บทที่ 4

การศึกษาสมรรถนะของหน่วยประมวลผล

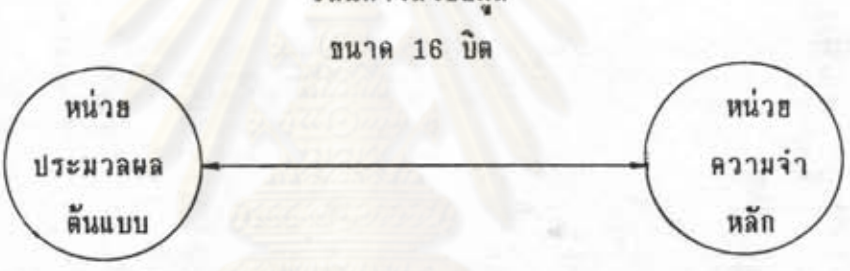
การศึกษาวิเคราะห์สมรรถนะการทำงานของหน่วยประมวลผล เป็นส่วนสำคัญในการออกแบบและพัฒนาระบบคอมพิวเตอร์ตลอดจนถึงหน่วยประมวลผลกลาง โดยพัฒนาตัวแบบจำลองระบบขึ้นมาเป็นเครื่องมือสำหรับช่วยให้ผู้ออกแบบได้แนวความคิดและความเข้าใจจากผลการศึกษาสมรรถนะการทำงานของระบบที่ออกแบบแล้วนำไปใช้ปรับปรุงระบบให้ทำงานได้ตามข้อกำหนด ในการพัฒนาตัวแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบนี้ เป็นการพิจารณาหน่วยประมวลผลในรูปลักษณะของกลุ่มทรัพยากรทางด้านฮาร์ดแวร์ของระบบและผลกระทบที่มีต่อกัน โดยมุ่งศึกษาการเข้าใช้ประโยชน์จากทรัพยากรและการตอบสนองการทำงานของทรัพยากรเหล่านั้น ซึ่งการรวบรวมและให้ข้อมูลเหล่านี้มีประโยชน์ในงานด้านการออกแบบ รวมถึงการตรวจสอบเพื่อยืนยันผลการวิเคราะห์และการประมาณสมรรถนะของหน่วยประมวลผลที่ได้ออกแบบ

งานวิจัยนี้นำเอาแนวความคิดการจำลองอิงกรรมวิธีมาประยุกต์ใช้ในการพัฒนาแบบจำลองเพื่อศึกษาสมรรถนะของหน่วยประมวลผล โดยกำหนดหน่วยประมวลผลต้นแบบสำหรับการทำงานวิจัย ในรูปลักษณะที่คล้ายคลึงไมโครโปรเซสเซอร์ 8086 ของบริษัทอินเทล เมื่อศึกษาแนวความคิดพื้นฐานทางด้านสถาปัตยกรรม ตลอดจนการติดต่อระหว่างกันขององค์ประกอบภายในหน่วยประมวลผล แล้วจึงออกแบบและพัฒนาเป็นโปรแกรมสำหรับใช้ดำเนินการทดสอบและศึกษาผลลัพธ์ที่ได้ตามต้องการโดยผู้ใช้สามารถกำหนดพารามิเตอร์บางอย่างได้ เช่น การเพิ่มหรือลดขนาดคิวคำสั่ง การกำหนดให้มีหน่วยความจำแคชในระบบ เป็นต้น จึงทำให้ใช้ตัวแบบจำลองได้หลายรูปลักษณะ

ในบทนี้ได้แบ่งหัวข้อการบรรยายออกเป็น 3 ส่วน โดยในหัวข้อแรกกล่าวถึงหน่วยประมวลผลต้นแบบที่ใช้เป็นแนวทางในการสร้างแบบจำลองสำหรับงานวิจัยนี้ หัวข้อถัดไป อธิบายถึงการพัฒนาแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบโดยอาศัยตัวจำลองอิงกรรมวิธี np และสุดท้าย กล่าวถึงกลุ่มโปรแกรมที่ใช้ในการทดสอบและศึกษาสมรรถนะของหน่วยประมวลผลต้นแบบ

