รูปที่ 3.1 ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
| ADD         two-op1
| ADDC        two-op1
| SUBB        two-op1
| XRL         two-op1
| XRL         two-op2
| ANL         two-op1
| ANL         two-op2
| ANL         two-op3
| ORL         two-op1
| ORL         two-op2
| ORL         two-op3
| XCH         two-op1
| INC         single-op1
| INC         DPTR
| DEC         single-op1
| DA          A
| DIV         AB
| JMP         '@'A'+'DPTR
| JMP         '@'DPTR'+'A
| MUL         AB
| RET
| RETI
| RL          A
| RLC         A
| RR          A
| RRC         A
| SWAP        A
| XCHD        two-op1
| CLR         single-op2
| CPL         single-op2
| SETB        single-op2
| PUST        data 8
| POP         data 8
| LJMP        addr 16
| LCALL       addr 16
| JC          relative
| JZ          relative
| JNC         relative
| JNZ         relative
| SJMP        relative
| CJNE        three-op1
| JB          two-op1
| JNB         two-op4
| JBC         two-op4
| DJNZ        two-op5
| MOV         two-op1
```

รูปที่ 3.1 (ต่อ) ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        | MOV         two-op2
                        | MOV         two-op6
                        | MOVC        A',''@'A'+'DPTR
                        | MOVC        A',''@'DPTR'+'
                        | MOVC        A',''@'A'+'PC
                        | MOVC        A','@'PC'+'A
                        | MOVX        A',''@'regi
                        | MOVX        A',''@'DPTR
                        | MOVX        '@'regi','A
                        | MOVX        '@'DPTR','A
two-op1         :       A','reg
                        | A ','data 8
                        | A ','@'regi
                        | A ','#'data 8
two-op2         :       data 8 ','A
                        | data 8 ','#'data 8
two-op3         :       C','bit
                        | C ','/'bit
                        | C ','!'bit
two-op 4        :       bit ',' rel
two-op 5        :       reg ','rel 2
                        | data 8 ',' rel
two-op 6        :       reg ',' A
                        | reg ',' data 8
                        | reg ','#'data 8
                        | data 8 ',' reg
                        | data 8 ','data 8
                        | data 8 ','@'regi
                        | '@'regi ','A
                        | '@'regi ','data 8
                        | '@'regi','#'data 8
                        | C ',' bit
                        | DPTR','#'data16
                        | data 8 ','C
                        | data 8 BITPOS ',' C
single-op 1     :       A
                        | reg
                        | data 8
                        | '@' regi
single-op 2     :       A
                        | C
                        | bit
three-op1       :       A','data 8','rel
                        | A','#'data 8','rel
                        | reg ','#data 8','rel
                        | '@'regi','#'data 8','rel
rel             :       expr
rel2            :       expr
```

รูปที่ 3.1 (ต่อ) ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

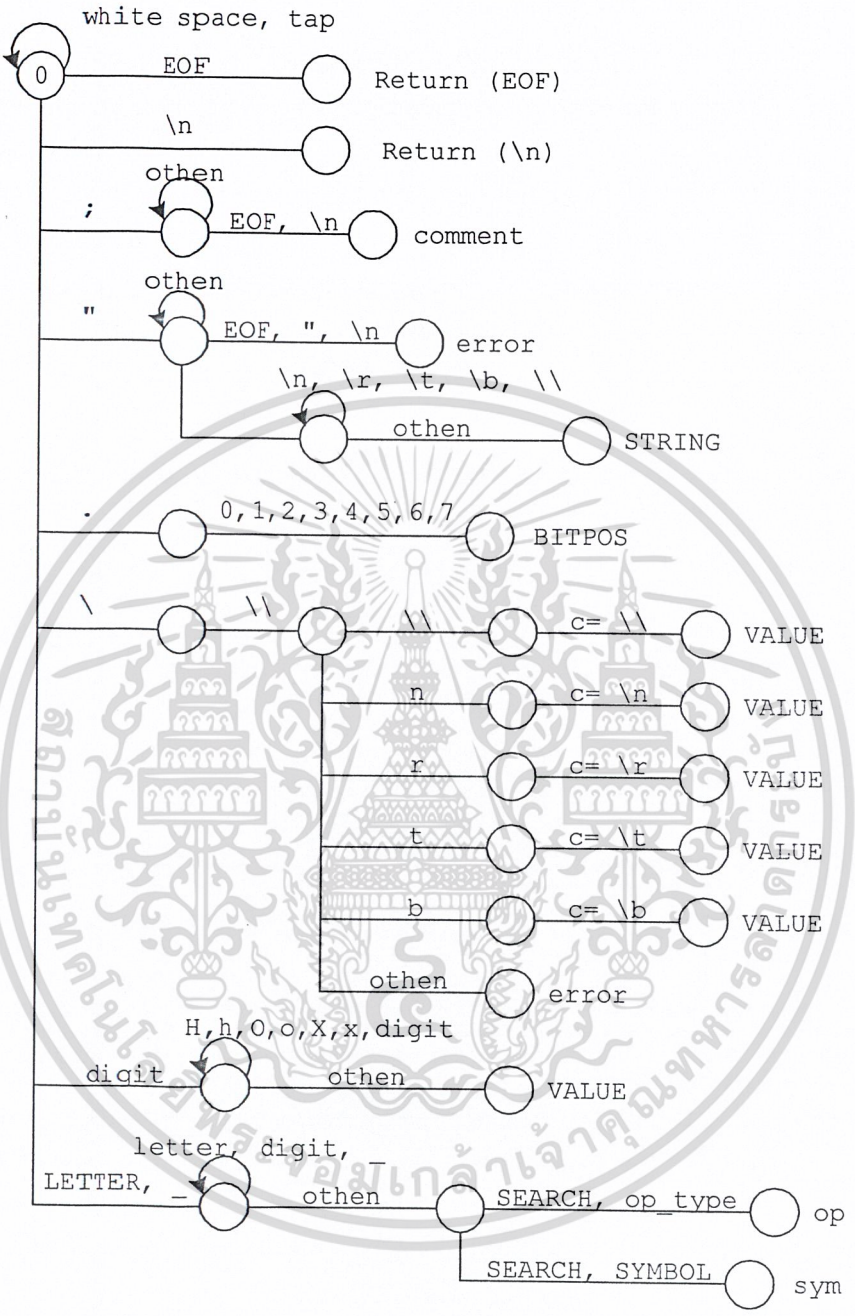
```
bit             :   bitv BITPOS
                |   bitv
bitv            :   SYMBOL
                |   VALUE
reg             :   R0
                |   R1
                |   R2
                |   R3
                |   R4
                |   R5
                |   R6
                |   R7
regi            :   R0
                |   R1
                |   R2
                |   R3
                |   R4
                |   R5
                |   R6
                |   R7
data 8          :   expr
data 16         :   expr
addr 11         :   expr
addr 16         :   expr
relative        :   expr
```

รูปที่ 3.1 (ต่อ) ไวยากรณ์คอนเทกซ์ฟรีของตัวแปลภาษา

3.3.1 ขั้นตอนการวิเคราะห์ภาษาต้นแบบของตัวแปลภาษา

การสร้างแผนภาพทรานซิชันเพื่อใช้แยกโทเคนจากโปรแกรมต้นแบบภาษาแอสเซมบลีได้ดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แผนภาพทรานซิชันเพื่อแยกโทเคนจากโปรแกรมต้นแบบภาษาแอสเซมบลี

3.3.2 ขั้นตอนการตรวจสอบความหมาย และการผลิตรหัสของตัวแปลภาษา

ไวยากรณ์ล่ามของตัวแปลภาษามีสัญลักษณ์แอกชันอยู่ในรูป #ddd เมื่อ ddd เป็นเลขจำนวนเต็ม และมีโปรดักชันต่างๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program     :   linelist
Linelist    :   linelist    line
                | line
line        :   undeFsym ' : 'linevest
                | linerest
linerest    :   divective '\n'
                | instr '\n'
                | '\n'
divetives   :   D_ORG [59] defexpr
                | D_BYTE [60]      blist
                | D_WORD [61]      wlist
                | D_SKIP [62]      defevexpr
                | D_EQU  [63]      Expr
                | D_FLAG [64]      SYMBOL      Flag
                | D_END  [65]
defexpr     :   expr
flag        :   flagv BITPOS
flagv       :   SYMBOL
                | VALUE
Undefsym    :   SYMBOL
Blist       :   blist ',' data 8
                | blist ',' STRING
                | data8
                | STRING
Wlist       :   wlist ',' data 16
                | data 16
expr        :   '*'
                | '('expr')'
                | '_'expr%prec '*'
                | expr ' | '  expr
                | expr ' & '  expr
                | expr ' * '  expr
                | expr ' / '  expr
                | expr ' % '  expr
                | expr ' _ - ' expr
                | expr ' + '  expr
                | SYMBOL [71]
                | VALUE  [72]
Instr       :   NOP [11]
                | ACALL [12] addr 11 [67]
                | AJMP  [13] addr 11 [67]
                | ADD   [14] two-op1
                | ADDC  [15] two-op1
                | SUBB  [16] two-op1
                | XRL   [17] two-op1
                | XRL   [17] two-op2
Wlist       :   wlist ',' data 16
                | data 16
```

รูปที่ 3.3 ไวยากรณ์ลำของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
expr        :     '*'
                | '('expr')'
                | '_'expr%prec '*'
                | expr ' | '    expr
                | expr ' & '    expr
                | expr ' * '    expr
                | expr ' / '    expr
                | expr ' % '    expr
                | expr ' _- '   expr
                | expr ' + '    expr
                | SYMBOL [71]
                | VALUE  [72]
Instr       :     NOP [11]
                | ACALL [12] addr 11 [67]
                | AJMP  [13] addr 11 [67]
                | ADD   [14] two-op1
                | ADDC  [15] two-op1
                | SUBB  [16] two-op1
                | XRL   [17] two-op1
                | XRL   [17] two-op2
                | ANL   [18] two-op1
                | ANL   [18] two-op2
                | ANL   [18] two-op3
                | ORL   [19] two-op1
                | ORL   [19] two-op2
                | ORL   [19] two-op3
                | XCH   [20] two-op1
                | INC   [21] single-op1
                | INC   [21] DPTR [91]
                | DEC   [22] single-op1
                | DA    [23] A [105]
                | DIV   [24] AB [106]
                | JMP   [25] '@' [107] A'+' [190] DPTR [269]
                | JMP   [25] '@'DPTR'+'A
                | MUL   [26] AB [108]
                | RET   [27]
                | RETI  [28]
                | RL    [29]   A [109]
                | RLC   [30]   A [110]
                | RR    [31]   A [111]
                | RRC   [32]   A [112]
                | SWAP  [33]   A [113]
                | XCHD  [34]       two-op1
                | CLR   [35]       single-op2
                | CPL   [36]       single-op2
                | SETB  [37]       single-op2
```

รูปที่ 3.3 (ต่อ) ไวยากรณ์ล่ามของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    | PUST  [38]        data 8
                    | POP   [39]        data 8
                    | LJMP  [40]        addr 16
                    | LCALL [41]        addr 16
                    | JC    [42]        relative
                    | JNC   [43]        relative
                    | JNZ   [44]        relative
                    | JZ    [45]        relative
                    | SJMP  [46]        relative
                    | CJNE  [47]        three-op1
                    | JB    [48]        two-op1
                    | JNB   [49]        two-op4
                    | JBC   [50]        two-op4
                    | DJNZ  [51]        two-op5
                    | MOV   [52]        two-op1
                    | MOV   [52]        two-op2
                    | MOV   [52]        two-op6
                    | MOVC[53]A[154]',’[205]’@’[252]A’+’[282]
                      DPTR[301]
                    | MOVC[53] A[154]',''@’[205]DPTR[301]’+’A
                      [282]
                    | MOVC [53] A[154]',’[205]’@’[252]A’+’[282]
                      PC[302]
                    | MOVC   [53] A[154]',’[205]@’[252]PC
                      [302]’+’A[282]
                    | MOVX   [54]   A[155]‘,’[206]’@’[253]regi
                    | MOVX   [54]   A[155]',’[206]’@’[253]DPTR
                      [286]
                    | MOVX   [54]   ‘@’[156]regi’,’[254]A[287]
                    | MOVX   [54]   ‘@’[156]DPTR[208]’,’[255]A
                      [288]
two-op1     :       A  [75] ’,’ [178] reg
                    | A [75] ’,’[178] data8
                    | A [75] ‘,’[178] ’@’ [223] regi
                    | A [75] ‘,’[178] ’#’ [224] data8
two-op2     :       data 8 ‘,’[80] A [225]
                    | data 8 ‘,’[80] ’#’ [226] data 8
two-op3     :       C [85] ’,’ [180] bit
                    | C  ‘,’’/’bit
                    | C ‘,’’!’bit
two-op 4    :       bit ‘,’[140] rel
two_op5     :       reg ',’[197] rel2
                    | data8 ',’[145] rel
two-op 6    :       reg ‘,’ [201] A [246]
                    | reg ‘,’[201] data 8
                    | reg ‘,’[201]’#’[248]data 8
                    | data 8 ‘,’[149] reg
                    | data 8 ‘,’[149]data 8
                    | data 8 ‘,’[149]’@’[243]regi
```

รูปที่ 3.3 (ต่อ) ไวยากรณ์ล่ามของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำ ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    | '@'[151]regi ','[249] A [278]
                    | '@'regi[151] ','[249]'data 8
                    | '@'[151]regi','[249]'#'[280]data 8
                    | DPTR[152]','[203]'#'[250]data16
                    | C[153] ','[204] bit
                    | data 8 ','[149] C [244]
                    | data 8 BITPOS ','[149] C [244]
single-op 1 :       A [92]
                    | reg
                    | data 8
                    | '@' [95] regi
single-op 2 :       A [116]
                    | C [117]
                    | bit
three-op1   :       A [136] ',' [193] data 8','rel
                    | A [136] ','[193]'#'[233] data 8','[272]
rel
                    | reg ','[194] '#'[234] data 8',' [273] rel
                    | @'[138]regi','[235]'#'[274]data8','[293]
rel
rel         :       expr
rel2        :       expr
bit         :       bitv BITPOS
                    | bitv
bitv        :       SYMBOL [120]
                    | VALUE [121]
reg         :       R0    [96]
                    | R1   [97]
                    | R2   [98]
                    | R3   [99]
                    | R4   [100]
                    | R5   [101]
                    | R6   [102]
                    | R7   [103]
regi        :       R0    [182]
                    | R1    [183]
                    | R2    [184]
                    | R3    [185]
                    | R4    [186]
                    | R5    [187]
                    | R6    [188]
                    | R7    [189]
data 8      :       expr
data 16     :       expr
addr 11     :       expr
addr 16     :       expr
relative    :       expr
```

รูปที่ 3.3 (ต่อ) ไวยากรณ์ล่ามของตัวแปลภาษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 การตรวจสอบวากยสัมพันธ์ของตัวแปลภาษา

ขั้นตอนการตรวจสอบวากยสัมพันธ์ของตัวแปลภาษา จะตรวจสอบไวยากรณ์ด้วยวิธีนอนรีเคอร์ซีฟพรีดิกทีฟพาซเซอร์ ซึ่งจะหา First และ Follow ของสัญลักษณ์นอนเทอร์มินอลต่างๆ

ตารางที่ 3.1 สัญลักษณ์นอนเทอร์มินอล และรหัส

สัญลักษณ์นอนเทอร์มินอล	รหัส
program	325
linelist	326
line	327
linerest	328
directive	329
defexpr	330
flag	331
flagv	332
undefsym	333
blist	334
wlist	335
expr	336
instr	337
two_op1	338
two_op2	339
two_op3	340
two_op4	341
two_op5	342
two_op6	343
single_op1	344
single_op2	345
three_op1	346

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 (ต่อ) สัญลักษณ์นอนเทอร์มินอล และรหัส

สัญลักษณ์นอนเทอร์มินอล	รหัส
rel	347
rel2	348
bit	349
bitv	350
reg	351
regi	352
data8	353
data16	354
addr11	355
addr16	356
relative	357

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

4.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการทดสอบชุดคำสั่ง และคำสั่งเทียมของไมโครคอนโทรลเลอร์ MCS-51 โดยจะมีการเปรียบเทียบผลการทดสอบที่ได้ระหว่างผลการทดสอบของ SXA51 กับผลการทดสอบของโปรแกรมที่ได้มีการเขียนขึ้นมาใช้งาน ว่ามีความแตกต่างกันหรือไม่ เพื่อเป็นการตรวจสอบความผิดพลาดของผลการทดสอบที่อาจเกิดขึ้น ซึ่งทำให้สามารถทราบถึงความผิดพลาดของโปรแกรมที่อาจเกิดขึ้น และจะส่งผลทำให้เกิดความไม่สมบูรณ์ของโปรแกรม

4.2 การทดสอบชุดคำสั่ง และคำสั่งเทียม

```
        ORG    0000H   ;       ORG   expr
LOOP1:  SJMP   LOOP1   ;       SJMP   relative
        EQU    TU 10H  ;       EQU    expr
        NOP            ;       NOP
        AJMP   TU      ;       AJMP  addr11
        LJMP   10H     ;       LJMP   addr16
        RR     A       ;       RR     A
        INC    A       ;       INC    A
        INC    10H     ;       INC    direct
        INC    @R1     ;       INC     @RI
        INC    @R0     ;       INC     @RI
        INC    R0      ;       INC    Rn
        INC    R1      ;       INC    Rn
        INC    R2      ;       INC    Rn
        INC    R3      ;       INC    Rn
        INC    R4      ;       INC    Rn
        INC    R5      ;       INC    Rn
        INC    R6      ;       INC    Rn
        INC    R7      ;       INC    Rn
        JBC    00H,10H;JBC    bit,rel
        ACALL  10H     ;       ACALL  addr11
        LCALL  10H     ;       LCALL  addr16
        RRC    A       ;       RRC    A
        DEC    A       ;       DEC    A
        DEC    10H     ;       DEC    direct
```

รูปที่ 4.1 โปรแกรมที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEC     @R0     ;       DEC     @RI
DEC     @R1     ;       DEC     @RI
DEC     R0      ;       DEC     Rn
DEC     R1      ;       DEC     Rn
DEC     R2      ;       DEC     Rn
DEC     R3      ;       DEC     Rn
DEC     R4      ;       DEC     Rn
DEC     R5      ;       DEC     Rn
DEC     R6      ;       DEC     Rn
DEC     R7      ;       DEC     Rn
JBC     00H,10H;JBC     bit,rel
AJMP    10H     ;       AJMP    addr11
RET             ;       RET
RL      A       ;       RL      A
ADD     A,#10H  ;ADD    A,#data8
ADD     A,10H   ;       ADD     direct
ADD     A,@R0   ;       ADD     A,@Ri
ADD     A,@R1   ;       ADD     A,@Ri
ADD     A,R0    ;       ADD     A,Rn
ADD     A,R1    ;       ADD     A,Rn
ADD     A,R2    ;       ADD     A,Rn
ADD     A,R3    ;       ADD     A,Rn
ADD     A,R4    ;       ADD     A,Rn
ADD     A,R5    ;       ADD     A,Rn
ADD     A,R6    ;       ADD     A,Rn
ADD     A,R7    ;       ADD     A,Rn
JNB     00H,10H;JNB     bit,rel
RET             ;       RET
RL      A       ;       RL      A
ADD     A,10H   ;       ADD     direct
ADD     A,@R0   ;       ADD     A,@Ri
ADD     A,@R1   ;       ADD     A,@Ri
ADD     A,R0    ;       ADD     A,Rn
ADD     A,R1    ;       ADD     A,Rn
ADD     A,R2    ;       ADD     A,Rn
ADD     A,R3    ;       ADD     A,Rn
ADD     A,R4    ;       ADD     A,Rn
ADD     A,R5    ;       ADD     A,Rn
ADD     A,R6    ;       ADD     A,Rn
ADD     A,R7    ;       ADD     A,Rn
JNB     00H,10H;JNB     bit,rel
ACALL   10H     ;       ACALL   addr11
RETI            ;       RETI
RLC     A       ;       RLC     A
ADDC    A,#10H  ;ADDC   A,#data8
ADDC    A,10H   ;       ADDC    A,direct
ADDC    A,@R0   ;       ADDC    A,@Ri
ADDC    A,@R1   ;       ADDC    A,@Ri
ADDC    A,R0    ;       ADDC    A,@Rn
ADDC    A,R1    ;       ADDC    A,@Rn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ

```
ADDC    A,R2    ;       ADDC    A,@Rn
ADDC    A,R3    ;       ADDC    A,@Rn
ADDC    A,R4    ;       ADDC    A,@Rn
ADDC    A,R5    ;       ADDC    A,@Rn
ADDC    A,R6    ;       ADDC    A,@Rn
ADDC    A,R7    ;       ADDC    A,@Rn
JC      10H     ;       JC      relative
AJMP    10H     ;       AJMP    addr11
ORL     10H,A   ;       ORL     direct,A
ORL     10H,#10H;       ORL     direct,#data8
ORL     A,#10H  ;       ORL     A,#data8
ORL     A,10H   ;       ORL     A,direct
ORL     A,@R0   ;       ORL     A,@Ri
ORL     A,@R1   ;       ORL     A,@Ri
ORL     A,R0    ;       ORL     A,Rn
ORL     A,R1    ;       ORL     A,Rn
ORL     A,R2    ;       ORL     A,Rn
ORL     A,R3    ;       ORL     A,Rn
ORL     A,R4    ;       ORL     A,Rn
ORL     A,R5    ;       ORL     A,Rn
ORL     A,R6    ;       ORL     A,Rn
ORL     A,R7    ;       ORL     A,Rn
JNC     10H     ;       JNC     relative
ACALL   10H     ;       ACALL   addr11
ANL     10H,A   ;       ANL     direct,A
ANL     10H,#10H;       ANL     direct,#data8
ANL     A,#10H  ;       ANL     A,#data8
ANL     A,10H   ;       ANL     A,direct
ANL     A,@R0   ;       ANL     A,@Ri
ANL     A,@R1   ;       ANL     A,@Ri
ANL     A,R0    ;       ANL     A,@Rn
ANL     A,R1    ;       ANL     A,@Rn
ANL     A,R2    ;       ANL     A,@Rn
ANL     A,R3    ;       ANL     A,@Rn
ANL     A,R4    ;       ANL     A,@Rn
ANL     A,R5    ;       ANL     A,@Rn
ANL     A,R6    ;       ANL     A,@Rn
ANL     A,R7    ;       ANL     A,@Rn
JZ      10H     ;       JZ      relative
AJMP    10H     ;       AJMP    addr11
XRL     10H,A   ;       XRL     direct,A
XRL     10H,#10H;       XRL     direct,#data8
XRL     A,#10H  ;       XRL     A,#data8
XRL     A,10H   ;       XRL     A,direct
XRL     A,@R0   ;       XRL     A,@Ri
XRL     A,@R1   ;       XRL     A,@Ri
XRL     A,R0    ;       XRL     A,Rn
XRL     A,R1    ;       XRL     A,Rn
XRL     A,R2    ;       XRL     A,Rn
XRL     A,R3    ;       XRL     A,Rn
```

รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
XRL     A,R4            ;XRL    A,Rn
XRL     A,R5            ;XRL    A,Rn
XRL     A,R6            ;XRL    A,Rn
XRL     A,R7            ;XRL    A,Rn
JNZ     0099H           ;JNZ    relative
ACALL   0099h           ;ACALL  addr11
ORL     C,01H           ;ORL    C,bit
JMP     @A+DPTR         ;JMP    @A+DPTR
MOV     A,#100          ;MOV    A,#data8
MOV     10H,#10H        ;MOV    direct,#data8
MOV     @R0,#10H        ;MOV    @Ri,#direct
MOV     @R1,#10H        ;MOV    @Ri,#direct
MOV     R0,#10H         ;MOV    Rn,#data8
MOV     R1,#10H         ;MOV    Rn,#data8
MOV     R2,#10H         ;MOV    Rn,#data8
MOV     R3,#10H         ;MOV    Rn,#data8
MOV     R4,#10H         ;MOV    Rn,#data8
MOV     R5,#10H         ;MOV    Rn,#data8
MOV     R6,#10H         ;MOV    Rn,#data8
MOV     R7,#10H         ;MOV    Rn,#data8
SJMP    00ADH           ;SJMP   relative
AJMP    00ADH           ;AJMP   addr11
ANL     C,00H           ;ANL    C,bit
MOVC    A,@A+PC         ;MOVC   A,@A+PC
DIV     AB              ;DIV    AB
MOV     10H,10H         ;MOV    direct,direct
MOV     10H,@R0         ;MOV    direct,@Ri
MOV     10H,@R1         ;MOV    direct,@Ri
MOV     10H,R0          ;MOV    direct,Rn
MOV     10H,R1          ;MOV    direct,Rn
MOV     10H,R2          ;MOV    direct,Rn
MOV     10H,R3          ;MOV    direct,Rn
MOV     10H,R4          ;MOV    direct,Rn
MOV     10H,R5          ;MOV    direct,Rn
MOV     10H,R6          ;MOV    direct,Rn
MOV     10H,R7          ;MOV    direct,Rn
SJMP    00ADH           ;SJMP   relative
AJMP    00ADH           ;AJMP   addr11
ANL     C,00H           ;ANL    C,bit
MOVC    A,@A+PC         ;MOVC   A,@A+PC
DIV     AB              ;DIV    AB
MOV     10H,10H         ;MOV    direct,direct
MOV     10H,@R0         ;MOV    direct,@Ri
MOV     10H,@R1         ;MOV    direct,@Ri
MOV     10H,R0          ;MOV    direct,Rn
MOV     10H,R1          ;MOV    direct,Rn
MOV     10H,R2          ;MOV    direct,Rn
MOV     10H,R3          ;MOV    direct,Rn
MOV     10H,R4          ;MOV    direct,Rn
MOV     10H,R5          ;MOV    direct,Rn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ

```
MOV     R6,#10H         ;MOV    Rn,#data8
MOV     R7,#10H         ;MOV    Rn,#data8
SJMP    00ADH           ;SJMP   relative
AJMP    00ADH           ;AJMP addr11
ANL     C,00H           ;ANL    C,bit
MOVC    A,@A+PC         ;MOVC   A,@A+PC
DIV     AB              ;DIV    AB
MOV     10H,10H         ;MOV    direct,direct
MOV     10H,@R0         ;MOV    direct,@Ri
MOV     10H,@R1         ;MOV    direct,@Ri
MOV     10H,R0          ;MOV    direct,Rn
MOV     10H,R1          ;MOV    direct,Rn
MOV     10H,R2          ;MOV    direct,Rn
MOV     10H,R3          ;MOV    direct,Rn
MOV     10H,R4          ;MOV    direct,Rn
MOV     10H,R5          ;MOV    direct,Rn
MOV     10H,R6          ;MOV    direct,Rn
MOV     10H,R7          ;MOV    direct,Rn
SJMP    00ADH           ;SJMP   relative
AJMP    00ADH           ;AJMP addr11
ANL     C,00H           ;ANL    C,bit
MOVC    A,@A+PC         ;MOVC   A,@A+PC
DIV     AB              ;DIV    AB
MOV     10H,10H         ;MOV    direct,direct
MOV     10H,@R0         ;MOV    direct,@Ri
MOV     10H,@R1         ;MOV    direct,@Ri
MOV     10H,R0          ;MOV    direct,Rn
MOV     10H,R1          ;MOV    direct,Rn
MOV     10H,R2          ;MOV    direct,Rn
MOV     10H,R3          ;MOV    direct,Rn
MOV     10H,R4          ;MOV    direct,Rn
MOV     10H,R5          ;MOV    direct,Rn
MOV     10H,R6          ;MOV    direct,Rn
MOV     10H,R7          ;MOV    direct,Rn
MOV     DPTR,#10H       ;MOV    DPTR,#DATA 16
ACALL   10H             ;ACALL addr11
MOV     00H,C           ;MOV    bit,C
MOVC    A,@A+DPTR       ;MOVC   A,@A+DPTR
SUBB    A,#10H          ;SUBB   A,#data8
SUBB    A,10H           ;SUBB   A,direct
SUBB    A,@R0           ;SUBB   A,@Ri
SUBB    A,@R1           ;SUBB   A,@Ri
SUBB    A,R0            ;SUBB   A,Rn
SUBB    A,R1            ;SUBB   A,Rn
SUBB    A,R2            ;SUBB   A,Rn
SUBB    A,R3            ;SUBB   A,Rn
SUBB    A,R4            ;SUBB   A,Rn
SUBB    A,R5            ;SUBB   A,Rn
SUBB    A,R6            ;SUBB   A,Rn
SUBB    A,R7            ;SUBB   A,Rn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ

