บทที่ 3

โครงสร้างของโปรแกรมสำเร็จรูปที่ใช้ในการจัดเรียงลำดับข้อมูล

1. องค์ประกอบในการเรียงลำดับข้อมูล

โปรแกรมสำเร็จรูปที่ใช้ในการจัดเรียงลำดับข้อมูลของเครื่องไอบีเอ็ม 370-138 เป็นโปรแกรมหมายเลข 5746-เอสเอ็ม 1 ทำงานภายใต้การควบคุมของระบบดอส-วีเอส สามารถจัดเรียงลำดับข้อมูลซึ่งจัดอยู่ในรูปแซม (SAM) หรือวีแซม (VSAM) ได้มากถึง 9 แฟ้มข้อมูลและสามารถนำข้อมูลที่เรียงลำดับแล้วไปเก็บอยู่ในแฟ้มข้อมูลแบบวีแซมได้ไม่ว่าข้อมูลที่นำเข้าจะเป็นแบบแซมหรือวีแซมก็ตาม มีคีย์ได้ 12 ตัว แต่ละตัวจะต้องมีตำแหน่งตรงกันในแต่ละระเบียน ความยาวของคีย์รวมกันไม่เกิน 256 ไบท์ และคีย์ต้องอยู่ภายใน 4092 ไบท์แรกของระเบียน

1.1 อุปกรณ์ที่ใช้ อุปกรณ์ที่สามารถใช้เป็นอุปกรณ์นำข้อมูลเข้าและออกได้คือ

1. เทปแม่เหล็กแบบ 2400 หรือ 3400
2. อุปกรณ์ที่ใช้เพื่อให้สามารถเข้าถึงข้อมูลได้โดยตรงแบบ 2311
3. อุปกรณ์ที่ใช้เพื่อให้สามารถเข้าถึงข้อมูลได้โดยตรงแบบ 2314/2319
4. อุปกรณ์ที่ใช้เพื่อให้สามารถเข้าถึงข้อมูลได้โดยตรงแบบ 3330/3333
5. อุปกรณ์ที่ใช้เพื่อให้สามารถเข้าถึงข้อมูลได้โดยตรงแบบ 3340

อุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราว จะเป็นข้อหนึ่งข้อใดก็ได้ ยกเว้น ข้อ 2

1.2 ความสัมพันธ์กับระบบควบคุมแบบดอส/วีเอส

โปรแกรมสำเร็จรูปออกแบบมาเพื่อให้กับระบบควบคุมการทำงานแบบดอส/วีเอส ดังนั้น รูปแบบของแฟ้มข้อมูลต่าง ๆ ที่ใช้จึงต้องเป็นไปตามมาตรฐานของระบบนี้

โปรแกรมนี้สามารถเรียงลำดับข้อมูลให้ทุกพาร์ทิชั่นที่เรียกใช้ โดยปกติเมื่อถูกนำเข้ามาภายในหน่วยความจำแล้ว สามารถจัดตำแหน่งส่วนต่าง ๆ ของโปรแกรมได้ด้วยตัวเอง[1]

1.3 โปรแกรมสำเร็จรูปที่ใช้ในการเรียงลำดับข้อมูล

1.3.1 การเลือกใช้ โปรแกรมสำเร็จรูปที่ใช้ในการจัดเรียงลำดับข้อมูลนี้ผู้ใช้จะเลือกติดตั้งได้ตามวัตถุประสงค์ของการใช้งาน เพื่อประหยัดเนื้อที่ในการเก็บโปรแกรม คือ

1. จัดเรียงลำดับข้อมูลโดยใช้จานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว
2. จัดเรียงลำดับข้อมูลโดยใช้เทปแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว
3. จัดเรียงลำดับข้อมูลโดยใช้เทปหรือจานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว
4. รวมแฟ้มข้อมูลที่เรียงลำดับไว้แล้วเป็นแฟ้มข้อมูลเดียวกัน (Merging) เท่านั้น
5. จัดเรียงลำดับข้อมูลและรวมแฟ้มข้อมูล โดยใช้จานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว
6. ตัวโปรแกรมทั้งหมด ซึ่งรวมทั้งการจัดเรียงลำดับข้อมูล การรวมแฟ้มข้อมูลที่เรียงลำดับแล้วเป็นแฟ้มข้อมูลเดียวกัน โดยใช้เทปหรือจานแม่เหล็กเป็นที่เก็บข้อมูลชั่วคราว

1.3.2 การใช้อุปกรณ์ที่กำหนดร่วมกัน ในการเรียงลำดับข้อมูลจะต้องใช้อุปกรณ์ประเภทที่กำหนดในข้อ 1.1 ร่วมกันดังนี้คือ

1. เทปหนึ่งม้วนหรือจานแม่เหล็ก (ชนิดที่กล่าวไว้แล้วในข้อ 1.1) เป็นที่ป้อนข้อมูลเข้าหรือนำข้อมูลที่เรียงลำดับแล้วออกให้ผู้เรียกใช้งาน

---

[1] IBM Nordic Laboratory, "DOS/VS Sort/Merge Logic" (IBM Thailand, 1973) p.7

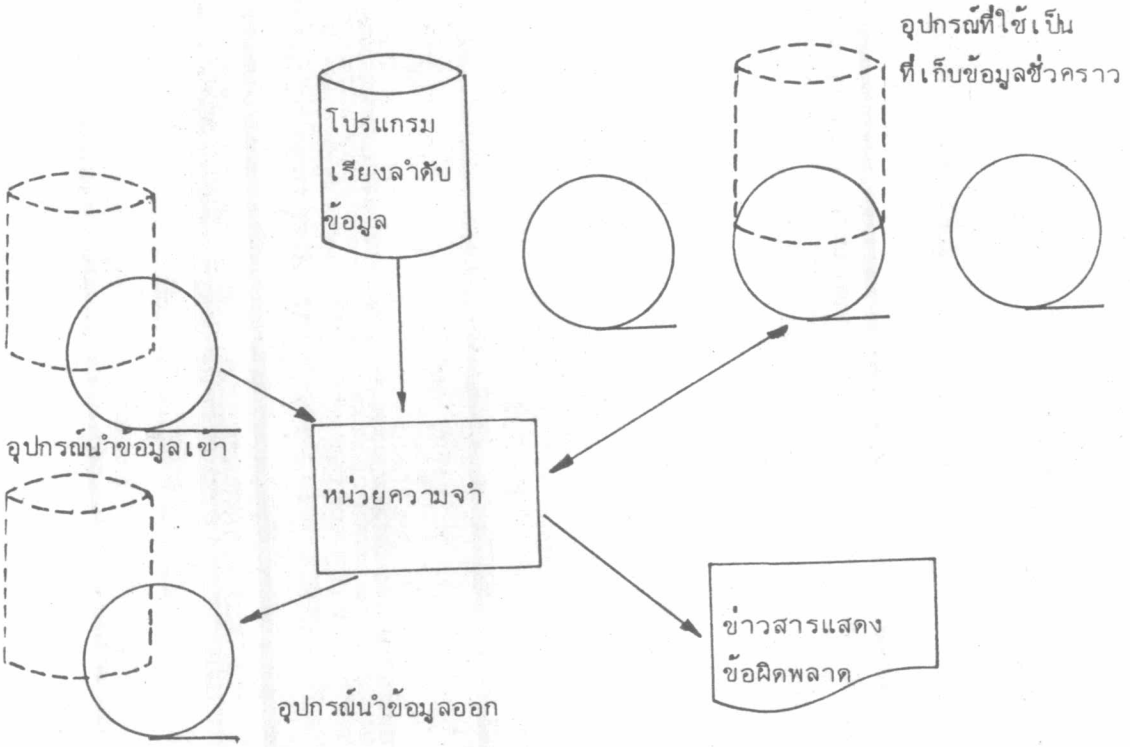
2. ที่เก็บข้อมูลชั่วคราว ซึ่งจะเป็นอย่างใดอย่างหนึ่งในข้อต่อไปนี้คือ

   2.1 เทปสามม้วน อย่างต่ำ 3 ม้วน อย่างมากไม่เกิน 9 ม้วน หรือ

   2.2 จานแม่เหล็กซึ่งอาจเป็นจานแม่เหล็กเดียวกับข้อ 1 ก็ได้ ถ้ามีเนื้อที่พอ และมีได้มากที่สุด 8 บริเวณ

3. จำนวนหน่วยความจำที่ใช้จะต้องเตรียมไว้อย่างน้อย 16 กิโลไบท์ ในกรณีที่ไม่มีโปรแกรมที่ผู้ใช้งานต้องการติดต่อกับโปรแกรมสำเร็จรูป และขนาดของบลอคข้อมูลที่นำเข้าและส่งออกไม่เกิน 1200 ไบท์

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.1 แสดงการใช้อุปกรณ์ที่กำหนดให้ร่วมกัน

1.3.3 การเรียกใช้ ทำได้ 2 วิธีคือ

1. การเรียกใช้โดยตรง คือการใช้คำสั่งควบคุม[1]

2. การเรียกใช้โดยผ่านโปรแกรมภาษาระดับสูง เช่นภาษาโคบอล[2]

1.3.4 การทำงานของโปรแกรม ตามปกติแล้ว โปรแกรมสำเร็จรูปนี้จะเก็บไว้ใน CIL (Core Image Library) เมื่อมีการเรียกใช้จะโลดที่ละโมดุลของแต่ละเฟส เพื่อไปทำงานดังนี้

1. เฟส 0 ทำหน้าที่อ่านคำสั่งควบคุมและกำหนดค่าพารามิเตอร์ที่ต้องใช้ในการทำงานตลอดโปรแกรม

2. เฟส1 ทำหน้าที่อ่านข้อมูลเข้า เรียงลำดับกลุ่มข้อมูลทีละกลุ่ม ต่อจากนั้นจึงกระจายไปเก็บบนอุปกรณ์ที่ใช้เก็บข้อมูลชั่วคราว

3. เฟส 2 ทำหน้าที่รวมข้อมูลที่กระจายไปเก็บบนที่เก็บข้อมูลชั่วคราวจนกระทั่งเหลือการรวมอีกเพียงครั้งเดียว ในกรณีที่ใช้การรวมเพียงครั้งเดียวได้ เฟสนี้จะไม่ทำงาน