4.1 หน่วยประมวลผลต้นแบบ

หน่วยประมวลผลต้นแบบสำหรับใช้ในการศึกษาสมรรถนะ เป็นหน่วยประมวลผลขนาดเล็กมาก (microprocessor) แบบ 16 บิต ซึ่งสามารถอ้างอิงตำแหน่งในหน่วยความจำหลักได้ถึง 1 เมกกะไบต์ (megabytes) และมีเส้นทางส่งข้อมูล (data path) ขนาด 16 บิต ซึ่งดำเนินการดูแลและควบคุมโดยส่วนประกอบสำคัญภายในหน่วยประมวลผลต้นแบบคือ หน่วยเชื่อมต่อบัส ทำให้หน่วยประมวลผลต้นแบบสามารถติดต่อกับหน่วยความจำหลักเพื่อรับและส่งข้อมูลได้อย่างถูกต้อง (ดูรูปที่ 4.1)



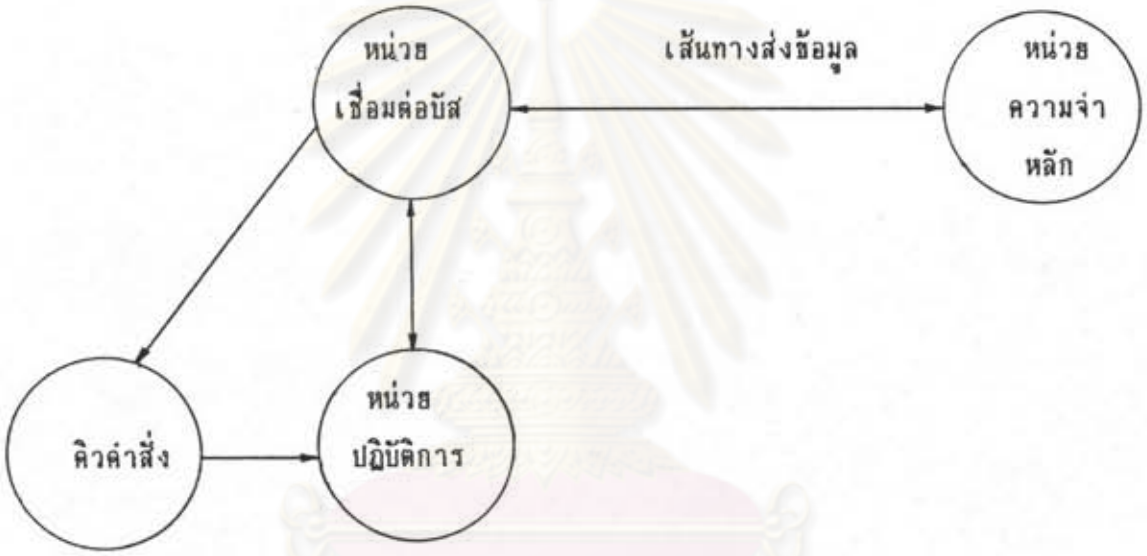
รูปที่ 4.1 การติดต่อระหว่างหน่วยประมวลผลต้นแบบกับหน่วยความจำหลัก

โดยทั่วไปหน้าที่หลักอันหนึ่งของหน่วยประมวลผลคือ ปฏิบัติการตามคำสั่งในโปรแกรมที่เก็บอยู่ในหน่วยความจำหลักตามลำดับ ดังนั้นในการออกแบบหน่วยประมวลผลต้องเกี่ยวข้องกับการติดตั้งใช้งานชุดคำสั่ง (instruction set) เป็นอย่างยิ่ง และยังเกี่ยวเนื่องรวมไปถึงวิธีการระบุหรืออ้างอิงตำแหน่งข้อมูลเพื่อให้การทำงานของหน่วยประมวลผลเป็นไปอย่างมีประสิทธิภาพ