```
        ORL     C,!00H          ;ORL    C,!bit
        AJMP    10H             ;AJMP   addr11
        MOV     C,00H           ;MOV    C,bit
        INC     DPTR            ;INC    DPTR
        MUL     AB              ;MUL    AB
        MOV     @R0,10H         ;MOV    @Ri',direct
        MOV     @R1,10H         ;MOV    @Ri',direct
        MOV     R0,10H          ;MOV    Rn,#direct
        MOV     R1,10H          ;MOV    Rn,#direct
        MOV     R2,10H          ;MOV    Rn,#direct
        MOV     R3,10H          ;MOV    Rn,#direct
        MOV     R4,10H          ;MOV    Rn,#direct
        MOV     R5,10H          ;MOV    Rn,#direct
        MOV     R6,10H          ;MOV    Rn,#direct
        MOV     R7,10H          ;MOV    Rn,#direct
        ANL     C,/00H          ;ANL    C,/bit
        ACALL   10H             ;ACALL  addr11
        CPL     00H             ;CPL    bit
loop:   CPL     C               ;CPL    C
        CJNE    A,30H,loop      ;CJNE   A,direct,rel
        CJNE    A,#30H,loop     ;CJNE   A,#data8,rel
        CJNE    @R0,#22H,loop   ;CJNE   @Ri,#data8,rel
        CJNE    @R1,#22H,loop   ;CJNE   @Ri,#data8,rel
        CJNE    R0,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R1,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R2,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R3,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R4,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R5,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R6,#22H,loop    ;CJNE   Rn,#data8,rel
        CJNE    R7,#22H,loop    ;CJNE   Rn,#data8,rel
        PUSH    10H             ;PUSH   direct
        AJMP    10H             ;AJMP   addr11
        CLR     00H             ;CLR    bit
        CLR     C               ;CLR    C
        SWAP    A               ;SWAP   A
        XCH     A,10H           ;XCH    A,direct
        XCH     A,@R0           ;XCH    A,@Ri
        XCH     A,@R1           ;XCH    A,@Ri
        XCH     A,R0            ;XCH    A,Rn
        XCH     A,R1            ;XCH    A,Rn
        XCH     A,R2            ;XCH    A,Rn
        XCH     A,R3            ;XCH    A,Rn
        XCH     A,R4            ;XCH    A,Rn
        XCH     A,R5            ;XCH    A,Rn
        XCH     A,R6            ;XCH    A,Rn
        XCH     A,R7            ;XCH    A,Rn
        POP     10H             ;POP    direct
        ACALL   10H             ;ACALL  addr11
        SETB    00H             ;SETB   bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ

```
SETB    C               ;SETB   C
DA      A               ;DA     A
DJNZ    30H,loop        ;DJNZ   direct,rel
XCHD    A,@R0           ;XCHD   A,@Ri
XCHD    A,@R1           ;XCHD   A,@Ri
DJNZ    R0,160H         ;DJNZ Rn,rel
DJNZ    R1,160H         ;DJNZ Rn,rel
DJNZ    R2,160H         ;DJNZ Rn,rel
DJNZ    R3,160H         ;DJNZ Rn,rel
DJNZ    R4,160H         ;DJNZ  Rn,rel
DJNZ    R5,160H         ;DJNZ  Rn,rel
DJNZ    R6,160H         ;DJNZ  Rn,rel
DJNZ    R7,160H         ;DJNZ  Rn,rel
MOVX    A,@DPTR         ;MOVX  A,@DPTR
AJMP    10H             ;AJMP  addr11
MOVX    A,@R0           ;MOVX  A,@Ri
MOVX    A,@R1           ;MOVX  A,@Ri
CLR     A               ;CLR   A
MOV     A,10H           ;MOV   A,direct
MOV     A,@R0           ;MOV   A,@Ri
MOV     A,@R1           ;MOV   A,@Ri
MOV     A,R0            ;MOV   A,@Rn
MOV     A,R1            ;MOV   A,@Rn
MOV     A,R2            ;MOV   A,@Rn
MOV     A,R3            ;MOV   A,@Rn
MOV     A,R4            ;MOV   A,@Rn
MOV     A,R5            ;MOV   A,@Rn
MOV     A,R6            ;MOV   A,@Rn
MOV     A,R7            ;MOV   A,@Rn
MOVX    @DPTR,A         ;MOVX  @DPTR,A
ACALL   10H             ;ACALL addr11
MOVX    @R0,A           ;MOVX  @Ri,A
MOVX    @R1,A           ;MOVX  @Ri,A
CPL     A               ;CPL   A
MOV     10H,A           ;MOV   direct,A
MOV     @R0,A           ;MOV   @Ri,A
MOV     @R1,A           ;MOV   @Ri,A
MOV     R0,A            ;MOV   Rn,A
MOV     R1,A            ;MOV   Rn,A
MOV     R2,A            ;MOV   Rn,A
MOV     R3,A            ;MOV   Rn,A
MOV     R4,A            ;MOV   Rn,A
MOV     R5,A            ;MOV   Rn,A
MOV     R6,A            ;MOV   Rn,A
MOV     R7,A            ;mOV   Rn,A
ANL     C,!00H          ;ANL   C,!bit
ORL     C,/01H          ;ORL   C,/bit
END                     ;END
```

รูปที่ 4.1 (ต่อ) โปรแกรมที่ใช้ในการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 ผลการทดสอบเมื่อเรียกใช้โปรแกรม SXA51

```
:1000000080FE000110020010030405100706080915
:100010000A0B0C0D0E0F1000F71110120010131424
:1000200015101617181918181C1D1E1F1000E101B0
:1000300010222324102510262728292A2B2C2D2E88
:100040002F3000CC11103233341035103637383998
:100050003A3B3C3D3E3F40B8011042104310104433
:1000600010451046474849 4A4B4C4D4E4F50A11140
:1000700010521053101054105510565758595A5BBF
:100080005C5D5E5F608A011062106310641065 31
:1000900010666768696A6B6C6D6E6F70FC1199729F
:1000A000017374647510107610771078107 9107AD7
:1000B000107B107C107D107E107F1080F001AD82CF
:1000C000083848510108610871088108 9108A108C
:1000D0008B108C108D108E108F10900010111092BC
:1000E000093941095109697989 99A9B9C9D9E9F2B
:1000F00072000110A200A3A4A610A710A810A910B6
:10010000AA10AB10AC10AD10AE10AF10B0001110B3
:10011000B200B3B530FCB430F9B622F6B722F3B86A
:1001200022F0B922EDBA22EABB22E7BC22E4BD22CA
:10013000E1BE22DEBF22DBC0100110C200C3C4C575
:1001400010C6C7C8C9CACBCCCDCECFD0101110D2E3
:1001500000D3D4D530BCD6D7D806D904DA02DB0018
:10016000DCFEDDFCDEFADFF8E00110E2E3E4E5109E
:10017000E6E7E8E9EAEBECEDEEEFF01110F2F3F46C
:10018000F510F6F7F8F9FAFBFCFDFEFF8200A0017E
:00000001FF
```

รูปที่ 4.2 ผลการทดสอบเมื่อเรียกใช้โปรแกรม SXA51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ผลการทดสอบเมื่อเรียกใช้โปรแกรม ECOM51

```
:1000000080FE000110020010030405100706080915
:100010000A0B0C0D0E0F1000F71110120010131424
:1000200015101617181918A1B1C1D1E1F1000E101B0
:10003000102223241025102627282928A2B2C2D2E88
:100040002F3000CC11103233341035103637383998
:100050003A3B3C3D3E3F40B8011042104310104433
:10006000104510464748494A4B4C4D4E4F50A11140
:10007000105210531010541055105657585 95A5BBF
:100080005C5D5E5F608A011062106310106 4106531
:10009000106667686 96A6B6C6D6E6F70FC1199729F
:1000A000017374647510107610771078107 9107AD7
:1000B000107B107C107D107E107F1080F001AD82CF
:1000C000008384851010861087108810891 08A108C
:1000D0008B108C108D108E108F10900010111092BC
:1000E000093941095109697989 99A9B9C9D9E9F2B
:1000F00072000110A200A3A4A610A710A810A910B6
:10010000AA10AB10AC10AD10AE10AF10B0001110B3
:10011000B200B3B530FCB430F9B622F6B722F3B86A
:1001200022F0B922EDBA22EABB22E7BC22E4BD22CA
:10013000E1BE22DEBF22DBC0100110C200C3C4C575
:1001400010C6C7C8C9CACBCCCDCECFD0101110D2E3
:1001500000D3D4D530BCD6D7D806D904DA02DB0018
:10016000DCFEDDFCDEFADFF8E00110E2E3E4E5109E
:10017000E6E7E8E9EAEBECEDEEEFF01110F2F3F46C
:10018000F510F6F7F8F9FAFBFCFDFEFF8200A0017E
:00000001FF
```

รูปที่ 4.3 ผลการทดสอบเมื่อเรียกใช้โปรแกรม ECOM51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 ผลการทดสอบพารามิเตอร์ –I

1) ผลการทดสอบเมื่อเรียกใช้โปรแกรม ECOM51

```
0000:                   ORG     0000H   ;       ORG
0000: 80 FE     LOOP1:  SJMP    LOOP1   ;       SJMP    relative
0002:                   EQU     TU 10H  ;       EQU     expr
0002: 00                NOP             ;       NOP
0003: 01 10             AJMP    TU      ;       AJMP    add
0005: 02 00 10          LJMP    10H     ;       LJMP    addr16
0008: 03                RR      A       ;       RR      A
0009: 04                INC     A       ;       INC     A
000A: 05 10             INC     10H     ;       INC     direct
000C: 07                INC     @R1     ;       INC     @RI
000D: 06                INC     @R0     ;       INC     @RI
000E: 08                INC     R0      ;       INC     Rn
000F: 09                INC     R1      ;       INC     Rn
0010: 0A                INC     R2      ;       INC     Rn
0011: 0B                INC     R3      ;       INC     Rn
0012: 0C                INC     R4      ;       INC     Rn
0013: 0D                INC     R5      ;       INC     Rn
0014: 0E                INC     R6      ;       INC     Rn
0015: 0F                INC     R7      ;       INC     Rn
0016: 10 00 F7          JBC     00H,10H ;       JBC     bit,rel
0019: 11 10             ACALL   10H     ;       ACALL   addr11
001B: 12 00 10          LCALL   10H     ;       LCALL   addr16
001E: 13                RRC     A       ;       RRC     A
001F: 14                DEC     A       ;       DEC     A
0020: 15 10             DEC     10H     ;       DEC     direct
0022: 16                DEC     @R0     ;       DEC     @RI
0023: 17                DEC     @R1     ;       DEC     @RI
0024: 18                DEC     R0      ;       DEC     Rn
0025: 19                DEC     R1      ;       DEC     Rn
0026: 1A                DEC     R2      ;       DEC     Rn
0027: 1B                DEC     R3      ;       DEC     Rn
0028: 1C                DEC     R4      ;       DEC     Rn
0029: 1D                DEC     R5      ;       DEC     Rn
002A: 1E                DEC     R6      ;       DEC     Rn
002B: 1F                DEC     R7      ;       DEC     Rn
002C: 10 00 E1          JBC     00H,10H ;       JBC     bit,rel
002F: 01 10             AJMP    10H     ;       AJMP    addr11
0031: 22                RET             ;       RET
0032: 23                RL      A       ;       RL      A
0033: 24 10             ADD     A,#10H  ;       ADD     A,#data8
0035: 25 10             ADD     A,10H   ;       ADD     direct
0037: 26                ADD     A,@R0   ;       ADD     A,@Ri
0038: 27                ADD     A,@R1   ;       ADD     A,@Ri
0039: 28                ADD     A,R0    ;       ADD     A,Rn
003A: 29                ADD     A,R1    ;       ADD     A,Rn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 ผลการทดสอบพารามิเตอร์ –I เมื่อเรียกใช้โปรแกรม ECOM51

```
003B: 2A                ADD     A,R2     ;       ADD     A,Rn
003C: 2B                ADD     A,R3     ;       ADD     A,Rn
003D: 2C                ADD     A,R4     ;       ADD     A,Rn
003E: 2D                ADD     A,R5     ;       ADD     A,Rn
003F: 2E                ADD     A,R6     ;       ADD     A,Rn
0040: 2F                ADD     A,R7     ;       ADD     A,Rn
0041: 30 00 CC          JNB     00H,10H  ;       JNB     bit,rel
0044: 11 10             ACALL   10H      ;       ACALL   addr11
0046: 32                RETI             ;       RETI
0047: 33                RLC     A        ;       RLC     A
0048: 34 10             ADDC    A,#10H   ;       ADDC    A,#data8
004A: 35 10             ADDC    A,10H    ;       ADDC    A,direct
004C: 36                ADDC    A,@R0    ;       ADDC    A,@Ri
004D: 37                ADDC    A,@R1    ;       ADDC    A,@Ri
004E: 38                ADDC    A,R0     ;       ADDC    A,@Rn
004F: 39                ADDC    A,R1     ;       ADDC    A,@Rn
0050: 3A                ADDC    A,R2     ;       ADDC    A,@Rn
0051: 3B                ADDC    A,R3     ;       ADDC    A,@Rn
0052: 3C                ADDC    A,R4     ;       ADDC    A,@Rn
0053: 3D                ADDC    A,R5     ;       ADDC    A,@Rn
0054: 3E                ADDC    A,R6     ;       ADDC    A,@Rn
0055: 3F                ADDC    A,R7     ;       ADDC    A,@Rn
0056: 40 B8             JC      10H      ;       JC      relative
0058: 01 10             AJMP    10H      ;       AJMP    addr11
005A: 42 10             ORL     10H,A    ;       ORL     direct,A
005C: 43 10 10          ORL     10H,#10H;       ORL  direct,#data
005F: 44 10             ORL     A,#10H   ;       ORL     A,#data8
0061: 45 10             ORL     A,10H    ;       ORL     A,direct
0063: 46                ORL     A,@R0    ;       ORL     A,@Ri
0064: 47                ORL     A,@R1    ;       ORL     A,@Ri
0065: 48                ORL     A,R0     ;       ORL     A,Rn
0066: 49                ORL     A,R1     ;       ORL     A,Rn
0067: 4A                ORL     A,R2     ;       ORL     A,Rn
0068: 4B                ORL     A,R3     ;       ORL     A,Rn
0069: 4C                ORL     A,R4     ;       ORL     A,Rn
006A: 4D                ORL     A,R5     ;       ORL     A,Rn
006B: 4E                ORL     A,R6     ;       ORL     A,Rn
006C: 4F                ORL     A,R7     ;       ORL     A,Rn
006D: 50 A1             JNC     10H      ;       JNC     relative
006F: 11 10             ACALL   10H      ;       ACALL   addr11
0071: 52 10             ANL     10H,A    ;       ANL     direct,A
0073: 53 10 10          ANL     10H,#10H;       ANL  direct,#data
0076: 54 10             ANL     A,#10H   ;       ANL     A,#data8
0078: 55 10             ANL     A,10H    ;       ANL     A,direct
007A: 56                ANL     A,@R0    ;       ANL     A,@Ri
007B: 57                ANL     A,@R1    ;       ANL     A,@Ri
007C: 58                ANL     A,R0     ;       ANL     A,@Rn
007D: 59                ANL     A,R1     ;       ANL     A,@Rn
007E: 5A                ANL     A,R2     ;       ANL     A,@Rn
007F: 5B                ANL     A,R3     ;       ANL     A,@Rn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งนี้ ...การนำไปใช้

รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51

```
0080: 5C          ANL     A,R4     ;      ANL     A,@Rn
0081: 5D          ANL     A,R5     ;      ANL     A,@Rn
0082: 5E          ANL     A,R6     ;      ANL     A,@Rn
0083: 5F          ANL     A,R7     ;      ANL     A,@Rn
0084: 60 8A       JZ      10H      ;      JZ      relative
0086: 01 10       AJMP    10H      ;      AJMP    addr11
0088: 62 10       XRL     10H,A    ;      XRL     direct,A
008A: 63 10 10    XRL     10H,#10H;      XRL  direct,#data
008D: 64 10       XRL     A,#10H   ;      XRL     A,#data8
008F: 65 10       XRL     A,10H    ;      XRL     A,direct
0091: 66          XRL     A,@R0    ;      XRL     A,@Ri
0092: 67          XRL     A,@R1    ;      XRL     A,@Ri
0093: 68          XRL     A,R0     ;      XRL     A,Rn
0094: 69          XRL     A,R1     ;      XRL     A,Rn
0095: 6A          XRL     A,R2     ;      XRL     A,Rn
0096: 6B          XRL     A,R3     ;      XRL     A,Rn
0097: 6C          XRL     A,R4     ;      XRL     A,Rn
0098: 6D          XRL     A,R5     ;      XRL     A,Rn
0099: 6E          XRL     A,R6     ;      XRL     A,Rn
009A: 6F          XRL     A,R7     ;      XRL     A,R
009B: 70 FC       JNZ     0099H    ;      JNZ     relative
009D: 11 99       ACALL   0099h    ;      ACALL   add
009F: 72 01       ORL     C,01H    ;      ORL     C,bit
00A1: 73          JMP     @A+DPTR  ;      JMP     @A+DPTR
00A2: 74 64       MOV     A,#100   ;      MOV     A,#data8
00A4: 75 10 10    MOV     10H,#10H;      MOV  direct,#data8
00A7: 76 10       MOV     @R0,#10H;      MOV  @Ri,#direct
00A9: 77 10       MOV     @R1,#10H;      MOV  @Ri,#direct
00AB: 78 10       MOV     R0,#10H  ;     MOV     Rn,#data8
00AD: 79 10       MOV     R1,#10H  ;     MOV     Rn,#data8
00AF: 7A 10       MOV     R2,#10H  ;     MOV     Rn,#data8
00B1: 7B 10       MOV     R3,#10H  ;     MOV     Rn,#data8
00B3: 7C 10       MOV     R4,#10H  ;     MOV     Rn,#data8
00B5: 7D 10       MOV     R5,#10H  ;     MOV     Rn,#data8
00B7: 7E 10       MOV     R6,#10H  ;     MOV     Rn,#data8
00B9: 7F 10       MOV     R7,#10H  ;     MOV     Rn,#data8
00BB: 80 F0       SJMP    00ADH    ;     SJMP    rel
00BD: 01 AD       AJMP    00ADH    ;     AJMP    add
00BF: 82 00       ANL     C,00H    ;     ANL     C,bit
00C1: 83          MOVC    A,@A+PC  ;     MOVC    A,@A+PC
00C2: 84          DIV     AB       ;     DIV     AB
00C3: 85 10 10    MOV     10H,10H  ;     MOV  direct,diret
00C6: 86 10       MOV     10H,@R0  ;     MOV    direct,@Ri
00C8: 87 10       MOV     10H,@R1  ;     MOV    direct,@Ri
00CA: 88 10       MOV     10H,R0   ;     MOV     direct,Rn
00CC: 89 10       MOV     10H,R1   ;     MOV     direct,Rn
00CE: 8A 10       MOV     10H,R2   ;     MOV     direct,Rn
00D0: 8B 10       MOV     10H,R3   ;     MOV     direct,Rn
00D2: 8C 10       MOV     10H,R4   ;     MOV     direct,Rn
00D4: 8D 10       MOV     10H,R5   ;     MOV     direct,Rn
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –l เมื่อเรียกใช้โปรแกรม ECOM51

```
00D6: 8E 10              MOV     10H,R6  ;       MOV     direct,Rn
00D8: 8F 10              MOV     10H,R7  ;       MOV     direct,Rn
00DA: 90 00 10           MOV     DPTR,#10H;      MOV DPTR,#DATA 16
00DD: 11 10              ACALL   10H     ;       ACALL   addr11
00DF: 92 00              MOV     00H,C   ;       MOV     bit,C
00E1: 93                 MOVC    A,@A+DPTR;      MOVC    A,@A+DPTR
00E2: 94 10              SUBB    A,#10H  ;       SUBB    A,#data8
00E4: 95 10              SUBB    A,10H   ;       SUBB    A,direct
00E6: 96                 SUBB    A,@R0   ;       SUBB    A,@Ri
00E7: 97                 SUBB    A,@R1   ;       SUBB    A,@Ri
00E8: 98                 SUBB    A,R0    ;       SUBB    A,Rn
00E9: 99                 SUBB    A,R1    ;       SUBB    A,Rn
00EA: 9A                 SUBB    A,R2    ;       SUBB    A,Rn
00EB: 9B                 SUBB    A,R3    ;       SUBB    A,Rn
00EC: 9C                 SUBB    A,R4    ;       SUBB    A,Rn
00ED: 9D                 SUBB    A,R5    ;       SUBB    A,Rn
00EE: 9E                 SUBB    A,R6    ;       SUBB    A,Rn
00EF: 9F                 SUBB    A,R7    ;       SUBB    A,Rn
00F0: A0 00              ORL     C,!00H  ;       ORL     C,!bit
00F2: 01 10              AJMP    10H     ;       AJMP    addr11
00F4: A2 00              MOV     C,00H   ;       MOV     C,bit
00F6: A3                 INC     DPTR    ;       INC     DPTR
00F7: A4                 MUL     AB      ;       MUL     AB
00F8: A6 10              MOV     @R0,10H ;       MOV     @Ri,direct
00FA: A7 10              MOV     @R1,10H ;       MOV     @Ri,direct
00FC: A8 10              MOV     R0,10H  ;       MOV     Rn,#direct
00FE: A9 10              MOV     R1,10H  ;       MOV     Rn,#direct
0100: AA 10              MOV     R2,10H  ;       MOV     Rn,#direct
0102: AB 10              MOV     R3,10H  ;       MOV     Rn,#direct
0104: AC 10              MOV     R4,10H  ;       MOV     Rn,#direct
0106: AD 10              MOV     R5,10H  ;       MOV     Rn,#direct
0108: AE 10              MOV     R6,10H  ;       MOV     Rn,#direct
010A: AF 10              MOV     R7,10H  ;       MOV     Rn,#direct
010C: B0 00              ANL     C,/00H  ;       ANL     C,/bit
010E: 11 10              ACALL   10H     ;       ACALL   addr11
0110: B2 00              CPL     00H     ;       CPL     bit
0112: B3           loop: CPL     C       ;       CPL     C
0113: B5 30 FC           CJNE    A,30H,loop;     CJNE    A,direct,rel
0116: B4 30 F9           CJNE    A,#30H,loop  ;CJNE   A,#data8,rel
0119: B6 22 F6           CJNE    @R0,#22H,loop;CJNE @Ri,#data8,rel
011C: B7 22 F3           CJNE    @R1,#22H,loop;CJNE @Ri,#data8,rel
011F: B8 22 F0           CJNE    R0,#22H,loop ;CJNE  Rn,#data8,rel
0122: B9 22 ED           CJNE    R1,#22H,loop ;CJNE  Rn,#data8,rel
0125: BA 22 EA           CJNE    R2,#22H,loop ;CJNE  Rn,#data8,rel
0128: BB 22 E7           CJNE    R3,#22H,loop ;CJNE  Rn,#data8,rel
012B: BC 22 E4           CJNE    R4,#22H,loop ;CJNE  Rn,#data8,rel
012E: BD 22 E1           CJNE    R5,#22H,loop ;CJNE  Rn,#data8,rel
0131: BE 22 DE           CJNE    R6,#22H,loop ;CJNE  Rn,#data8,rel
0134: BF 22 DB           CJNE    R7,#22H,loop ;CJNE  Rn,#data8,rel
0137: C0 10              PUSH    10H          ;      PUSH    direct
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –I เมื่อเรียกใช้โปรแกรม ECOM51

```
0139: 01 10          AJMP    10H       ;       AJMP    addr11
013B: C2 00          CLR     00H       ;       CLR     bit
013D: C3             CLR     C         ;       CLR     C
013E: C4             SWAP    A         ;       SWAP    A
013F: C5 10          XCH     A,10H     ;       XCH     A,direct
0141: C6             XCH     A,@R0     ;       XCH     A,@Ri
0142: C7             XCH     A,@R1     ;       XCH     A,@Ri
0143: C8             XCH     A,R0      ;       XCH     A,Rn
0144: C9             XCH     A,R1      ;       XCH     A,Rn
0145: CA             XCH     A,R2      ;       XCH     A,Rn
0146: CB             XCH     A,R3      ;       XCH     A,Rn
0147: CC             XCH     A,R4      ;       XCH     A,Rn
0148: CD             XCH     A,R5      ;       XCH     A,Rn
0149: CE             XCH     A,R6      ;       XCH     A,Rn
014A: CF             XCH     A,R7      ;       XCH     A,Rn
014B: D0 10          POP     10H       ;       POP     direct
014D: 11 10          ACALL   10H       ;       ACALL   addr11
014F: D2 00          SETB    00H       ;       SETB    bit
0151: D3             SETB    C         ;       SETB    C
0152: D4             DA      A         ;       DA      A
0153: D5 30 BC       DJNZ    30H,loop;       DJNZ    direct,rel
0156: D6             XCHD    A,@R0     ;       XCHD    A,@Ri
0157: D7             XCHD    A,@R1     ;       XCHD    A,@Ri
0158: D8 06          DJNZ    R0,160H   ;       DJNZ    Rn,addr11
015A: D9 04          DJNZ    R1,160H   ;       DJNZ    Rn,addr11
015C: DA 02          DJNZ    R2,160H   ;       DJNZ    Rn,addr11
015E: DB 00          DJNZ    R3,160H   ;       DJNZ    Rn,addr11
0160: DC FE          DJNZ    R4,160H   ;       DJNZ    Rn,addr11
0162: DD FC          DJNZ    R5,160H   ;       DJNZ    Rn,addr11
0164: DE FA          DJNZ    R6,160H   ;       DJNZ    Rn,addr11
0166: DF F8          DJNZ    R7,160H   ;       DJNZ    Rn,addr11
0168: E0             MOVX    A,@DPTR   ;       MOVX    A,@DPTR
0169: 01 10          AJMP    10H       ;       AJMP    addr11
016B: E2             MOVX    A,@R0     ;       MOVX    A,@Ri
016C: E3             MOVX    A,@R1     ;       MOVX    A,@Ri
016D: E4             CLR     A         ;       CLR     A
016E: E5 10          MOV     A,10H     ;       MOV     A,direct
0170: E6             MOV     A,@R0     ;       MOV     A,@Ri
0171: E7             MOV     A,@R1     ;       MOV     A,@Ri
0172: E8             MOV     A,R0      ;       MOV     A,@Rn
0173: E9             MOV     A,R1      ;       MOV     A,@Rn
0174: EA             MOV     A,R2      ;       MOV     A,@Rn
0175: EB             MOV     A,R3      ;       MOV     A,@Rn
0176: EC             MOV     A,R4      ;       MOV     A,@Rn
0177: ED             MOV     A,R5      ;       MOV     A,@Rn
0178: EE             MOV     A,R6      ;       MOV     A,@Rn
0179: EF             MOV     A,R7      ;       MOV     A,@Rn
017A: F0             MOVX    @DPTR,A   ;       MOVX    @DPTR,A
017B: 11 10          ACALL   10H       ;       ACALL   addr11
017D: F2             MOVX    @R0,A     ;       MOVX    @Ri,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –1 เมื่อเรียกใช้โปรแกรม ECOM51

