บทที่ 3

การวิเคราะห์และออกแบบระบบ

จากการศึกษาและทำความเข้าใจเกี่ยวกับการทำงานของระบบเครื่องคอมพิวเตอร์ PDP-8 ได้ข้อสรุปว่างานวิทยานิพนธ์นี้จะเกี่ยวข้องกับการทำงานหลักสองส่วน ส่วนแรกเป็นส่วนการติดต่อกับผู้ใช้และส่วนที่สองเป็นส่วนจำลองการทำงานของเครื่องคอมพิวเตอร์ ในบทนี้และบทถัด ๆ ไป จะเรียกโปรแกรมจำลองการทำงานของระบบคอมพิวเตอร์ที่พัฒนาในงานวิทยานิพนธ์นี้อย่างสั้น ๆ ว่า "โปรแกรมจำลองฯ"

ส่วนการติดต่อกับผู้ใช้จะให้อยู่ในรูปแบบที่เป็นมิตรและง่ายต่อการใช้งาน ส่วนการจำลองการทำงานของเครื่องคอมพิวเตอร์จะประกอบด้วยโปรแกรมซึ่งดำเนินงานแปลโปรแกรมต้นฉบับ (source program) ที่เขียนด้วยภาษาแอสเซมบลี (assembly language) ไปเป็นโปรแกรมภาษาเครื่อง (object program or machine language program) และโปรแกรมซึ่งดำเนินงานตามคำสั่งต่าง ๆ ของภาษาเครื่อง

3.1 แนวคิดพื้นฐานในการออกแบบ

โปรแกรมจำลองฯ ต้องการออกแบบให้ผู้ใช้สามารถสร้างโปรแกรมต้นฉบับในรูปโปรแกรมภาษาแอสเซมบลีของเครื่องคอมพิวเตอร์ PDP-8 แล้วนำมาประมวลผลด้วยตัวแปรภาษาในโปรแกรมจำลองฯ ดังนั้นข้อมูลเข้าของโปรแกรมจำลองฯ จึงเป็นโปรแกรมต้นฉบับซึ่งถูกเขียนด้วยภาษาแอสเซมบลีของเครื่องคอมพิวเตอร์ PDP-8 โดยผู้ใช้สามารถสร้างโปรแกรมต้นฉบับได้ 2 วิธี ดังนี้

1. ใช้โปรแกรมบรรณาธิกร (editor) ใด ๆ ที่อยู่บนระบบปฏิบัติการของเครื่องคอมพิวเตอร์ที่ใช้
2. ใช้โปรแกรมบรรณาธิกรที่มีอยู่ในโปรแกรมจำลองฯ

เมื่อผู้ใช้ได้โปรแกรมต้นฉบับและนำมาประมวลผลกับตัวแปรภาษาในโปรแกรมจำลองฯ จะได้โปรแกรมภาษาเครื่อง (object program) ที่สามารถนำมาประมวลผลกับโปรแกรมจำลองฯ ได้

ในส่วนการดำเนินงานให้ผู้ใช้ติดต่อกับโปรแกรมจำลองฯ เพื่อสร้างโปรแกรมต้นฉบับและนำมาประมวลผลได้โปรแกรมภาษาเครื่อง โปรแกรมจำลองฯ ในส่วนนี้ถูกออกแบบให้อยู่ในรูปเมนูที่เป็นกราฟิก (Graphic User Interface - GUI)

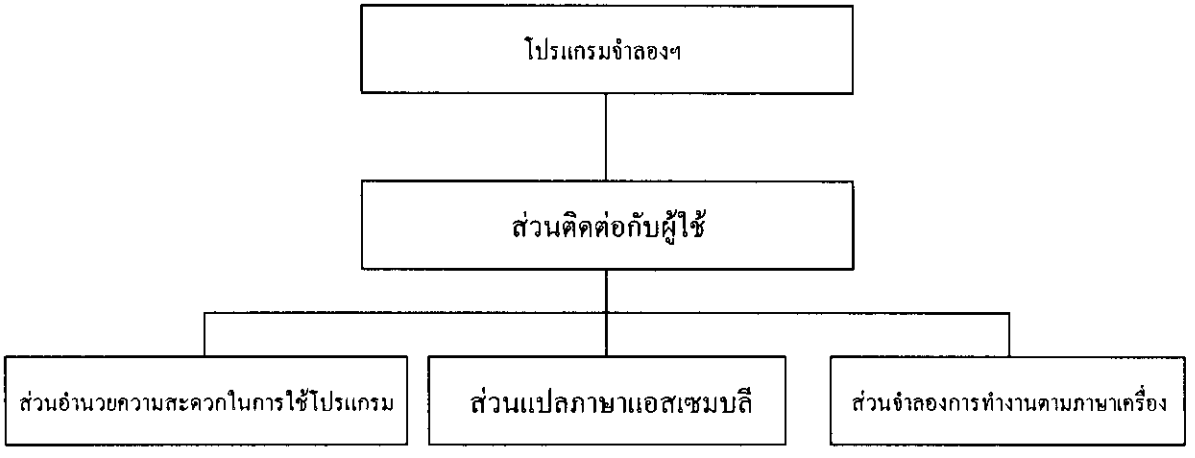
ในส่วนการจำลองการทำงานของเครื่องคอมพิวเตอร์ จะแบ่งการทำงานออกเป็น 2 ส่วน