4.1.1 การจัดองค์กรภายในหน่วยประมวลผลต้นแบบ

โครงสร้างภายในของหน่วยประมวลผลต้นแบบ (ดูรูปที่ 4.2) แบ่งออกเป็น 2 ส่วนหลัก คือหน่วยเชื่อมต่อบัสและหน่วยปฏิบัติการ โดยหน่วยเชื่อมต่อบัส ทำหน้าที่ดึงเอาข้อมูลมาจากหน่วยความจำหลัก ซึ่งถ้าข้อมูลเป็นคำสั่งจะถูกเก็บไว้ในคิวคำสั่ง หรือถ้าข้อมูลเป็นตัวถูกกระทำการ (operand) จะถูกส่งไปยังหน่วยปฏิบัติการ และยังทำหน้าที่รับข้อมูลที่เป็นผลลัพธ์จากการทำงานของคำสั่งไปบันทึกในหน่วยความจำหลัก สำหรับหน่วยปฏิบัติการ ทำหน้าที่ปฏิบัติการตามคำสั่งที่ดึงมาจากคิวคำสั่ง และติดต่อกับหน่วยเชื่อมต่อบัส เมื่อต้องการข้อมูลในหน่วยความจำสำหรับกระทำการ

ส่วนประกอบย่อยอื่นๆภายในหน่วยประมวลผลต้นแบบที่เกี่ยวข้องกับการทำงานที่เกิดขึ้นได้แก่คิวคำสั่ง ซึ่งเป็นเนื้อที่ที่ใช้เก็บคำสั่งที่ดึงมาจากหน่วยความจำโดยหน่วยเชื่อมต่อบัส และเมื่อหน่วยปฏิบัติการต้องการคำสั่งถัดไปสำหรับการทำงาน ก็สามารถดึงจากคิวคำสั่งนี้โดยตรงทำให้ไม่ต้องเสียเวลาในการติดต่อกับหน่วยความจำหลัก นอกจากนี้ยังเกี่ยวข้องกับเส้นทางส่งข้อมูล ซึ่งเป็นเส้นทางสำหรับการส่งผ่านข้อมูลระหว่างหน่วยประมวลผลต้นแบบและหน่วยความจำหลัก



รูปที่ 4.2 องค์ประกอบภายในหน่วยประมวลผลต้นแบบ

4.1.2 <u>การปฏิบัติการคำสั่ง</u>

โดยทั่วไปลำดับเหตุการณ์สำหรับการปฏิบัติการตามคำสั่งในโปรแกรม จะเป็นดังนี้

1) ดึงคำสั่งถัดไปมาจากหน่วยความจำหลัก (fetch instruction)
2) อ่านค่าข้อมูลของตัวถูกกระทำการ เมื่อคำสั่งนั้นจำเป็นต้องใช้
3) ปฏิบัติการตามคำสั่ง
4) บันทึกข้อมูลผลลัพธ์ เมื่อคำสั่งนั้นระบุไว้

หน่วยประมวลผลต้นแบบนี้ ออกแบบขึ้นมาให้สามารถกระทำการเหล่านี้ได้พร้อมกันในบางขณะ โดยจัดสรรไปยังส่วนที่ทำการประมวลผลภายในคือ หน่วยเชื่อมต่อบัส และหน่วยปฏิบัติการ ทั้ง 2 ส่วนนี้ทำงานอย่างอิสระต่อกันและโดยมากมักจะเกิดการทับซ้อน

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

(overlap) กันระหว่างการดึงคำสั่งกับการปฏิบัติการ ซึ่งทำให้หน่วยปฏิบัติการสามารถดำเนินการตามคำสั่งที่หน่วยเชื่อมต่อบัสดึงมาจากหน่วยความจำหลักและเก็บไว้ในคิวคำสั่งแล้ว ดังนั้นเวลาที่ใช้ในการดึงคำสั่งจากหน่วยความจำหลัก จึงถูกกำจัดไปได้เป็นบางส่วน