```
017E: F3          MOVX    @R1,A   ;       MOVX    @Ri,A
017F: F4          CPL     A       ;       CPL     A
0180: F5 10       MOV     10H,A   ;       MOV     direct,A
0182: F6          MOV     @R0,A   ;       MOV     @Ri'',A'
0183: F7          MOV     @R1,A   ;       MOV     @Ri'',A'
0184: F8          MOV     R0,A    ;       MOV     Rn,A
0185: F9          MOV     R1,A    ;       MOV     Rn,A
0186: FA          MOV     R2,A    ;       MOV     Rn,A
0187: FB          MOV     R3,A    ;       MOV     Rn,A
0188: FC          MOV     R4,A    ;       MOV     Rn,A
0189: FD          MOV     R5,A    ;       MOV     Rn,A
018A: FE          MOV     R6,A    ;       MOV     Rn,A
018B: FF          MOV     R7,A    ;       mOV     Rn,A
018C: 82 00       ANL     C,00H   ;       ANL     C,!bit
018E: A0 01       ORL     C,/01H  ;       ORL     C,/bit
0190:             END             ;       END
```

รูปที่ 4.4 (ต่อ) ผลการทดสอบพารามิเตอร์ –l เมื่อเรียกใช้โปรแกรม ECOM51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

โปรแกรมแปลภาษาเป็นโปรแกรมที่ทำหน้าที่แปลจากโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ไปเป็นรหัสคำสั่ง (Opcode) โดยใช้หลักการของคอมไพเลอร์ในการเขียนโปรแกรม โดยจะทำการแยกโปรแกรมออกเป็นส่วนๆ ประกอบด้วยโปรแกรมที่ทำหน้าที่วิเคราะห์ภาษาต้นแบบ, โปรแกรมที่ทำหน้าที่ตรวจสอบวากยสัมพันธ์, โปรแกรมที่ทำหน้าที่ตรวจสอบความหมาย, โปรแกรมที่ทำหน้าที่เปลี่ยนแปลงรหัส และโปรแกรมที่ทำหน้าที่จัดการตารางสัญลักษณ์ เมื่อนำซอร์สโปรแกรมที่มีนามสกุล ASM มาทำการแปลแล้วจะได้ไฟล์ประเภท Hex File ที่มีชื่อเดียวกับซอร์สโปรแกรมแต่มีนามสกุลเป็น HEX เกิดขึ้น และถ้ามีการใช้พารามิเตอร์ –l จะได้ Listing File ที่มีนามสกุลเป็น LST เกิดขึ้น

5.2 ปัญหา และแนวทางแก้ไข

1) ปัญหา โปรแกรมมีขนาดใหญ่ทำให้เกิดความผิดพลาดได้ง่าย

แนวทางแก้ไข แยกโปรแกรมออกเป็นโปรแกรมย่อยๆแล้วจึงนำมารวมกันภายหลัง

2) ปัญหา การตรวจสอบผลการทำงานของโปรแกรมว่าถูกต้องหรือไม่ทำได้ยาก

แนวทางแก้ไข นำเอาโปรแกรมที่มีความสามารถในการเปรียบเทียบไฟล์มาใช้ในการเปรียบเทียบผลการทำงานของไฟล์ที่ทำการแปลด้วย SXA51 และไฟล์ที่ทำการแปลด้วยตัวแปลภาษา

3) ปัญหา การแยกโปรแกรมออกเป็นโปรแกรมย่อยๆ ให้ผู้เขียนโปรแกรมแต่ละคนเขียนมีการส่งค่าของแต่ละโปรแกรมต่อเนื่องกัน ทำให้ทดสอบการทำงานทั้งหมดทำได้ยาก เพราะต้องนำเอาโปรแกรมมารวมกัน

แนวทางแก้ไข ให้ผู้เขียนแต่ละคนทำการกำหนดว่าต้องการค่าอะไรบ้าง ทำการทดลองแล้วมาปรับโปรแกรมให้มีค่าตามที่ต้องการ

4) ปัญหา ขั้นตอนการคอมไพเลอร์บางขั้นตอนไม่สามารถทำได้

แนวทางแก้ไข บางขั้นตอนสามารถข้ามไปได้ ปรึกษาอาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ปัญหา ขั้นตอนของคอมไพเลอร์สามารถเขียนได้หลายแบบ

แนวทางแก้ไข เขียนโปรแกรมตามความถนัดของผู้เขียนแต่ควรให้ถูกต้องตามหลักที่ใช้ในการอ้างอิง

5.3 แนวทางในการพัฒนา

1) สามารถใช้เป็นโปรแกรมต้นแบบเพื่อใช้ในการพัฒนาโปรแกรมแปลภาษาให้สามารถทำการแปลภาษาของไมโครคอนโทรลเลอร์ตระกูลอื่นๆ ได้

2) เขียนโปรแกรมจำลองการทำงานของไมโครคอนโทรลเลอร์ MCS-51 ควบคู่ไปด้วย

3) ปรับปรุงอัลกอริทึมให้มีความซับซ้อนน้อยลงเมื่อต้องการพัฒนาโปรแกรมที่ดี

4) ทดสอบการทำงานให้มากและมีรูปแบบที่หลากหลาย เพื่อตรวจสอบความผิดพลาดของโปรแกรม และความถูกต้องของโปรแกรม

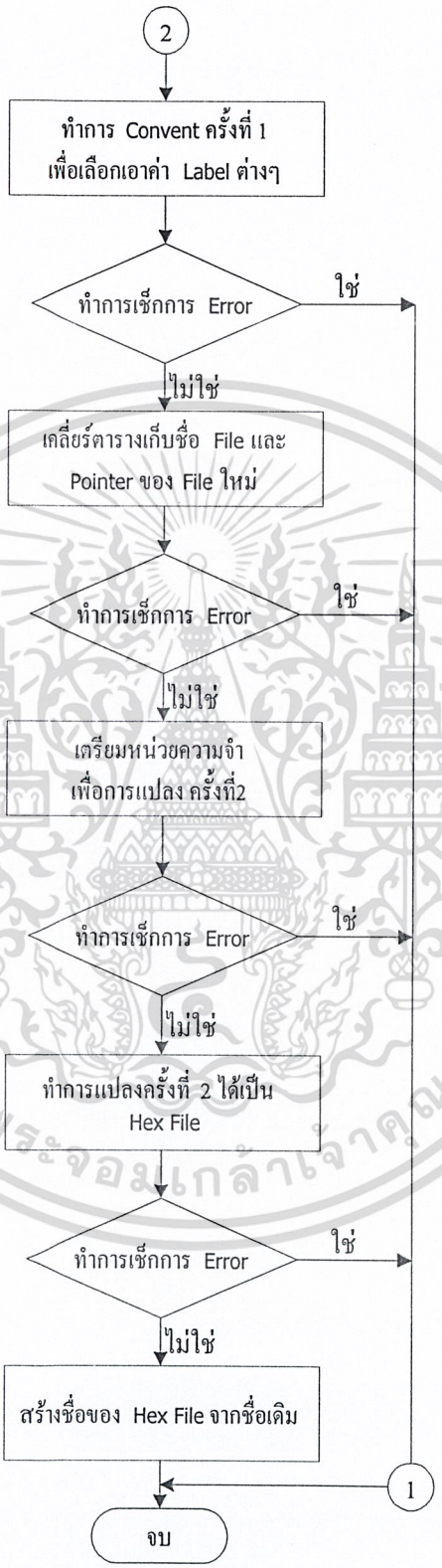
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
ผังการทำงานและโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

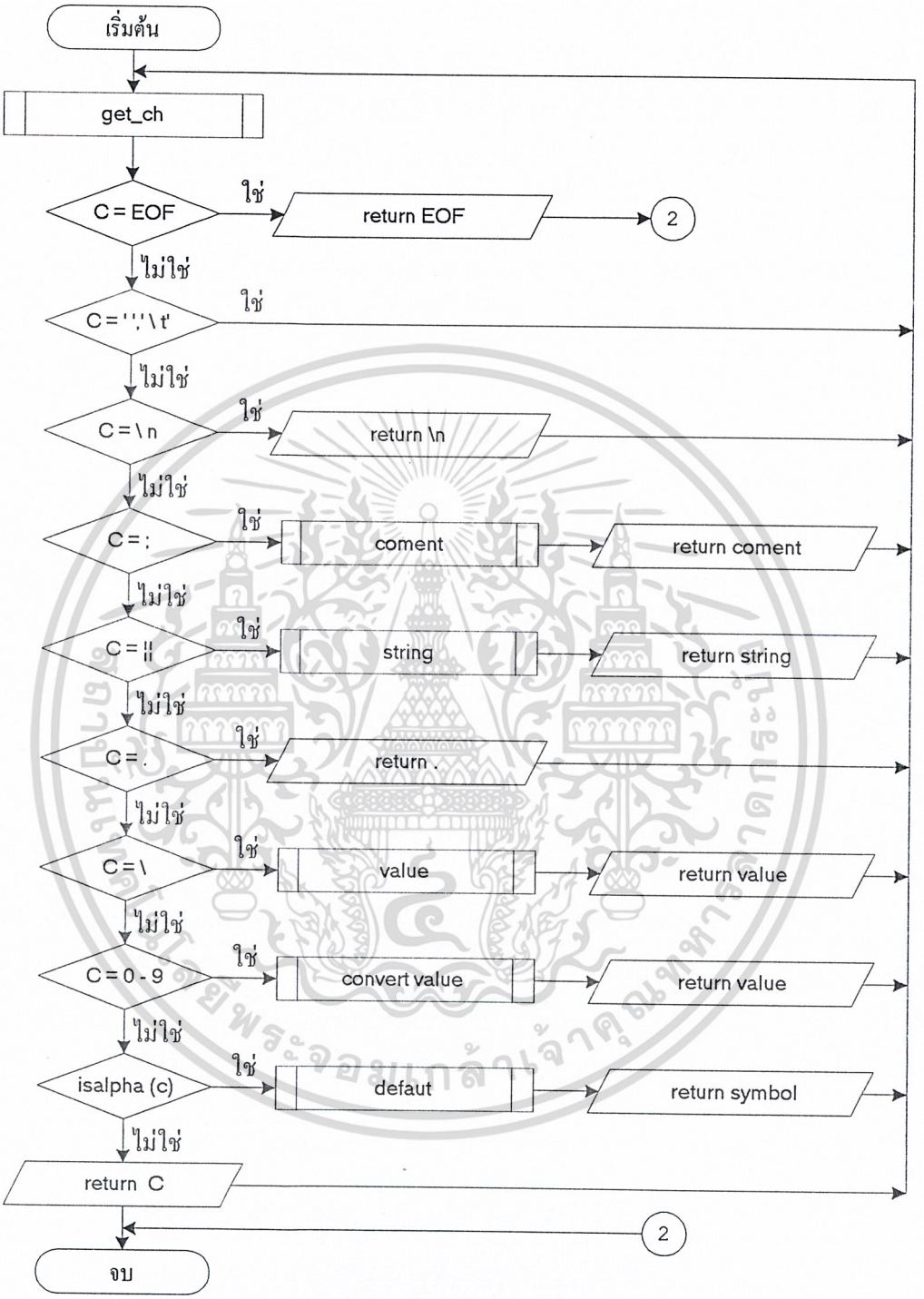
รูปที่ ก.1 ผังงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.1 (ต่อ) ผังงานของโปรแกรมหลัก



รูปที่ ก.2 ผังงานของโปรแกรม Lexer.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*
 * FILE: MAIN.C
 * ไฟล์ศูนย์กลางในการจัดการข้อความตรวจสอบประโยคคำสั่งในเบื้องต้น
 * ที่มีการเรียกใช้งาน และค่าของพารามิเตอร์ต่างๆ
 * และการทำงานนั้นทำ PASS1 ก่อนก่อนทำ PASS 2
 */
#include <stdio.h>
#include <setjmp.h>
extern int lineno;
extern int pass,fatal;
extern unsigned long lc;
jmp_buf main_env;
char *asmfile;
int dashl=0;
FILE *listing;

char *checkext(s,e)
char *s,*e;
{
      register char *ps = s, *pe = e;
      // ตรวจสอบแฟ้มข้อมูลว่าเป็น FILE.ASM หรือไม่
      // โดยการใช้ตัวแปร s และ e ในการตรวจสอบ
      while( *ps ) ps++;          // ถ้าไม่ใช่ FILE.ASM จะได้ค่า NULL กลับมา
      while( *pe ) pe++;
      for( ; ps>=s && pe>=e && *ps == *pe; ps--, pe-- )
            if( pe == e ) return(ps);
      return(NULL);
}
main(argc,argv) /* ตรวจสอบค่าคำสั่งพารามิเตอร์*/
char *argv[];
{
      FILE *fin;
      char *p,*dashF=NULL, *dashA=NULL;
      char hexfile[100];
      char lstfile[100];
      int i;
      if( argc < 2 ) { /* กรณีที่เรียกใช้โปรแกรมแต่ไม่ได้มี FILE.ASM ตามมาด้วย */
            fprintf(stderr,"Usage: %s [-l]  infile.asm\n",argv[0]);
            emitusage();
            exit(1);
      }
      for(i=1; i<argc; i++ ) {
            if( argv[i][0] != '-' ) break;
            if( argv[i][1] == 'l' ) dashl = 1;   /* มีการใช้พารามิเตอร์ -l */
            else {
                  fprintf(stderr,"Duplicate or unknown flag.\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

```
                exit(1);
            }
        }
        if( i == argc ) { /* ตรวจสอบ input file */
            fprintf(stderr,"Missing input file.\n");
            exit(1);
        }
        if( (p=checkext(argv[i],".asm")) == NULL ) { /*ชื่อไฟล์ที่จะทำการแปล
ต้องมีนามสกุลเป็น ".ASM" */
            fprintf(stderr,"Input file \"%s\" must end with .asm\n",
                        argv[i]);
            exit(1);
        }
        asmfile = argv[i];
        if( (fin = freopen(argv[i],"r",stdin)) == NULL ) { /* ตรวจสอบการ
เปิดไฟล์ ".ASM" */
            fprintf(stderr,"Cannot open input file: %s\n",argv[i]);
            exit(1);
        }
        if( dashl ) { /* ทำการสร้างไฟล์รองรับกับคำสั่งที่มีการใช้ พารามิเตอร์ -l */
            strcpy(lstfile,argv[i]);
            strcpy(lstfile+(p-argv[i]),".lst");
            listing = fopen(lstfile,"w");
            if( listing == NULL ) { /* ตรวจสอบการสร้างไฟล์ */
                fprintf(stderr,"Cannot open file: %s for
writting.\n",
                        lstfile);
                exit(1);
            }
        }
        strcpy(hexfile,argv[i]);
        strcpy(hexfile+(p-argv[i]),".hex"); /* กำหนด "FILE.HEX" */
        emitopen(hexfile,dashF,dashA);
        syminit();
        fatal = 0;
        lineno = 1;
        pass=0;
        lc = 0x0000;
        if( setjmp(main_env) ) {
            fclose(fin);
            emitclose();
            unlink(hexfile);
            exit(1);
        }
        /*
        ** P A S S    1
        */
        yyparse();
        if( fatal ) longjmp(main_env,1);
        rewind(fin);
        lineno = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pass++;
        lc = 0x0000;
        emitaddr(lc);

        /*
        ** P A S S    2
        */
        yyparse();
        emitclose();
        fclose(fin);
        if( dashl )
              fclose(listing);
        exit(0);
}
```

รูปที่ ก.3 โปรแกรม MAIN.C