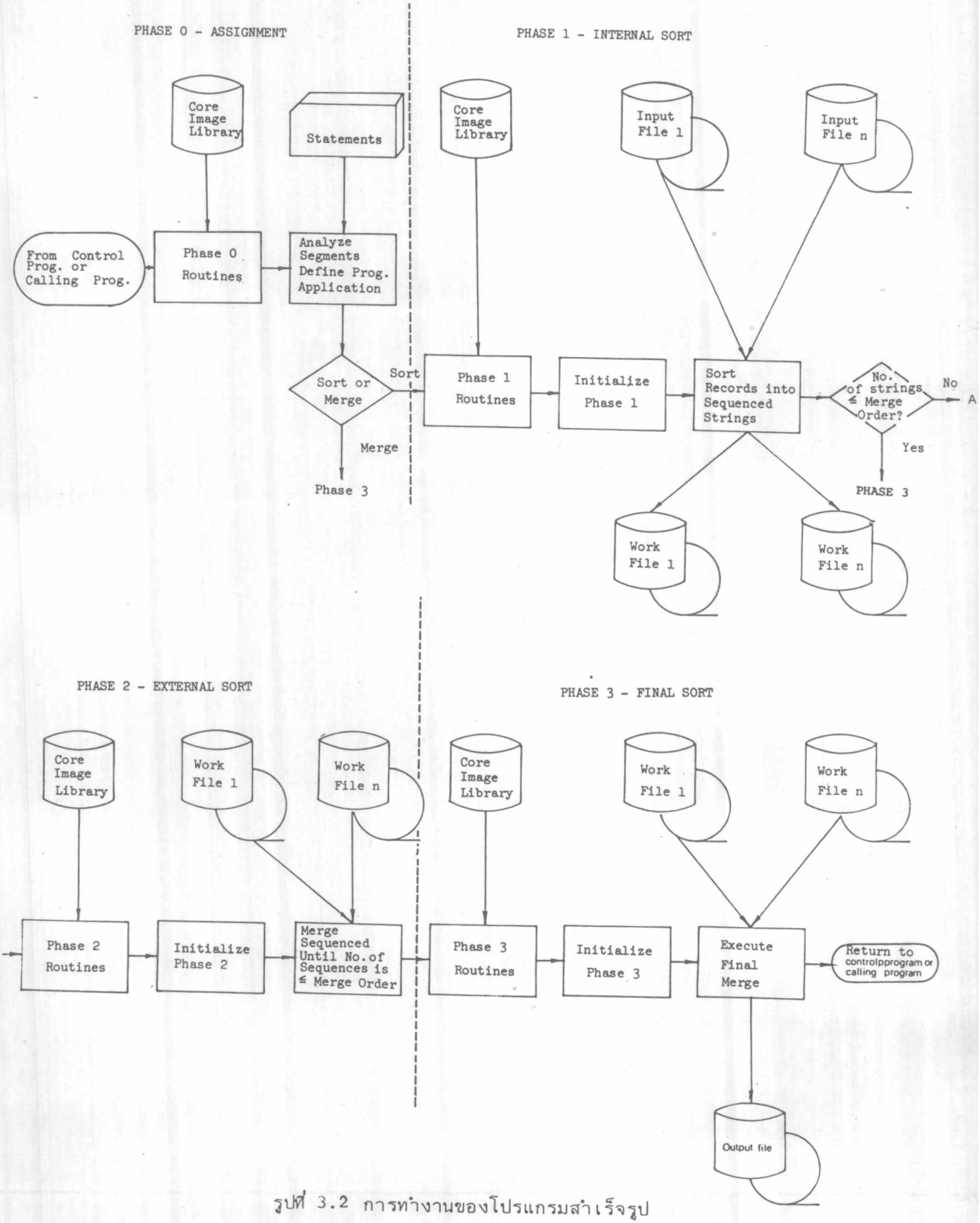
4. เฟส 3 ทำหน้าที่รวมข้อมูลเข้าด้วยกันเป็นครั้งสุดท้าย หลังจากเฟสนี้แล้วจะโอนการควบคุมการทำงานกลับไปยังโปรแกรมระบบหรือโปรแกรมที่เรียกใช้งาน

ในการดำเนินการจะมีลักษณะดังแสดงในรูป 3.2

---

[1] IBM Nordic Laboratory, "DOS/VS Sort/Merge Programmer's Guide" (IBM Thailand, 1973) p. 71

[2] IBM Nordic Laboratory, "IBM DOS Full American National Standard COBOL" (IBM Thailand, 1976), pp. 65 - 67.



รูปที่ 3.2 การทำงานของโปรแกรมสำเร็จรูป

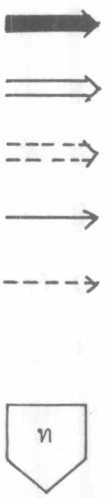
1.3.5 กลวิธีที่ใช้ ในการเรียงลำดับ โปรแกรมสำเร็จรูปใช้กลวิธีการเรียงลำดับแบบเลือกแทนที่ (รายละเอียดดูจากภาคผนวก จ.) และการรวมแฟ้มข้อมูลแบบไบนารี

2. ขั้นตอนการทำงานของเฟสและโมดุลต่าง ๆ

ในการแสดงขั้นตอนการทำงานของเฟสและโมดุลต่าง ๆ กำหนดให้

1. PPI (ภาคผนวก ค) คือบริเวณที่เก็บข้อมูลที่ใช้ร่วมกันทุกเฟส จะแสดงในภาพเมื่อมีการเปลี่ยนแปลงข้อมูลเท่านั้น

2. สัญญลักษณ์ที่ใช้ในแผนภาพ



แสดงลำดับการทำงาน ไปตามหัวลูกศร

แสดงการเคลื่อนย้ายข้อมูล ไปตามหัวลูกศร

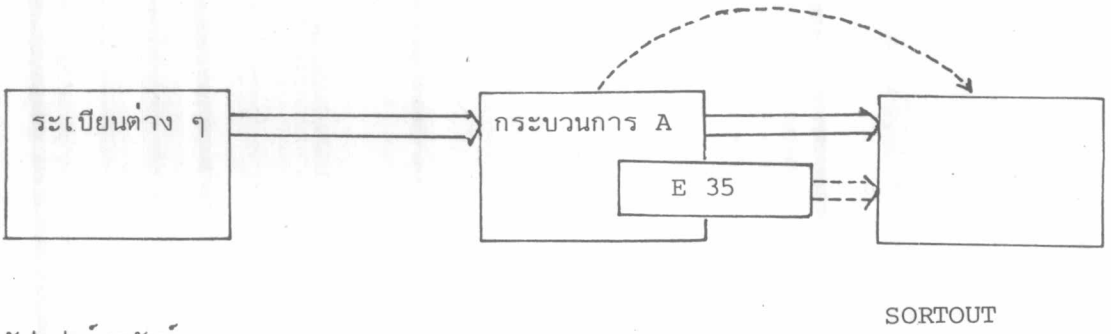
แสดงการเคลื่อนย้ายข้อมูลลำดับต่อมา

แสดงตำแหน่งที่อยู่ของข้อมูล ซึ่งจะอยู่ ณ ตำแหน่งที่หัวลูกศรชี้

แสดงการอ้างถึงข้อมูล หัวลูกศรแสดงทิศทางการสั่งการดึงข้อมูล

แสดงการอ้างถึงแผนภาพการอธิบาย กระบวนการ อย่างละเอียด ในแผนภาพหมายเลข n

ตัวอย่างเช่น



รูปที่ 3.3 ตัวอย่างการใช้สัญญลักษณ์

หมายความว่ากระบวนการ A เคลื่อนระเบียนต่าง ๆ จากบัฟเฟอร์ผลลัพธ์ไปเก็บใน SORTOUT พร้อมทั้งส่งการควบคุมไปให้กระบวนการ E35 ซึ่งกระบวนการ E35 จะทำหน้าที่เคลื่อนข้อมูลไปเก็บที่ SORTOUT เช่นกัน นอกจากนี้ในการทำงานของกระบวนการ A ยังจำเป็นจะต้องอาศัยข้อมูลจาก SORTOUT อีกด้วย

2.1 โครงสร้างระบบโปรแกรมสำเร็จรูป

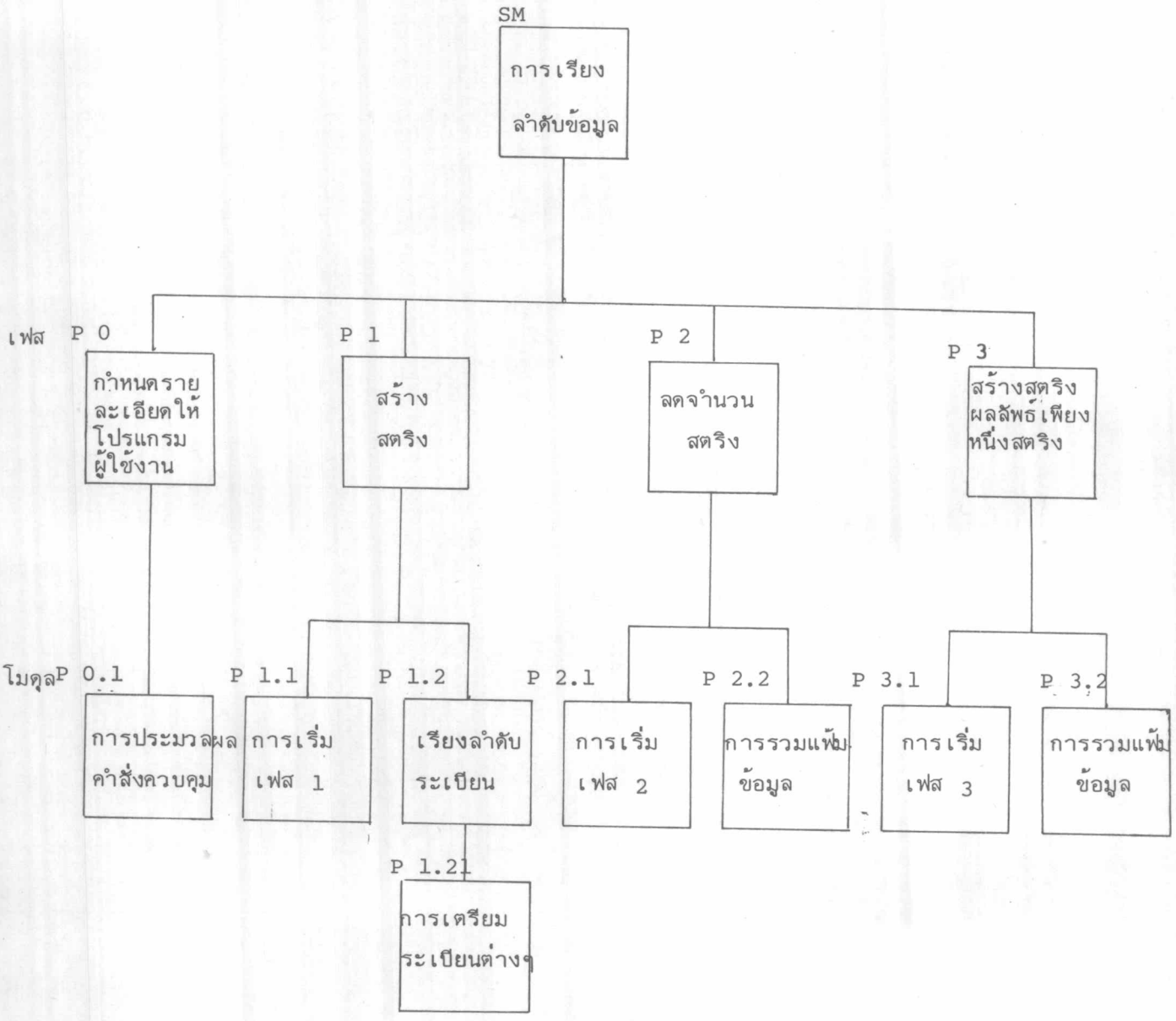
โปรแกรมสำเร็จรูปประกอบด้วยเฟสต่าง ๆ 4 เฟส ซึ่งแต่ละเฟสจะมีหน้าที่ที่สัมพันธ์กันดังนี้คือ