ส่วนแรก เมื่อผู้ใช้นำโปรแกรมต้นฉบับมาให้โปรแกรมจำลองฯ ทำงาน ตัวแปรภาษาในโปรแกรมจำลองฯ จะทำการรวบรวมคำและตรวจสอบไวยากรณ์ของภาษา ถ้าการตรวจสอบถูกต้องจะแปลโปรแกรมต้นฉบับไปเป็นโปรแกรมภาษาเครื่องและเก็บไว้ในแฟ้มข้อมูลที่สามารถถูกนำไปประมวลผลต่อไป นอกจากนี้จะมีการแสดงรายละเอียดข้อมูลที่ถูกเก็บในความจำหลักและในรีจิสเตอร์ของซีพียูที่ได้จากการดำเนินงานที่ผู้ใช้สามารถเรียกดูได้

ส่วนที่สอง เมื่อผู้ใช้นำโปรแกรมภาษาเครื่องมาประมวลผล ผู้ใช้สามารถกำหนดให้โปรแกรมจำลองฯ ดำเนินการประมวลผลทั้งโปรแกรมรวดเดียวจบ หรือกำหนดให้โปรแกรมจำลองฯ ดำเนินการทีละคำสั่ง โดยการดำเนินการทีละคำสั่งนี้ จะมีการแสดงรายละเอียดของรีจิสเตอร์ต่าง ๆ เช่น PC, IR และ ACC ที่ใช้ในการประมวลผลคำสั่งนั้น ๆ

3.2 โครงสร้างโปรแกรม

จากการวิเคราะห์การดำเนินงานที่ควรจะเป็นในหัวข้อ 3.1 โปรแกรมจำลองฯ ถูกออกแบบให้มีโครงสร้างหลักในการดำเนินงาน ดังแสดงในภาพประกอบที่ 3.1