```
/*
 * FILE: LEXER.C
 * แฟ้มนี้บรรจุ Tokenizer สำหรับขั้นตอนของการ Assembler
 * เมื่อขั้นตอนของ yacc เริ่มต้นขั้นตอนของ Lexer ถูกเรียกว่า yylex ()
 * เพื่อรวบรวมรายรายละเอียดต่างๆ Record ของผู้ใช้ เส้นต้นกำเนิดถูกเก็บ
 * สิ่งนี้ทำโดยการเพิ่ม get_ch ()และ unget_ch () ซึ่งจะมีการคืนค่าต่างๆ
 * หรือมีการเก็บเอาตัวอักษรต่างๆ เข้าไปในอะเรย์
 */
#include <stdio.h>
#include <ctype.h>
#include "as31.h"
extern union ystack yylval;
extern int pass;
struct symbol *looksym();
struct opcode *lookop();
char *malloc();
int lineno;
static char line[100],*lineptr=line;
/*
 * get_ch:
 * ทำหน้าที่อ่านอักขระตัวต่อไป จากแฟ้มข้อมูลที่เก็บโปรแกรมต้นแบบ
 * แล้วส่งอักขระที่อ่านได้กลับไป ในขณะที่อ่านได้จะมีการจำตำแหน่งสดมภ์
 * และบรรทัดของอักขระที่อ่านได้
 */get_ch()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        register int c;
        c = getchar();
        if( c != EOF && lineptr - line < sizeof(line) )
                *lineptr++ = c;
        return(c);
}
/*
 * unget_ch:
 * มีอาร์กิวเมนต์หนึ่งตัวคือ C ฟังก์ชันนี้ทำหน้าที่เอาอักขระที่เก็บในตัวแปร c
 * ใส่กลับเข้าไปในแฟ้มข้อมูลที่เก็บโปรแกรมต้นแบบ เพื่อเป็นการ Retract
 * พร้อมทั้งแก้ไขตำแหน่งบรรทัดและสดมภ์ให้ถูกต้อง
 */
unget_ch(c)
int c;
{
        ungetc(c,stdin);
        if( lineptr > line )
                lineptr--;
}
/* ---------------------------------------------------------------
 *              yylval.value
 *                      ตัวเลข (พื้นฐานใดๆ)
 *                      สตริง(in pass 1).
 *                      bit position.0, .1, .2, ...
 *              yylval.str
 *                      สตริง (in pass 2).
 *                      '\n' (both passes).
 *              yylval.sym
 *                      สัญลักษณที่ผู้ใช้กำหนดเอง
 *              yylval.op
 *              คำสงวนไว้ (Opcode/Directive/Misc.)
 *      ไม่ใช่ส่วน Field อื่นๆ ใน yylval ที่ถูกใช้โดย yylex ().
 *              Characters that do not have an attribute do
 *              not set anything in the yylval variable.
 */
yylex()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      static nl_flag=0;  /* ข่าวสารข้อผิดพลาด */
      register int c;
      char buf[120];     /* บัพเฟอร์ชั่วคราว */
      char *p;           /* ตัวชี้ทั่วไป */
      struct symbol *sym; /* Symbol */
      struct opcode *op;   /* Opcode */
int octal=0,hex=0,decimal=0,binary=0; /*เพื่อใช้ในกรณีที่รับเอาค่าตัวเลขเข้ามา */
      register long value = 0;
      if( nl_flag ) {
            nl_flag = 0;
            lineno++;
      }
for(;;) {
      c = get_ch();
      switch(c) {
      case EOF: return(EOF); /*สิ้นสุดไฟล์ */
      case ' ':                  /*ช่องว่าง*/
      case '\t':                 /* TAB */
            break;
      case '\n':                 /* ขึ้นบรรทัดใหม่ */
            nl_flag = 1;
            yylval.str = line;
            *lineptr = '\0';
            lineptr = line;
            return('\n');
      case ';':                  /* หมายเหตุ ( COMMENT) */
            while((c=get_ch()) != EOF && c!='\n');
            nl_flag= 1;
            yylval.str = line;
            *lineptr = '\0';
            lineptr = line;
            return(c);
      case '"': /* STRING */
            p = buf;
            while((c=get_ch()) != EOF && c!='"' && c!='\n') {
                  if( c == '\\' ) {
                        switch(c=get_ch()) {
                        case 'n': c = '\n'; break;
                        case 'r': c = '\r'; break;
                        case 't': c = '\t'; break;
                        case 'b': c = '\b'; break;
                        case '"': c = '"'; break;
                        case '\\': c = '\\'; break;
                        default:
                          error("Invalid escape character: \\%c",c);
                          break;
                        }
                  }
                  if( p-buf<sizeof(buf)-1 )
                        *p++ = c;
                  else {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        error("String constant longer than %d bytes",
                                sizeof(buf));
                  }
            }
            *p = '\0';
            if( c == '\n' || c == EOF ) {
                  error("String terminated improperly.");
                  unget_ch(c);
            }
            if(pass1)
                  yylval.value = strlen(buf);
            else {
                  if( (p = malloc(strlen(buf)+1)) == NULL ) /*ทำการ
จองหน่วยความจำเพื่อการใช้งาน*/
                  error("Cannot allocate %d bytes",strlen(buf)+1);
                  strcpy(p,buf);
                  yylval.str = p;
            }
            return(STRING);
      case '.': /* BITPOS */
            if( (c=get_ch())>='0' && c<='7' ) {
                  yylval.value = c-'0';
                  return(BITPOS);
            }
            unget_ch(c);
            return('.');
      case '\'':
            c = get_ch();
            if( c=='\\' ) {
                  switch(c=get_ch()) {
                  case 'n': c = '\n'; break;
                  case 'r': c = '\r'; break;
                  case 't': c = '\t'; break;
                  case 'b': c = '\b'; break;
                  case '\\': c = '\\'; break;
                  case '\'': c = '\''; break;
                  default:
                        error("Invalid escape character: \\%c",c);
                  }
            }
            if( get_ch() != '\'' )
                  error("Missing quote in character constant");
            yylval.value = c;
            return(VALUE);
      case '0':   /* วิเคราะห์คำในไวยากรณ์กรณีตัวเลข    */
      case '1':   /* สามารถถูกปฏิบัติตามโดยกรณี a:     */
      case '2':   /*     'b','B' - Binary               */
      case '3':   /*     'h','H' - Hex                  */
      case '4':   /*     'd','D' - Decimal              */
      case '5':   /*     'o','O' - Octal                */
      case '6':   /* ตัวเลขต้องเริ่มกับกรณีหลัก   */
      case '7':   /* ตัวเลขสามารถ Preceeded ได้ */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      case '8':   /*    0x     - Hex,       0b      - Binary */
      case '9':   /*    0      - Octal                 */
            p = buf;
            do {
                  if( p-buf<sizeof(buf)-1 )
                        *p++ = c;
                  c = get_ch();
      } while( c=='H' || c=='h' || c=='O' || c=='o' ||/* ตรวจสอบ
ตัวอักษรที่ต่อท้ายตัวเลข */
                        c=='x' || c=='X' || isxdigit(c) );
            unget_ch(c);
            *p = '\0';
            /* Check any preceeding chars */
      if( buf[0]=='0' && (buf[1]=='x' || buf[1]=='X') ) { /* HEX */
                        hex++;
                        buf[1] = '0';
            } else if( buf[0]=='0' &&
                  (buf[1]=='b' || buf[1]=='B') ) { /* binary */
                        binary++;
                        buf[1] = '0';
                  }
            else if( buf[0]=='0' ) octal++; /* Octal++; */
            /* check any trailing chars */
            c = *(p-1);
            if( !hex && (c=='b' || c=='B') ) /* Binary */
                  { binary++; *(p-1) = '\0'; }
            else if( c=='H' || c=='h' )          /* Hex++ */
                  { hex++; *(p-1) = '\0'; }
            else if( !hex && (c=='D' || c=='d') ) /* Decimal */
                  { decimal++; *(p-1) = '\0'; }
            else if( c=='O' || c=='o' ) /* Octal++ */
                  { octal++; *(p-1) = '\0'; }
            else if( !hex && !octal && !binary) decimal++; /*
Decimal */
            if( binary ) { /* Binary */
                  for(p=buf; *p; p++ ) {
                        if( *p=='1' ) value = value<<1 + 1;
                        else if( *p=='0' ) value = value<<1;
                        else
                          error("Invalid binary digit: %c",*p);
                  }
                  yylval.value = value;
                  return(VALUE);
            }
            if( hex ) { /* Hex  */
                  for(p=buf; *p; p++ ) {
                        value <<= 4;
                        if( isdigit(*p) )
                              value += *p-'0';
                        else if( *p>='a' && *p<='f' )
                              value += *p-'a'+ 10;
                        else if( *p>='A' && *p<='F' )
                              value += *p-'A'+ 10;
                        else
                          error("Invalid hex digit: %c",*p);
                  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                yylval.value = value;
                return(VALUE);
            }
            if( decimal ) { /* Decimal */
                for(p=buf; *p; p++ ) {
                    if( isdigit(*p) )
                        value = value*10 + *p-'0';
                    else
                       error("Invalid decimal digit: %c",*p);
                }
                yylval.value = value;
                return(VALUE);
            }
            if( octal ) { /* Octal */
                for(p=buf; *p; p++ ) {
                    if( *p>='0' && *p<='7' )
                        value = value<<3 + *p-'0';
                    else
                       error("Invalid octal digit: %c",*p);
                }
                yylval.value = value;
                return(VALUE);
            }
        default: /* ตัวอักษรใดๆ ก็ตามที่ไม่ตรงกับเงื่อนไขข้างบน*/
            if( isalpha(c) || c=='_' ) {
                p = buf;
                do {
                    if( p-buf<sizeof(buf)-1 )
                        *p++ = c;
                    c = get_ch();
                } while( isalnum(c) || c=='_' );
                *p = '\0';
                unget_ch(c);
                if( op = lookop(buf) ) {
                    yylval.op = op;
                    return(op->type);
                }
                sym = looksym(buf);
                yylval.sym = sym;
                return(SYMBOL);
            } else
                return(c);
        } /* switch */
    } /* for */
    } /* yylex */
```

รูปที่ ก.4 โปรแกรม Lexer.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
* FILE: EMITTER.C
* แฟ้มนี้บรรจุรหัสต่างๆ เพื่อไว้ในขั้นตอนของ Code to Generate เช่น
* รูปแบบรหัสวัตถุข้อกำหนดต่างๆ การสนับสนุนชนิดจำนวนมากมายของผลลัพธ์
* การจัดรูปแบบภายในที่ทำงานด้วยกัน...
*/
#include <stdio.h>
/*
 * DECLARE your own open(), close(), addr(), and byte() routines
here.
 */
static int open1(), close1(), addr1(), byte1();
static int open2(), close2(), addr2(), byte2();
static int open3(), close3(), addr3(), byte3();
/*
* เพิ่มรายการที่อยู่ในตารางให้ไปอยู่ในรูปของ เอาท์พุท
* ให้รูปแบบ ซึ่งถูกเจาะจงกับพารามิเตอร์ –L
*/
unsigned int c[17];
int w,z,upbit,downbit,dd ;
static int format;
static struct {
      char *name;
      int (*e_open)();
      int (*e_close)();
      int (*e_addr)();
      int (*e_byte)();
} formtab[] = {
      { "tdr",    open1, close1, addr1, byte1 },
      { "byte",   open2, close2, addr2, byte2 },
      { "od",     open3, close3, addr3, byte3 },
      { "srec2", open4, close4, addr4, byte4 }
};
#define FORMTABSIZE     (sizeof(formtab)/sizeof(formtab[0]))
emitusage() /*สำหรับพารามิเตอร์ */
{
      int i;
      fprintf(stderr, "\tfmt is one of:");
      for(i=0; i<FORMTABSIZE; ) {
            fprintf(stderr, "%s", formtab[i].name);
            if( ++i < FORMTABSIZE)
                  fprintf(stderr, ", ");
      }
      fprintf(stderr, ".\n");
}
emitopen(file,ftype,arg)
char *file,*ftype,*arg;
{
      int i;
      if( ftype ) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            for(i=0; i<FORMTABSIZE; i++ ) {
                  if( !strcmp(formtab[i].name,ftype) ) {
                        format = i;
                        (*formtab[format].e_open)(file,ftype,arg);
                        return;
                  }
            }
            fprintf(stderr, "warning: no format \"%s\", using
\"%s\"\n",
                  ftype, formtab[0].name);
      }
      /*
       * 0th entry is the default format type
       */
      format = 0;
      (*formtab[format].e_open)(file,ftype,arg);
}

emitclose() /* ปิดไฟล์ */
{
      (*formtab[format].e_close)();
}
emitaddr(a)
unsigned long int a;
{
      (*formtab[format].e_addr)(a);
}
emitbyte(b)
int b;
{
      (*formtab[format].e_byte)(b);
}
/*  รูปแบบแฟ้มข้อมูล:
 * รูปแบบแฟ้มแต่ละอันต้องจำกัดดังต่อไปนี้:
 * open() ถูกเรียกใช้เป็นลำดับแรกก่อนที่จะเรียกใช้ฟังก์ชั่นอื่นต่อไป
 * ซึ่งเป็นส่วนหนึ่งของชื่อแฟ้มและรูปแบบอย่างจำเพาะเจาะจง
 * close() ถูกเรียกใช้เมื่อฟังก์ชั่น  emit_byte() จบการทำงาน
 * addr () ถูกเรียกใช้เมื่อมี Address ใหม่เกิดขึ้นในการทำงาน (ie. .org, .skip)
 * ซึ่ง Routine จะถูกเรียกเมื่อ ตัวนับตำแหน่ง ( Location Counter ) ถูกตั้งค่า
 * ให้เป็น 0 ที่จุดเริ่มต้นของการเริ่ม Address ใหม่
 * byte()  ถูกเรียกใช้เมื่อไบท์ถูกนำไปเป็นผลลัพธ์ ( Output ) ในการทำงาน
 * รูปแบบ สำหรับระบบ  Intel Hex File
 * ระบบทำการสร้างแฟ้มสคริปที่อ่านได้โดยดีบักเกอร์
 * [addr] : : [byte] [byte] ..
*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
static unsigned long addr;
static FILE *fout;
static long int offset;
static int newaddr;
static int pos= -666; /* เพื่อใช้ในการตรวจสอบค่า */
static open1(file,ftype,arg)
char *file, *ftype, *arg;
{     fout = fopen(file,"w");
      if( fout == NULL ) {
            fprintf(stderr,"Cannot open %s for writting.\n",file);
            exit(1);
      }
      if( arg ) {
            offset = atoi(arg);
      } else
            offset = 0X0000;
}
static close1()
{
      int u ;
      if( pos != 15 ) /* ทำการขึ้นบรรทัดใหม่ให้อยู่ในรูปแบบของ INTEL HEX FILE
*/
      {     fprintf(fout,"\n:%02X",w); /* ไบท์ข้อมูลในเรคคอร์ด */
            for(u=0;u<=w;u++)
            {
                  if (u!=0)   z+=c[u];
                  fprintf(fout,"%02X",c[u]);
            if (u==0) fprintf(fout,"00"); /* ชนิดของเรคคอร์ดข้อมูล */
            }
                  dd=c[0]&0Xffff;
                  downbit = dd&0Xff;  /* แปลง Address  */
                  upbit = dd&0Xff00 ; /* แปลง Address  */
                  upbit>>=8; /*shif bit  */
                  z+=downbit+upbit+w ; /* ผลรวมทั้งหมดของทุกไบท์ */
                  fprintf(fout,"%02X",(~z+1)&0Xff); /*ค่า Checksum */
                  z=w=0;
      }
fprintf(fout,"\n:00000001FF"); /*บรรทัดสุดท้ายในรูปแบบของ INTEL HEX
FILE*/
      fclose(fout);
}
static addr1(a)
unsigned long int a;
{     addr = a; /* กำหนดค่า address */
      newaddr = 1;
}
static byte1(b)
unsigned char b;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  int u ;
      if( newaddr )
      {
            if( pos != -666 )
            {     fprintf(fout,"\n:%02X",w);/* ค่า Address */
                  for(u=0;u<=w;u++)
                  {
                   if (u!=0)   z+=c[u];     /* ชนิดข้องเรคคอร์ดของข้อมูล */
                  fprintf(fout,"%02X",c[u]);      /* 00 เป็นเรคคอร์ดข้อมูล */
      if (u==0) fprintf(fout,"00"); /* 01 เป็นเรคคอร์ดสุดท้ายของไฟล์ */
                  }
                  dd=c[0]&0Xffff;
                  downbit = dd&0Xff; /* แปลง Address   */
                  upbit = dd&0Xff00 ; /* แปลง Address   */
                  upbit>>=8; /*shif bit   */
                  z+=downbit+upbit+w ; /* ผลรวมทั้งหมดของทุกไบท์ */
   fprintf(fout,"%02X",(~z+1)&0Xff);
/* ค่า Checksum ( รูปแบบของ Intel Hex File )*/
                  z=w=0;            }
            newaddr = 0;
            pos = 15;
            c[0] = addr+offset ;
      }
      else if( pos == 15 )
      {
            c[0] = addr+offset ;
                  for(u=0;u<w;u++)
                  fprintf(fout,"%02X",c[u]);
                  w=0;
      }
      c[++w] = (b&0Xff) ;
      if( pos-- == 0 ) {
            pos = 15;
            fprintf(fout,"\n:%02X",w);
            for(u=0;u<=w;u++)
            {
                  if (u!=0)   z+=c[u];
                  fprintf(fout,"%02X",c[u]);
                  if (u==0) fprintf(fout,"00");
            }
                 dd=c[0]&0Xffff;
                 downbit = dd&0Xff;
                 upbit = dd&0Xff00 ;
                 upbit>>=8;
                 z+=downbit+upbit+w ;
                 fprintf(fout,"%02X",(~z+1)&0Xff);
                 z=w=0;
      }
      addr += 1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*  รูปแบบของ "byte"
 *  ลักษณะเหมือน "tdr"แต่แต่ละไบท์อยู่บนเส้นโดยตัวเอง
 *  สิ่งนี้ดีสำหรับดีบักคือ -A ไม่ถูกใช้
 */
static open2(file,ftype,arg)
char *file, *ftype, *arg;
{
      fout = fopen(file,"w");
      if( fout == NULL ) {
            fprintf(stderr,"Cannot open %s for writting.\n",file);
            exit(1);
      }
}
static close2()
{
      fclose(fout);
}
static addr2(a)
unsigned long int a;
{
      addr = a;
}
static byte2(b)
unsigned char b;
{
      fprintf(fout,"%04X: %02X\n", addr, b&0Xff );
      addr += 1;
}
/* " od " รูปแบบนี้แสดง 16 ไบท์ต่อเส้นกับตำแหน่ง Address
 * มันยังรวมถึง ascii บนด้านข้างเดียวกัน
 * รูปแบบซึ่งคล้ายกับ od (1) ซึ่งเป็นโปรแกรมที่อยู่ภายใต้ระบบปฏิบัติการยูนิกซ์
 */
static int pos3;
static unsigned char buf[16];
static unsigned long saveaddr;
static open3(file,ftype,arg)
char *file, *arg;
{
      fout = fopen(file,"w");
      if( fout == NULL ) {
            fprintf(stderr,"Cannot open %s for writting.\n",file);
            exit(1);
      }
}
static close3()
{
      dumpline(saveaddr,buf,pos3-1);
      fclose(fout);
}
static addr3(a)
unsigned long int a;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        newaddr = 1;
        addr = a;
}
static byte3(b)
unsigned char b;
{
        if( newaddr ) {
                dumpline(saveaddr,buf,pos3-1);
                pos3 = 0;
                newaddr = 0;
                saveaddr = addr;
        } else if( pos3 == 16 ) {
                dumpline(saveaddr,buf,pos3-1);
                pos3 = 0;
                saveaddr = addr;
        }
        buf[pos3++] = b & 0X00ff;

        addr += 1;
}
dumpline(a,b,len)
unsigned long a;
unsigned char *b;
int len;
{
        int i;
        if(len <= 0 ) return;
        fprintf(fout,"%04X: ",a);
        for(i=0; i<8; i++ ) {
                if( i <= len )
                        fprintf(fout,"%02X ",b[i]);
                else
                        fprintf(fout,"   ");
        }
        fprintf(fout,"- ");

        for(i=8; i<16; i++ ) {
                if( i <= len )
                        fprintf(fout,"%02X ",b[i]);
                else
                        fprintf(fout,"   ");
        }
        fprintf(fout,"   ");

        for(i=0; i<16; i++ ) {
                if( i <= len )
                        fprintf(fout,"%c",
                                (b[i]>=' ' && b[i]<='~') ? b[i] : '.' );
                else
                        break;
        }
        fprintf(fout,"\n");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ลอกเลียนแบบ ดัดแปลง ก่อนได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.5 โปรแกรม EMITTER.C

```
*   FILE: SYMBOL.C
*   ไฟล์นี้ภายในประกอบไปด้วยตารางสัญลักษณ์ การค้นหาค่าในตารางสัญลักษณ์
*   การแทรก Routines ที่เกี่ยวข้องกับสัญลักษณ์ที่ต่างๆ
*   คำสงวน (Instructions/Directives) การค้นหา Routinesต่างๆ ที่ถูกกำหนดไว้
*   ตาราง Opcode สำหรับ Instructions ที่ถูกกำหนดให้อยู่ในไฟล์นี้
*/
#include "as31.h"
#define NULL            (0)
#define B(a)            (0xF0+(a))
#define ACC(a)          (0xE0+(a))
#define PSW(a)          (0xD0+(a))
#define T2CON(a)        (0xC8+(a))
#define IP(a)           (0xB8+(a))
#define P3(a)           (0xB0+(a))
#define IE(a)           (0xA8+(a))
#define P2(a)           (0xA0+(a))
#define SCON(a)         (0x98+(a))
#define P1(a)           (0x90+(a))
#define TCON(a)         (0x88+(a))
#define P0(a)           (0x80+(a))
/*  สัญลักษณ์เหล่านี้ไม่ใช่ Reserved Keywords.
 * อะเรย์นี้บรรจุรายการที่มีอยู่ในตารางสัญลักษณ์เริ่มต้น
*สำหรับตารางสัญลักษณ์ สัญลักษณ์ต่างๆ นี้เป็นสัญลักษณ์พื้นฐานที่มีการแปลความหมายโดย
*ง่าย ฟังก์ชัน syminit () ทำหน้าที่ใส่รายการเหล่านี้เข้าไปในตารางสัญลักษณ์ในหน่วยความจำ
 */
static struct symbol sinit[] = {
        { "AC",     LABEL,          PSW(6),NULL },
        { "ACC",    LABEL,          ACC(0),NULL },
        { "B",      LABEL,          B(0), NULL },
        { "CY",     LABEL,          PSW(7),NULL },
        { "DPH",    LABEL,          0x83, NULL },
        { "DPL",    LABEL,          0x82, NULL },
        { "EA",     LABEL,          IE(7),NULL },
        { "ES",     LABEL,          IE(4),NULL },
        { "ET0",    LABEL,          IE(1),NULL },
        { "ET1",    LABEL,          IE(3),NULL },
        { "ET2",    LABEL,          IE(5),NULL },
        { "EX0",    LABEL,          IE(0),NULL },
        { "EX1",    LABEL,          IE(2),NULL },
        { "EXEN2",  LABEL,          T2CON(3),NULL },
        { "EXF2",   LABEL,          T2CON(6),NULL },
        { "F0",     LABEL,          PSW(5),NULL },
        { "IE",     LABEL,          IE(0),NULL },
        { "IE0",    LABEL,          TCON(1),NULL },
        { "IE1",    LABEL,          TCON(3),NULL },
        { "IP",     LABEL,          IP(0),NULL },
        { "IT0",    LABEL,          TCON(0),NULL },
        { "IT1",    LABEL,          TCON(2),NULL },
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        { "OV",      LABEL,              PSW(2),NULL },
        { "P",       LABEL,              PSW(0),NULL },
        { "P0",      LABEL,              P0(0),NULL },
        { "P1",      LABEL,              P1(0),NULL },
        { "P2",      LABEL,              P2(0),NULL },
        { "P3",      LABEL,              P3(0),NULL },
        { "PCON",    LABEL,              0x87, NULL },
        { "PS",      LABEL,              IP(4),NULL },
        { "PSW",     LABEL,              PSW(0),NULL },
        { "PT0",     LABEL,              IP(1),NULL },
        { "PT1",     LABEL,              IP(3),NULL },
        { "PT2",     LABEL,              IP(5),NULL },
        { "PX0",     LABEL,              IP(0),NULL },
        { "PX1",     LABEL,              IP(2),NULL },
        { "RB8",     LABEL,              SCON(2),NULL },
        { "RCAP2H",  LABEL,              0xCB, NULL },
        { "RCAP2L",  LABEL,              0xCA, NULL },
        { "RCLK",    LABEL,              T2CON(5),NULL },
        { "REN",     LABEL,              SCON(4),NULL },
        { "RI",      LABEL,              SCON(0),NULL },
        { "RL2",     LABEL,              T2CON(0),NULL },
        { "RS0",     LABEL,              PSW(3),NULL },
        { "RS1",     LABEL,              PSW(4),NULL },
        { "SBUF",    LABEL,              0x99, NULL },
        { "SCON",    LABEL,              SCON(0),NULL },
        { "SM0",     LABEL,              SCON(7),NULL },
        { "SM1",     LABEL,              SCON(6),NULL },
        { "SM2",     LABEL,              SCON(5),NULL },
        { "SP",      LABEL,              0x81, NULL },
        { "T2CON",   LABEL,              T2CON(0),NULL },
        { "TB8",     LABEL,              SCON(3),NULL },
        { "TCLK",    LABEL,              T2CON(4),NULL },
        { "TCON",    LABEL,              TCON(0),NULL },
        { "TF0",     LABEL,              TCON(5),NULL },
        { "TF1",     LABEL,              TCON(7),NULL },
        { "TF2",     LABEL,              T2CON(7),NULL },
        { "TH0",     LABEL,              0x8C, NULL },
        { "TH1",     LABEL,              0x8D, NULL },
        { "TH2",     LABEL,              0xCD, NULL },
        { "TI",      LABEL,              SCON(1),NULL },
        { "TL0",     LABEL,              0x8A, NULL },
        { "TL1",     LABEL,              0x8B, NULL },
        { "TL2",     LABEL,              0xCC, NULL },
        { "TMOD",    LABEL,              0x89, NULL },
        { "TR0",     LABEL,              TCON(4),NULL },
        { "TR1",     LABEL,              TCON(6),NULL },
        { "TR2",     LABEL,              T2CON(2),NULL }
};
#define SINITSIZE (sizeof(sinit)/sizeof(sinit[0]))
```

/* อะเรย์เหล่านี้บรรจุ Opcodes คำสั่งต่างๆ

* คำสั่งของ Opcodes เหล่านี้สำคัญมากสำหรับ

* Fuctioning ที่จะใช้ในการทำการแปล

* เมื่อรูปแบบของโครงสร้างของข้อมูลถูกวิเคราะห์ไวยากรณ์ การวิเคราะห์คำใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* ไวยากรณ์ดัชนีของอะเรย์เหล่านี้โดยจำนวนที่ถูกต้อง และดังเช่น

* รับ Opcode ที่ถูกต้องสำหรับรูปแบบที่เจาะจง*/

```
static unsigned char acall[]= { 0x11 };
static unsigned char add[]=   { 0x28, 0x25, 0x26, 0x24 };
static unsigned char addc[]=  { 0x38, 0x35, 0x36, 0x34 };
static unsigned char ajmp[]=  { 0x01 };
static unsigned char anl[]=   { 0x58, 0x55, 0x56, 0x54, 0x52, 0x53,
0x82, 0xb0 };
static unsigned char cjne[]=  { 0xb5, 0xb4, 0xb8, 0xb6 };
static unsigned char clr[]=   { 0xe4, 0xc3, 0xc2 };
static unsigned char cpl[]=   { 0xf4, 0xb3, 0xb2 };
static unsigned char da[]=    { 0xd4 };
static unsigned char dec[]=   { 0x14, 0x18, 0x15, 0x16 };
static unsigned char div[]=   { 0x84 };
static unsigned char djnz[]=  { 0xd8, 0xd5 };
static unsigned char inc[]=   { 0x04, 0x08, 0x05, 0x06, 0xa3 };
static unsigned char jb[]=    { 0x20 };
static unsigned char jbc[]=   { 0x10 };
static unsigned char jc[]=    { 0x40 };
static unsigned char jmp[]=   { 0x73 };
static unsigned char jnb[]=   { 0x30 };
static unsigned char jnc[]=   { 0x50 };
static unsigned char jnz[]=   { 0x70 };
static unsigned char jz[]=    { 0x60 };
static unsigned char lcall[]= { 0x12 };
static unsigned char ljmp[]=  { 0x02 };
static unsigned char mov[]=   { 0xe8, 0xe5, 0xe6, 0x74, 0xf5, 0x75,
0xf8, 0xa8, 0x78, 0x88, 0x85, 0x86, 0xf6, 0xa6, 0x76, 0x90, 0xa2,
0x92 };
static unsigned char movc[]=  { 0x93, 0x83 };
static unsigned char movx[]=  { 0xe2, 0xe3, 0xe0, 0xf2, 0xf3, 0xf0
};
static unsigned char mul[]=   { 0xa4 };
static unsigned char nop[]=   { 0x00 };
static unsigned char orl[]=   { 0x48, 0x45, 0x46, 0x44, 0x42, 0x43,
0x72, 0xa0 };
static unsigned char pop[]=   { 0xd0 };
static unsigned char push[]=  { 0xc0 };
static unsigned char ret[]=   { 0x22 };
static unsigned char reti[]=  { 0x32 };
static unsigned char rl[]=    { 0x23 };
static unsigned char rlc[]=   { 0x33 };
static unsigned char rr[]=    { 0x03 };
static unsigned char rrc[]=   { 0x13 };
static unsigned char setb[]=  { 0xd3, 0xd2 };
static unsigned char sjmp[]=  { 0x80 };
static unsigned char subb[]=  { 0x98, 0x95, 0x96, 0x94 };
static unsigned char swap[]=  { 0xc4 };
static unsigned char xch[]=   { 0xc8, 0xc5, 0xc6 };
static unsigned char xchd[]=  { 0xd6 };
static unsigned char xrl[]=   { 0x68, 0x65, 0x66, 0x64, 0x62, 0x63
};
```

/*ตารางนี้บรรจุ Opcodes, และสัญลักษณ์

*ส่วน Field ที่สองคือค่าของ Token

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*ถ้ามิใช่สัญลักษณ์ ก็คือ Opcode Field ที่สามจะได้ค่า NULL

*ส่วน Field ที่สามคือ ตัวชี้อะเรย์ของ Opcode ไบท์

*/

```
static struct opcode optable[] = {
      {"a",             A,     NULL         },
      {"ab",            AB,    NULL         },
      {"acall",         ACALL,       acall  },
      {"add",           ADD,   add          },
      {"addc",          ADDC,  addc         },
      {"ajmp",          AJMP,  ajmp         },
      {"anl",           ANL,   anl          },
      {"byte",          D_BYTE,      NULL   },
      {"c",             C,     NULL         },
      {"cjne",          CJNE,  cjne         },
      {"clr",           CLR,   clr          },
      {"cpl",           CPL,   cpl          },
      {"da",            DA,    da           },
      {"dec",           DEC,   dec          },
      {"div",           DIV,   div          },
      {"djnz",          DJNZ,  djnz         },
      {"dptr",          DPTR,  NULL         },
      {"end",           D_END,       NULL   },
      {"equ",           D_EQU,       NULL   },
      {"flag",          D_FLAG,      NULL   },
      {"inc",           INC,   inc          },
      {"jb",            JB,    jb           },
      {"jbc",           JBC,   jbc          },
      {"jc",            JC,    jc           },
      {"jmp",           JMP,   jmp          },
      {"jnb",           JNB,   jnb          },
      {"jnc",           JNC,   jnc          },
      {"jnz",           JNZ,   jnz          },
      {"jz",            JZ,    jz           },
      {"lcall",         LCALL,       lcall  },
      {"ljmp",          LJMP,  ljmp         },
      {"mov",           MOV,   mov          },
      {"movc",          MOVC,  movc         },
      {"movx",          MOVX,  movx         },
      {"mul",           MUL,   mul          },
      {"nop",           NOP,   nop          },
      {"org",           D_ORG,       NULL   },
      {"orl",           ORL,   orl          },
      {"pc",            PC,    NULL         },
      {"pop",           POP,   pop          },
      {"push",          PUSH,  push         },
      {"r0",            R0,    NULL         },
      {"r1",            R1,    NULL         },
      {"r2",            R2,    NULL         },
      {"r3",            R3,    NULL         },
      {"r4",            R4,    NULL         },
      {"r5",            R5,    NULL         },
      {"r6",            R6,    NULL         },
      {"r7",            R7,    NULL         },
      {"ret",           RET,   ret          },
      {"reti",          RETI,  reti         },
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {"rl",          RL,     rl          },
        {"rlc",         RLC,    rlc         },
        {"rr",          RR,     rr          },
        {"rrc",         RRC,    rrc         },
        {"setb",        SETB,   setb        },
        {"sjmp",        SJMP,   sjmp        },
        {"skip",        D_SKIP,     NULL    },
        {"subb",        SUBB,   subb        },
        {"swap",        SWAP,   swap        },
        {"word",        D_WORD,     NULL    },
        {"xch",         XCH,    xch         },
        {"xchd",        XCHD,   xchd        },
        {"xrl",         XRL,    xrl         }
};

#define OPTABSIZE (sizeof(optable)/sizeof(struct opcode))
/* strcase:
 *     A case IN-sensitive string compare.
 */
strcase(s,t)
char *s,*t;
{       for( ; (*s|040) == (*t|040); s++, t++)
                if( *s == '\0') return(0);
        return( (*s|040) - (*t|040) );
}
```

/* ทำการค้นหาแบบไบนารี่ผ่าน Optable [] สำหรับการจับคู่

* สัญลักษณ์คืนสัญลักษณ์ที่ค้นพบหรือได้ค่าเป็น NULL

*/