1. เฟส 0 ทำหน้าที่กำหนดรายละเอียดให้โปรแกรมผู้ใช้งาน ประกอบด้วยกลุ่มโมดุล PO.1 ซึ่งทำหน้าที่ประมวลผลคำสั่งควบคุม

2. เฟส 1 ทำหน้าที่สร้างสตริง ประกอบด้วยกลุ่มโมดุล P1.1 P1.2 P1.21 ซึ่งทำหน้าที่เริ่มเฟส 1 เรียงลำดับระเบียน เตรียมระเบียนต่าง ๆ ตามลำดับ

3. เฟส 2 ทำหน้าที่ลดจำนวนสตริง ประกอบด้วยกลุ่มโมดุล P2.1 P2.2 ซึ่งทำหน้าที่เริ่มเฟส 2 และรวมแฟ้มข้อมูล

4. เฟส 3 ทำหน้าที่สร้างสตริงผลลัพธ์เพียงหนึ่งสตริง ประกอบด้วยกลุ่มโมดุล P3.1 และ P3.2 ซึ่งทำหน้าที่ เริ่มเฟส 3 และรวมแฟ้มข้อมูล



รูปที่ 3.4 แสดงโครงสร้างระบบโปรแกรม[1]

---

[1] IBM Nordic Laboratory, "DOS/VS Sort/Merge Logic" (IBM Thailand, 1973), p.

2.2 การทำงานของระบบโปรแกรมสำเร็จรูป (ดังแสดงในรูป 3.4 มีลำดับการทำงาน ดังนี้คือ

1. เฟส 0 (กำหนดรายละเอียดต่าง ๆ ให้โปรแกรมผู้ใช้งาน) ประกอบด้วยขั้นตอนต่าง ๆ คือ

- การประมวลผลคำสั่งควบคุม
- การประมวลผลข่าวสารของระบบ
- การสร้างรหัสและตารางต่าง ๆ
- การคำนวณค่าที่เหมาะสม

2. เฟส 1 (เรียงลำดับข้อมูลให้อยู่ในรูปสตริง) ประกอบด้วยขั้นตอนต่าง ๆ คือ

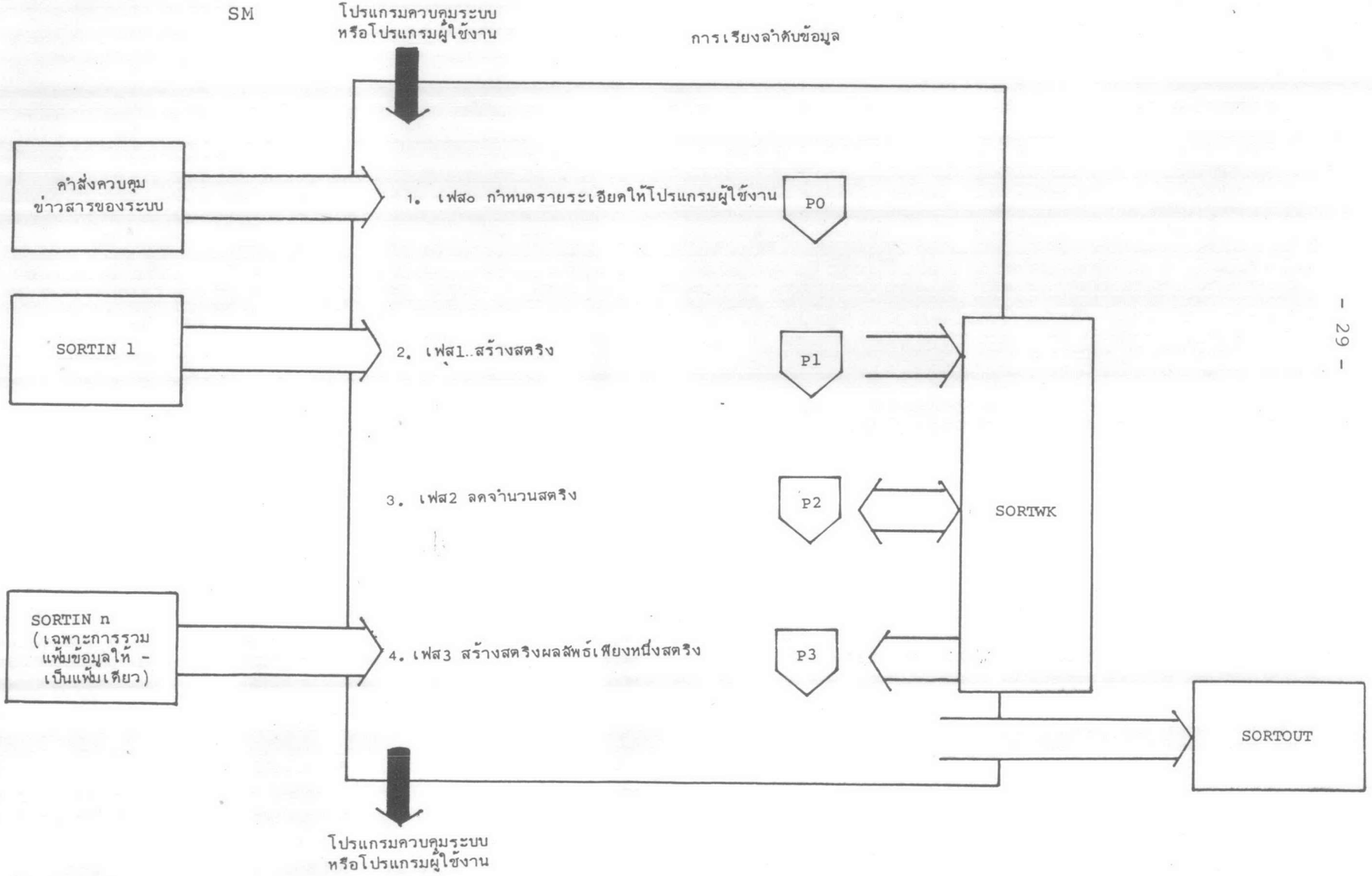
- การเริ่มเฟส 1
- การอ่านข้อมูลจากแฟ้มข้อมูลนำเข้าตามจำนวนที่ผู้ใช้งานกำหนด
- การเรียงลำดับระเบียนต่าง ๆ
- การกระจายสตริงไปเก็บบนที่เก็บข้อมูลชั่วคราว
- การพิจารณาความจำเป็นในการใช้งานของเฟส 2

3. เฟส 2 (ลดจำนวนสตริง) ประกอบด้วยขั้นตอนต่าง ๆ คือ

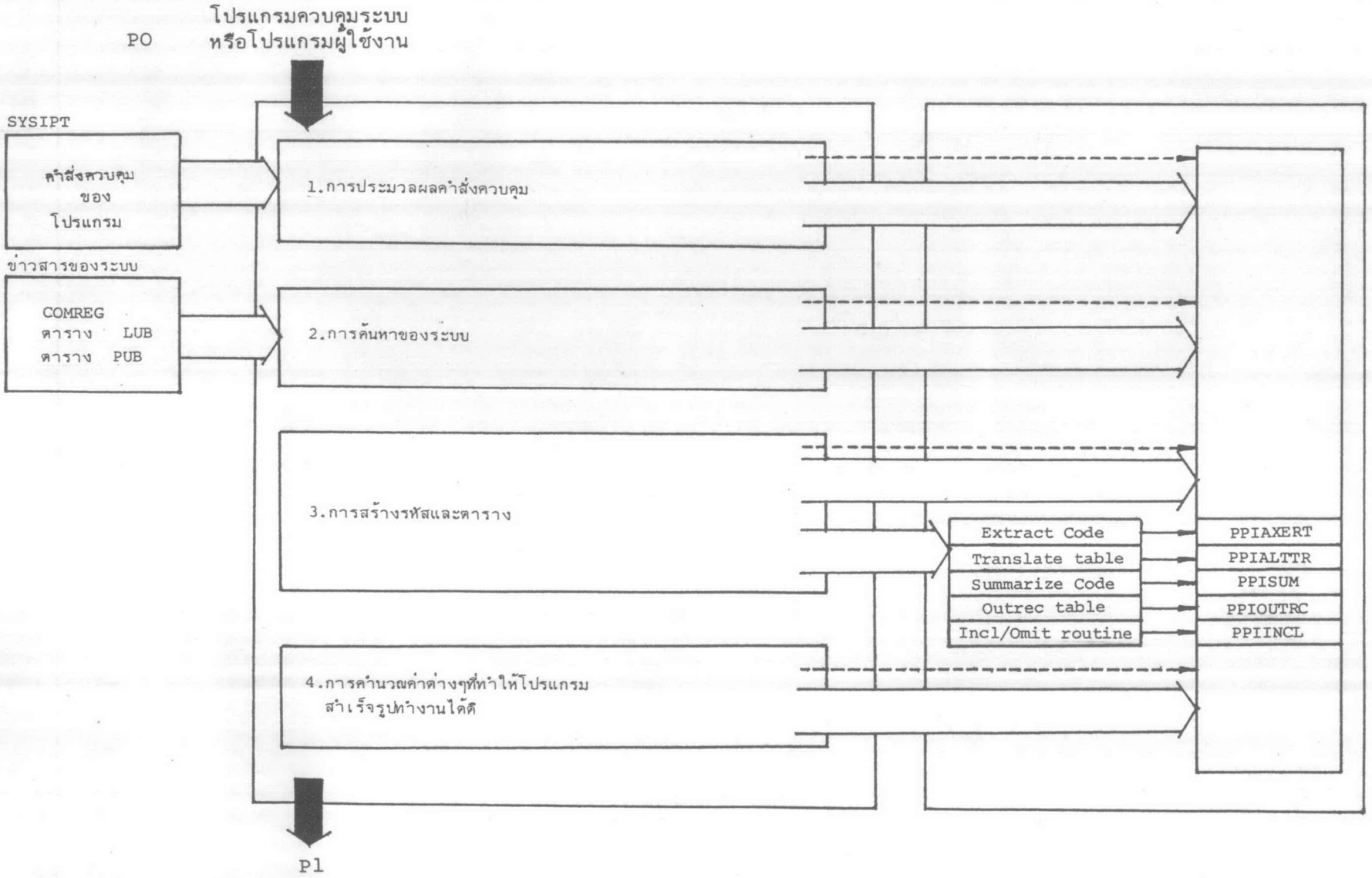
- การเริ่มเฟส 2
- การรวมสตริงบนที่เก็บข้อมูลชั่วคราว