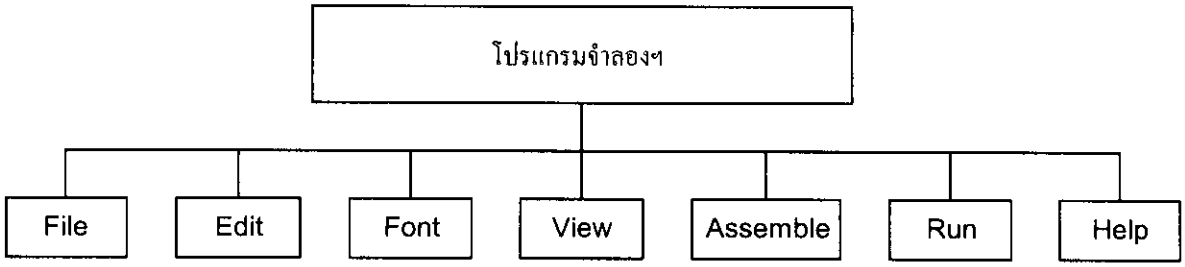


ภาพประกอบ 3.1 โครงสร้างโปรแกรม

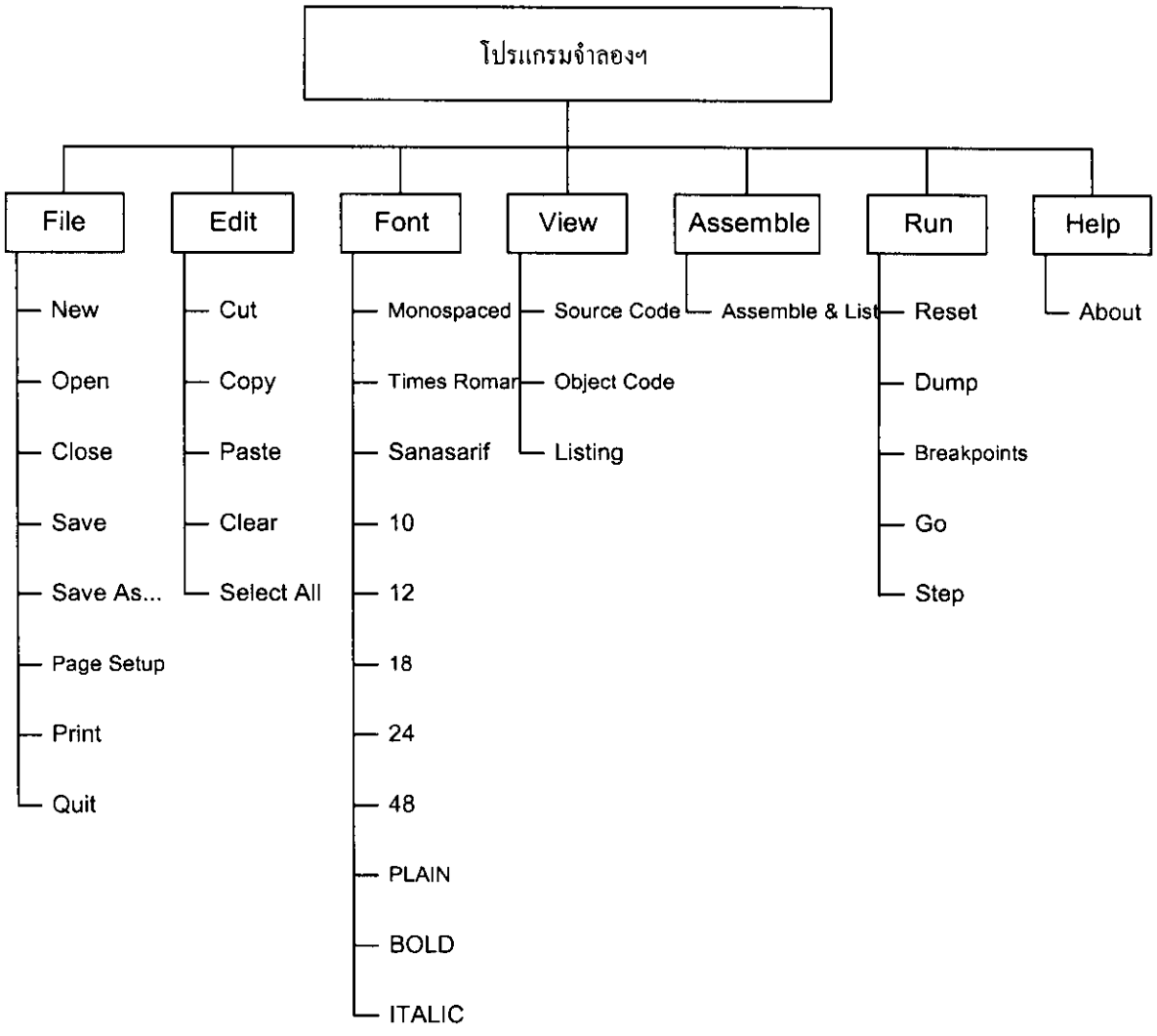
3.2.1 ส่วนติดต่อกับผู้ใช้

ส่วนติดต่อกับผู้ใช้ เป็นส่วนที่ออกแบบเพื่อให้ผู้ใช้ดำเนินการกับคำสั่งต่าง ๆ ที่ครอบคลุมการดำเนินงานทั้งหมดของโปรแกรมจำลองฯ อาทิเช่น การเปิดแฟ้มข้อมูล การบันทึกแฟ้มข้อมูล การสั่งพิมพ์ การแปลภาษาแอสเซมบลี การสั่งจำลองการทำงานตามภาษาเครื่อง เป็นต้น โดยมีการดำเนินการแสดงผลบนจอภาพในรูปเมนูที่เป็นกราฟิก โครงสร้างหลัก ๆ ของโปรแกรมจะอยู่ในรูป

การดำเนินงานของคำสั่งต่าง ๆ ในเมนูหลัก แต่ละเมนูหลักจะประกอบด้วยเมนูย่อยต่าง ๆ ตามกลุ่มของคำสั่งใช้งานตามเมนูหลัก ดังแสดงในภาพประกอบ 3.2 และภาพประกอบ 3.3 ตามลำดับ



ภาพประกอบ 3.2 เมนูหลักของโปรแกรม



ภาพประกอบ 3.3 เมนูการทำงานทั้งหมดของโปรแกรม

จากภาพประกอบ 3.2 แสดงเมนูการทำงานทั้งหมดของโปรแกรม จากส่วนอำนวยความสะดวกในการใช้โปรแกรมจากโครงสร้างหลัก ประกอบด้วยเมนู File , Edit , Font , View และ Help ในส่วนแปลภาษาแอสเซมบลีประกอบด้วยเมนู Assemble และส่วนจำลองการทำงานตามภาษาเครื่องประกอบด้วยเมนู Run โดยมีรายละเอียดในการออกแบบในแต่ละเมนูโดยสังเขปมีดังนี้

- File ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ที่เกี่ยวข้องกับแฟ้มข้อมูล โดยมีรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.1
- Edit ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ที่เกี่ยวข้องกับการแก้ไขข้อความบนความจำหลักขณะกำลังใช้งาน โดยมีรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.2
- Font ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ที่เกี่ยวข้องกับการเลือกใช้รูปแบบและขนาดของตัวอักขระสำหรับแสดงข้อมูลบนความจำหลักที่กำลังใช้งาน โดยมีรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.3
- View ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ที่เกี่ยวข้องกับการแสดงประเภทของข้อความที่จะแสดงในความจำหลัก โดยมีรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.4
- Assemble ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ที่เกี่ยวข้องกับการแปลโปรแกรมภาษาแอสเซมบลีปัจจุบันบนความจำหลักให้เป็นภาษาเครื่อง โดยมีรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.5
- Run ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ตามโปรแกรมภาษาเครื่องที่ได้จากการแปลโปรแกรมภาษาแอสเซมบลี โดยมีรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.6
- Help ถูกออกแบบไว้สำหรับการดำเนินการต่าง ๆ ที่เกี่ยวข้องกับตัวโปรแกรมจำลองฯ สำหรับรายละเอียดของเมนูย่อย ดังแสดงในตาราง 3.7

ตาราง 3.1 รายละเอียดของเมนูย่อยในเมนู File

เมนูย่อย	**การดำเนินงาน**
New	บันทึกโปรแกรมภาษาแอสเซมบลีใหม่บนความจำหลัก
Open	เปิดแฟ้มข้อมูลที่มีอยู่
Close	ปิดแฟ้มข้อมูลปัจจุบันที่กำลังใช้งานอยู่
Save	บันทึกข้อมูลบนเนื้อที่ความจำหลักส่วนที่ผู้ใช้กำลังใช้งานอยู่ เก็บลงในแฟ้มข้อมูลเดิมที่กำลังเปิดใช้อยู่
Save As	บันทึกข้อมูลบนเนื้อที่ความจำหลักส่วนที่ผู้ใช้กำลังใช้งานอยู่ เก็บลงในแฟ้มข้อมูลใหม่
Page Setup	กำหนดรายละเอียดต่าง ๆ ของหน้ากระดาษที่จะพิมพ์
Print	พิมพ์รายละเอียดของความจำหลักที่ผู้ใช้กำลังใช้งานบนกระดาษผ่านเครื่องพิมพ์
Quit	ออกจากโปรแกรมจำลอง ฯ