```
struct opcode *lookop(s)
char *s;
{       register int low,high,mid,cond;
        low = 0;
        high = OPTABSIZE-1;
        while( low<=high ) {
                mid = (low+high)/2;
                if( (cond = strcase(s,optable[mid].name)) < 0 )
                        high = mid-1;
                else if(cond > 0 )
                        low = mid+1;
                else
                        return(&optable[mid]);
        }
        return(NULL);
}
/*  symtab, hash, looksym:
 *  User symbol table routines.
 *  symtab is the hash table for the user symbols.
 * (chaining is used for collision resolution).
 */
static struct symbol *symtab[HASHTABSIZE];
static hash(s)
char *s;
{       register char *p;
        register unsigned h=0,g;
        for(p=s; *p; p++) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            h = (h<<4) + *p;
            if( g = h&0xf0000000 ) {
                  h = h ^ (g >> 24);
                  h = h ^ g;
            }
      }
      return( h % HASHTABSIZE );
}
struct symbol *looksym(s)
char *s;
{     register struct symbol *ptr,*prev;
      char *malloc(),*p;
      register int hv;
      hv = hash(s);
      prev = NULL;
      for(ptr=symtab[hv]; ptr; ptr = ptr->next) {
            if( !strcmp(ptr->name,s) ) {
                  if( prev != NULL ) {
                        prev->next = ptr->next;
                        ptr->next = symtab[hv];
                        symtab[hv] = ptr;
                  }
                  return(ptr);
            }
            prev = ptr;
      }
      if( p = malloc(strlen(s)+1) )
            strcpy(p,s);
      else
            error("Cannot allocate %d bytes",strlen(s)+1);
      ptr = (struct symbol *) malloc( sizeof(struct symbol) );
      if( ptr == NULL )
            error("Cannot allocate %d bytes",sizeof(struct symbol));
      ptr->name = p;
      ptr->type = UNDEF;
      ptr->next = symtab[hv];
      symtab[hv] = ptr;
      return(ptr);
}
/* syminit
 * ทำการเริ่มต้น Hash Table, กับสัญลักษณ์เริ่มต้นจาก sinit []
 */
syminit()
{     register int i,hv;
      for(i=0; i<SINITSIZE; i++ ) {
            hv = hash(sinit[i].name);
            if( symtab[hv] )
                  sinit[i].next = symtab[hv];
            symtab[hv] = &sinit[i];
      }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.6 โปรแกรม SYMBOL.C

```
#line 22 "ec51.y"
#include <setjmp.h>
#include <stdio.h>
#define NOPE
#include "as31.h"
#undef NOPE
#define YYSTYPE union ystack
extern int lineno;
extern int dash1;
extern char *asmfile;
extern jmp_buf main_env;
extern FILE *listing;
int pass,fatal;
unsigned long lc;
static unsigned char bytebuf[1024];        /* used by dumplist() */
static int bytecount;
/* ------------------------ G R A M M E R ---------------------- */
/* ค่าของ TOKEN ต่างๆ */
# define STRING 257
# define D_ORG 258
# define D_BYTE 259
# define D_WORD 260
# define D_SKIP 261
# define D_EQU 262
# define D_FLAG 263
# define D_END 264
# define ACALL 265
# define ADD 266
# define ADDC 267
# define AJMP 268
# define ANL 269
# define CJNE 270
# define CLR 271
# define CPL 272
# define DA 273
# define DEC 274
# define DIV 275
# define DJNZ 276
# define INC 277
# define JB 278
# define JBC 279
# define JC 280
# define JMP 281
# define JNB 282
# define JNC 283
# define JNZ 284
# define JZ 285
# define LCALL 286
# define LJMP 287
# define MOV 288
# define MOVC 289
# define MOVX 290
# define NOP 291
# define MUL 292
# define ORL 293
# define POP 294
# define PUSH 295
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ทำสำเนา ดัดแปลง หรือกระทำการใดๆ ที่ละเมิดลิขสิทธิ์ของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# define RET 296
# define RETI 297
# define RL 298
# define RLC 299
# define RR 300
# define RRC 301
# define SETB 302
# define SJMP 303
# define SUBB 304
# define SWAP 305
# define XCH 306
# define XCHD 307
# define XRL 308
# define AB 309
# define A 310
# define C 311
# define PC 312
# define DPTR 313
# define BITPOS 314
# define R0 315
# define R1 316
# define R2 317
# define R3 318
# define R4 319
# define R5 320
# define R6 321
# define R7 322
# define VALUE 323
# define SYMBOL 324
#define yyclearin yychar = -1
#define yyerrok yyerrflag = 0
extern int yychar;
extern short yyerrflag;
#ifndef YYMAXDEPTH
#define YYMAXDEPTH 150
#endif
#ifndef YYSTYPE
#define YYSTYPE int
#endif
YYSTYPE yylval, yyval;
# define YYERRCODE 256
# line 949 "ec51.y"
yyerror(s)
char *s;
{
      error(s);
}
/* ฟังก์ชันแจ้งข้อผิดพลาด:
 *      ทำการแจ้งข้อผิดพลาดที่เกิดขึ้นสิ่งนี้เป็นสาเหตุให้โปรแกรมจบการทำงานทันที
 */
*/
error(cs,a1,a2,a3,a4,a5,a6)
char *cs,*a1,*a2,*a3,*a4,*a5,*a6;
{
      fprintf(stderr,"File: %s, line: %d, ",asmfile,lineno);
      fprintf(stderr,cs,a1,a2,a3,a4,a5,a6);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fprintf(stderr,".\n");
        longjmp(main_env,1);
}
/* ฟังก์ชันแจ้งคำเตือน
 * ทำการแจ้งคำเตือนถึงสิ่งที่ผิดพลาดเล็กๆ น้อยๆ
 * ที่โปรแกรมสามารถดำเนินการต่อไปได้
 */
warning(cs,a1,a2,a3,a4,a5,a6)
char *cs,*a1,*a2,*a3,*a4,*a5,*a6;
{
        fatal++;
        fprintf(stderr,"File: %s, line: %d, ",asmfile,lineno);
        fprintf(stderr,cs,a1,a2,a3,a4,a5,a6);
        fprintf(stderr,".\n");
}
/* makeop
 *      ฟังก์ชันนี้สร้าง Opcode โดยอาศัยสัญลักษณ์จากตารางสัญลักษณ์ (SYMBOL.C)
 *      จากรายการที่ได้กระทำเอาไว้ในขั้นตอนต่างๆ และโครงสร้างโหมดที่ตำแหน่งที่อยู่
 *      ฟังก์ชันนี้ถูกเรียกใช้, แต่จะทำการสร้างรหัส (Opcode) ใน PASS 2 เท่านั้น
 *      Opcode ซึ่งเป็นผลไบท์ผ่านถึง genbyte () ในขั้นตอนการผลิตรหัสใน PASS 2
 *      คืนค่าของไบทเพื่อใช้ในการผลิตรหัส
 */
makeop(op,m,add)
struct opcode *op;
struct mode *m;
{       register unsigned int newop;
        if( m == NULL ) {
                if(pass2) genbyte(op->bytes[0+add]);
                return(1);
        }
        if( pass2 ) {
                newop = op->bytes[ get_md(*m)+add ] | get_ov(*m);
                genbyte(newop);
                if( get_sz(*m) > 0 ) genbyte( get_b1(*m) );
                if( get_sz(*m) > 1 ) genbyte( get_b2(*m) );
        }
        return( get_sz(*m)+1 );
}
/* inclc :
 *      จำนวนค่าที่เพิ่มขึ้นโดยมีตัวนับตำแหน่งโดย 'i' จำนวน
 *      ตรวจสอบดูถ้า 'i' ว่าเกิน 64K หรือไม่
 *      ตรวจสอบการใช้พื้นที่หน่วยความจำของรหัส (การใช้ส่วนพื้นที่บิท).
 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define indx(a) ( (a)/(sizeof(long)*8) )
#define bit(a)    ( 1 << ((a)%(sizeof(long)*8)) )
#define getloc(a) (regions[indx(a)] & bit(a))
#define setloc(a) (regions[indx(a)] |= bit(a))

inclc(i)
{
      static unsigned long regions[ 0x10000/(sizeof(long)*8) ];
      while(i-- > 0) {
            if( pass2 && getloc(lc) )
                  error("Location counter overlaps");
            if( pass2 ) setloc(lc);
            lc += 1;
      }
            if( lc > 0xffff )
            error("Location counter has exceeded 16-bits");
}
/* padline :
 *      Routine นี้ทำการคืนค่าสตริงใหม่ ซึ่งคือ Equivilant ถึง
 *      ' line ' ยกเว้นแท็ปทั้งหมดนั้นถูกขยายให้ที่ว่างและ
 *      ความยาวทั้งหมดถูกตัดให้ยาวถึง 60 ตัวอักษร
 */
char *padline(line)
char *line;
{     static char newline[61];
      char *p1;
      int pos=0,nxtpos;
      for(p1=line; pos<sizeof(newline)-1 && *p1; p1++ ) {
            if( *p1 == '\t' ) {
                  nxtpos = pos+8-pos%8;
                  while(pos<sizeof(newline)-1 && pos <= nxtpos)
                        newline[pos++] = ' ';
            } else if( *p1 != '\n' )
                  newline[pos++]= *p1;
      }
      newline[pos] = '\0';
      return(newline);
}
/* dumplist :
 *      ผลลัพธ์ของตัวนับตำแหน่งปัจจุบัน bytebuf [] อะเรย์, และ
 *      สตริง' txt 'ถึงแฟ้มการรวบรวมรายละเอียด (Listing File)
 *      Routine นี้ถูกเรียกใช้นับทุกเส้นต้นกำเนิดที่เจอในแฟ้มต้นฉบับ
 * (ใน PASS 2 เท่านั้น และถ้าการรวบรวมรายชื่อเปิด)
 */
dumplist(txt,show)
char *txt;
{     int i,j;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fprintf(listing,show?"%04X: ":"      ",lc); /* ส่วนของ Listing File
(File.Lst)*/
        j=0;
        for(i=0; i<bytecount; i++ ) {
                fprintf(listing,"%02X ",bytebuf[I]);
                if( ++j >= 4 ) {
                        j = 0;
                        fprintf(listing,"\n      ");
                }
        }
        while(++j <= 4)
                fprintf(listing,"   ");
        fprintf(listing," %s\n",padline(txt));
}
/* gen routines:
 *      สถานที่ที่นำข้อมูลเก็บเข้าไปในเข้าไปใน Bytebuf [] อะเรย์ และ
 *      เรียก Emitbyte กับไบท์
 */
genbyte(b)
int b;
{       if( bytecount < sizeof(bytebuf) )
                bytebuf[bytecount++] = b;
        emitbyte(b);
}
genstr(s)
char *s;
{       while( *s )
                genbyte(*s++);
}
genword(w)
unsigned long w;
{       genbyte( (w & 0xff00) >> 8 );
        genbyte( (w & 0x00ff) );
}
short yyexca[] ={
        -1, 1,
        0, -1,
        -2, 0,
        };
# define YYNPROD 161
# define YYLAST 742
short yyact[]={
    9, 286, 168, 182, 183, 184, 185, 186, 187, 188,
  189, 208, 229, 182, 183, 184, 185, 186, 187, 188,
  189,   6, 121, 120, 226, 289, 228, 263, 262,  69,
  282,  68, 284, 283,  70, 192,  10, 182, 183, 184,
  185, 186, 187, 188, 189, 116, 117, 190, 302, 301,
  191, 269, 276, 243, 304, 156, 303, 288, 121, 120,
    9, 287, 270, 154,  75, 113, 112, 111, 110, 109,
  105, 108, 106,  59,  60,  61,  62,  63,  64,  65,
  170, 253, 252, 173, 170, 107, 226,   4, 171,  69,
   56,  68, 158, 172,  70,   5,  10, 139, 300, 292,
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ให้ปฏิบัติตามข้อกำหนดของข้อตกลงการใช้งานเอกสารทุกครั้งที่มีการนำไปใช้

```
291, 271, 255,  79,  69, 199,  68, 254, 297,  70,
224, 249, 245, 151, 280,  69, 235,  68, 212,  69,
 70,  68,  83,  87,  70, 211, 210, 209,  95, 206,
 69, 205,  68, 204, 296,  70, 203, 201, 198, 223,
 69, 197,  68, 196, 295,  70, 194, 141, 142, 193,
129, 167, 157, 248,  95, 166, 147, 233,  69, 231,
 68, 180,  69,  70,  68, 179, 169,  70, 178,  69,
169,  68,  74,  69,  70,  68, 126, 230,  70, 138,
 69, 274,  68, 250, 234,  70,  90,  66,  76,  77,
 78,  82,  86,  89, 131, 132, 133, 134,  58,  84,
115,  73,  57,   3, 119, 238,  55, 114, 148, 104,
 69, 143,  68, 135, 261,  70, 260, 163, 128,  88,
160,  69,   8,  68,   7, 146,  70,   2,   1,   0,
236,   0,   0,   0,   0,   0, 164, 122, 123,   0,
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0,   0,  12,  14,  15,  13,  18,
 47,  35,  36,  23,  22,  24,  51,  21,  48,  50,
 42,  25,  49,  43,  44,  45,  41,  40,  52,  53,
 54,  11,  26,  19,  39,  38,  27,  28,  29,  30,
 31,  32,  37,  46,  16,  33,  20,  34,  17, 225,
244, 155, 121, 120,  96,  97,  98,  99, 100, 101,
102, 103,  72,  71,   6,  12,  14,  15,  13,  18,
 47,  35,  36,  23,  22,  24,  51,  21,  48,  50,
 42,  25,  49,  43,  44,  45,  41,  40,  52,  53,
 54,  11,  26,  19,  39,  38,  27,  28,  29,  30,
 31,  32,  37,  46,  16,  33,  20,  34,  17,  75,
153, 225, 152,   0,  96,  97,  98,  99, 100, 101,
102, 103,  72,  71,  92, 200,   0,  91,   0,  96,
 97,  98,  99, 100, 101, 102, 103,  72,  71, 278,
 96,  97,  98,  99, 100, 101, 102, 103,  72,  71,
 92,   0,  72,  71, 181,  96,  97,  98,  99, 100,
101, 102, 103,  72,  71,  96,  97,  98,  99, 100,
101, 102, 103,  72,  71, 136,  81, 257, 246, 240,
 96,  97,  98,  99, 100, 101, 102, 103, 162,  67,
 67,  72,  71,  75,  85,  72,  71, 258,   0,   0,
 75,   0,  72,  71,   0,   0,  72,  71,   0, 118,
 80,   0,   0,  72,  71,   0,  93, 127, 127, 130,
130, 130, 130, 130,   0,   0,   0,   0,   0,   0,
  0,   0,  94,  94,   0,   0, 159, 281, 165, 159,
  0,   0,   0,  72,  71,   0, 176, 177,   0, 124,
125,   0, 290,   0,  72,  71,   0,   0, 140, 140,
140,   0, 145, 149, 137,   0,   0,   0, 144, 150,
  0, 161, 298, 299, 173, 170,   0,   0, 220, 171,
175, 305, 174,   0, 172,   0,   0,   0,   0, 173,
170,   0,   0, 195, 171, 175,   0, 174,   0, 172,
  0,   0,   0,   0,   0,   0, 202,   0,   0,   0,
  0, 207,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0,   0,   0,   0, 213, 214, 215, 216,
217, 218, 219,   0,   0,   0,   0,   0,   0,   0,
  0, 169,   0,   0,   0,   0,   0,   0,   0,   0,
  0,   0,   0, 237, 239, 237, 169,   0, 264,   0,
  0,   0,   0,   0,   0,   0,   0, 165, 259, 222,
227,   0,   0,   0,   0, 221,   0,   0, 275,   0,
  0,   0,   0,   0, 232,   0,   0,   0, 285,   0,
242,   0, 247,   0, 251,   0, 241,   0,   0,   0,
256,   0,   0,   0,   0,   0,   0, 165,   0,   0,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   0,   0,   0,   0,   0, 265,   0, 266, 267, 268,
   0,   0,   0,   0, 272, 273,   0,   0, 237,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0, 277,
 279,   0,   0,   0,   0,   0,   0,   0, 237, 237,
   0,   0,   0,   0,   0,   0,   0, 237,   0,   0,
   0,   0,   0,   0,   0, 293,   0,   0,   0,   0,
   0, 294 };
short yypact[]={

 -10,-1000, -10,-1000,  32,-1000,-1000, 192, 188,-1000,
-185,-1000, 129, 129,-246,-246,-246, 140, 133, 133,
-246,  64,  90,-240,-237,  21,-238,-1000,-1000,-241,
-242,-243,-244,-245,-246,-265,-265,-265, 129, 129,
 129, 129, 129, 129, 129, 129, 129, 115,-301,-301,
-301, 100,  49,-247,  -9,-1000,  50,-1000,-1000, 129,
 181, 129, 129,-303,-322,-1000,-1000, 502,-1000, 129,
 129,-1000,-1000,-1000,-1000, 124,-1000,-1000,-1000,-1000,
 121, 502,-1000,-1000,-1000, 117,-1000,-1000,-1000,-1000,
-1000,-1000,-1000,-1000,-1000,-278,-1000,-1000,-1000,-1000,
-1000,-1000,-1000,-1000,-1000,-1000,-1000,-263,-1000,-1000,
-1000,-1000,-1000,-1000,-1000,-1000,-1000,-1000,-1000,-279,
-1000,-1000,-1000,-1000,-1000,-1000,-1000, 502,-1000,-1000,
 502,-1000,-1000,-1000,-1000,-1000, 105, 102,-278,-1000,
  99,-1000,-1000,-1000,  97,  94,-1000,-1000,-1000,  61,
  93,-278,  92,  89,  87,  85,-302,-1000,-1000, 502,
  83,-1000,-1000,  82,-1000, 502,-1000,  81,  74, 129,
 129, 129, 129, 129, 129, 129, 487,  42,  75,  51,
 -21,-1000,-1000,-1000,-1000,-1000,-1000,-1000,-1000,-1000,
 134, 116,-1000, 122, 149,  72, 129, 129, 129, -11,
  68, 118,  67, 148,-301,  18,  17,  63,  58, 170,
 129, 129,-296,-1000,-1000,  42,  42,  42,  46,  46,
-1000,-1000,-1000,-278, 129,-1000, 129,-1000,-301,-301,
-262,-248,  57, 129, 129, 146,-1000, 502,-1000, 502,
-1000,-1000,-1000,-278,-1000,-259,-1000,-1000, 129,  79,
 129,-1000,-280,-312,-249,-253,-1000,-1000,-1000, 502,
-1000,-289,-1000,-1000,-1000,-1000,-1000,-1000,-1000,-1000,
-1000, 129,  56,  55, 129,-1000,-1000,-1000,-1000,-1000,
 129,-1000, 101,  91,  65,-1000,-1000,-1000,-1000,-1000,
-1000, 129, 129,  54,-1000,-264,-254,-256,-1000,-1000,
 129,-1000,-1000,-1000,-1000,-1000 };
short yypgo[]={

   0, 228, 227, 203,  87,  95, 224, 222,  92, 220,
 217, 426, 216, 214, 460, 236, 187, 172, 103, 199,
 186, 200, 176, 150, 213,  97, 211, 208, 404, 466,
 459, 230, 205, 204 };
short yyr1[]={

   0,   1,   2,   2,   3,   3,   5,   5,   5,   6,
   6,   6,   6,   6,   6,   6,   8,  12,  13,  13,
   4,   9,   9,   9,   9,  10,  10,  11,  11,  11,
  11,  11,  11,  11,  11,  11,  11,  11,  11,   7,
   7,   7,   7,   7,   7,   7,   7,   7,   7,   7,
   7,   7,   7,   7,   7,   7,   7,   7,   7,   7,
   7,   7,   7,   7,   7,   7,   7,   7,   7,   7,
   7,   7,   7,   7,   7,   7,   7,   7,   7,   7,
   7,   7,   7,   7,   7,   7,   7,   7,  17,  17,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  17,  17,  18,  18,  19,  19,  19,  25,  26,  26,
  27,  27,  27,  27,  27,  27,  27,  27,  27,  27,
  27,  27,  27,  20,  20,  20,  20,  21,  21,  21,
  24,  24,  24,  24,  31,  32,  30,  30,  33,  33,
  29,  29,  29,  29,  29,  29,  29,  29,  28,  28,
  28,  28,  28,  28,  28,  28,  14,  15,  16,  22,
  23 };
short yyr2[]={

   0,   1,   2,   1,   3,   1,   2,   2,   1,   3,
   3,   3,   3,   5,   5,   2,   1,   2,   1,   1,
   1,   3,   3,   1,   1,   3,   1,   1,   3,   2,
   3,   3,   3,   3,   3,   3,   3,   1,   1,   1,
   2,   2,   2,   2,   2,   2,   2,   2,   2,   2,
   2,   2,   2,   2,   2,   2,   2,   2,   2,   5,
   5,   2,   1,   1,   2,   2,   2,   2,   2,   2,
   2,   2,   2,   2,   2,   2,   2,   2,   2,   2,
   2,   2,   2,   2,   2,   2,   2,   2,   2,   2,
   7,   7,   7,   7,   5,   5,   5,   5,   3,   3,
   4,   4,   3,   4,   3,   4,   4,   3,   3,   3,
   3,   3,   4,   3,   3,   4,   4,   4,   5,   4,
   3,   3,   4,   1,   1,   1,   2,   1,   1,   1,
   5,   6,   6,   7,   1,   1,   2,   1,   1,   1,
   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,
   1,   1,   1,   1,   1,   1,   1,   1,   1,   1,
   1 };
short yychk[]={

-1000,  -1,  -2,  -3,  -4,  -5, 324,  -6,  -7,  10,
  46, 291, 265, 268, 266, 267, 304, 308, 269, 293,
 306, 277, 274, 273, 275, 281, 292, 296, 297, 298,
 299, 300, 301, 305, 307, 271, 272, 302, 295, 294,
 287, 286, 280, 283, 284, 285, 303, 270, 278, 282,
 279, 276, 288, 289, 290,  -3,  58,  10,  10, 258,
 259, 260, 261, 262, 263, 264, -16, -11,  42,  40,
  45, 324, 323, -16, -17, 310, -17, -17, -17, -18,
 -14, -11, -17, -18, -19, 311, -17, -18, -19, -17,
 -20, 313, 310, -29, -14,  64, 315, 316, 317, 318,
 319, 320, 321, 322, -20, 310, 309,  64, 309, 310,
 310, 310, 310, 310, -17, -21, 310, 311, -30, -33,
 324, 323, -21, -21, -14, -14, -22, -11, -22, -23,
 -11, -23, -23, -23, -23, -24, 310, -29,  64, -25,
 -30, -25, -25, -26, -29, -14, -17, -18, -27, -14,
 -29,  64, 313, 311, 310, 310,  64,  -5,  -8, -11,
  -9, -14, 257, -10, -15, -11,  -8,  -4, 324, 124,
  38,  42,  47,  37,  45,  43, -11, -11,  44,  44,
  44, -28, 315, 316, 317, 318, 319, 320, 321, 322,
 310, 313, 314,  44,  44, -28,  44,  44,  44,  44,
 314,  44, -28,  44,  44,  44,  44, -28, 313,  44,
  44,  44,  44, -11, -11, -11, -11, -11, -11, -11,
  41, -29, -14,  64,  35, 310,  35, -30,  47,  33,
  43,  43, -14,  35,  35,  44, -31, -11, -32, -11,
 -31, -29, -14,  64, 311,  44, 310, -14,  35,  44,
  35, -30,  64,  64,  44,  44, -14, 257, -15, -11,
 -12, -13, 324, 323, -28, -14, -14, -30, -30, 313,
 310,  44, -14, -14,  35, -28, 311, -14, 310, -14,
  35, -15, 310, 313, 312, -28, 313, 310, 310, 314,
 -31,  44,  44, -14, -14,  43,  43,  43, -31, -31,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  44, 313, 312, 310, 310, -31 };
short yydef[]={

   0,  -2,   1,   3,   0,   5,  20,   0,   0,   8,
   0,  39,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,  62,  63,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   2,   0,   6,   7,   0,
   0,   0,   0,   0,   0,  15,  40, 158,  27,   0,
   0,  37,  38,  41,  42,   0,  43,  44,  45,  46,
   0, 156,  47,  48,  49,   0,  50,  51,  52,  53,
  54,  55, 123, 124, 125,   0, 140, 141, 142, 143,
 144, 145, 146, 147,  56,  57,  58,   0,  61,  64,
  65,  66,  67,  68,  69,  70, 127, 128, 129, 137,
 138, 139,  71,  72,  73,  74,  75, 159,  76,  77,
 160,  78,  79,  80,  81,  82,   0,   0,   0,  83,
   0,  84,  85,  86,   0,   0,  87,  88,  89,   0,
   0,   0,   0,   0,   0,   0,   0,   4,   9,  16,
  10,  23,  24,  11,  26, 157,  12,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,  29,   0,   0,
   0, 126, 148, 149, 150, 151, 152, 153, 154, 155,
   0,   0, 136,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
   0,   0,   0,  30,  31,  32,  33,  34,  35,  36,
  28,  98,  99,   0,   0, 102,   0, 104,   0,   0,
   0,   0,   0,   0,   0,   0, 107, 134, 108, 135,
 109, 113, 114,   0, 121,   0, 110, 111,   0,   0,
   0, 120,   0,   0,   0,   0,  21,  22,  25,  13,
  14,   0,  18,  19, 100, 101, 103, 105, 106,  59,
  60,   0,   0,   0,   0, 115, 122, 112, 116, 117,
   0, 119,   0,   0,   0,  94,  95,  96,  97,  17,
 130,   0,   0,   0, 118,   0,   0,   0, 131, 132,
   0,  90,  92,  91,  93, 133 };
#ifndef lint
static      char yaccpar_sccsid[] = "@(#)yaccpar 1.6 88/02/08 SMI";
/* from UCB 4.1 83/02/11 */
#endif
#define YYFLAG -1000
#define YYERROR goto yyerrlab
#define YYACCEPT return(0)
#define YYABORT return(1)
/*     การวิเคราะห์คำในไวยากรณ์สำหรับผลลัพธ์ yacc   */