4. เฟส 3 (สร้างสตริงผลลัพธ์เพียงหนึ่งสตริง) ประกอบด้วยขั้นตอนต่าง ๆ คือ

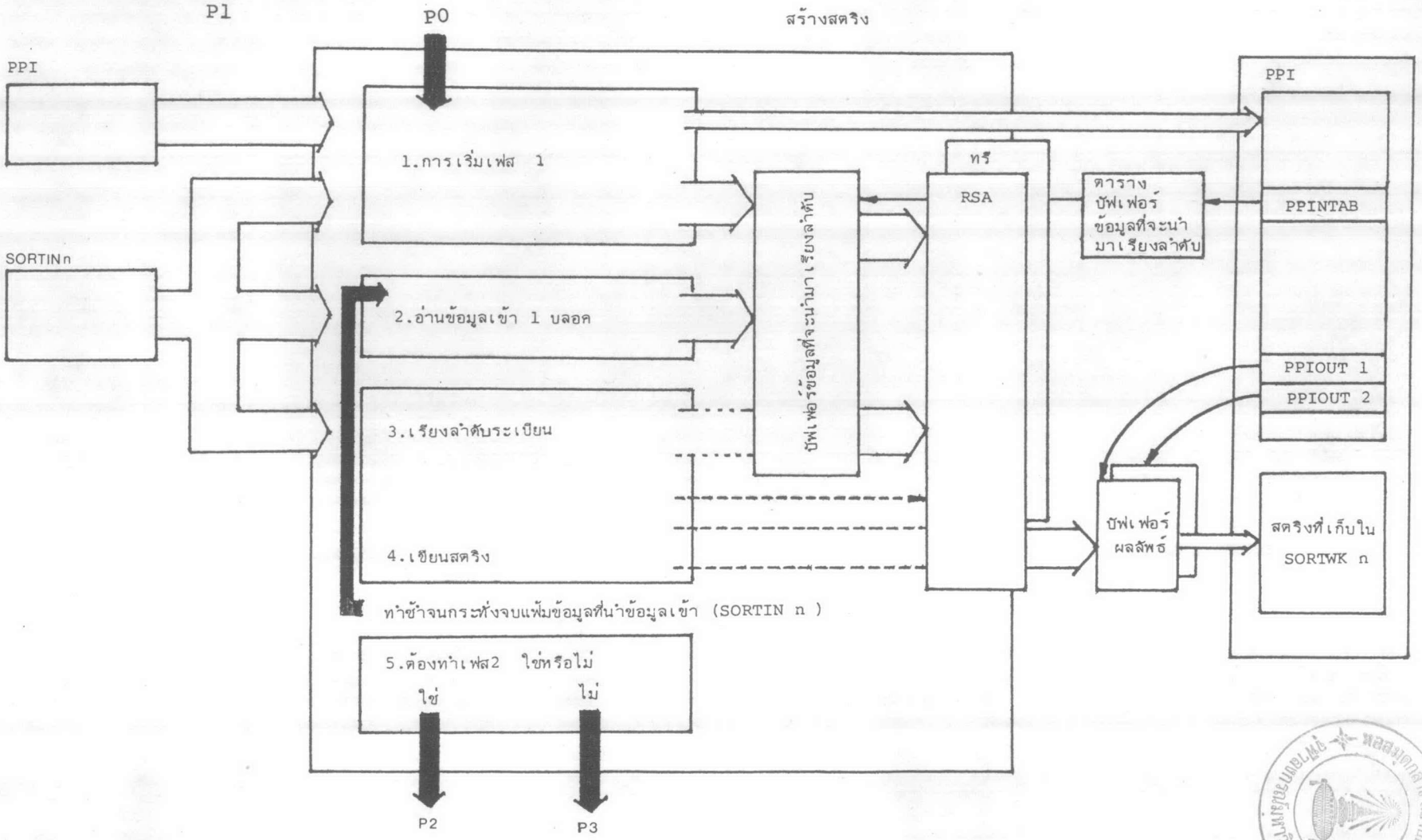
- การเริ่มเฟส 3
- การรวมสตริงบนที่เก็บข้อมูลชั่วคราวให้เป็นสตริงเดียวกัน และนำไปเก็บบนอุปกรณ์ที่ผู้ใช้งานกำหนดให้



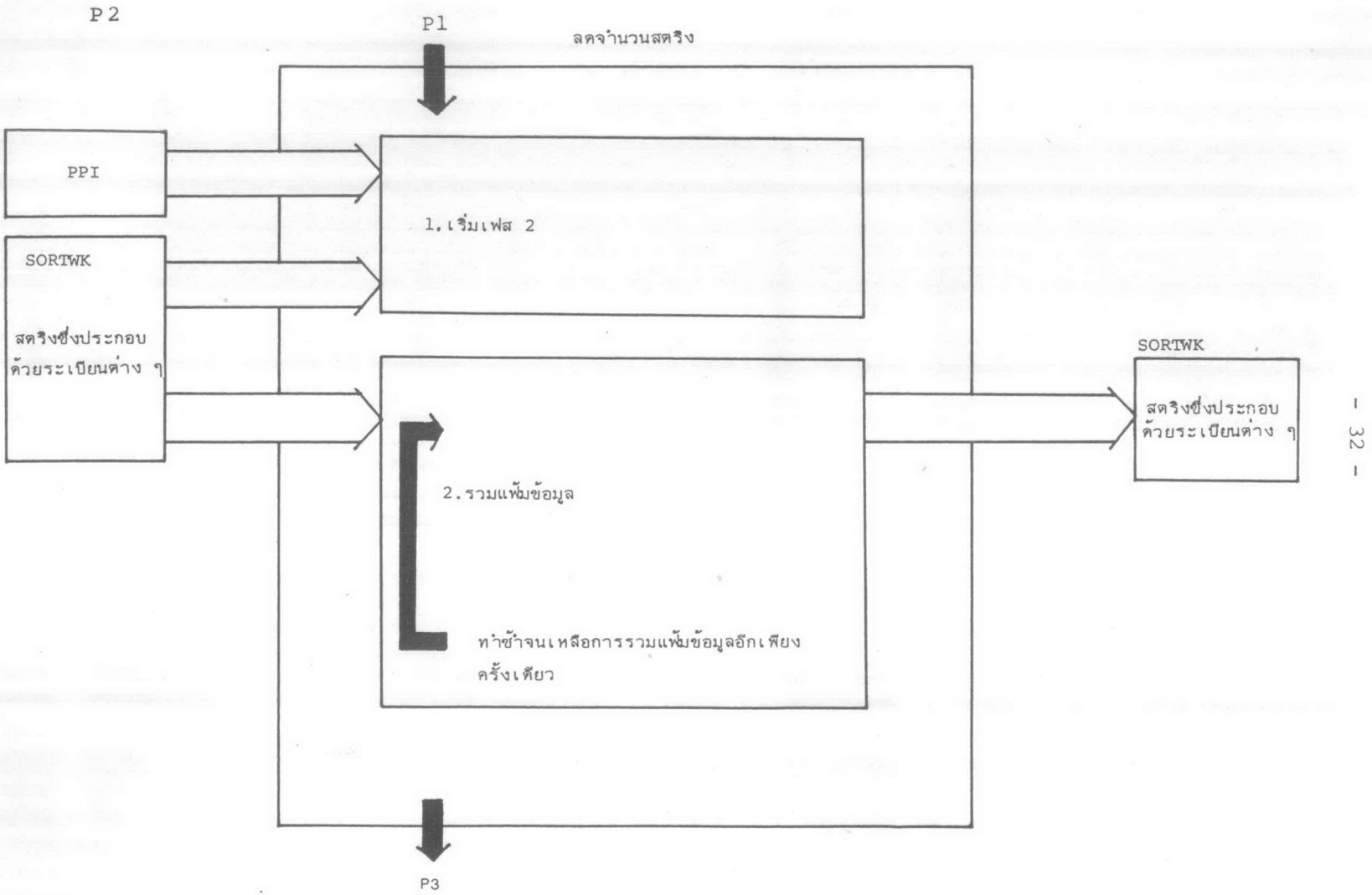
รูปที่ 3.5 แสดงการทำงานของระบบโปรแกรมสำเร็จรูป