ตาราง 3.2 รายละเอียดของเมนูย่อยในเมนู Edit

เมนูย่อย	**การดำเนินงาน**
Cut	ลบข้อความที่ถูกเลือกและคัดลอกข้อความเหล่านั้นไปเก็บไว้ในหน่วยความจำหลักส่วนที่เก็บข้อมูลชั่วคราวที่เรียกว่าบัฟเฟอร์ (Buffer)
Copy	คัดลอกข้อความที่ถูกเลือกไปเก็บไว้ในบัฟเฟอร์
Paste	นำข้อความจากบัฟเฟอร์มาวางในตำแหน่งบนความจำหลักที่เคอเซอร์ (cursor) ชี้
Clear	ลบข้อความทั้งหมดออกจากพื้นที่ที่ใช้งาน
Select All	เลือกข้อความทั้งแฟ้ม

ตาราง 3.3 รายละเอียดของเมนูย่อยในเมนู Font

เมนูย่อย	**การดำเนินงาน**
Monospaced	กำหนดรูปแบบตัวอักขระเป็นแบบ Monospaced
Time Roman	กำหนดรูปแบบตัวอักขระเป็นแบบ Time Roman
San Sarif	กำหนดรูปแบบตัวอักขระเป็นแบบ San Sarif
10	กำหนดขนาดอักขระให้มีขนาด 10
12	กำหนดขนาดอักขระให้มีขนาด 12
18	กำหนดขนาดอักขระให้มีขนาด 18
24	กำหนดขนาดอักขระให้มีขนาด 24
48	กำหนดขนาดอักขระให้มีขนาด 48
PLAIN	กำหนดรูปแบบตัวอักขระเป็นแบบตัวปกติ
BOLD	กำหนดรูปแบบตัวอักขระเป็นแบบตัวหนา
ITALIC	กำหนดรูปแบบตัวอักขระเป็นแบบตัวเอียง

ตาราง 3.4 รายละเอียดของเมนูย่อยในเมนู View

เมนูย่อย	**การดำเนินงาน**
Source Code	แสดงโปรแกรมภาษาแอสเซมบลีฉบับปัจจุบันที่อยู่บนความจำหลัก
Object Code	แสดงโปรแกรมภาษาเครื่องในรูปเลขจำนวนฐานแปดจากแฟ้มข้อมูลภาษาเครื่อง
Listing	แสดงรายละเอียดต่าง ๆ ซึ่งเกิดจากการแปลโปรแกรมภาษาแอสเซมบลีปัจจุบัน

ตาราง 3.5 รายละเอียดของเมนูย่อยในเมนู Assemble

เมนูย่อย	**การดำเนินงาน**
Assemble & List	แปลโปรแกรมภาษาแอสเซมบลีให้เป็นภาษาเครื่องของเครื่อง PDP-8 ที่จำลองและแสดงรายละเอียดต่าง ๆ จากการแปลโปรแกรมภาษาแอสเซมบลี

ตาราง 3.6 รายละเอียดของเมนูย่อยในเมนู Run

เมนูย่อย	การดำเนินงาน
Reset	ปรับสภาพแวดล้อมเพื่อเตรียมความพร้อมในการดำเนินงานใหม่กับโปรแกรม
Dump	แสดงเนื้อหาของความจำหลัก/รีจิสเตอร์ที่ได้จากการประมวลผลโปรแกรมปัจจุบัน
Breakpoints	กำหนดจุดหยุดการทำงานของคำสั่งภาษาเครื่องที่ต้องการพิจารณาผลการดำเนินงานหลังเสร็จสิ้นการดำเนินงานของคำสั่งนั้น
Go	ดำเนินงานตามโปรแกรมภาษาเครื่องปัจจุบัน
Step	ดำเนินงานตามคำสั่งภาษาเครื่องปัจจุบันทีละคำสั่งโดยผู้ใช้กำหนด