YYSTYPE yyv[YYMAXDEPTH]; /* ที่เก็บค่าต่างๆ */
int yychar = -1; /* Current input token number. */
int yynerrs = 0;  /* Number of errors. */
short yyerrflag = 0;  /* Error recovery flag. */
yyparse() {
      short yys[YYMAXDEPTH];
      short yyj, yym;
      register YYSTYPE *yypvt;
      register short yystate, *yyps, yyn;
      register YYSTYPE *yypv;
      register short *yyxi;
      yystate = 0;
      yychar = -1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      yynerrs = 0;
      yyerrflag = 0;
      yyps= &yys[-1];
      yypv= &yyv[-1];
 yystack:    /* put a state and value onto the stack */
            if( ++yyps>= &yys[YYMAXDEPTH] ) { yyerror( "yacc stack
overflow" ); return(1); }
            *yyps = yystate;
            ++yypv;
            *yypv = yyval;
 yynewstate:
      yyn = yypact[yystate];
      if( yyn<= YYFLAG ) goto yydefault; /* simple state */
      if( yychar<0 ) if( (yychar=yylex())<0 ) yychar=0;
            if ((yyn==-10)&&((yychar==258)||(yychar==259)||
(yychar==260)||(yychar==261)||
      (yychar==262)||(yychar==263)||(yychar==264))) { yyn=-185;  }
      if( (yyn += yychar)<0 || yyn >= YYLAST ) goto yydefault;
      if( yychk[ yyn=yyact[ yyn ] ] == yychar ){ /* valid shift */
            yychar = -1;
            yyval = yylval;
            yystate = yyn;
            if( yyerrflag > 0 ) --yyerrflag;
            goto yystack;
            }
 yydefault:
      /* default state action */
      if( (yyn=yydef[yystate]) == -2 ) {
            if( yychar<0 ) if( (yychar=yylex())<0 ) yychar = 0;
/* ข้อยกเว้น*/

            for( yyxi=yyexca; (*yyxi!= (-1)) || (yyxi[1]!=yystate) ;
yyxi += 2 ) ; /* VOID */
            while( *(yyxi+=2) >= 0 ){
                  if( *yyxi == yychar ) break;
                  }

            if( (yyn = yyxi[1]) < 0 ) return(0);   /* ยอมรับ */
            }
      if( yyn == 0 ){ /* error */
/* ข้อผิดพลาด พยายามเพื่อการวิเคราะห์คำต่อไปในไวยากรณ์*/
            switch( yyerrflag ){
            case 0:   /* brand new error */
                  yyerror( "syntax error" );
            yyerrlab:
                  ++yynerrs;
            case 1:
            case 2:  /* ข้อผิดพลาดอย่างไม่สมบูรณ์ ทดลองอีกครั้ง*/
            yyerrflag = 3;
                              /* ค้นหาสถานะที่ "เกิดข้อผิดพลาดขึ้น" */
                  while ( yyps >= yys ) {
                     yyn = yypact[*yyps] + YYERRCODE;
 if( yyn>= 0 && yyn < YYLAST && yychk[yyact[yyn]] == YYERRCODE )
{
                     yystate = yyact[yyn];  /* simulate a shift of "error"*/
                              goto yystack;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                    yyn = yypact[*yyps];
                    --yyps;
                    --yypv;
                    }
                yyabort:
                return(1);
        case 3: /* ไม่มีการเปลี่ยนแปลง */
    if( yychar == 0 ) goto yyabort; /* don't discard EOF, quit */
                yychar = -1;
                goto yynewstate; /* ทดลองอีกครั้งในสถานะเช่นเดียวกัน */
                }
            }
    /* การลดลงโดยการผลิต yyn */
            yyps -= yyr2[yyn];
            yypvt = yypv;
            yypv -= yyr2[yyn];
            yyval = yypv[1];
            yym=yyn;
    /*ค้นหาสถานะถัดไป*/
            yyn = yyr1[yyn];
            yyj = yypgo[yyn] + *yyps + 1;
            if( yyj>=YYLAST || yychk[ yystate = yyact[yyj] ] != -yyn
) yystate = yyact[yypgo[yyn]];
            switch(yym){
case 1:
# line 124 "ec51.y" /*  program          :      linelist */
{
} break;
case 4:
# line 133 "ec51.y" /*  line             : undefsym ':' linerest */
{
      if( pass1 ) {
            yypvt[-2].sym->type = LABEL;
            yypvt[-2].sym->value = lc;
      }
      inclc(yypvt[-0].value);
      bytecount = 0;
} break;
case 5:
# line 141 "ec51.y" /*  | linerest      */
{ inclc(yypvt[-0].value); bytecount = 0; } break;
case 6:
# line 144 "ec51.y" /*linerest       : directive '\n' */
{
                                    yyval.value = yypvt[-1].value;
                                    if( dashl && pass2 )
                                          dumplist(yypvt[-0].str,1);
                                } break;
case 7:
# line 149 "ec51.y" /* | instr '\n' */
{
                                    yyval.value = yypvt[-1].value;
                                    if( dashl && pass2 )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

```
                                        dumplist(yypvt[-0].str,1);

                                } break;
case 8:
# line 155 "ec51.y" /* |  '\n'        */
{
                                        yyval.value = 0;
                                        if( dashl && pass2 )
                                                dumplist(yypvt[-0].str,0);
                                } break;
case 9:
# line 172 "ec51.y" /* directive    :  D_ORG defexpr  */
{
        lc = yypvt[-0].val.v;
        if( pass2 ) emitaddr(lc);
        bytecount = 0;
        yyval.value = 0;
} break;
case 10:
# line 178 "ec51.y" /*|  D_BYTE blist  */
{ yyval.value = yypvt[-0].value; } break;
case 11:
# line 179 "ec51.y" /*|  D_WORD wlist */
{ yyval.value = yypvt[-0].value; } break;
case 12:
# line 180 "ec51.y" /* |  D_SKIP defexpr */
{ yyval.value = yypvt[-0].val.v;
                        if( pass2 )
                                emitaddr(lc+yyval.value); }
break;
case 13:
# line 184 "ec51.y" /* | D_EQU undefsym  expr */
{
        if( yypvt[-0].val.d == 0 )
                error("Expression is undefined in pass 1");
        yypvt[-2].sym->type = LABEL;
        yypvt[-2].sym->value = yypvt[-0].val.v;
        yyval.value = 0;
} break;
case 14:
# line 193 "ec51.y" /*  | D_FLAG SYMBOL ',' flag  */
{
        yypvt[-2].sym->type = LABEL;
        yypvt[-2].sym->value = yypvt[-0].value;
        yyval.value = 0;
} break;
case 15:
# line 198 "ec51.y" /*  | D_END  */
{ yyval.value = 0; } break;
case 16:
# line 202 "ec51.y" /* defexpr             : expr  */
{
                if( yypvt[-0].val.d == 0 )
                        error("Expression is undefined in pass 1");
                if( !(isbit16(yypvt[-0].val.v)) )
                        error("Value greater than 16-bits");
                yyval.value = yypvt[-0].val.v;
} break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 17:
# line 212 "ec51.y" /* flag          : flagv BITPOS   */
{
      if( !isbit8(yypvt[-1].value) )
            warning("Bit address exceeds 8-bits");
      if( isbmram(yypvt[-1].value) )
            yyval.value = (yypvt[-1].value-0x20)*8+ yypvt[-0].value;
      else if( isbmsfr(yypvt[-1].value) )
            yyval.value = yypvt[-1].value + yypvt[-0].value;
      else
            warning("Invalid bit addressable RAM location");
} break;
case 18:
# line 225 "ec51.y" /* flagv         : SYMBOL*/
{
      if( yypvt[-0].sym->type == UNDEF )
            error("Symbol %s must be defined in pass 1",yypvt[-
0].sym->name);
      yyval.value = yypvt[-0].sym->value;
} break;
case 19:
# line 230 "ec51.y" /*   | VALUE */
{ yyval.value = yypvt[-0].value; } break;
case 20:
# line 235 "ec51.y" /* undefsym       : SYMBOL   */
{
      if( yypvt[-0].sym->type != UNDEF && pass1)
            error("Attempt to redefine symbol: %s",yypvt[-0].sym->
name);
      yyval.sym = yypvt[-0].sym;
} break;
case 21:
# line 243 "ec51.y"  /* blist         : blist ',' data8   */
{
      if( pass2 ) genbyte(yypvt[-0].value);
      yyval.value = yypvt[-2].value + 1;
} break;
case 22:
# line 248 "ec51.y" /*   | blist ',' STRING   */
{
      if( pass1 )
            yyval.value = yypvt[-2].value + yypvt[-0].value;
      else {
            yyval.value = yypvt[-2].value + strlen(yypvt[-0].str);
            genstr(yypvt[-0].str);

            free(yypvt[-0].str);
      }
} break;
case 23:
# line 259 "ec51.y" /*   | data8   */
{
      if( pass2 ) genbyte(yypvt[-0].value);
      yyval.value = 1;
} break;
case 24:
# line 264 "ec51.y" /*   | STRING */
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if( pass1 )
            yyval.value = yypvt[-0].value;
      else {
            yyval.value = strlen(yypvt[-0].str);
            genstr(yypvt[-0].str);
            free(yypvt[-0].str);
      }
} break;
case 25:
# line 276 "ec51.y" /*wlist         : wlist ',' data16  */
{
      if( pass2 ) genword(yypvt[-0].value);
      yyval.value = yypvt[-2].value + 2;
} break;
case 26:
# line 281 "ec51.y" /*| data16  */
{
      if( pass2 ) genword(yypvt[-0].value);
      yyval.value = 2;
} break;
case 27:
# line 294 "ec51.y" /*expr          : '*'  */
{ yyval.val.v = 1c;
                              yyval.val.d = 1; } break;
case 28:
# line 297 "ec51.y" /* | '(' expr ')' */
{ yyval.val.v = yypvt[-1].val.v;
                              yyval.val.d = yypvt[-1].val.d; }
break;
case 29:
# line 300 "ec51.y" /* | '-' expr %prec '*' */
{ yyval.val.v = -yypvt[-0].val.v;
                              yyval.val.d = yypvt[-0].val.d;  }
break;
case 30:
# line 303 "ec51.y" /*| expr '|' expr       */
{ yyval.val.v = yypvt[-2].val.v | yypvt[-0].val.v;
                              yyval.val.d = yypvt[-2].val.d &&
yypvt[-0].val.d; } break;
case 31:
# line 306 "ec51.y" /*| expr '&' expr  */
{ yyval.val.v = yypvt[-2].val.v & yypvt[-0].val.v;
                              yyval.val.d = yypvt[-2].val.d &&
yypvt[-0].val.d; } break;
case 32:
# line 309 "ec51.y" /* | expr '*' expr     */
{ yyval.val.v = yypvt[-2].val.v * yypvt[-0].val.v;
                              yyval.val.d = yypvt[-2].val.d &&
yypvt[-0].val.d; } break;
case 33:
# line 312 "ec51.y" /*| expr '/' expr  */
{ yyval.val.v = yypvt[-2].val.v / yypvt[-0].val.v;
                              yyval.val.d = yypvt[-2].val.d &&
yypvt[-0].val.d; } break;
case 34:
# line 315 "ec51.y" /*  | expr '%' expr     */
{ yyval.val.v = yypvt[-2].val.v % yypvt[-0].val.v;
                              yyval.val.d = yypvt[-2].val.d &&
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
yypvt[-0].val.d; } break;
case 35:
# line 318 "ec51.y" /*| expr '-' expr  */
{ yyval.val.v = yypvt[-2].val.v - yypvt[-0].val.v;
                                yyval.val.d = yypvt[-2].val.d &&
yypvt[-0].val.d; } break;
case 36:
# line 321 "ec51.y" /*  | expr '+' expr    */
{ yyval.val.v = yypvt[-2].val.v + yypvt[-0].val.v;
                                yyval.val.d = yypvt[-2].val.d &&
yypvt[-0].val.d; } break;
case 37:
# line 324 "ec51.y" /*  | SYMBOL  */
{
      if( pass1 ) {
            yyval.val.v = yypvt[-0].sym->value;
            yyval.val.d = (yypvt[-0].sym->type != UNDEF);
      }
      else {
            if( yypvt[-0].sym->type == UNDEF )
                  error("Undefined symbol %s",yypvt[-0].sym->name);
            yyval.val.v = yypvt[-0].sym->value;
            yyval.val.d = 1;
      }
} break;
case 38:
# line 336 "ec51.y" /* | VALUE */
{ yyval.val.v = yypvt[-0].val.v; yyval.val.d=1; } break;
case 39:
# line 349 "ec51.y" /* instr          : NOP */
{ yyval.value = makeop(yypvt[-0].op,NULL,0); } break;
case 40:
# line 351 "ec51.y" /*  | ACALL addr11   */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 41:
# line 353 "ec51.y" /* | AJMP addr11 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 42:
# line 355 "ec51.y" /*  | ADD two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 43:
# line 357 "ec51.y" /* | ADDC two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 44:
# line 359 "ec51.y" /* | SUBB two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 45:
# line 361 "ec51.y" /* | XRL two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 46:
# line 363 "ec51.y" /* | XRL two_op2 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,4); } break;
case 47:
# line 365 "ec51.y" /* | ANL two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 48:
# line 367 "ec51.y" /* | ANL two_op2 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,4); } break;
```

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 49:
# line 369 "ec51.y" /* | ANL two_op3 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,6); } break;
case 50:
# line 371 "ec51.y" /* | ORL two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 51:
# line 373 "ec51.y" /* | ORL two_op2 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,4); } break;
case 52:
# line 375 "ec51.y" /* | ORL two_op3 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,6); } break;
case 53:
# line 377 "ec51.y" /*  | XCH two_op1 */
{ if( get_md(yypvt[-0].mode) == 3 )
                error("Immediate mode is illegal");
            yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0);
                    } break;
case 54:
# line 382 "ec51.y" /* | INC single_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 55:
# line 384 "ec51.y" /* | INC DPTR */
{ yyval.value = makeop(yypvt[-1].op,NULL,4); } break;
case 56:
# line 386 "ec51.y" /* | DEC single_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 57:
# line 388 "ec51.y" /*  | DA A */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 58:
# line 390 "ec51.y" /* | DIV AB */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 59:
# line 392 "ec51.y" /* | JMP '@' A '+' DPTR */
{ yyval.value = makeop(yypvt[-4].op,NULL,0); } break;
case 60:
# line 394 "ec51.y" /* | JMP '@' DPTR '+' A */
{ yyval.value = makeop(yypvt[-4].op,NULL,0); } break;
case 61:
# line 396 "ec51.y" /* | MUL AB */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 62:
# line 398 "ec51.y" /* | RET */
{ yyval.value = makeop(yypvt[-0].op,NULL,0); } break;
case 63:
# line 400 "ec51.y" /* | RETI */
{ yyval.value = makeop(yypvt[-0].op,NULL,0); } break;
case 64:
# line 402 "ec51.y" /* | RL A */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 65:
# line 404 "ec51.y" /* | RLC A */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 66:
# line 406 "ec51.y" /* | RR A */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 67:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# line 408 "ec51.y" /* | RRC A */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 68:
# line 410 "ec51.y" /* | SWAP A */
{ yyval.value = makeop(yypvt[-1].op,NULL,0); } break;
case 69:
# line 412 "ec51.y" /* | XCHD two_op1 */
{ if( get_md(yypvt[-0].mode) != 2 )
                    error("Invalid addressing mode");
                yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,-2);
} break;
case 70:
# line 416 "ec51.y" /* | CLR single_op2 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 71:
# line 418 "ec51.y" /* | CPL single_op2 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 72:
# line 420 "ec51.y" /*  | SETB single_op2 */
{ if( get_md(yypvt[-0].mode) == 0 )
                    error("Invalid addressing mode");
                yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,-1);
} break;
case 73:
# line 424 "ec51.y" /*| PUSH data8  */
{
                struct mode tmp;
                    set_md(tmp,0);
                    set_ov(tmp,0);
                    set_sz(tmp,1);
                    set_bl(tmp,yypvt[-0].value);
                    yyval.value = makeop(yypvt[-1].op,&tmp,0);
                        } break;
case 74:
# line 433 "ec51.y" /* | POP data8 */
{
                struct mode tmp;
                    set_md(tmp,0);
                    set_ov(tmp,0);
                    set_sz(tmp,1);
                    set_bl(tmp,yypvt[-0].value);
                    yyval.value = makeop(yypvt[-1].op,&tmp,0);
                        } break;
case 75:
# line 442 "ec51.y" /*  | LJMP addr16  */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 76:
# line 444 "ec51.y" /* | LCALL addr16 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 77:
# line 446 "ec51.y" /* | JC relative */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 78:
# line 448 "ec51.y" /* | JNC relative */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 79:
# line 450 "ec51.y" /* | JNZ relative */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
```

เอกสารนี้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 80:
# line 452 "ec51.y" /* | JZ relative */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 81:
# line 454 "ec51.y" /* | SJMP relative */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 82:
# line 456 "ec51.y" /* | CJNE three_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 83:
# line 458 "ec51.y" /* | JB two_op4 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 84:
# line 460 "ec51.y" /* | JNB two_op4 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 85:
# line 462 "ec51.y" /* | JBC two_op4 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 86:
# line 464 "ec51.y" /* | DJNZ two_op5 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 87:
# line 466 "ec51.y" /* | MOV two_op1 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,0); } break;
case 88:
# line 468 "ec51.y" /* | MOV two_op2 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,4); } break;
case 89:
# line 470 "ec51.y" /* | MOV two_op6 */
{ yyval.value = makeop(yypvt[-1].op,&yypvt[-0].mode,6); } break;
case 90:
# line 474 "ec51.y" /* | MOVC A ',' '@' A '+' DPTR */
{ yyval.value = makeop(yypvt[-6].op,NULL,0); } break;
case 91:
# line 476 "ec51.y" /* | MOVC A ',' '@' DPTR '+' A */
{ yyval.value = makeop(yypvt[-6].op,NULL,0); } break;
case 92:
# line 478 "ec51.y" /* | MOVC A ',' '@' A '+' PC */
{ yyval.value = makeop(yypvt[-6].op,NULL,1); } break;
case 93:
# line 480 "ec51.y" /* | MOVC A ',' '@' PC '+' A */
{ yyval.value = makeop(yypvt[-6].op,NULL,1); } break;
case 94:
# line 483 "ec51.y" /* | MOVX A ',' '@' regi */
{ yyval.value = makeop(yypvt[-4].op,NULL,yypvt[-0].value); } break;
case 95:
# line 485 "ec51.y" /*  | MOVX A ',' '@' DPTR  */
{ yyval.value = makeop(yypvt[-4].op,NULL,2); } break;
case 96:
# line 487 "ec51.y" /* | MOVX '@' regi ',' A */
{ yyval.value = makeop(yypvt[-4].op,NULL,yypvt[-2].value+3); }break;
case 97:
# line 489 "ec51.y" /*  | MOVX '@' DPTR ',' A  */
{ yyval.value = makeop(yypvt[-4].op,NULL,5); } break;
case 98:
# line 501 "ec51.y" /* two_op1      : A ',' reg */
{
                set_md(yyval.mode,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    set_ov(yyval.mode, yypvt[-0].value);
                    set_sz(yyval.mode, 0);
                          } break;
case 99:
# line 507 "ec51.y" /*| A ',' data8  */
{
                    set_md(yyval.mode,1);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;

case 100:
# line 514 "ec51.y" /* | A ',' '@' regi */
{
                    set_md(yyval.mode,2);
                    set_ov(yyval.mode,yypvt[-0].value);
                    set_sz(yyval.mode,0);
                          } break;
case 101:
# line 520 "ec51.y" /*  | A ',' '#' data8 */
{
                    set_md(yyval.mode,3);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;

case 102:
# line 529 "ec51.y" /* two_op2              : data8 ',' A */
{
                    set_md(yyval.mode,0);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-2].value);
                          } break;
case 103:
# line 536 "ec51.y" /* | data8 ',' '#' data8 */
{
                    set_md(yyval.mode,1);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,2);
                    set_b1(yyval.mode,yypvt[-3].value);
                    set_b2(yyval.mode,yypvt[-0].value);
                          } break;
case 104:
# line 546 "ec51.y" /* two_op3             : C ',' bit */
{
                    set_md(yyval.mode,0);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 105:
# line 553 "ec51.y" /*  | C ',' '/' bit */
{
                    set_md(yyval.mode,1);
                    set_ov(yyval.mode,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 106:
# line 560 "ec51.y" /* | C ',' '!' bit */
{
                    set_md(yyval.mode,1);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 107:
# line 569 "ec51.y" /* two_op4              : bit ',' rel  */
{
                    set_md(yyval.mode,0);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,2);
                    set_b1(yyval.mode,yypvt[-2].value);
                    set_b2(yyval.mode,yypvt[-0].value);
                          } break;
case 108:
# line 579 "ec51.y" /* two_op5                : reg ',' rel2 */
{
                    set_md(yyval.mode,0);
                    set_ov(yyval.mode,yypvt[-2].value);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 109:
# line 586 "ec51.y" /* | data8 ',' rel */
{
                    set_md(yyval.mode,1);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,2);
                    set_b1(yyval.mode,yypvt[-2].value);
                    set_b2(yyval.mode,yypvt[-0].value);
                          } break;
case 110:
# line 596 "ec51.y" /* two_op6          : reg ',' A */
{
                    set_md(yyval.mode,0);
                    set_ov(yyval.mode,yypvt[-2].value);
                    set_sz(yyval.mode,0);
                    } break;
case 111:
# line 602 "ec51.y" /*  | reg ',' data8  */
{
                    set_md(yyval.mode,1);
                    set_ov(yyval.mode,yypvt[-2].value);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 112:
# line 609 "ec51.y" /* | reg ',' '#' data8 */
{
                    set_md(yyval.mode,2);
                    set_ov(yyval.mode,yypvt[-3].value);
                    set_sz(yyval.mode,1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 113:
# line 616 "ec51.y" /*| data8 ',' reg  */
{
                    set_md(yyval.mode,3);
                    set_ov(yyval.mode,yypvt[-0].value);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-2].value);
                          } break;
case 114:
# line 623 "ec51.y" /*| data8 ',' data8  */
{
                    set_md(yyval.mode,4);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,2);
                    set_b1(yyval.mode,yypvt[-0].value);
                    set_b2(yyval.mode,yypvt[-2].value);
                          } break;
case 115:
# line 631 "ec51.y" /* | data8 ',' '@' regi */
{
                    set_md(yyval.mode,5);
                    set_ov(yyval.mode,yypvt[-0].value);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-3].value);
                          } break;
case 116:
# line 638 "ec51.y" /* | '@' regi ',' A */
{
                    set_md(yyval.mode,6);
                    set_ov(yyval.mode,yypvt[-2].value);
                    set_sz(yyval.mode,0);
                          } break;
case 117:
# line 644 "ec51.y" /*  | '@' regi ',' data8 */
{
                    set_md(yyval.mode,7);
                    set_ov(yyval.mode,yypvt[-2].value);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 118:
# line 651 "ec51.y" /* | '@' regi ',' '#' data8 */
{
                    set_md(yyval.mode,8);
                    set_ov(yyval.mode,yypvt[-3].value);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                          } break;
case 119:
# line 658 "ec51.y" /*| DPTR ',' '#' data16  */
{
               set_md(yyval.mode,9);
               set_ov(yyval.mode,0);
               set_sz(yyval.mode,2);
               set_b1(yyval.mode, (yypvt[-0].value & 0xff00) >> 8 );
               set_b2(yyval.mode, (yypvt[-0].value & 0x00ff) );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    } break;
case 120:
# line 666 "ec51.y" /*| C ',' bit  */
{
                    set_md(yyval.mode,10);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-0].value);
                    } break;

case 121:
# line 682 "ec51.y" /* | data8 ',' C */
{
                    set_md(yyval.mode,11);
                    set_ov(yyval.mode,0);
                    set_sz(yyval.mode,1);
                    set_b1(yyval.mode,yypvt[-2].value);
                         } break;
case 122:
# line 689 "ec51.y" /*  | data8 BITPOS ',' C */
{     if( pass2 ) {
            if( !isbit8(yypvt[-3].value) )
                    warning("Bit address exceeds 8-bits");
            if( isbmram(yypvt[-3].value) )
      set_b1(yyval.mode, (yypvt[-3].value-0x20)*8+ yypvt[-2].value
);
            else if( isbmsfr(yypvt[-3].value) )
      set_b1(yyval.mode, yypvt[-3].value + yypvt[-2].value );
            else
                    warning("Invalid bit addressable RAM location");
      }
      set_md(yyval.mode,11);
      set_ov(yyval.mode,0);
      set_sz(yyval.mode,1);
} break;
case 123:
# line 708 "ec51.y" /* single_op1    : A */
{                           set_md(yyval.mode,0);
                            set_ov(yyval.mode,0);
                            set_sz(yyval.mode,0);
                         } break;
case 124:
# line 715 "ec51.y" /*  | reg */
{                           set_md(yyval.mode,1);
                            set_ov(yyval.mode,yypvt[-0].value);
                            set_sz(yyval.mode,0);
                        } break;
case 125:
# line 721 "ec51.y" /*  | data8 */
{                           set_md(yyval.mode,2);
                            set_ov(yyval.mode,0);
                            set_sz(yyval.mode,1);
                            set_b1(yyval.mode,yypvt[-0].value);
                        } break;
case 126:
# line 728 "ec51.y" /* | '@' regi */
{                           set_md(yyval.mode,3);
                            set_ov(yyval.mode,yypvt[-0].value);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
                                set_sz(yyval.mode,0);
                            } break;
case 127:
# line 736 "ec51.y" /* single_op2   : A */
{                               set_md(yyval.mode,0);
                                set_ov(yyval.mode,0);
                                set_sz(yyval.mode,0);
                            } break;
case 128:
# line 742 "ec51.y" /*  | C */
{                               set_md(yyval.mode,1);
                                set_ov(yyval.mode,0);
                                set_sz(yyval.mode,0);
                            } break;
case 129:
# line 748 "ec51.y" /* | bit */
{                               set_md(yyval.mode,2);
                                set_ov(yyval.mode,0);
                                set_sz(yyval.mode,1);
                                set_b1(yyval.mode,yypvt[-0].value);
                            } break;
case 130:
# line 757 "ec51.y" /* three_op1     : A ',' data8 ',' rel */
{                               set_md(yyval.mode,0);
                                set_ov(yyval.mode,0);
                                set_sz(yyval.mode,2);
                                set_b1(yyval.mode,yypvt[-2].value);
                                set_b2(yyval.mode,yypvt[-0].value);
                            } break;
case 131:
# line 765 "ec51.y" /* | A ',' '#' data8 ',' rel */
{                               set_md(yyval.mode,1);
                                set_ov(yyval.mode,0);
                                set_sz(yyval.mode,2);
                                set_b1(yyval.mode,yypvt[-2].value);
                                set_b2(yyval.mode,yypvt[-0].value);
                            } break;
case 132:
# line 773 "ec51.y" /* | reg ',' '#' data8 ',' rel */
{                               set_md(yyval.mode,2);
                                set_ov(yyval.mode,yypvt[-5].value);
                                set_sz(yyval.mode,2);
                                set_b1(yyval.mode,yypvt[-2].value);
                                set_b2(yyval.mode,yypvt[-0].value);
                            } break;
case 133:
# line 781 "ec51.y" /* | '@' regi ',' '#' data8 ',' rel */
{                               set_md(yyval.mode,3);
                                set_ov(yyval.mode,yypvt[-5].value);
                                set_sz(yyval.mode,2);
                                set_b1(yyval.mode,yypvt[-2].value);
                                set_b2(yyval.mode,yypvt[-0].value);
                            } break;
case 134:
# line 791 "ec51.y" /* rel           : expr */
{       long offset;
        if( pass2 ) {
                offset = yypvt[-0].val.v - (lc+3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if( offset > 127 || offset < -128 )
                   warning("Relative offset exceeds -128 / +127");
                yyval.value = offset;
            }
} break;
case 135:
# line 808 "ec51.y" /* rel2          : expr  */
{           long offset;
            if( pass2 ) {
                offset = yypvt[-0].val.v - (lc+2); /* different!
*/
                if( offset > 127 || offset < -128 )
                   warning("Relative offset exceeds -128 / +127");
                yyval.value = offset;
            }
} break;
case 136:
# line 821 "ec51.y" /* bit             : bitv BITPOS  */
{     if( pass2 ) {
            if( !isbit8(yypvt[-1].value) )
                warning("Bit address exceeds 8-bits");
            if( isbmram(yypvt[-1].value) )
            yyval.value = (yypvt[-1].value-0x20)*8+yypvt[-0].value;
            else if( isbmsfr(yypvt[-1].value) )
                yyval.value = yypvt[-1].value + yypvt[-0].value;
            else
                warning("Invalid bit addressable RAM location");
      }
} break;
case 137:
# line 834 "ec51.y" /* | bitv */
{     if( pass2 ) {
            if( !isbit8(yypvt[-0].value) )
                warning("Bit address exceeds 8-bits");
            yyval.value = yypvt[-0].value;
      }
} break;
case 138:
# line 844 "ec51.y" /* bitv          : SYMBOL  */
{     if( yypvt[-0].sym->type == UNDEF && pass2 )
            error("Symbol %s undefined",yypvt[-0].sym->name);
      yyval.value = yypvt[-0].sym->value;
} break;
case 139:
# line 849 "ec51.y" /* | VALUE        */
{ yyval.value = yypvt[-0].value; } break;
case 140:
# line 852 "ec51.y" /* reg           : R0  */
{ yyval.value = 0; } break;
case 141:
# line 853 "ec51.y" /* | R1 */
{ yyval.value = 1; } break;
case 142:
# line 854 "ec51.y" /* | R2 */
{ yyval.value = 2; } break;
case 143:
# line 855 "ec51.y" /* | R3 */
{ yyval.value = 3; } break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 144:
# line 856 "ec51.y" /* | R4 */
{ yyval.value = 4; } break;
case 145:
# line 857 "ec51.y" /* | R5 */
{ yyval.value = 5; } break;
case 146:
# line 858 "ec51.y" /* | R6 */
{ yyval.value = 6; } break;
case 147:
# line 859 "ec51.y" /* | R7 */
{ yyval.value = 7; } break;
case 148:
# line 862 "ec51.y" /* regi        : R0 */
{ yyval.value = 0; } break;
case 149:
# line 863 "ec51.y" /* | R1 */
{ yyval.value = 1; } break;
case 150:
# line 865 "ec51.y" /* | R2 */
{ yyval.value = 0;
                        warning("Illegal indirect register: @r2");
} break;
case 151:
# line 868 "ec51.y" /* | R3 */
{ yyval.value = 0;
                        warning("Illegal indirect register: @r3");
} break;
case 152:
# line 871 "ec51.y" /* | R4 */
{ yyval.value = 0;
                        warning("Illegal indirect register: @r4");
} break;
case 153:
# line 874 "ec51.y" /* | R5 */
{ yyval.value = 0;
                        warning("Illegal indirect register: @r5");
} break;
case 154:
# line 877 "ec51.y" /* | R6 */
{ yyval.value = 0;
                        warning("Illegal indirect register: @r6");
} break;
case 155:
# line 880 "ec51.y" /* | R7 */
{ yyval.value = 0;
                        warning("Illegal indirect register: @r7");
} break;
case 156:
# line 885 "ec51.y" /* data8        : expr */
{
      if( pass2 ) {
            if( !isbit8(yypvt[-0].val.v) )
                  warning("Expression greater than 8-bits");
      }
      yyval.value = yypvt[-0].val.v;
} break;
case 157:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# line 895 "ec51.y" /* data16        : expr */
{
      if( pass2 ) {
            if( !isbit16(yypvt[-0].val.v) )
                  warning("Expression greater than 16-bits");
      }
      yyval.value = yypvt[-0].val.v;
} break;
case 158:
# line 905 "ec51.y" /* addr11        : expr */
{
            if( pass2 ) {
                  if( !isbit16(yypvt[-0].val.v)  )
                        warning("Address greater than 16-bits");
                  if( (yypvt[-0].val.v & size11) != ((lc+2) &
size11) )                     warning("Address outside current 2K
page");
            }
            set_md(yyval.mode,0);
            set_ov(yyval.mode, (yypvt[-0].val.v&0x0700)>>3 );
            set_sz(yyval.mode,1);
            set_b1(yyval.mode,yypvt[-0].val.v&0x00ff);
} break;
case 159:
# line 920 "ec51.y" /* addr16        : expr */
{
            if( pass2 ) {
                  if( !isbit16(yypvt[-0].val.v)  )
                        warning("Address greater than 16-bits");
            }
            set_md(yyval.mode,0);
            set_ov(yyval.mode, 0 );
            set_sz(yyval.mode,2);
            set_b1(yyval.mode, (yypvt[-0].val.v & 0xff00 ) >> 8 );
            set_b2(yyval.mode, (yypvt[-0].val.v & 0x00ff ) );
} break;
case 160:
# line 934 "ec51.y" /* relative      : expr */
{
            long offset;
            if( pass2 ) {
                  offset = yypvt[-0].val.v - (lc+2);
                  if( offset>127 || offset<-128 )
                     warning("Relative offset exceeds -128 / +127");
            }
            set_md(yyval.mode,0);
            set_ov(yyval.mode,0);
            set_sz(yyval.mode,1);
            set_b1(yyval.mode,offset);
} break;
            }
            goto yystack;  /* Stack new state and value. */
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.7 โปรแกรม EC51.C