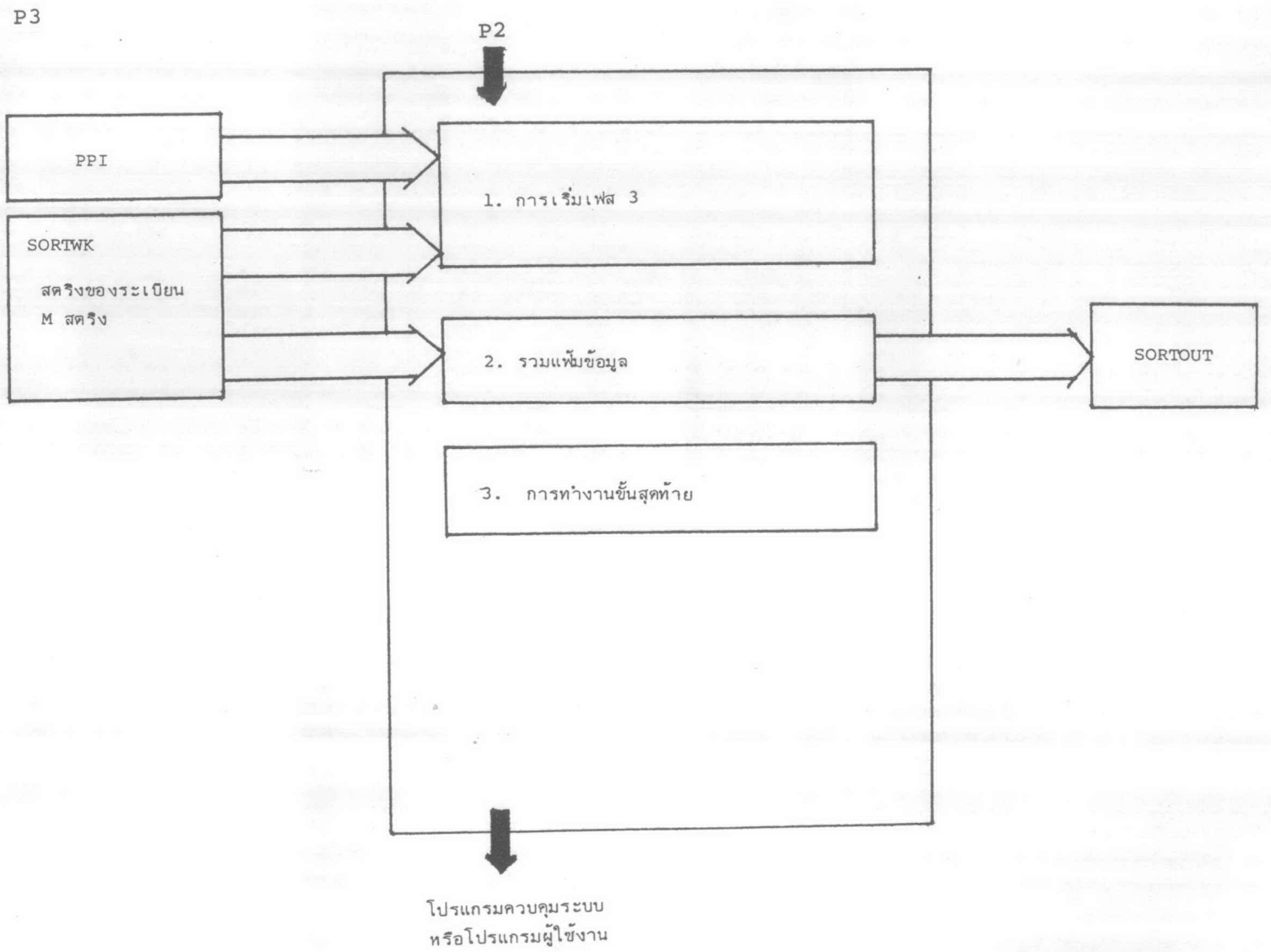
รูปที่ 3.6 แสดงขั้นตอนการทำงานของเฟส 0



รูปที่ 3.7 แสดงขั้นตอนการทำงานของเฟส 1



รูปที่ 3.8 แสดงขั้นตอนการทำงานของเฟส 2



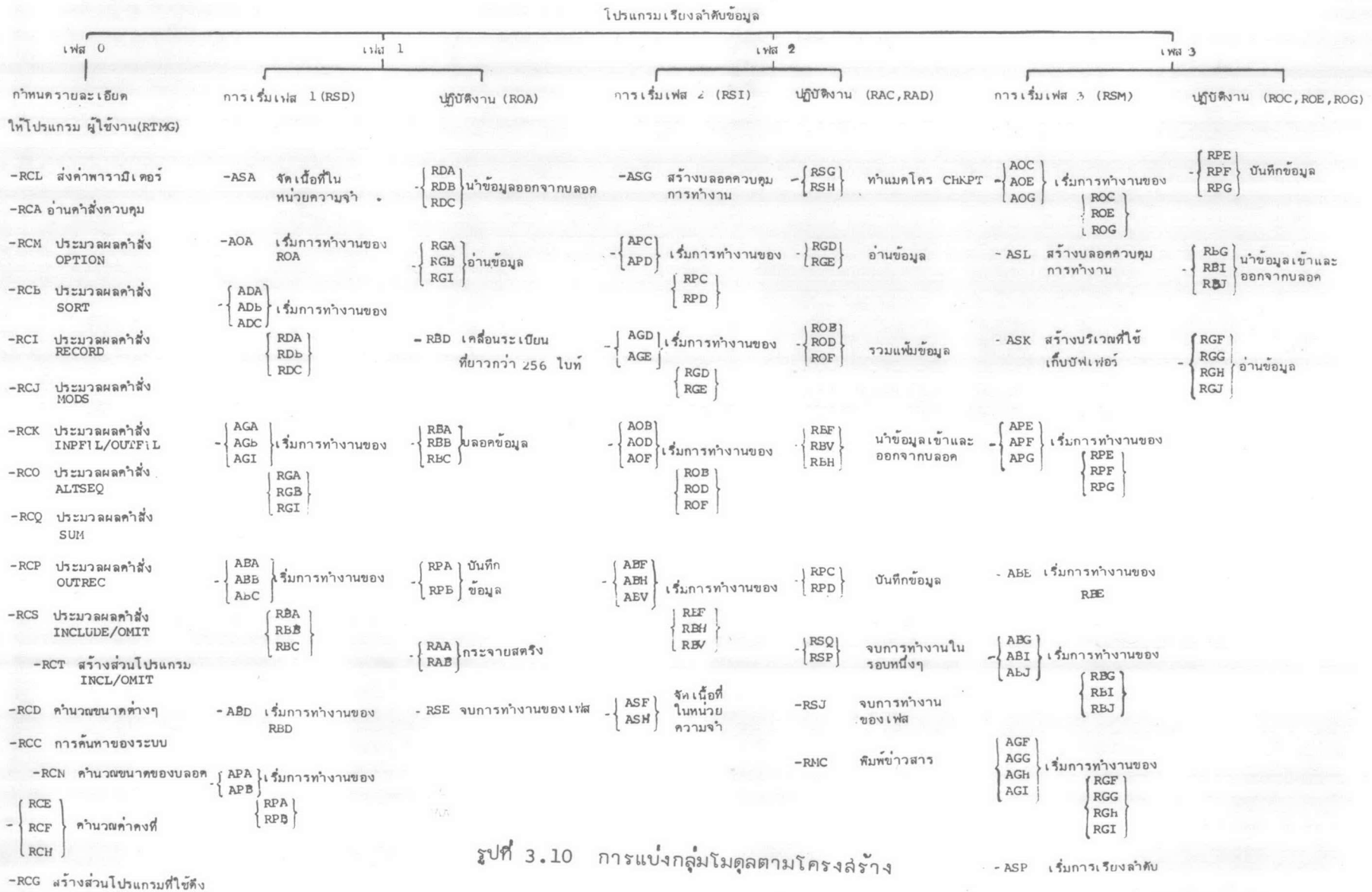
รูปที่ 3.9 แสดงขั้นตอนการทำงานของเฟส 3

3. รายละเอียดของโปรแกรมสำเร็จรูป

โปรแกรมในแต่ละเฟสมีลักษณะวิธีดำเนินการแบบเดียวกันคือ มีโมดุลหนึ่งเป็นโปรแกรมหลัก ทำหน้าที่เรียกโมดุลอื่น ๆ เข้ามาทำงาน โดยที่โมดุลที่เป็นโปรแกรมหลักจะอยู่ในหน่วยความจำตลอดการทำงานของเฟส แต่โมดุลที่ถูกเรียกเข้ามาทำงาน จะคงอยู่ตลอดการทำงาน หรือออกจากหน่วยความจำทันทีที่เสร็จการทำงาน ขึ้นกับหน้าที่ของแต่ละโมดุลนั้น

3.1 การแบ่งโมดุลตามโครงสร้าง

โมดุลต่าง ๆ จะแบ่งตามโครงสร้างออกเป็น 4 กลุ่ม คือกลุ่มโมดุลในเฟส 0 เฟส 1 เฟส 2 และเฟส 3 ซึ่งแต่ละกลุ่มจะมีโมดุลที่เป็นโปรแกรมหลักและย่อย ดังแสดงในรูป 3.10(ส่วนรายละเอียดการทำงานของแต่ละโมดุล ดูจากภาคผนวก ง.)