ตาราง 3.7 รายละเอียดของเมนูย่อยในเมนู Help

เมนูย่อย	การดำเนินงาน
About	แสดงข้อมูลทั่วไปเกี่ยวกับโปรแกรมจำลองฯ

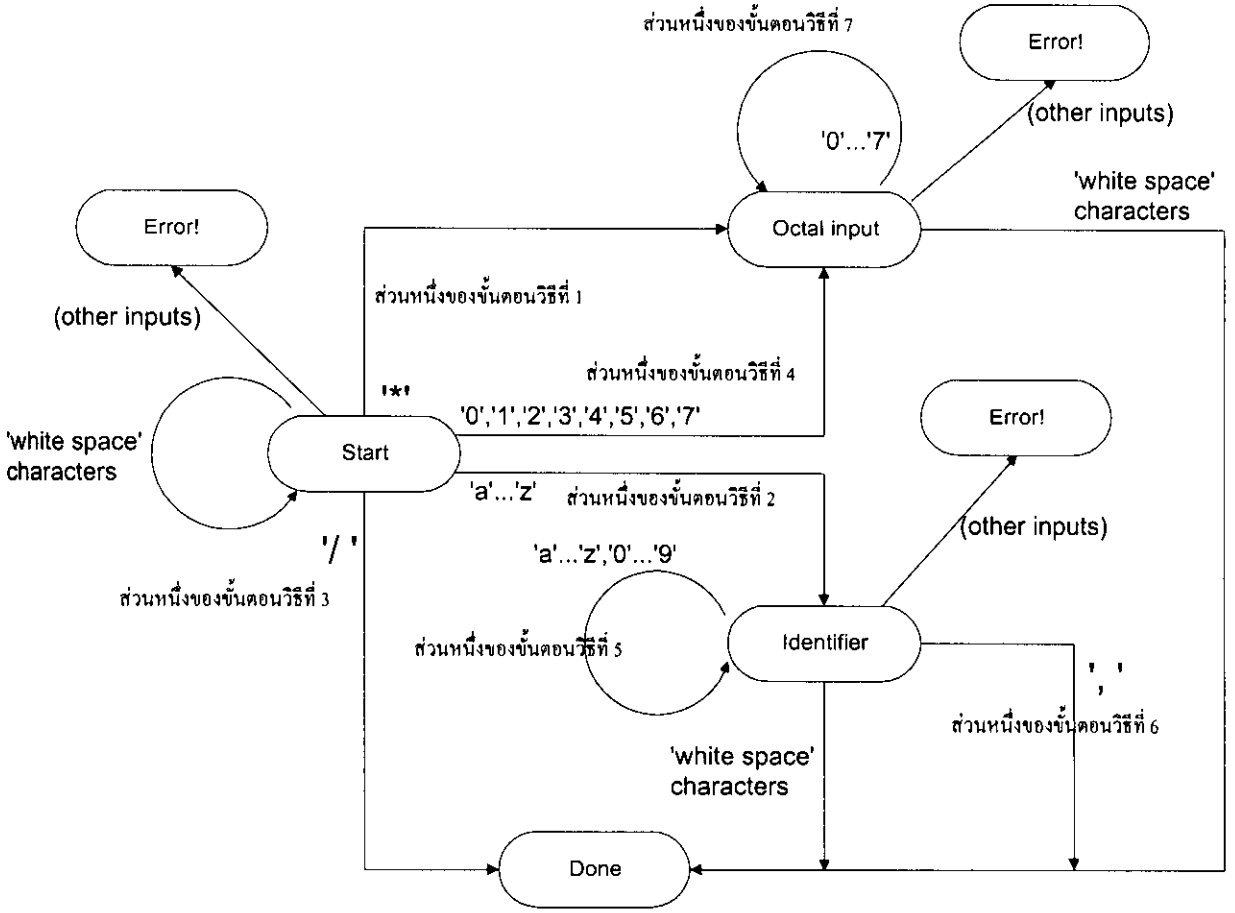
3.2.2 ส่วนแปลภาษาแอสเซมบลี

ส่วนแปลภาษาแอสเซมบลี เป็นส่วนที่ออกแบบเพื่อดำเนินการแปลข้อมูลชุดอักขระที่เป็นโปรแกรมภาษาแอสเซมบลีที่เป็นโปรแกรมต้นฉบับให้เป็นโปรแกรมภาษาเครื่อง โดยออกแบบให้มีการดำเนินงานอ่านอักขระที่เป็นโปรแกรมภาษาแอสเซมบลี 2 รอบ (two pass assembler) ในส่วนแปลภาษาแอสเซมบลีนี้ได้ใช้ขั้นตอนวิธีที่ออกแบบไว้แล้วโดย GRAY [Gray, N.A.B. (1987)]

ในการแปลรอบที่ 1 จะเป็นการวิเคราะห์คำ โดยจะมีการแยกและรวบรวมคำต่าง ๆ ที่พบในโปรแกรมภาษาแอสเซมบลีให้เป็น token ซึ่งประกอบด้วย comment , origin , label , ending , octal , identifier , null และ errors ดังแสดงในภาพประกอบ 3.4 โดยที่

- null จะใช้บอกว่าเป็นการสิ้นสุดในแต่ละบรรทัดของคำสั่งภาษาแอสเซมบลี
- error เป็นการแจ้งว่าพบข้อผิดพลาดจากโปรแกรมภาษาแอสเซมบลีที่ทำการแปล
- ending เป็นการบอกการสิ้นสุดโปรแกรมภาษาแอสเซมบลี โดยในโปรแกรมภาษาแอสเซมบลีที่ทำการแปลจะต้องกำหนดจุดสิ้นสุดด้วยอักขระ "$"
- comment จะเป็นหมายเหตุของโปรแกรม โดยถ้าพบอักขระ "/" จะถือว่าเป็นหมายเหตุของโปรแกรมจนสิ้นสุดบรรทัดคำสั่ง

ส่วนการแปลรอบที่ 2 จะเป็นการแปลให้เป็นภาษาเครื่องโดยเก็บลงในแฟ้มข้อมูลชื่อ object อย่างอัตโนมัติ



ภาพประกอบ 3.4 การวิเคราะห์คำ

[Gray, N.A.B. (1987)]

จากภาพประกอบ 3.4 เป็นตัวอย่างการแยกคำจากโปรแกรมต้นฉบับในแต่ละบรรทัด หลังจากที่พบอักขระตัวแรกในบรรทัดของโปรแกรมต้นฉบับจะมีการดำเนินการเพื่อหาชนิดของ token แล้วจัดเก็บด้วยตัวแปร symboltype แล้วดำเนินการค้นหาคำจนครบ token ซึ่งตัวอย่างส่วนหนึ่งของขั้นตอนวิธีมีดังนี้

- ส่วนหนึ่งของขั้นตอนวิธีที่ 1

  symboltype ← origin

  value ← 0

- ส่วนหนึ่งของขั้นตอนวิธีที่ 2

  symboltype ← indentifier

  name ← ch

- ส่วนหนึ่งของขั้นตอนวิธีที่ 3

  symboltype ← comment

  (discard rest of input line)