ภายในหน่วยปฏิบัติการ มีเส้นทางส่งข้อมูลเพื่อการติดต่อกับรีจิสเตอร์ต่างๆ ขนาด 16 บิต โดยที่หน่วยปฏิบัติการนี้จะไม่มีการติดต่อกับหน่วยความจำหลักโดยตรง ต้องกระทำแทนโดยหน่วยเชื่อมต่อบัส เมื่อคำสั่งที่กำลังปฏิบัติการต้องการข้อมูลจากหน่วยความจำหลักหรืออุปกรณ์นำข้อมูลเข้า/ออก หน่วยปฏิบัติการส่งข่าวสารไปยังหน่วยเชื่อมต่อบัสให้อ่านหรือบันทึกข้อมูลนั้น โดยที่หน่วยปฏิบัติการเพียงแต่จัดสร้างเฉพาะตำแหน่งขนาด 16 บิตเพื่อใช้เป็นค่าออฟเซต (offset) แล้วส่งไปให้หน่วยเชื่อมต่อบัสนำไปใช้ในการอ้างอิงตำแหน่งในหน่วยความจำหลัก นั่นคือหน่วยเชื่อมต่อบัสทำการปฏิบัติการที่เกี่ยวข้องกับบัสทั้งหมดให้แก่หน่วยปฏิบัติการ รวมทั้งการส่งผ่านข้อมูลทั้งหลายระหว่างหน่วยปฏิบัติการกับหน่วยความจำหลักหรืออุปกรณ์ต่อรอบนอกต่าง ๆ เมื่อหน่วยปฏิบัติการต้องการ

ในขณะที่หน่วยปฏิบัติการกำลังปฏิบัติการคำสั่งใดๆอยู่ หน่วยเชื่อมต่อบัสก็จะดึงคำสั่งถัดไปตามลำดับจากหน่วยความจำหลักเข้ามาเพิ่มโดยเก็บไว้ในคิวคำสั่ง ทั้งนี้การดึงคำสั่งเข้ามาบรรจุไว้ในคิวคำสั่งของหน่วยเชื่อมต่อบัส ต้องเป็นไปตามเงื่อนไขเหล่านี้คือ

1) เมื่อมีที่ว่างในคิวคำสั่งไม่น้อยกว่าจำนวนไบต์สูงสุดในการส่งผ่านข้อมูลภายในเส้นทางส่งข้อมูล

2) เมื่อไม่มีการขอใช้บัสของหน่วยปฏิบัติการรออยู่ แต่ถ้าหน่วยปฏิบัติการต้องการเข้าใช้บัสในขณะที่หน่วยเชื่อมต่อบัสกำลังอยู่ในระหว่างการดึงคำสั่งแล้ว หน่วยเชื่อมต่อบัสต้องทำการนี้ให้เสร็จก่อนแล้วจึงดำเนินการตอบสนองแก่หน่วยปฏิบัติการ

3) ในกรณีที่มีการย้ายการควบคุมตามการปฏิบัติการของคำสั่งขณะนั้น ไปยังตำแหน่งอื่นของหน่วยความจำหลัก หน่วยเชื่อมต่อบัสจะเริ่มต้นการบรรจุข้อมูลใหม่ทั้งหมดให้แก่คิวคำสั่ง

4.1.3 <u>วิธีการอ้างตำแหน่ง</u> (Addressing mode)