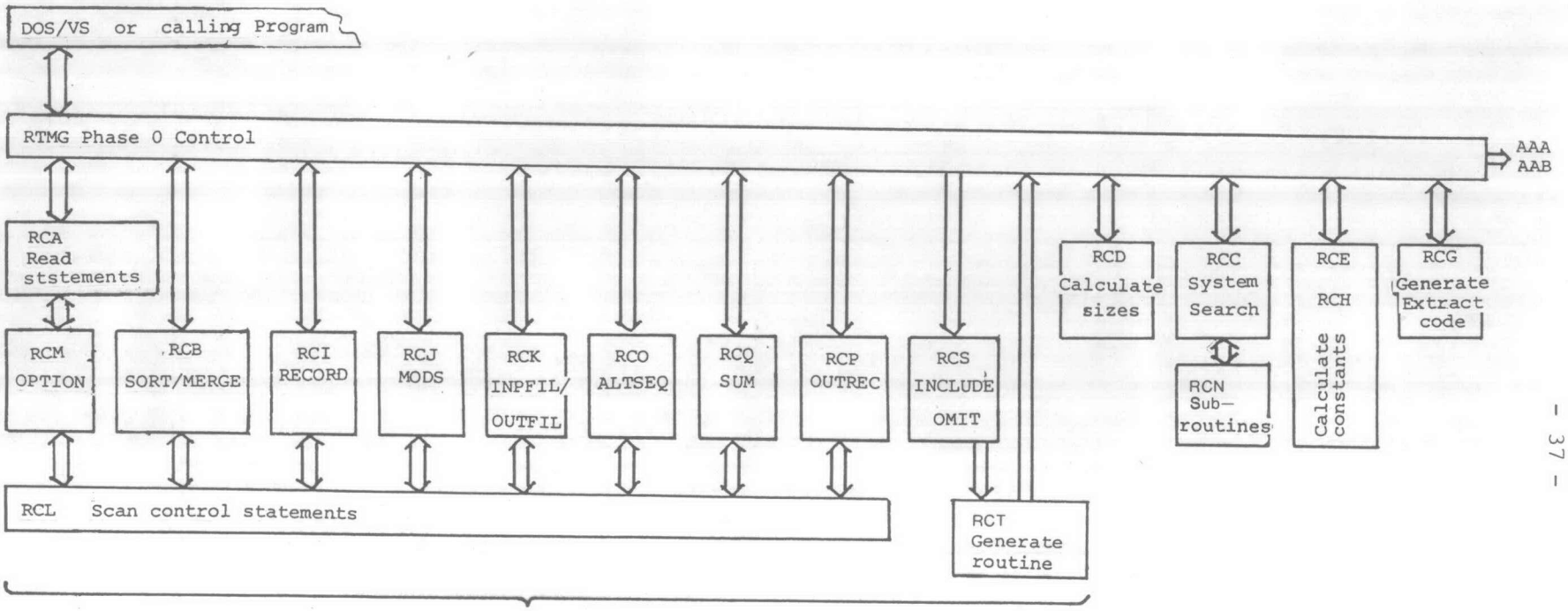


รูปที่ 3.10 การแบ่งกลุ่มโมดุลตามโครงสร้าง

3.2 ความสัมพันธ์ของโมดุลต่าง ๆ ในแต่ละเฟส

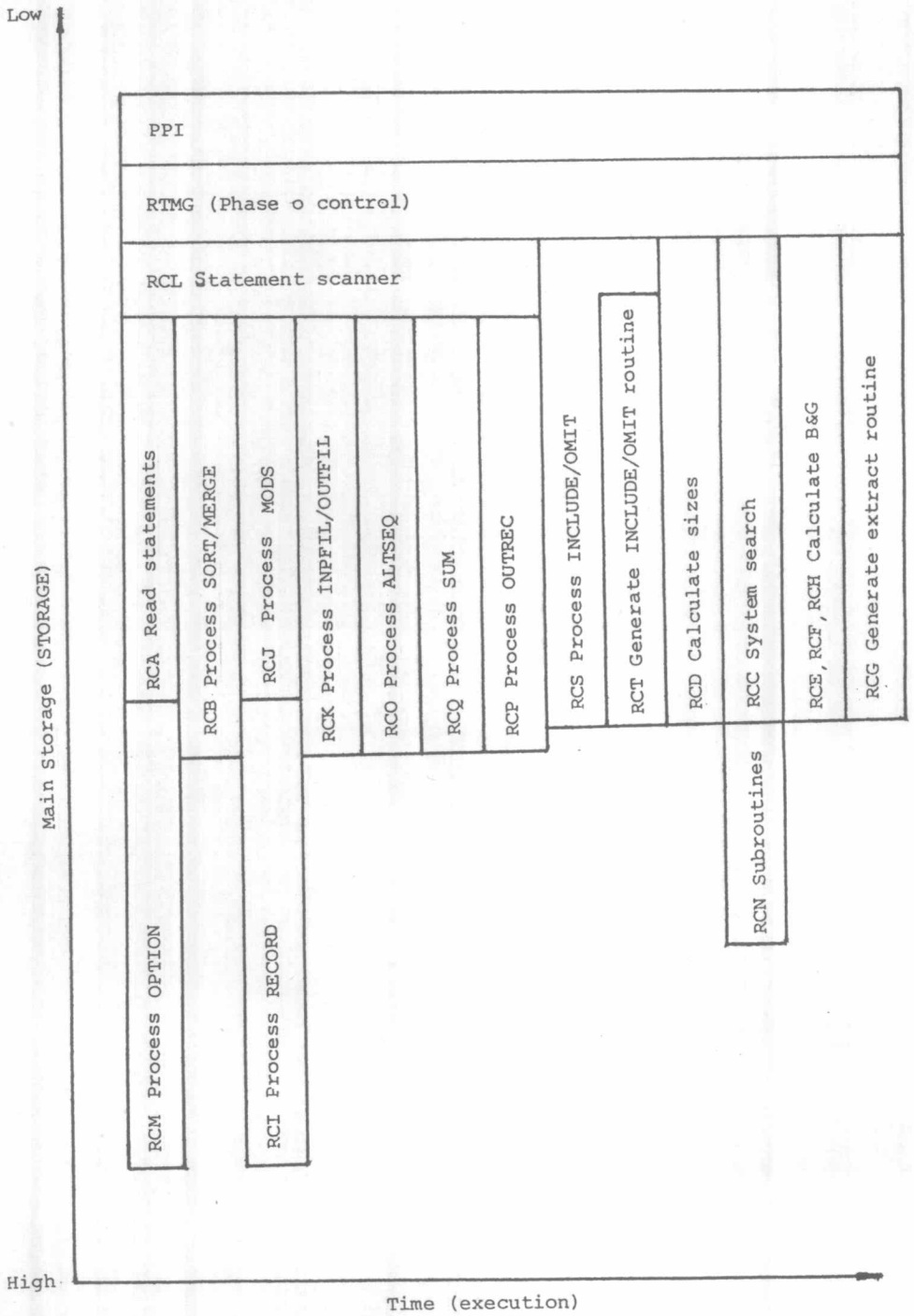
1) เฟส 0 (แสดงในรูป 3.11)โมดุล RTMG ซึ่งเป็นโมดุลควบคุมการทำงานของเฟส 0 จะถูกเรียกจากโปรแกรมระบบหรือโปรแกรมของผู้ใช้งาน ต่อจากนั้นจึงเรียกโมดุลต่าง ๆ เข้ามาเพื่อประมวลผลคำสั่งควบคุมทีละคำสั่ง โมดุลต่าง ๆ จะถูกวางลงในหน่วยความจำดังแสดงในรูป3.12เมื่อประมวลผลคำสั่งควบคุมเสร็จเรียบร้อยแล้ว จึงเริ่มคำนวณค่าคงที่ สร้างส่วนโปรแกรมและตารางต่าง ๆ ไว้ใช้งานต่อไป

2) เฟส 1 2 3 (แสดงในรูป3.13 3.15 3.17)แบ่งการทำงานออกเป็น 2 ส่วน คือ การกำหนดส่วนโปรแกรมที่ใช้ในการปฏิบัติงาน และการปฏิบัติงานจริง โมดุลที่ใช้ในการกำหนดส่วนโปรแกรมที่ใช้ในการปฏิบัติงานจะทำหน้าที่ตัดส่วนโปรแกรมให้เหลือเฉพาะที่เหมาะกับการทำงานในแต่ละครั้งเท่านั้น นอกจากนี้โมดุลยังทำหน้าที่เลือกโมดุลต่อไปที่จะนำเข้ามาในหน่วยความจำอีกด้วย ส่วนการวางส่วนโปรแกรมในหน่วยความจำเป็นหน้าที่ของโมดุลที่ทำหน้าที่นี้โดยเฉพาะของแต่ละเฟส การจัดวางโมดุลในหน่วยความจำดังแสดงในรูป 3.14 3.16 และ3.18ตามลำดับ

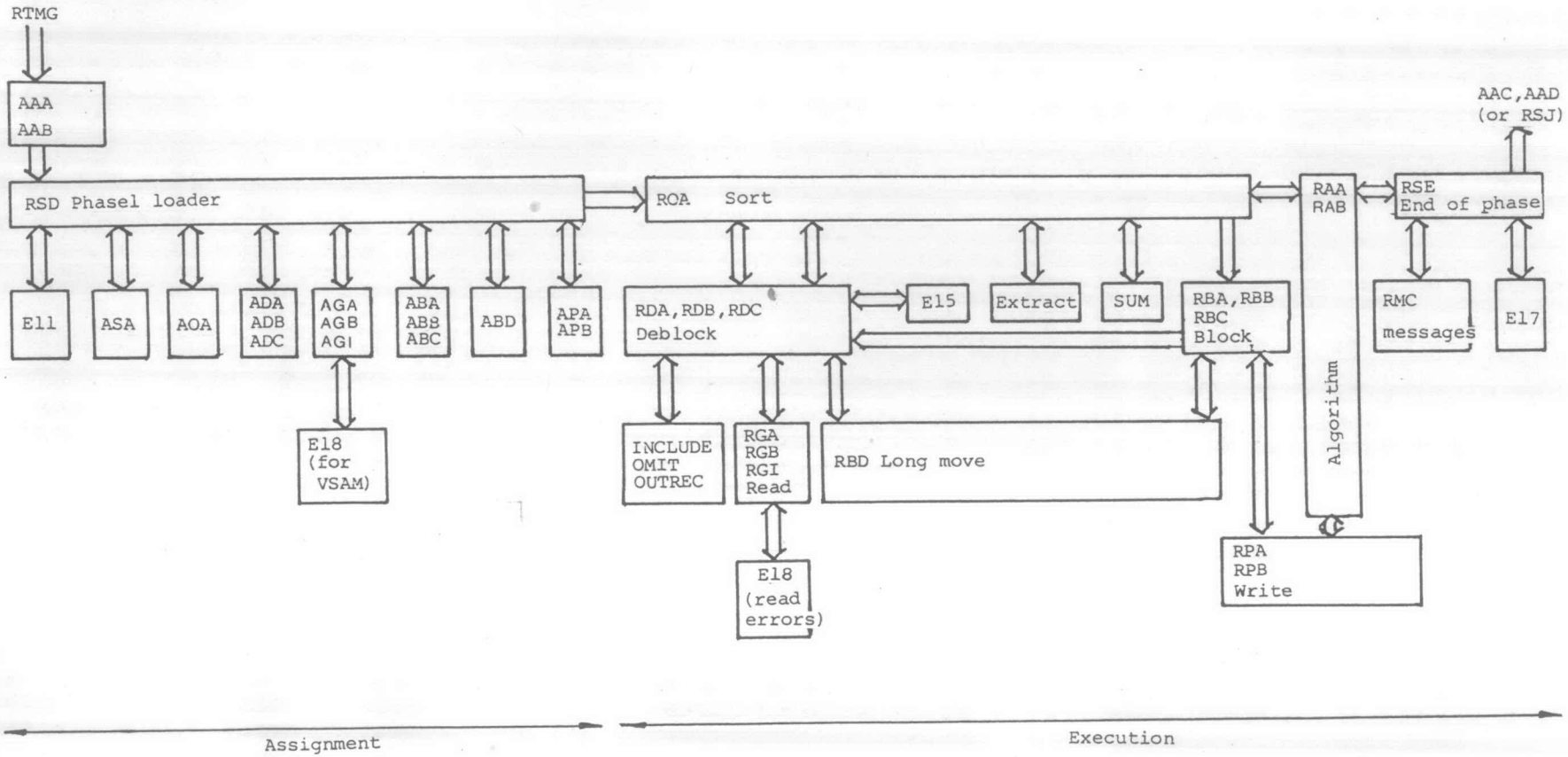


Program control statement processing

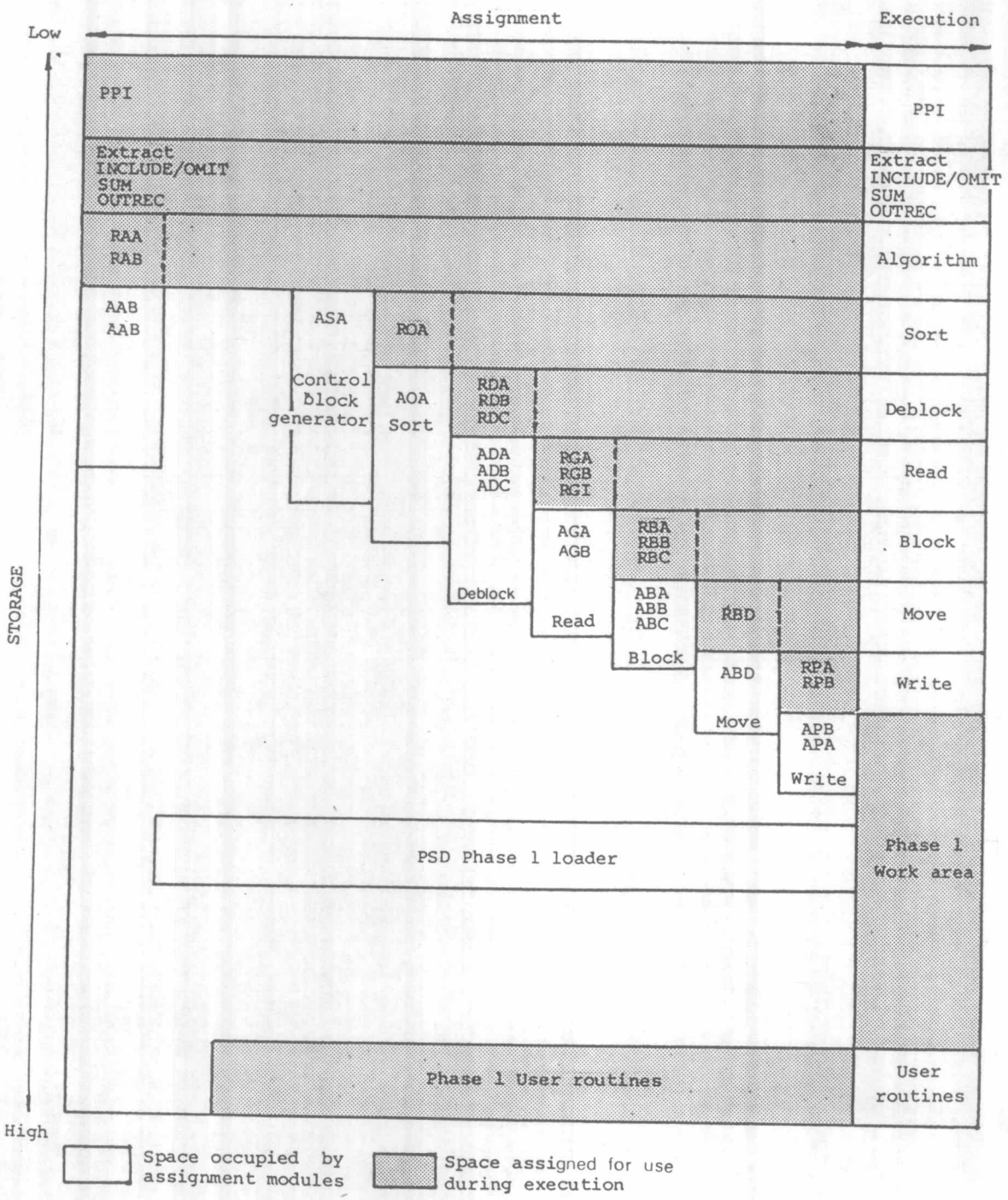
รูปที่ 3.11 ความสัมพันธ์ของโมดุลต่างๆในเฟส 0