- ส่วนหนึ่งของขั้นตอนวิธีที่ 4

  symboltype ← octal

  value ← ch – '0'

- ส่วนหนึ่งของขั้นตอนวิธีที่ 5

  name ← concat(name,ch)

- ส่วนหนึ่งของขั้นตอนวิธีที่ 6

  symboltype ← label

- ส่วนหนึ่งของขั้นตอนวิธีที่ 7

  value ← value + ch – '0'

โดยที่

- symboltype เป็นตัวแปรที่เก็บชนิดของ token ที่พบ
- value เป็นตัวแปรเลขจำนวนที่เก็บค่าที่พบเมื่อเป็นตัวเลข
- name เป็นตัวแปรสายอักขระที่เก็บสายอักขระที่มีชนิดของ token เป็น identifier

3.2.3 ส่วนจำลองการทำงานตามภาษาเครื่อง

ส่วนจำลองการทำงานตามภาษาเครื่อง เป็นส่วนที่ออกแบบให้ทำการจำลองการประมวลผลคำสั่งแต่ละคำสั่งภาษาเครื่องที่แปลได้ โดยการดำเนินงานจะประกอบด้วย 2 ช่วง

- ช่วงแรก เป็นการนำคำสั่ง (fetch phase) จากความจำหลักที่ตำแหน่งถูกบ่งบอกในรีจิสเตอร์ PC มาเก็บในรีจิสเตอร์ IR และดำเนินการแยกแยะส่วนรหัสคำสั่ง (opcode) และส่วนรายละเอียดอื่น ๆ ในคำสั่ง
- ช่วงที่สอง เป็นการดำเนินงานกับคำสั่ง (execution phase) ซึ่งแต่ละคำสั่งการดำเนินงานเลขคณิตและตรรกะของ PDP-8 จะเป็นการดำเนินงานของรูปแบบคำสั่งเลขที่เดียว (one – address format instruction) กล่าวคือ ข้อมูลตัวแรกสำหรับการดำเนินงานจะอยู่ในรีจิสเตอร์ ACC เสมอ

3.3 โครงสร้างข้อมูล

โปรแกรมจำลองฯ ที่ถูกออกแบบตามแนวคิดในหัวข้อ 3.1 และตามการออกแบบโครงสร้างโปรแกรมในหัวข้อ 3.2 เพื่อให้ผู้ใช้โปรแกรมจำลองฯ สามารถสร้างโปรแกรมภาษาแอสเซมบลีซึ่งเป็นโปรแกรมต้นฉบับ แล้วนำมาประมวลผลด้วยโปรแกรมจำลองฯ จนได้ผลลัพธ์ โครงสร้างข้อมูลที่ใช้ในระหว่างการประมวลผลสามารถประมวลได้ดังต่อไปนี้

3.3.1 ค่าคงที่และรายละเอียดของชุดคำสั่ง

คำสั่งภาษาแอสเซมบลีในกลุ่มต่าง ๆ ทั้งหมดของ PDP-8 แต่ละคำสั่งจะมีภาษาเครื่องหนึ่งคำสั่งที่สมนัยกันเสมอ คำสั่งภาษาแอสเซมบลีพร้อมทั้งค่าเลขจำนวนฐานแปดซึ่งแทนรูปแบบบิตของคำสั่งในภาษาเครื่องของแต่ละคำสั่ง ที่ต้องใช้ในการแปลโปรแกรมต้นฉบับ แสดงในตาราง 3.8

ตาราง 3.8 รายละเอียดของคำสั่งแอสเซมบลี

คำสั่ง	กลุ่มคำสั่ง	ค่าเลขจำนวนฐานแปดซึ่งแทนรูปแบบบิตของคำสั่งในภาษาเครื่อง
adcrb	input/output	6601
adsf	input/output	6602
adstrt	input/output	6604
and	memory reference	0000
cia	operate	7041
cla	operate	7200
clkcf	input/output	6501
clksf	input/output	6502
clkt	input/output	6504
cll	operate	7100
cma	operate	7040
cml	operate	7020
dca	memory reference	3000
dlma	input/output	6410
dlsk	input/output	6400

ตาราง 3.8 รายละเอียดของคำสั่งแอสเซมบลี (ต่อ)

คำสั่ง	กลุ่มคำสั่ง	ค่าเลขจำนวนฐานแปดซึ่งแทนรูปแบบบิตของคำสั่งในภาษาเครื่อง
drd	input/output	6411
dscf	input/output	6402
dssf	input/output	6401
dtcf	input/output	6414
dtsf	input/output	6412
dwrt	input/output	6415
hlt	operate	7402
i	indirect	0400
iac	operate	7001
iof	input/output	6002
ion	input/output	6001
iot	input/output	6000
isz	memory reference	2000
jmp	memory reference	5000
jms	memory reference	4000
krb	input/output	6036
ksf	input/output	6031
nop	operate	7000
opr	operate	7000
ral	operate	7004
rar	operate	7010
rtl	operate	7006
rtr	operate	7012
skp	operate	7410
sma	operate	7500
sna	operate	7450
snl	operate	7420

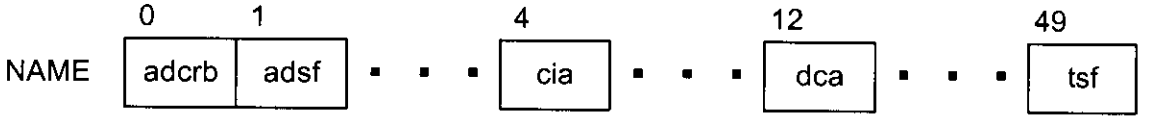
ตาราง 3.8 รายละเอียดของคำสั่งแอสเซมบลี (ต่อ)