```
/*  FILE: EC51.Y
 *  แฟ้มนี้บรรจุขั้นตอนของ Yacc Parser สำหรับโปรแกรมแปลภาษา
 *  โดยกำหนดให้มีฟังชันก์ดังต่อไปนี้
 *  error(), warning(), yyerror()
 *  genbyte(), genword(), genstr(), makeop()
 */
%{
#include <setjmp.h>
#include <stdio.h>
#define NOPE
#include "as31.h"
#undef NOPE
#define YYSTYPE union ystack
extern int lineno;
extern int dashl;
extern char *asmfile;
extern jmp_buf main_env;
extern FILE *listing;
int pass,fatal;
unsigned long lc;
static unsigned char bytebuf[1024];      /* used by dumplist() */
static int bytecount;
/* ---------------------- G R A M M E R ------------------------- */
/* ค่าของ TOKEN ต่างๆ */
%}
%token STRING
%token D_ORG
%token D_BYTE
%token D_WORD
%token D_SKIP
%token D_EQU
%token D_FLAG
%token D_END
%token ACALL
%token ADD
%token ADDC
%token AJMP
%token ANL
%token CJNE
%token CLR
%token CPL
%token DA
%token DEC
%token DIV
%token DJNZ
%token INC
%token JB
%token JBC
%token JC
%token JMP
%token JNB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%token JNC
%token JNZ
%token JZ
%token LCALL
%token LJMP
%token MOV
%token MOVC
%token MOVX
%token NOP
%token MUL
%token ORL
%token POP
%token PUSH
%token RET
%token RETI
%token RL
%token RLC
%token RR
%token RRC
%token SETB
%token SJMP
%token SUBB
%token SWAP
%token XCH
%token XCHD
%token XRL
%token AB
%token A
%token C
%token PC
%token DPTR
%token BITPOS
%token R0
%token R1
%token R2
%token R3
%token R4
%token R5
%token R6
%token R7
%token VALUE
%token SYMBOL
%left '+' '-'
%left '*' '/' '%'
%left '|' '&'
%start program
%%
program     :     linelist
{
}           ;
linelist    : linelist line
            | line
            ;
line        : undefsym ':' linerest
{
        if( pass1 ) {
            $1.sym->type = LABEL;
            $1.sym->value = lc;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        inclc($3.value);
        bytecount = 0;
}
                | linerest          { inclc($1.value); bytecount = 0; }
                ;
linerest        : directive '\n'    {
                                        $$.value = $1.value;
                                        if( dashl && pass2 )
                                            dumplist($2.str,1);
                                    }
                | instr '\n'                {
                                            $$.value = $1.value;
                                            if( dashl && pass2 )
                                                dumplist($2.str,1);
                                    }
                | '\n'                      {
                                            $$.value = 0;
                                            if( dashl && pass2 )
                                                dumplist($1.str,0);
                                    };
/* --------------------
 * DIRECTIVES:
 *
 */
directive       :  D_ORG defexpr
{
        lc = $3.val.v;
        if( pass2 ) emitaddr(lc);
        bytecount = 0;
        $$.value = 0;
}
                | D_BYTE blist      { $$.value = $2.value; }
                | D_WORD wlist      { $$.value = $2.value; }
                | D_SKIP defexpr    { $$.value = $2.val.v;
                                      if( pass2 )
                                          emitaddr(lc+$$.value); }
                | D_EQU undefsym  expr
{
        if( $3.val.d == 0 )
                error("Expression is undefined in pass 1");
        $2.sym->type = LABEL;
        $2.sym->value = $3.val.v;
        $$.value = 0;
}
        | D_FLAG SYMBOL ',' flag
{
        $2.sym->type = LABEL;
        $2.sym->value = $5.value;
        $$.value = 0;
}
                | D_END                     { $$.value = 0; }
                ;
defexpr             : expr
{
                if( $1.val.d == 0 )
                        error("Expression is undefined in pass 1");
                if( !(isbit16($1.val.v)) )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                  error("Value greater than 16-bits");
            $$.value = $1.val.v;
}
            ;
flag        : flagv BITPOS
{
      if( !isbit8($1.value) )
            warning("Bit address exceeds 8-bits");
      if( isbmram($1.value) )
            $$.value = ($1.value-0x20)*8+ $2.value;
      else if( isbmsfr($1.value) )
            $$.value = $1.value + $2.value;
      else
            warning("Invalid bit addressable RAM location");
}
            ;
flagv       : SYMBOL
{
      if( $1.sym->type == UNDEF )
            error("Symbol %s must be defined in pass 1",$1.sym->
name);
      $$.value = $1.sym->value;
}
            | VALUE                  { $$.value = $1.value; }
            ;
undefsym    : SYMBOL
{
      if( $1.sym->type != UNDEF && pass1)
            error("Attempt to redefine symbol: %s",$1.sym->name);
      $$.sym = $1.sym;
}
            ;
blist       : blist ',' data8
{
      if( pass2 ) genbyte($3.value);
      $$.value = $1.value + 1;
}
            | blist ',' STRING
{
      if( pass1 )
            $$.value = $1.value + $3.value;
      else {
            $$.value = $1.value + strlen($3.str);
            genstr($3.str);
            free($3.str);
      }
}
            | data8
{
      if( pass2 ) genbyte($1.value);
      $$.value = 1;
}
            | STRING
{
      if( pass1 )
            $$.value = $1.value;
      else {
            $$.value = strlen($1.str);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            genstr($1.str);
            free($1.str);
      }
}
            ;
wlist       : wlist ',' data16
{
      if( pass2 ) genword($3.value);
      $$.value = $1.value + 2;
}
            | data16
{
      if( pass2 ) genword($1.value);
      $$.value = 2;
}
            ;
/* --------------------
 * EXPRESSIONS:
 *
 */
expr        : '*'                 { $$.val.v = lc;
                                    $$.val.d = 1; }
            | '(' expr ')'              { $$.val.v = $2.val.v;
                                    $$.val.d = $2.val.d; }

            | '-' expr %prec '*'        { $$.val.v = -$2.val.v;
                                    $$.val.d = $2.val.d;  }

            | expr '|' expr             { $$.val.v = $1.val.v |
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }

            | expr '&' expr             { $$.val.v = $1.val.v &
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }

            | expr '*' expr             { $$.val.v = $1.val.v *
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }

            | expr '/' expr             { $$.val.v = $1.val.v /
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }

            | expr '%' expr             { $$.val.v = $1.val.v %
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }

            | expr '-' expr             { $$.val.v = $1.val.v -
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }

            | expr '+' expr             { $$.val.v = $1.val.v +
$3.val.v;
                                    $$.val.d = $1.val.d && $3.val.d; }
            | SYMBOL
{
      if( pass1 ) {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            $$.val.v = $1.sym->value;
            $$.val.d = ($1.sym->type != UNDEF);
      }
      else {
            if( $1.sym->type == UNDEF )
                  error("Undefined symbol %s",$1.sym->name);
            $$.val.v = $1.sym->value;
            $$.val.d = 1;
      }
}
            | VALUE           { $$.val.v = $1.val.v; $$.val.d=1; }
            ;
/* --------------------
 * INSTRUCTIONS:
 *
 */
instr       : NOP
                        { $$.value = makeop($1.op,NULL,0); }
            | ACALL addr11
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | AJMP addr11
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | ADD two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | ADDC two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | SUBB two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | XRL two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | XRL two_op2
                        { $$.value = makeop($1.op,&$2.mode,4); }
            | ANL two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | ANL two_op2
                        { $$.value = makeop($1.op,&$2.mode,4); }
            | ANL two_op3
                        { $$.value = makeop($1.op,&$2.mode,6); }
            | ORL two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | ORL two_op2
                        { $$.value = makeop($1.op,&$2.mode,4); }
            | ORL two_op3
                        { $$.value = makeop($1.op,&$2.mode,6); }
            | XCH two_op1
                        { if( get_md($2.mode) == 3 )
                              error("Immediate mode is illegal");
                          $$.value = makeop($1.op,&$2.mode,0);
                        }
            | INC single_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | INC DPTR
                        { $$.value = makeop($1.op,NULL,4); }
            | DEC single_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
            | DA A
                        { $$.value = makeop($1.op,NULL,0); }
            | DIV AB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                { $$.value = makeop($1.op,NULL,0); }
        | JMP '@' A '+' DPTR
                { $$.value = makeop($1.op,NULL,0); }
        | JMP '@' DPTR '+' A
                { $$.value = makeop($1.op,NULL,0); }
        | MUL AB
                { $$.value = makeop($1.op,NULL,0); }
        | RET
                { $$.value = makeop($1.op,NULL,0); }
        | RETI
                { $$.value = makeop($1.op,NULL,0); }
        | RL A
                { $$.value = makeop($1.op,NULL,0); }
        | RLC A
                { $$.value = makeop($1.op,NULL,0); }
        | RR A
                { $$.value = makeop($1.op,NULL,0); }
        | RRC A
                { $$.value = makeop($1.op,NULL,0); }
        | SWAP A
                { $$.value = makeop($1.op,NULL,0); }
        | XCHD two_op1
                { if( get_md($2.mode) != 2 )
                    error("Invalid addressing mode");
                  $$.value = makeop($1.op,&$2.mode,-2); }
        | CLR single_op2
                { $$.value = makeop($1.op,&$2.mode,0); }
        | CPL single_op2
                { $$.value = makeop($1.op,&$2.mode,0); }
        | SETB single_op2
                { if( get_md($2.mode) == 0 )
                    error("Invalid addressing mode");
                  $$.value = makeop($1.op,&$2.mode,-1); }
        | PUSH data8
                {
                  struct mode tmp;
                    set_md(tmp,0);
                    set_ov(tmp,0);
                    set_sz(tmp,1);
                    set_b1(tmp,$2.value);
                    $$.value = makeop($1.op,&tmp,0);
                }
        | POP data8
                {
                   struct mode tmp;
                     set_md(tmp,0);
                     set_ov(tmp,0);
                     set_sz(tmp,1);
                     set_b1(tmp,$2.value);
                     $$.value = makeop($1.op,&tmp,0);
                }
        | LJMP addr16
                { $$.value = makeop($1.op,&$2.mode,0); }
        | LCALL addr16
                { $$.value = makeop($1.op,&$2.mode,0); }
        | JC relative
                { $$.value = makeop($1.op,&$2.mode,0); }
        | JNC relative
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | JNZ relative
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | JZ relative
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | SJMP relative
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | CJNE three_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | JB two_op4
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | JNB two_op4
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | JBC two_op4
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | DJNZ two_op5
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | MOV two_op1
                        { $$.value = makeop($1.op,&$2.mode,0); }
              | MOV two_op2
                        { $$.value = makeop($1.op,&$2.mode,4); }
              | MOV two_op6
                        { $$.value = makeop($1.op,&$2.mode,6); }
              | MOVC A ',' '@' A '+' DPTR
                        { $$.value = makeop($1.op,NULL,0); }
              | MOVC A ',' '@' DPTR '+' A
                        { $$.value = makeop($1.op,NULL,0); }
              | MOVC A ',' '@' A '+' PC
                        { $$.value = makeop($1.op,NULL,1); }
              | MOVC A ',' '@' PC '+' A
                        { $$.value = makeop($1.op,NULL,1); }
              | MOVX A ',' '@' regi
                        { $$.value = makeop($1.op,NULL,$5.value); }
              | MOVX A ',' '@' DPTR
                        { $$.value = makeop($1.op,NULL,2); }
              | MOVX '@' regi ',' A
                        { $$.value = makeop($1.op,NULL,$3.value+3);
}
              | MOVX '@' DPTR ',' A
                        { $$.value = makeop($1.op,NULL,5); }
              ;

/* --------------------
 * ADDRESSING MODES:
 *
 */
two_op1           : A ',' reg
                        {
                              set_md($$.mode,0);
                              set_ov($$.mode, $3.value);
                              set_sz($$.mode, 0);
                        }
              | A ',' data8
                        {
                              set_md($$.mode,1);
                              set_ov($$.mode,0);
                              set_sz($$.mode,1);
                              set_bl($$.mode,$3.value);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
            | A ',' '@' regi
                        {
                                set_md($$.mode,2);
                                set_ov($$.mode,$4.value);
                                set_sz($$.mode,0);
                        }
            | A ',' '#' data8
                        {
                                set_md($$.mode,3);
                                set_ov($$.mode,0);
                                set_sz($$.mode,1);
                                set_b1($$.mode,$4.value);
                        }
            ;
two_op2             : data8 ',' A
                        {
                                set_md($$.mode,0);
                                set_ov($$.mode,0);
                                set_sz($$.mode,1);
                                set_b1($$.mode,$1.value);
                        }
            | data8 ',' '#' data8
                        {
                                set_md($$.mode,1);
                                set_ov($$.mode,0);
                                set_sz($$.mode,2);
                                set_b1($$.mode,$1.value);
                                set_b2($$.mode,$4.value);
                        }
            ;
two_op3             : C ',' bit
                        {
                                set_md($$.mode,0);
                                set_ov($$.mode,0);
                                set_sz($$.mode,1);
                                set_b1($$.mode,$3.value);
                        }
            | C ',' '/' bit
                        {
                                set_md($$.mode,1);
                                set_ov($$.mode,0);
                                set_sz($$.mode,1);
                                set_b1($$.mode,$4.value);
                        }
            | C ',' '!' bit
                        {
                                set_md($$.mode,1);
                                set_ov($$.mode,0);
                                set_sz($$.mode,1);
                                set_b1($$.mode,$4.value);
                        }
            ;
two_op4             : bit ',' rel
                        {
                                set_md($$.mode,0);
                                set_ov($$.mode,0);
                                set_sz($$.mode,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    set_b1($$.mode,$1.value);
                                    set_b2($$.mode,$3.value);
                              }
            ;
two_op5           : reg ',' rel2
                              {
                                    set_md($$.mode,0);
                                    set_ov($$.mode,$1.value);
                                    set_sz($$.mode,1);
                                    set_b1($$.mode,$3.value);
                              }
            | data8 ',' rel
                              {
                                    set_md($$.mode,1);
                                    set_ov($$.mode,0);
                                    set_sz($$.mode,2);
                                    set_b1($$.mode,$1.value);
                                    set_b2($$.mode,$3.value);
                              }
            ;
two_op6           : reg ',' A
                              {
                                    set_md($$.mode,0);
                                    set_ov($$.mode,$1.value);
                                    set_sz($$.mode,0);
                              }
            | reg ',' data8
                              {
                                    set_md($$.mode,1);
                                    set_ov($$.mode,$1.value);
                                    set_sz($$.mode,1);
                                    set_b1($$.mode,$3.value);
                              }
            | reg ',' '#' data8
                              {
                                    set_md($$.mode,2);
                                    set_ov($$.mode,$1.value);
                                    set_sz($$.mode,1);
                                    set_b1($$.mode,$4.value);
                              }
            | data8 ',' reg
                              {
                                    set_md($$.mode,3);
                                    set_ov($$.mode,$3.value);
                                    set_sz($$.mode,1);
                                    set_b1($$.mode,$1.value);
                              }
            | data8 ',' data8
                              {
                                    set_md($$.mode,4);
                                    set_ov($$.mode,0);
                                    set_sz($$.mode,2);
                                    set_b1($$.mode,$3.value);
                                    set_b2($$.mode,$1.value);
                              }
            | data8 ',' '@' regi
                              {
                                    set_md($$.mode,5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        set_ov($$.mode,$4.value);
                        set_sz($$.mode,1);
                        set_b1($$.mode,$1.value);
                    }
        | '@' regi ',' A
                    {
                        set_md($$.mode,6);
                        set_ov($$.mode,$2.value);
                        set_sz($$.mode,0);
                    }
        | '@' regi ',' data8
                    {
                        set_md($$.mode,7);
                        set_ov($$.mode,$2.value);
                        set_sz($$.mode,1);
                        set_b1($$.mode,$4.value);
                    }
        | '@' regi ',' '#' data8
                    {
                        set_md($$.mode,8);
                        set_ov($$.mode,$2.value);
                        set_sz($$.mode,1);
                        set_b1($$.mode,$5.value);
                    }
        | DPTR ',' '#' data16
                {
                    set_md($$.mode,9);
                    set_ov($$.mode,0);
                    set_sz($$.mode,2);
                    set_b1($$.mode, ($4.value & 0xff00) >> 8 );
                    set_b2($$.mode, ($4.value & 0x00ff) );
                }
        | C ',' bit
                    {
                        set_md($$.mode,10);
                        set_ov($$.mode,0);
                        set_sz($$.mode,1);
                        set_b1($$.mode,$3.value);
                    }
/*
 * Productions 2 สิ่งดังต่อไปนี้ไม่สามารถถูกแสดงโดย:
 *      bit ',' C เพราะว่า yacc ให้ลด/ลดข้อผิดพลาดต่างๆ ที่จะเกิดขึ้น
 */
| data8 ',' C
                    {
                        set_md($$.mode,11);
                        set_ov($$.mode,0);
                        set_sz($$.mode,1);
                        set_b1($$.mode,$1.value);
                    }
        | data8 BITPOS ',' C
{
    if( pass2 ) {
        if( !isbit8($1.value) )
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                warning("Bit address exceeds 8-bits");
            if( isbmram($1.value) )
                set_b1($$.mode, ($1.value-0x20)*8+ $2.value );
            else if( isbmsfr($1.value) )
                set_b1($$.mode, $1.value + $2.value );
            else
                warning("Invalid bit addressable RAM location");
      }
      set_md($$.mode,11);
      set_ov($$.mode,0);
      set_sz($$.mode,1);
};
single_op1  : A
                        {
                              set_md($$.mode,0);
                              set_ov($$.mode,0);
                              set_sz($$.mode,0);
                        }
            | reg
                        {
                              set_md($$.mode,1);
                              set_ov($$.mode,$1.value);
                              set_sz($$.mode,0);
                        }
            | data8
                        {
                              set_md($$.mode,2);
                              set_ov($$.mode,0);
                              set_sz($$.mode,1);
                              set_b1($$.mode,$1.value);
                        }
            | '@' regi
                        {
                              set_md($$.mode,3);
                              set_ov($$.mode,$2.value);
                              set_sz($$.mode,0);
                        }
            ;
single_op2  : A
                        {
                              set_md($$.mode,0);
                              set_ov($$.mode,0);
                              set_sz($$.mode,0);
                        }
            | C
                        {
                              set_md($$.mode,1);
                              set_ov($$.mode,0);
                              set_sz($$.mode,0);
                        }
            | bit
                        {
                              set_md($$.mode,2);
                              set_ov($$.mode,0);
                              set_sz($$.mode,1);
                              set_b1($$.mode,$1.value);
                        }
            ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
three_op1    : A ',' data8 ',' rel
                        {
                              set_md($$.mode,0);
                              set_ov($$.mode,0);
                              set_sz($$.mode,2);
                              set_b1($$.mode,$3.value);
                              set_b2($$.mode,$5.value);
                        }
             | A ',' '#' data8 ',' rel
                        {
                              set_md($$.mode,1);
                              set_ov($$.mode,0);
                              set_sz($$.mode,2);
                              set_b1($$.mode,$4.value);
                              set_b2($$.mode,$6.value);
                        }
             | reg ',' '#' data8 ',' rel
                        {
                              set_md($$.mode,2);
                              set_ov($$.mode,$1.value);
                              set_sz($$.mode,2);
                              set_b1($$.mode,$4.value);
                              set_b2($$.mode,$6.value);
                        }
             | '@' regi ',' '#' data8 ',' rel
                        {
                              set_md($$.mode,3);
                              set_ov($$.mode,$2.value);
                              set_sz($$.mode,2);
                              set_b1($$.mode,$5.value);
                              set_b2($$.mode,$7.value);
                        };
rel          : expr
{
             long offset;
             if( pass2 ) {
                  offset = $1.val.v - (lc+3);
                  if( offset > 127 || offset < -128 )
                       warning("Relative offset exceeds -128 / +127");
                  $$.value = offset;
             }
};
/*
 * This production แตกต่างจาก above, 1 ตัว !
 */
rel2         : expr
{
             long offset;
             if( pass2 ) {
                  offset = $1.val.v - (lc+2); /* different! */
                  if( offset > 127 || offset < -128 )
                       warning("Relative offset exceeds -128 / +127");
                  $$.value = offset;
             }
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
bit         : bitv BITPOS
{
      if( pass2 ) {
            if( !isbit8($1.value) )
                  warning("Bit address exceeds 8-bits");
            if( isbmram($1.value) )
                  $$.value = ($1.value-0x20)*8+$2.value;
            else if( isbmsfr($1.value) )
                  $$.value = $1.value + $2.value;
            else
                  warning("Invalid bit addressable RAM location");
      }
}
            | bitv
{
      if( pass2 ) {
            if( !isbit8($1.value) )
                  warning("Bit address exceeds 8-bits");
            $$.value = $1.value;
      }
};
bitv        : SYMBOL
{
      if( $1.sym->type == UNDEF && pass2 )
            error("Symbol %s undefined",$1.sym->name);
      $$.value = $1.sym->value;
}
            | VALUE              { $$.value = $1.value; };
reg         : R0          { $$.value = 0; }
            | R1          { $$.value = 1; }
            | R2          { $$.value = 2; }
            | R3          { $$.value = 3; }
            | R4          { $$.value = 4; }
            | R5          { $$.value = 5; }
            | R6          { $$.value = 6; }
            | R7          { $$.value = 7; };
regi        : R0          { $$.value = 0; }
            | R1          { $$.value = 1; }
            | R2          { $$.value = 0;
                            warning("Illegal indirect register: @r2");
}
            | R3
                        { $$.value = 0;
                          warning("Illegal indirect register: @r3");
}
            | R4
                        { $$.value = 0;
                          warning("Illegal indirect register: @r4");
}
            | R5
                        { $$.value = 0;
                          warning("Illegal indirect register: @r5");
}
            | R6
                        { $$.value = 0;
                          warning("Illegal indirect register: @r6");
}
            | R7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        { $$.value = 0;
                          warning("Illegal indirect register: @r7");
}
            ;
data8       : expr
{
      if( pass2 ) {
            if( !isbit8($1.val.v) )
                  warning("Expression greater than 8-bits");
      }
      $$.value = $1.val.v;
};
data16            : expr
{
      if( pass2 ) {
            if( !isbit16($1.val.v) )
                  warning("Expression greater than 16-bits");
      }
      $$.value = $1.val.v;
};
addr11            : expr
{
            if( pass2 ) {
                  if( !isbit16($1.val.v)  )
                        warning("Address greater than 16-bits");
                  if( ($1.val.v & size11) != ((lc+2) & size11) )
                        warning("Address outside current 2K page");
            }
            set_md($$.mode,0);
            set_ov($$.mode, ($1.val.v&0x0700)>>3 );
            set_sz($$.mode,1);
            set_b1($$.mode,$1.val.v&0x00ff);
};
addr16            : expr
{
            if( pass2 ) {
                  if( !isbit16($1.val.v)  )
                        warning("Address greater than 16-bits");
            }
            set_md($$.mode,0);
            set_ov($$.mode, 0 );
            set_sz($$.mode,2);
            set_b1($$.mode, ($1.val.v & 0xff00 ) >> 8 );
            set_b2($$.mode, ($1.val.v & 0x00ff ) );
};
relative    : expr
{
            long offset;
            if( pass2 ) {
                  offset = $1.val.v - (lc+2);
                  if( offset>127 || offset<-128 )
                     warning("Relative offset exceeds -128 / +127");
            }
            set_md($$.mode,0);
            set_ov($$.mode,0);
            set_sz($$.mode,1);
            set_b1($$.mode,offset);
};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
%%
/* ---------------------------------------------------------- */
yyerror(s)
char *s;
{
      error(s);
}
/* ฟังก์ชันแจ้งข้อผิดพลาด:
 * ทำการแจ้งข้อผิดพลาดที่เกิดขึ้นสิ่งนี้เป็นสาเหตุให้โปรแกรมจบการทำงานทันที
 */
error(cs,a1,a2,a3,a4,a5,a6)
char *cs,*a1,*a2,*a3,*a4,*a5,*a6;
{
      fprintf(stderr,"File: %s, line: %d, ",asmfile,lineno);
      fprintf(stderr,cs,a1,a2,a3,a4,a5,a6);
      fprintf(stderr,".\n");
      longjmp(main_env,1);
}
/* ฟังก์ชันแจ้งคำเตือน
 * ทำการแจ้งคำเตือนถึงสิ่งที่ผิดพลาดเล็กๆ น้อยๆ
 * ที่โปรแกรมสามารถดำเนินการต่อไปได้
 */
warning(cs,a1,a2,a3,a4,a5,a6)
char *cs,*a1,*a2,*a3,*a4,*a5,*a6;
{
      fatal++;
      fprintf(stderr,"File: %s, line: %d, ",asmfile,lineno);
      fprintf(stderr,cs,a1,a2,a3,a4,a5,a6);
      fprintf(stderr,".\n");
}
/* makeop
 *      ฟังก์ชันนี้สร้าง Opcode โดยอาศัยสัญลักษณ์จากตารางสัญลักษณ์ (SYMBOL.C)
 *      จากรายการที่ได้กระทำเอาไว้ในขั้นตอนต่างๆ , และโครงสร้างโหมดที่ตำแหน่งที่อยู่
 *      ฟังก์ชันนี้ถูกเรียกใช้ แต่จะทำการสร้างรหัส (Opcode)ใน PASS 2 เท่านั้น
 *      Opcode ซึ่งเป็นผลไบท์ผ่านถึง genbyte () ในขั้นตอนการผลิตรหัสใน PASS 2
 *      คืนค่าของไบท์เพื่อใช้ในการผลิตรหัส
 */
makeop(op,m,add)
struct opcode *op;
struct mode *m;
{
      register unsigned int newop;
      if( m == NULL ) {
            if(pass2) genbyte(op->bytes[0+add]);
            return(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        if( pass2 ) {
                newop = op->bytes[ get_md(*m)+add ] | get_ov(*m);
                genbyte(newop);
                if( get_sz(*m) > 0 ) genbyte( get_b1(*m) );
                if( get_sz(*m) > 1 ) genbyte( get_b2(*m) );
        }
        return( get_sz(*m)+1 );
}
```

/* inclc :

* จำนวนค่าที่เพิ่มขึ้นโดยมีตัวนับตำแหน่งโดย 'i' จำนวน

* ตรวจสอบดูถ้า 'i' ว่าเกิน 64K หรือไม่

* ตรวจสอบการใช้พื้นที่หน่วยความจำของรหัส (การใช้ส่วนพื้นที่บิทมากมาย).

*/