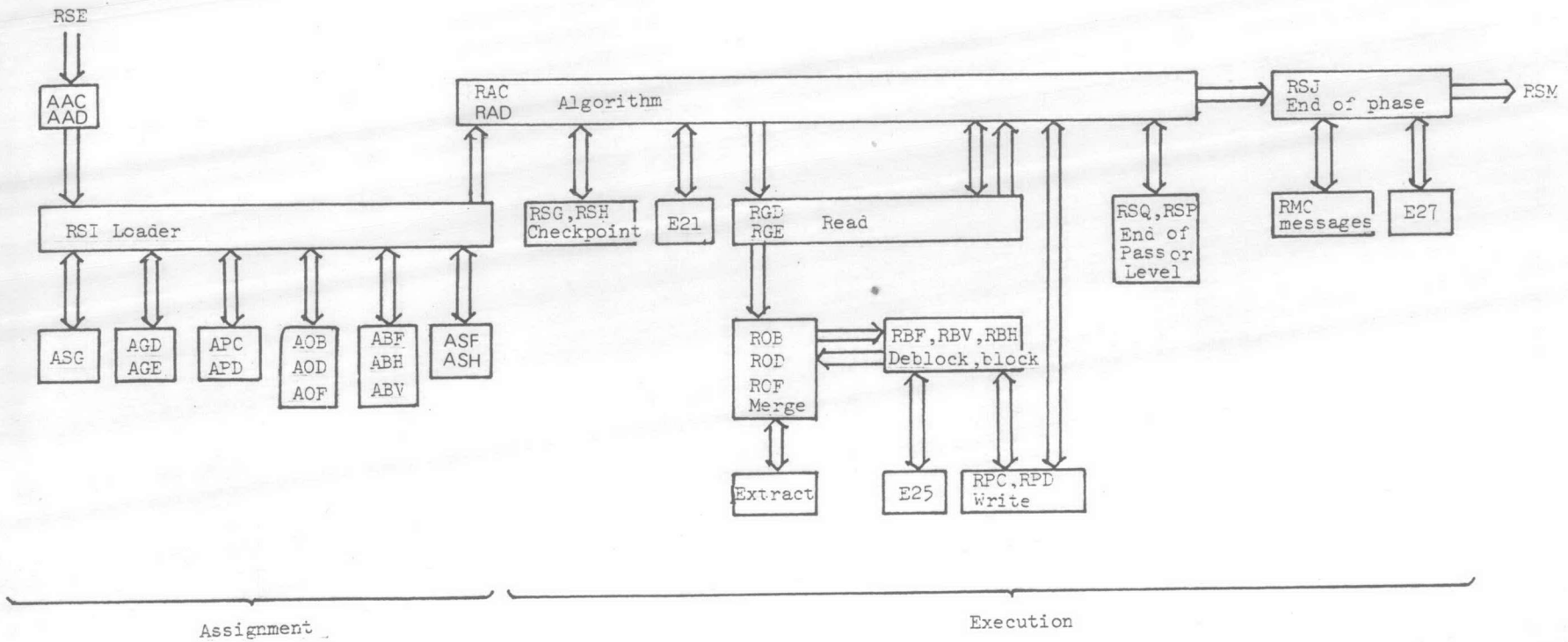
รูปที่ 3.12 การจัดวางโมดูลต่างๆในหน่วยความจำของเฟส 0



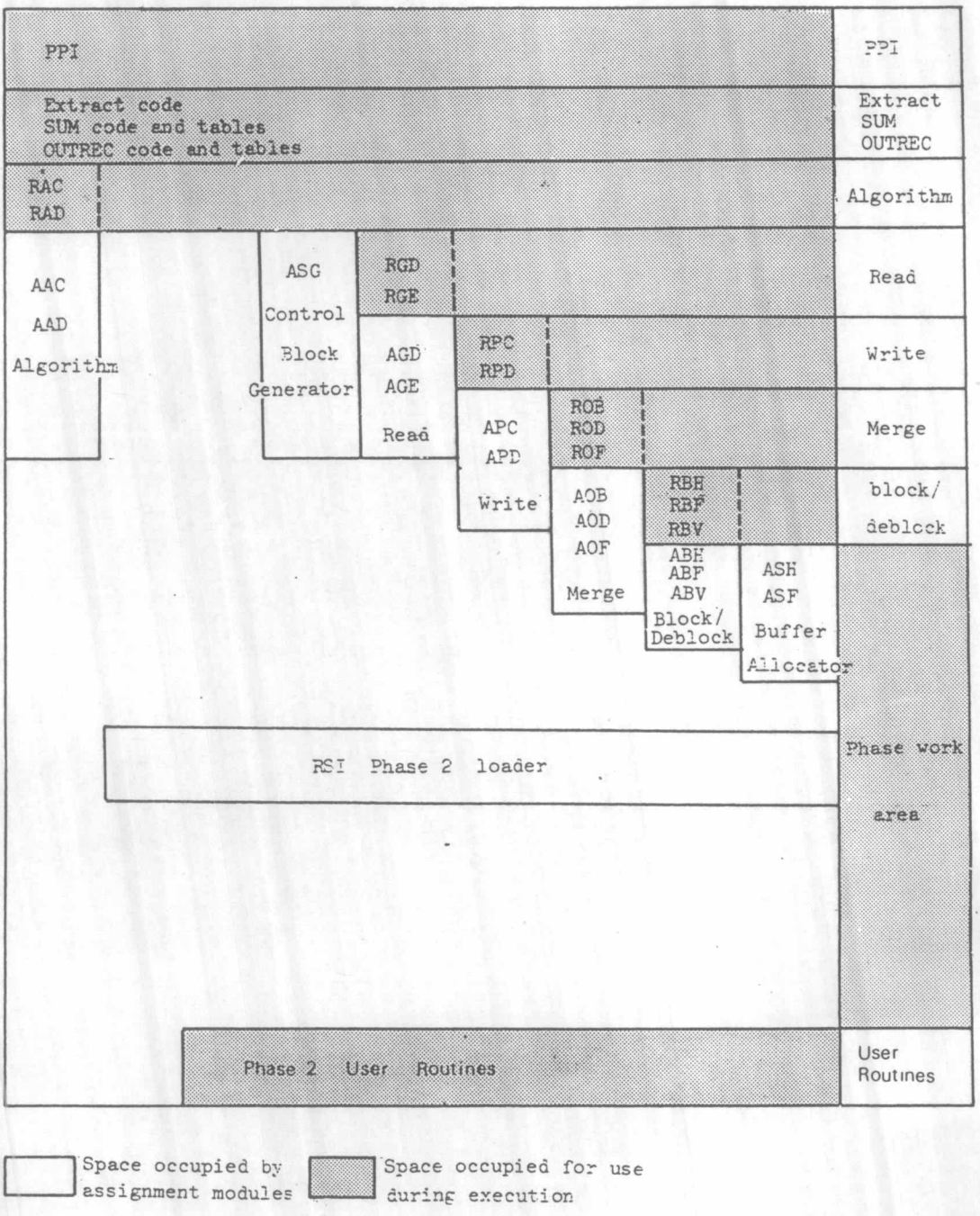
รูปที่ 3.13 ความสัมพันธ์ของโมดุลต่างๆในเฟส 1



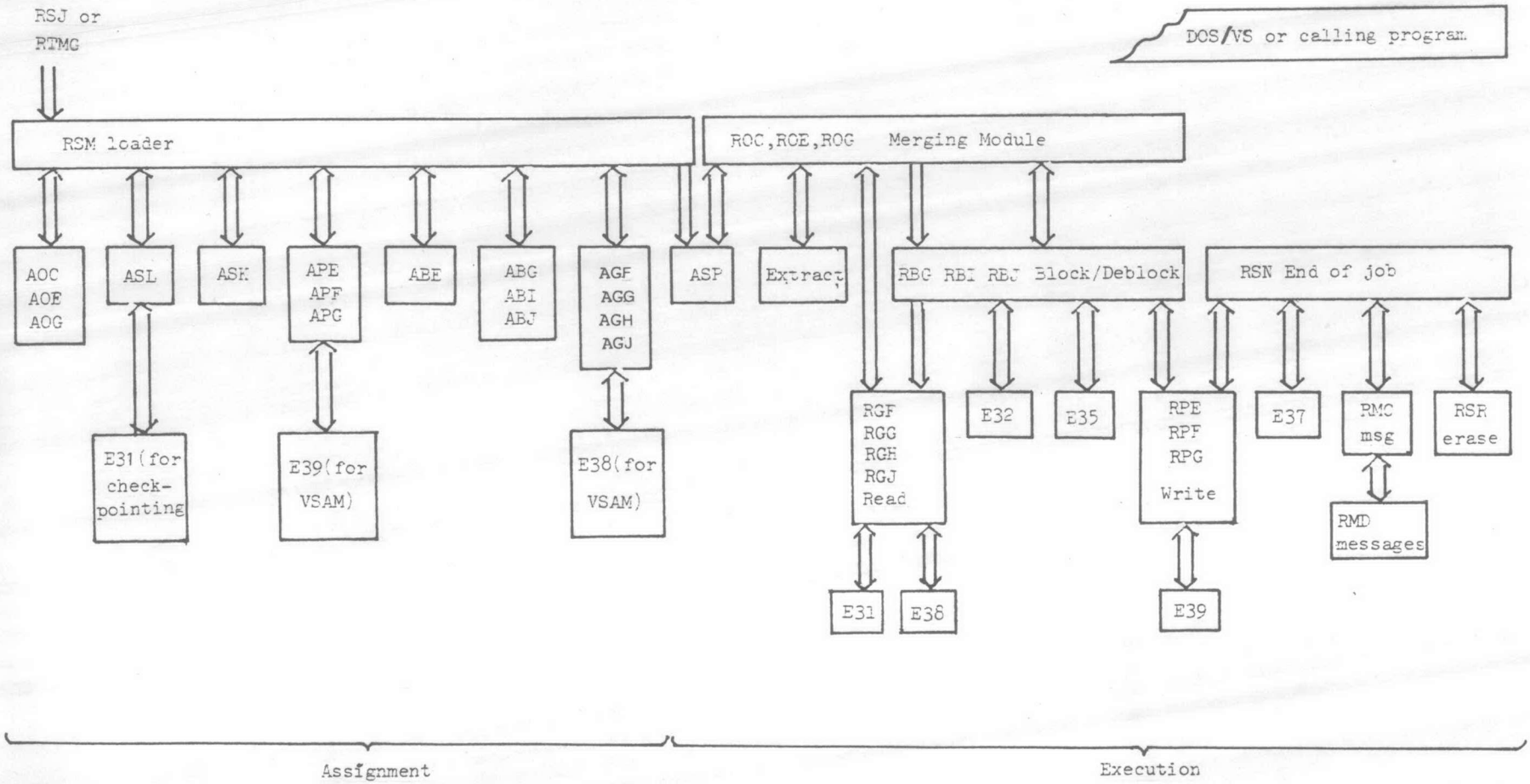
รูปที่ 3.14 การจัดวางโมดูลต่างๆในหน่วยความจำของเฟส 1