คำสั่ง	กลุ่มคำสั่ง	ค่าเลขจำนวนฐานแปดซึ่งแทนรูปแบบบิตของคำสั่งในภาษาเครื่อง
spa	operate	7510
sza	operate	7440
szl	operate	7430
tad	memory reference	1000
tcf	input/output	6042
tls	input/output	6046
tpc	input/output	6044
tsf	input/output	6041

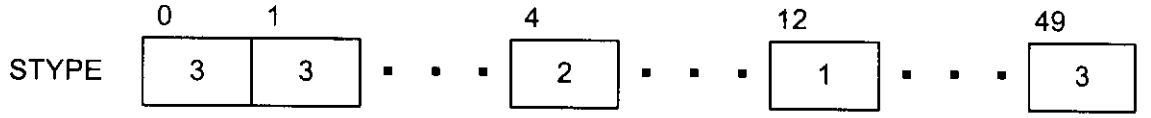
ในการแปลโปรแกรมต้นฉบับ การจัดการข้อมูลเข้าซึ่งเป็นภาษาแอสเซมบลี โปรแกรมจำลองฯอ่านคำสั่งในโปรแกรมต้นฉบับทีละคำสั่งซึ่งบรรจุอยู่ในหนึ่งบรรทัด ดำเนินการแยกคำและนำส่วนของคำแรกของคำสั่งมาเปรียบเทียบกับสัญญลักษณ์แทนคำสั่งในตารางชุดคำสั่งในตาราง 3.8 ว่าคำสั่งที่อ่านเข้ามาเป็นคำสั่งที่ถูกต้องหรือไม่ การจัดการค่าคงที่ในการเก็บชื่อคำสั่ง, ชนิด และค่าเลขจำนวนฐานแปดซึ่งแทนรหัสคำสั่งในภาษาเครื่อง ตามตาราง 3.8 ใช้โครงสร้างข้อมูลแบบแถวลำดับ (array) 1 มิติจำนวน 3 ชุด โดยแยกเก็บเป็น ชื่อคำสั่ง ชนิดของคำสั่ง และค่าเลขจำนวนฐานแปดซึ่งแทนรหัสคำสั่งในภาษาเครื่องของแต่ละคำสั่ง ดังตัวอย่างแสดงในภาพประกอบ 3.5 ภาพประกอบ 3.6 และภาพประกอบ 3.7 ตามลำดับ โดยมีรายละเอียดดังนี้

- แถวลำดับ NAME เป็นแถวลำดับ 1 มิติ ขนาด 50 หน่วย เก็บข้อมูลชนิดสายอักขระ (string) ที่แทนคำสั่งต่าง ๆ ของ PDP-8 โดยเก็บเรียงตามตาราง 3.8
- แถวลำดับ STYPE เป็นแถวลำดับ 1 มิติ ขนาด 50 หน่วย เก็บข้อมูลชนิดเลขจำนวนเต็มที่แทนประเภทของคำสั่งต่าง ๆ ของ PDP-8 โดยใช้ดัชนีที่สอดคล้องกับแถวลำดับ NAME ตามตาราง 3.8 แทนค่าในการจัดเก็บได้ดังนี้
  - 1 แทนคำสั่งประเภท memory reference
  - 2 แทนคำสั่งประเภท operate
  - 3 แทนคำสั่งประเภท input/output
  - 4 แทนตัวอักขระที่เป็น indirect

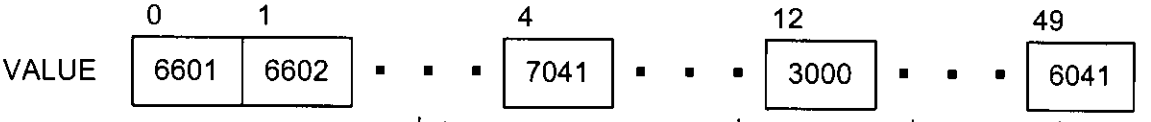
- แถวลำดับ VALUE เป็นแถวลำดับ 1 มิติ ขนาด 50 หน่วย เก็บข้อมูลชนิดเลขจำนวนเต็มที่แทนค่าเลขจำนวนฐานแปดซึ่งแทนรหัสคำสั่งในภาษาเครื่องของแต่ละคำสั่งของ PDP-8 โดยใช้ดัชนีที่สอดคล้องกับแถวลำดับ NAME ตามตาราง 3.8



ภาพประกอบ 3.5 โครงสร้างข้อมูลที่เก็บชื่อคำสั่ง



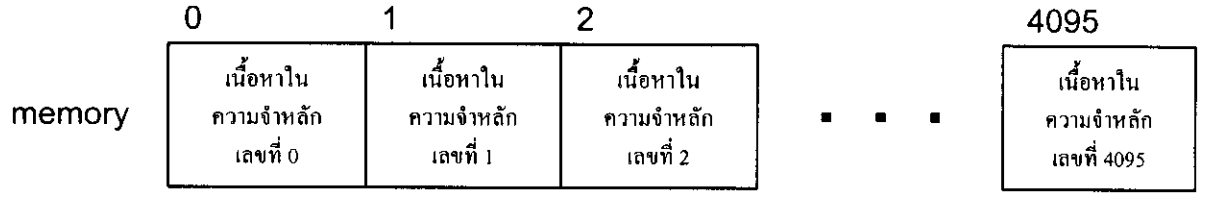
ภาพประกอบ 3.6 โครงสร้างข้อมูลที่เก็บชนิดคำสั่ง