```
#define indx(a)  ( (a)/(sizeof(long)*8) )
#define bit(a)   ( 1 << ((a)%(sizeof(long)*8)) )
#define getloc(a) (regions[indx(a)] & bit(a))
#define setloc(a) (regions[indx(a)] |= bit(a))
inclc(i)
{
        static unsigned long regions[ 0x10000/(sizeof(long)*8) ];
        while(i-- > 0) {
                if( pass2 && getloc(lc) )
                        error("Location counter overlaps");
                if( pass2 ) setloc(lc);
                lc += 1;
        }
                if( lc > 0xffff )
                error("Location counter has exceeded 16-bits");
}
```

/* padline :

* Routine นี้ทำการคืนค่าสตริงใหม่ ซึ่งคือ Equivilant ถึง

* ' line ' ยกเว้นแท็ปทั้งหมดนั้นถูกขยายให้ที่ว่าง และ

* ความยาวทั้งหมดถูกตัดให้ยาวถึง 60 ตัวอักษร

*/

```
char *padline(line)
char *line;
{
        static char newline[61];
        char *p1;
        int pos=0,nxtpos;
        for(p1=line; pos<sizeof(newline)-1 && *p1; p1++ ) {
                if( *p1 == '\t' ) {
                        nxtpos = pos+8-pos%8;
                        while(pos<sizeof(newline)-1 && pos <= nxtpos)
                                newline[pos++] =' ';
                } else if( *p1 != '\n' )
                        newline[pos++]= *p1;
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        newline[pos] = '\0';
        return(newline);
}
/* dumplist :
 *      ผลลัพธ์ของตัวนับตำแหน่งปัจจุบัน bytebuf []อะเรย์, และ
 *      สตริง 'txt' ถึงแฟ้มการรวบรวมรายละเอียด (Listing File )
 *      Routine นี้ถูกเรียกใช้นับทุกเส้นต้นกำเนิดที่เจอในแฟ้มต้นฉบับ
 * (ใน PASS 2 เท่านั้น และถ้าการรวบรวมรายชื่อเปิด).
 */
dumplist(txt,show)
char *txt;
{
        int i,j;
        fprintf(listing,show?"%04X: ":"      ",lc);      j=0;
        for(i=0; i<bytecount; i++ ) {
                fprintf(listing,"%02X ",bytebuf[i]);
                if( ++j >= 4 ) {  j = 0;
                        fprintf(listing,"\n      "); }      }
        while(++j <= 4)
                fprintf(listing,"   ");
        fprintf(listing," %s\n",padline(txt));
}
/* gen routines:
 *      สถานที่ที่นำข้อมูลเก็บเข้าไปในเข้าไปใน bytebuf [] อะเรย์ และ
 *      เรียก emitbyte() กับไบท์
 */
genbyte(b)
int b;
{
        if( bytecount < sizeof(bytebuf) )
                bytebuf[bytecount++] = b;
        emitbyte(b);
}
genstr(s)
char *s;
{
        while( *s )
                genbyte(*s++);
}
genword(w)
unsigned long w;
{
        genbyte( (w & 0xff00) >> 8 );
        genbyte( (w & 0x00ff) );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก.8 โปรแกรม EC51.Y

```
/* ไฟล์ " Y_TAB.H" ภายในเป็นการกำหนดค่าของ TOKEN ต่างๆ */

# define STRING 257
# define D_ORG 258
# define D_BYTE 259
# define D_WORD 260
# define D_SKIP 261
# define D_EQU 262
# define D_FLAG 263
# define D_END 264
# define ACALL 265
# define ADD 266
# define ADDC 267
# define AJMP 268
# define ANL 269
# define CJNE 270
# define CLR 271
# define CPL 272
# define DA 273
# define DEC 274
# define DIV 275
# define DJNZ 276
# define INC 277
# define JB 278
# define JBC 279
# define JC 280
# define JMP 281
# define JNB 282
# define JNC 283
# define JNZ 284
# define JZ 285
# define LCALL 286
# define LJMP 287
# define MOV 288
# define MOVC 289
# define MOVX 290
# define NOP 291
# define MUL 292
# define ORL 293
# define POP 294
# define PUSH 295
# define RET 296
# define RETI 297
# define RL 298
# define RLC 299
# define RR 300
# define RRC 301
# define SETB 302
# define SJMP 303
# define SUBB 304
# define SWAP 305
# define XCH 306
# define XCHD 307
# define XRL 308
# define AB 309
# define A 310
# define C 311
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
# define PC 312
# define DPTR 313
# define BITPOS 314
# define R0 315
# define R1 316
# define R2 317
# define R3 318
# define R4 319
# define R5 320
# define R6 321
# define R7 322
# define VALUE 323
# define SYMBOL 324
```

รูปที่ ก.9 โปรแกรม Y_TAB.H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*FILE: EC51.H
* บรรจุโครงสร้างต่างๆ เอาไว้ เช่น Opcode, Symbol
*/

struct opcode {
      char *name;
      int type;
      unsigned char *bytes;
};

struct symbol {
      char *name;
      int type;
      long value;
      struct symbol *next;
};

#define UNDEF     0
#define LABEL     1
struct mode {
      unsigned char mode;              /* value to index with */
      unsigned char size;              /* # of bytes used */
      unsigned char orval;             /* value OR'd to obcode */
      unsigned char byte1;             /* extra byte 1 */
      unsigned char byte2;             /* extra byte 2 */
};

#define set_md(m,a)      ((m).mode=(a))
#define set_sz(m,a)      ((m).size=(a))
#define set_ov(m,a)      ((m).orval=(a))
#define set_b1(m,a)      ((m).byte1=(a))
#define set_b2(m,a)      ((m).byte2=(a))
#define get_md(m)  ((m).mode)
#define get_sz(m)  ((m).size)
#define get_ov(m)  ((m).orval)
#define get_b1(m)  ((m).byte1)
#define get_b2(m)  ((m).byte2)

struct value {
      long v;
      unsigned char d;          /* Expression defined flag. */
};
union ystack {
      long value;
      struct value val;
      struct opcode *op;
      struct symbol *sym;
      struct mode mode;
      char *str;
};
#define isbmram(a)       (((a)&0xf0)==0x20)
#define isbmsfr(a)       (((a)&0x80) && !((a) & 0x07))
#define size8            (~0x00ff)
#define size11           (~0x07ff)
#define size13           (~0x1fff)
#define size16           (~0xffff)
#define size10           (~0x03ff)
```

เอกสารนี้จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define size12          (~0x0fff)
#define size15          (~0x7fff)
#define isbit8(v) ( !( ((v)>=0) ? (v)&size8 : -(v)>=128) )
#define isbit11(v)( !( ((v)>=0) ? (v)&size11 : (-(v))&size10 ) )
#define isbit13(v)( !( ((v)>=0) ? (v)&size13 : (-(v))&size12 ) )
#define isbit16(v)( !( ((v)>=0) ? (v)&size16 : (-(v))&size15 ) )
#define HASHTABSIZE          1000
#define pass1                (!pass)
#define pass2                (pass)
#ifdef NOPE
#else
#include "Y_TAB.H"
#endif
```

รูปที่ ก.10 โปรแกรม EC51.H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานโปรแกรม ECOM51 Assembler

1.1 รายละเอียดทั่วไปของ ECOM51 Assembler

ECOM51 เป็นโปรแกรม 8051 แอสเซมเบลอร์ ใช้สำหรับทำการแปล Source Program ซึ่งเขียนเป็นภาษาแอสเซมบลีของ MCS-51 (File.Asm) จะได้เอาต์พุตไฟล์เป็น Intel Hex File (File.Hex) การเขียน Source File ให้กระทำจากโปรแกรม Editor ใดๆ ก็ได้ จากนั้นจึงนำมาแปลด้วย ECOM51 ซึ่งทำได้โดยการเรียกใช้โปรแกรม ECOM51.EXE และตามด้วยชื่อ Source Program ที่เขียนขึ้น การแปลของโปรแกรม ECOM51 จะกระทำในแบบ 2 Pass Assembler ซึ่งหากมีข้อผิดพลาดใดๆ ECOM51 จะแสดงให้ทราบในขณะที่แปล ลักษณะของ Source Program จะเป็นไปตามมาตรฐานของการเขียนโปรแกรม Assembly ทั่วไป

1.2 การใช้งาน ECOM51 Assembler

โปรแกรมแอสเซมเบลอร์ที่ใช้งานคือ ECOM51.EXE โดย Assembler จะทำการแปล Source File ซึ่งถูกกำหนดไว้เป็น File.Asm เมื่อทำการแปลแล้ว จะได้ File Output เป็น Intel - Hex File ซึ่งถูกกำหนดไว้เป็น File.Hex โดย File.Hex นี่เอง ที่ผู้ใช้สามารถนำไปโหลดเข้าบอร์ดต่างๆ หรืออัดเข้าตัว EPROM เพื่อนำไปทดสอบโปรแกรมตามที่เขียนได้ นอกจากนี้ ECOM51 ยังสามารถสร้าง Listing File ต่างๆ ได้ตามรูปแบบของ Option ดังนี้

A>ECOM51 <-Option> Source File

โดยที่

Source File : คือ ชื่อ File ของโปรแกรมนามสกุลของ File กำหนดให้เป็น .Asm เท่านั้น

<-Option> : เป็นส่วนของการกำหนด Option ต่างๆ ประกอบด้วยเครื่องหมาย "-" และตามด้วยตัวอักษรย่อเพียงตัวเดียว ซึ่งการกำหนด Option นี้สามารถกำหนดได้ 1 ตัวในการแปลแต่ละครั้งรายของ Option จะมีดังนี้

-L หมายถึง ให้สร้าง File แสดงรายละเอียด (Listing File) โดย Output File ที่ได้คือ File.Lst

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A:\>ECOM51 Test.Asm

หมายถึง Source : Test.Asm

Listing : ไม่สร้าง

Hex : สร้าง Hex File ชื่อ Test.Hex

A:\>ECOM51 –L Testl.Asm

หมายถึง Source : Testl.Asm

Listing : สร้าง Listing File ชื่อ Testl.Lst

Hex : สร้าง Hex File ชื่อ Testl.Hex

1.2.1 ข้อกำหนดของ Source File

1) ในแต่ละบรรทัดต้องมีคำสั่งในภาษาแอสเซมบลี หรือ Assembly Directive เพียงหนึ่งคำสั่งเท่านั้น

2) Label Declaration ทุก Label ต้องปิดท้ายด้วย ":"

3) โปรแกรม ECOM51 จะมองตัวอักษรตัวเล็กหรือตัวใหญ่เหมือนกัน เช่น hh: จะเหมือนกับ HH:

4) บรรทัดว่างในโปรแกรมต้นแบบจะยังคงปรากฏอยู่ใน File แสดงรายละเอียด (File.Lst) แต่ใน File แสดงรหัสภาษาเครื่อง (File.Hex) จะถูกตัดทิ้งไป

5) Source File ในแต่ละบรรทัดต้องประกอบด้วยตัวอักษรไม่เกิน 128 ตัวอักษร ซึ่งใน แต่ละบรรทัดต้องมีรูปแบบดังนี้

[Label:] Operation [Operand1],[Operand2] [;Comment]

Label เป็นกลุ่มของตัวอักษร และตัวเลขที่จะใช้เป็นตัวแปร หรือตำแหน่งหน่วยความจำสำหรับอ้างอิงในโปรแกรม Label ต้องขึ้นต้นด้วยตัวอักษร A-Z หรือ "_" "." "?" และต้องปิดท้ายด้วย ":" เสมอ

Operation ส่วนนี้จะเป็นคำสั่งมาตรฐานของ MCS51 เช่น คำสั่ง MOV INC DEC ADD

Operand เป็นกลุ่มของตัวอักษร และตัวเลขซึ่งอาจจะอยู่ในรูปของการกระทำทางคณิตศาสตร์หรือการกระทำทางด้าน Logic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Comment หมายถึง ประโยคหรือข้อความที่ตามหลังเครื่องหมาย ";" โดยทั่วไปมักจะใช้อธิบายการทำงานของโปรแกรมสั้นๆ เพื่อให้ทำความเข้าใจได้ง่ายขึ้น เครื่องหมาย ";" จะอยู่ส่วนใดของโปรแกรมก็ได้ ซึ่งข้อความที่ตามหลังเครื่องหมาย ";" จะไม่มีผลต่อการแปลของ Assembler

1.2.2 Assembler Directive

1) DB Expression [,Expression...]

DB (Define Byte) คำสั่งนี้ใช้กำหนดค่าข้อมูลเป็นไบต์ลงในหน่วยความจำโปรแกรม สามารถใส่ข้อความรหัสแอสกี้ (Ascii String) ลงตำแหน่งในหน่วยความจำได้ โดยข้อความนั้นต้องอยู่ในเครื่องหมาย Quotes '_' และยังกำหนดค่าข้อมูลที่เป็นตัวเลข และตัวอักษรในบรรทัดเดียวกันได้โดยการคั่นด้วยเครื่องหมาย "," เช่น

DB 'This is a string',0

2) DW Expression [,Expression...]

DW (Define Word) ใช้ในการเขียนข้อมูล 16 บิท ของหน่วยความจำโปรแกรมโดยการกำหนดค่าข้อมูลไบท์ต่ำก่อนแล้วตามด้วยข้อมูลไบต์สูง โดยข้อมูลแต่ละ Word ต้องคั่นด้วยเครื่องหมาย "," เช่นเดียวกับคำสั่ง DB

3) ORG Expression

ORG (Origin) ใช้สำหรับกำหนด Address เริ่มต้นของโปรแกรม (Origin Program Location)

4) END

END คำสั่งนี้มีไว้เพื่อบอกให้ Assembler ทราบถึงจุดสิ้นสุดของโปรแกรมว่าอยู่ที่ตำแหน่งใดคำสั่งนี้จำเป็นต้องใส่ไว้ท้ายโปรแกรมด้วย

5) EQU Symbol_name Expression

Symbol_name Bit Expression สำหรับการกำหนดค่าให้กับ Symbol_name (หรือที่เรียกกันว่า Label) โดยสามารถใช้คำสั่ง EQU หรือคำสั่ง BIT ก็ได้ ซึ่งจะทำงานเหมือนกันทุกประการ

6) Expression

การกำหนดค่าใน Expression ของแต่ละคำสั่งในภาษา Assembler นั้น สามารถใส่ตัว Operator ต่างๆ เพื่อความสะดวกในการใช้งานได้ รวมทั้งยังกำหนดเป็นเลขฐานต่างๆ ได้ด้วย สรุปได้ดังนี้

1) โดยปกติตัวเลขจะถือว่าเป็นเลขฐานสิบเสมอ แต่ถ้าต้องการกำหนดเป็นเลขฐานสิบหก (Hexadecimal) จะต้องเขียนนำหน้าด้วยตัวเลข และลงท้ายด้วยตัวอักษร H เช่น 12H, 0F7H และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าต้องการกำหนดเป็นเลขฐานแปด (Octal) ให้ลงท้ายด้วยตัวอักษร O หรือ Q และถ้าต้องการเป็นเลขฐานสอง (Binary) ให้ลงท้ายด้วย B

2) สามารถกำหนด Operator ในการคำนวณได้ คือ + - * / !

3) สามารถกระทำทางด้าน Logic ได้ คือ AND OR

7) Bit Operation

ECOM51 มีความสามารถอ้างตัวแปรในระดับ Bit ได้ โดยใช้หลักการ h.m ซึ่ง n คือ ค่า Byte Address และ m คือ หมายเลข Bit Address เท่านั้น เช่น 22H.3 จะมีค่าเท่ากับ 13H รวมทั้งจะใช้กับชื่อมาตรฐานได้ด้วย เช่น ACC.7 P1.2

8) ECOM51 Error Massage

"Value greater than 16-bits"

"Bit address exceeds 8-bits"

"Expression greater than 16-bits"

"Address greater than 16-bits"

"Address outside current 2K page"

"Expression is undefined in pass 1"

"Invalid addressing mode"

"Invalid bit addressable RAM location"

"Relative offset exceeds -128 / +127"

"Symbol must be defined in pass 1"

"Attempt to redefine symbol"

"Undefined symbol"

"Illegal indirect register: @r2"

"Illegal indirect register: @r3"

"Illegal indirect register: @r4"

"Illegal indirect register: @r5"

"Illegal indirect register: @r6"

"Illegal indirect register: @r7"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

สมยศ จุณณะปิยะ. **การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51**. กรุงเทพฯ : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2537

วิบูลย์ ชื่นแยก. **ไมโครโปรเซสเซอร์**. กรุงเทพฯ: สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ. 2525

อัชฌา ธีรเชษฐมงคล, ดร. **โครงสร้างข้อมูล**. ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ มหาวิทยาลัยสงขลานครินทร์ หาดใหญ่. 2527

พ.อ. เจนวิทย์ เหลืองอร่าม และปิยวิทย์ เหลืองอร่าม. **การเขียนโปรแกรมสำหรับ Applications ด้วย C/C++** บริษัท ธรรมสาร จำกัด พ.ศ. 2543

Aho, A.V., Ravi Sethi , and J.D. Ullman, **Compilers Principles, Techniques, and Tools**, Addisos – Wesley Pulishing Company, 1986

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้แต่งปริญญานิพนธ์	นายกิตติคุณ อินอ่อน
วัน เดือน ปีเกิด	23 มีนาคม พ.ศ. 2523
สถานที่เกิด	จังหวัดอุทัยธานี
ภูมิลำเนาเดิม	5 หมู่ที่4 ตำบลดอนกลอย อำเภอหนองขาหย่าง จังหวัดอุทัยธานี 61130
ที่อยู่ปัจจุบัน	5 หมู่ที่4 ตำบลดอนกลอย อำเภอหนองขาหย่าง จังหวัดอุทัยธานี 61130
โทรศัพท์	0-5659-7179

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเก้านิ้ว
มัธยมศึกษาตอนต้น	โรงเรียนหนองขาหย่างวิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคอุทัยธานี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคอุทัยธานี
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คติพจน์ ไม่มีคำว่าสายสำหรับการเริ่มต้นใหม่

ประวัติผู้แต่ง

ชื่อผู้แต่งปริญญานิพนธ์	นายอิทธิพล ไชยถาวร
วัน เดือน ปีเกิด	4 มีนาคม พ.ศ. 2522
สถานที่เกิด	จังหวัดสงขลา
ภูมิลำเนาเดิม	46/130 ถ. นิพัทธ์สงเคราะห์ 3 อำเภอหาดใหญ่ จังหวัดสงขลา 90110
ที่อยู่ปัจจุบัน	46/130 ถ. นิพัทธ์สงเคราะห์ 3 อำเภอหาดใหญ่ จังหวัดสงขลา 90110
โทรศัพท์	0-7422-0535

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนพลวิทยา
มัธยมศึกษาตอนต้น	โรงเรียนหาดใหญ่อำนวยวิทย์
ประกาศนียบัตรวิชาชีพ (ปวช.)	โรงเรียนสงขลาเทคโนโลยี
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	โรงเรียนสงขลาเทคโนโลยี
ปริญญาตรี	สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	พรุ่งนี้ยังมีหวังจงสู่ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อผู้แต่งปริญญานิพนธ์	นายนิติพงษ์ ไคขุนทด
วัน เดือน ปีเกิด	15 ตุลาคม พ.ศ. 2523
สถานที่เกิด	จังหวัดนครราชสีมา
ภูมิลำเนาเดิม	78 หมู่ที่ 16 ตำบลบ้านเพชร อำเภอบำเหน็จณรงค์ จังหวัดชัยภูมิ 36160
ที่อยู่ปัจจุบัน	78 หมู่ที่ 16 ตำบลบ้านเพชร อำเภอบำเหน็จณรงค์ จังหวัดชัยภูมิ 36160
โทรศัพท์	0-4484-2646

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนชุมชนบ้านเพชร
มัธยมศึกษาตอนต้น	โรงเรียนบุญวัฒนา
ประกาศนียบัตรวิชาชีพ (ปวช.)	-
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคลวิทยาเขต ภาคตะวันออกเฉียงเหนือ สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ปริญญาตรี	-
ผลงานที่ได้รับรางวัล	-
ทุนการศึกษา	-
คติพจน์	สูงสุดคืนสู่สามัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้