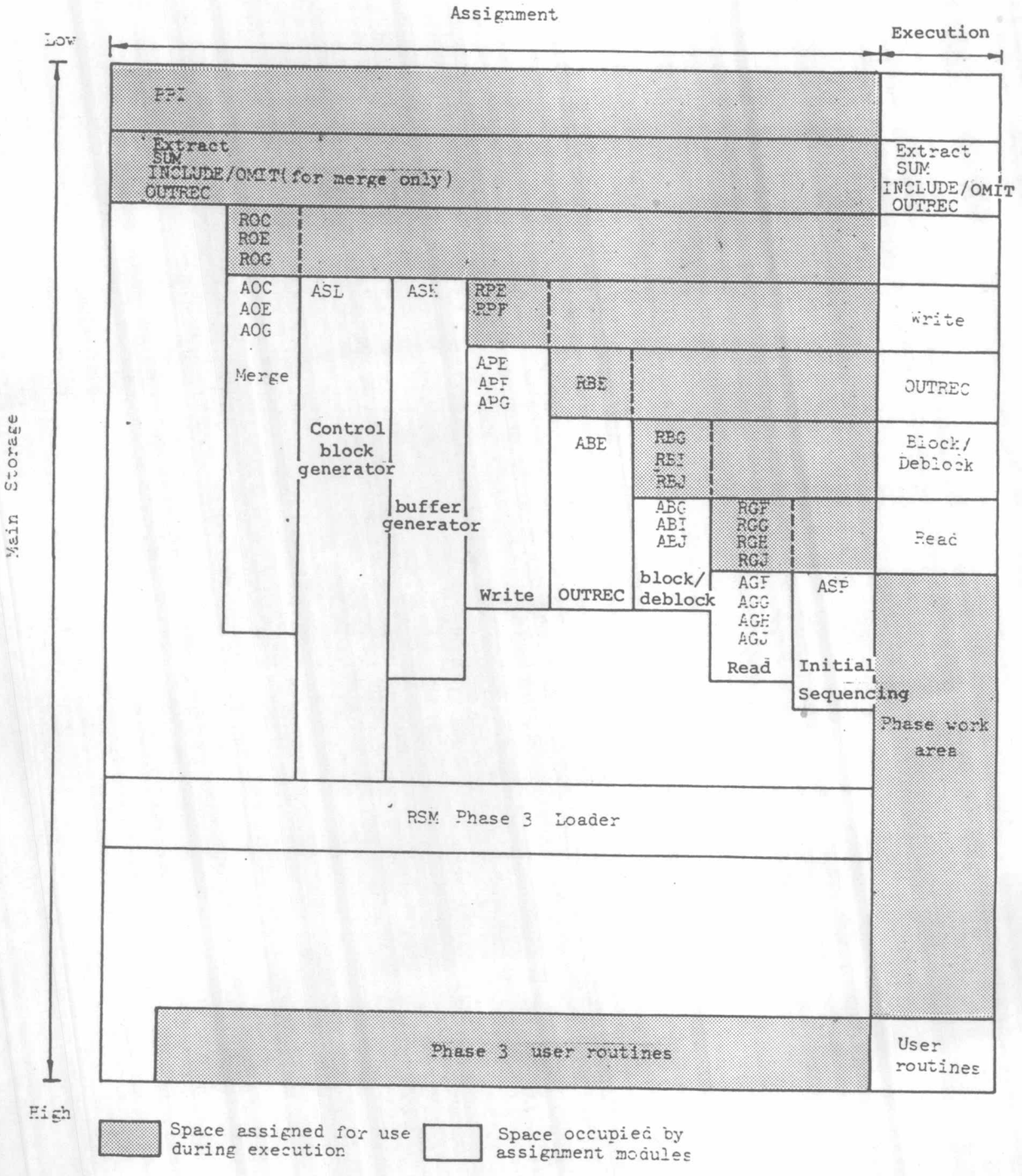
รูปที่ 3.15 ความสัมพันธ์ของโมดุลต่างๆในเฟส 2



รูปที่ 3.16 การจัดวางโมดูลต่างๆในหน่วยความจำของเฟส 2



รูปที่ 3.17 ความสัมพันธ์ของโมดูลต่างๆในเฟส 3



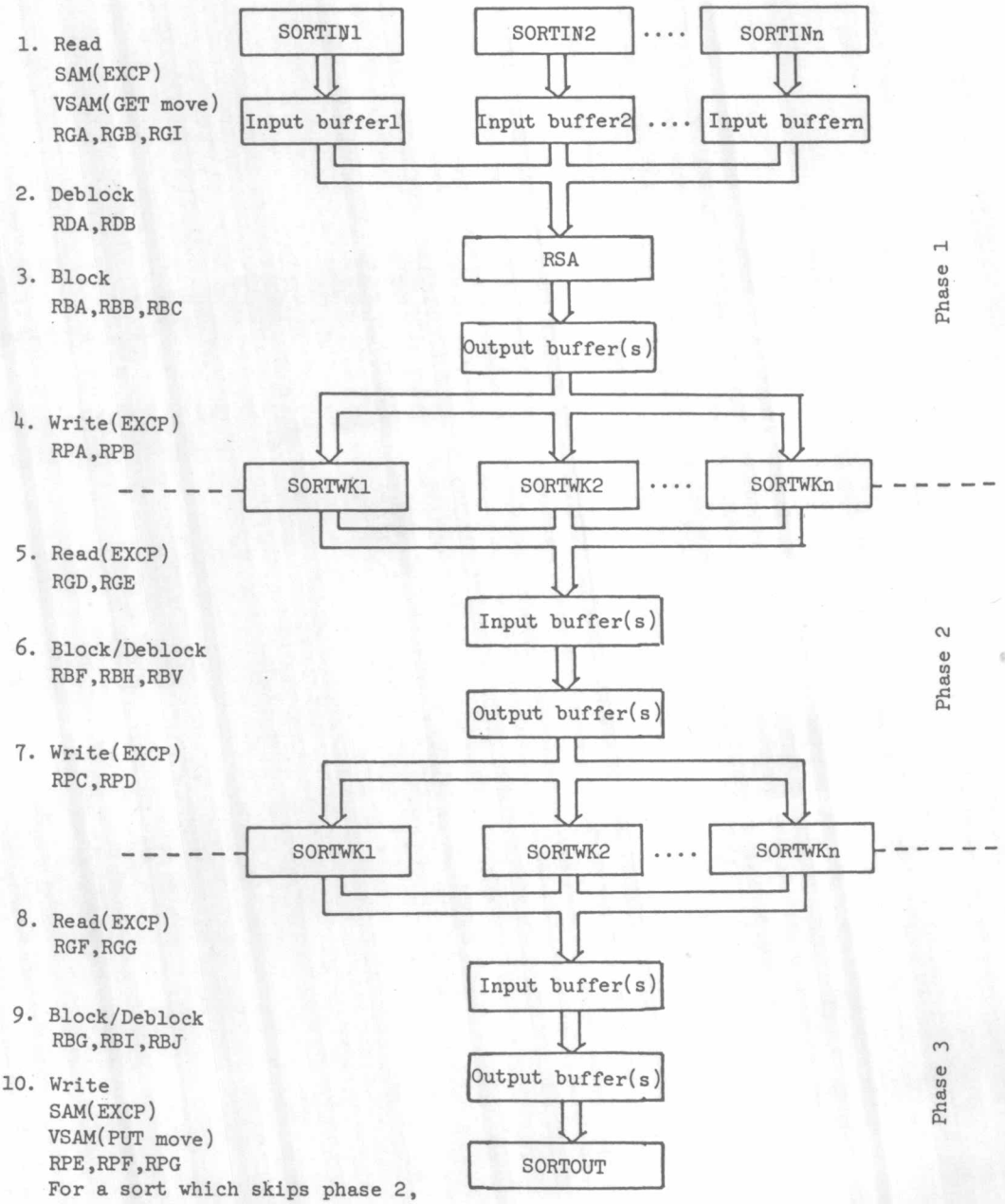
รูปที่ 3.18 การจัดวางโมดูลต่างๆในหน่วยความจำของเฟส 3

4. การเคลื่อนระเบียนข้อมูลในการทำงานของโปรแกรม

ระเบียนจะเริ่มมีการเคลื่อนย้ายตำแหน่งตั้งแต่การทำงานในเฟส 1 เป็นต้นไป ในเฟส 1 นั้น ระเบียนข้อมูลจะเคลื่อนจากแฟ้มข้อมูลนำเข้าไปเก็บในบัฟเฟอร์ข้อมูลนำเข้า ต่อจากนั้นนำไปเก็บใน RSA (Record Storage Area) ซึ่งเป็นบริเวณที่เก็บข้อมูลที่กำลังเรียงลำดับเมื่อได้วินเนอร์ (รายละเอียดในภาคผนวก จ.) จะนำวินเนอร์ไปเก็บในบัฟเฟอร์ผลลัพธ์จนเต็ม บันทึกบัฟเฟอร์ผลลัพธ์ตามหลักการกระจายบนอุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราว

ในเฟส 2 และ 3 เป็นการรวมแฟ้มข้อมูล ซึ่งทำหน้าที่คล้ายกัน คือนำข้อมูลจากอุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราวใส่ในบัฟเฟอร์ข้อมูลนำเข้า รวมแฟ้มข้อมูล นำไปเก็บในบัฟเฟอร์ผลลัพธ์ และบันทึกข้อมูล ในเฟส 2 จะบันทึกข้อมูลลงบนอุปกรณ์ที่ใช้เป็นที่เก็บข้อมูลชั่วคราว ส่วนเฟส 3 จะบันทึกข้อมูลบนอุปกรณ์ที่ผู้ใช้งานกำหนดให้เป็นที่เก็บผลลัพธ์

การเคลื่อนย้ายระเบียนข้อมูลดังแสดงในรูป 3.19



รูปที่ 3.19 การเคลื่อนระเบียนข้อมูล