ภาพประกอบ 3.7 โครงสร้างข้อมูลที่เก็บค่าเลขจำนวนฐานแปดซึ่งแทนรหัสคำสั่งในภาษาเครื่อง

3.3.2 การจำลองหน่วยความจำหลัก

คำสั่งภาษาเครื่องที่ได้จากการแปลคำสั่งภาษาแอสเซมบลีจะถูกบันทึกไว้ที่ความจำหลักจำลองซึ่งใช้โครงสร้างข้อมูลแบบแถวลำดับ 1 มิติ ขนาด 4096 หน่วย โดยใช้หมายเลขดัชนี (Index) ของแถวลำดับแทนเลขที่ของแต่ละตำแหน่งความจำหลัก ค่าที่เก็บภายในแถวลำดับแทนเนื้อหา (content) ของความจำหลัก ณ ตำแหน่งนั้น ๆ ภาพประกอบที่ 3.8 แสดงความจำหลักจำลองนี้



ภาพประกอบ 3.8 โครงสร้างข้อมูลที่แทนความจำหลักจำลอง

3.3.3 การจำลองรีจิสเตอร์ภายในซีพียู

ในการจำลองรีจิสเตอร์ต่าง ๆ ของ PDP-8 มีโครงสร้างข้อมูล 2 ชนิดตามลักษณะของรีจิสเตอร์ คือ รีจิสเตอร์ที่มีขนาด 12 บิต จะใช้ตัวแปรที่มีชนิดข้อมูลเลขจำนวนเต็มในการจัดเก็บโดยเก็บเป็นเลขฐานสิบ ก่อนการจัดเก็บจะมีการตรวจสอบค่าที่จัดเก็บเพื่อไม่ให้มีค่าเกินขนาดของรีจิสเตอร์ที่มีขนาด 12 บิต คือ 4095 ส่วนรีจิสเตอร์ขนาด 1 บิต จะใช้ตัวแปรที่มีชนิดข้อมูลบูลีนโครงสร้างข้อมูลที่ใช้แทนรีจิสเตอร์ภายในซีพียูของ PDP-8 ดังแสดงในตาราง 3.9

ตาราง 3.9 รายละเอียดโครงสร้างข้อมูลที่ใช้แทนรีจิสเตอร์ภายในซีพียูของ PDP-8

รีจิสเตอร์	**ชนิดข้อมูล**
accumulator	เลขจำนวนเต็ม
program counter	เลขจำนวนเต็ม
instruction register	เลขจำนวนเต็ม
link	บูลีน
interrupt-enabled	บูลีน

3.3.4 การจำลองรีจิสเตอร์ของหน่วยรับเข้า/ส่งออก

ในการจำลองรีจิสเตอร์ต่าง ๆ ของหน่วยรับเข้า/ส่งออก มีโครงสร้างข้อมูล 2 ชนิดตามลักษณะของรีจิสเตอร์ คือ รีจิสเตอร์ข้อมูล (data register) ที่มีขนาด 12 บิต จะใช้ตัวแปรที่มีชนิดข้อมูลเลขจำนวนเต็ม ส่วนรีจิสเตอร์แฟลก (flag register) ขนาด 1 บิต ซึ่งใช้แสดงสถานะในการทำงานจะใช้ตัวแปรที่มีชนิดข้อมูลบูลีนโครงสร้างข้อมูลที่ใช้แทนรีจิสเตอร์ต่าง ๆ ของหน่วยรับเข้า/ส่งออก ดังแสดงในตาราง 3.10

ตาราง 3.10 รายละเอียดโครงสร้างข้อมูลที่ใช้แทนรีจิสเตอร์ของหน่วยรับเข้า/ส่งออก

รีจิสเตอร์	**ชนิดข้อมูล**
teletype data register	เลขจำนวนเต็ม
teletype flag register	บูลีน
keyboard data register	เลขจำนวนเต็ม
keyboard flag register	บูลีน

ตาราง 3.10 รายละเอียดโครงสร้างข้อมูลที่ใช้แทนรีจิสเตอร์ของหน่วยรับเข้า/ส่งออก (ต่อ)

รีจิสเตอร์	**ชนิดข้อมูล**
disk controller data register	เลขจำนวนเต็ม
disk transmit flag register	บูลีน
disk memory register	เลขจำนวนเต็ม
disk seek flag register	บูลีน

3.4 โครงสร้างแฟ้มข้อมูล

โปรแกรมจำลองการทำงานของระบบคอมพิวเตอร์ได้ถูกออกแบบให้มีการบันทึกงานจากความจำหลักที่กำลังใช้งานของโปรแกรมเก็บไว้ในแฟ้มข้อมูลในรูปแบบของข้อความ (text file) โดยแฟ้มข้อมูลต่าง ๆ ที่เกี่ยวข้องขณะประมวลผลประกอบด้วย

- แฟ้มข้อมูลข้อความที่เก็บโปรแกรมภาษาเครื่องที่ได้จากการแปลโปรแกรมภาษาแอสเซมบลี
- แฟ้มข้อมูลข้อความที่จำลองดิสก์ซึ่งเป็นความจำสำรองของระบบ

โดยทุกแฟ้มข้อมูลมีโครงสร้างแบบตามลำดับ (sequential file)