หน่วยประมวลผลต้นแบบนี้ได้กำหนดใช้หน่วยความจำหลักของระบบเป็นแบบปล้อง (segmentation) ซึ่งสามารถมีตำแหน่งแท้จริงของหน่วยความจำหลักที่อ้างอิงได้ถึง 1 เมกกะไบต์ ดังนั้นในการอ้างถึงแต่ละตำแหน่งของหน่วยความจำหลักต้องใช้จำนวนบิตข้อมูลอย่างน้อย 20 บิต โดยต้องอาศัยการทำงานของรีจิสเตอร์ 2 ตัวคือ รีจิสเตอร์กำหนดเซกเมนต์และรีจิสเตอร์ตัวชี้

ในการคำนวณหาตำแหน่งแท้จริงของข้อมูลที่อ้างถึงในหน่วยความจำหลัก ได้จากการรวมกันของค่าเซกเมนต์ (segment) ซึ่งได้จากรีจิสเตอร์กำหนดเซกเมนต์ และค่าออฟเซต ซึ่งได้จากการคำนวณตามวิธีการอ้างตำแหน่งของข้อมูล โดยเริ่มจากการเลื่อนบิตข้อมูลในรีจิสเตอร์กำหนดเซกเมนต์ไปทางซ้าย 4 บิต แล้วจึงบวกกับค่าออฟเซต ตัวอย่างเช่น การดึงคำสั่งจากหน่วยความจำหลักสำหรับปฏิบัติการ ต้องใช้ค่าในรีจิสเตอร์ CS กับค่าในรีจิสเตอร์ IP ซึ่งเก็บค่าออฟเซต สมมติให้ รีจิสเตอร์ CS มีค่า E89FH และรีจิสเตอร์ IP มีค่า 0003H ดังนั้นตำแหน่งแท้จริงของคำสั่งที่ต้องการดึงคือ E89F3H

วิธีการอ้างตำแหน่งในหน่วยความจำหลัก เป็นวิธีการที่หน่วยประมวลผลต้นแบบใช้ในการเข้าถึงตัวถูกกระทำการตามที่ระบุในคำสั่ง ข้อมูลสำหรับตัวถูกกระทำการอาจเป็นส่วนหนึ่งของคำสั่ง อยู่ในรีจิสเตอร์ ในหน่วยความจำ หรือในหน่วยข้อมูลเข้า/ออก วิธีการอ้างตำแหน่งของข้อมูลสำหรับหน่วยประมวลผลต้นแบบนี้ มี 7 วิธีการคือ (ดูรูปที่ 4.3)

1) แบบรีจิสเตอร์ (Register)

ค่าของตัวถูกกระทำการได้มาจากรีจิสเตอร์โดยตรง ไม่ต้องอ้างอิงหน่วยความจำหลัก เวลาที่ใช้ในการเข้าถึงข้อมูลในรีจิสเตอร์น้อยกว่าในหน่วยความจำหลัก

2) แบบทันที (Immidiate)

เป็นวิธีการที่ง่ายที่สุดในการกระทำการ เนื่องจากไม่ต้องมีการอ้างถึงตำแหน่งในหน่วยความจำหลัก ตัวถูกกระทำการเป็นค่าคงที่ซึ่งอาจเป็นข้อมูลขนาด 8 บิตหรือ 16 บิตโดยค่าคงที่นั้นอยู่ในไบต์ถัดไปจากคำสั่ง จึงทำให้ประหยัดช่วงเวลารอบเก็บข้อมูล (memory cycle)

3) แบบโดยตรง (Direct)

วิธีการนี้ทำการอ้างอิงหน่วยความจำหลัก 1 ครั้ง โดยคำนวณตำแหน่งแท้จริงของข้อมูลที่ต้องการ ได้จากค่าตำแหน่งที่อยู่ของตัวถูกกระทำการ (effective address) ซึ่งอยู่ในไบต์ถัดไปจากคำสั่ง รวมกับค่าเซกเมนต์

4) แบบรีจิสเตอร์โดยอ้อม (Register Indirect)

วิธีการนี้ทำการอ้างอิงหน่วยความจำหลักเพียงครั้งเดียว โดยไม่มีการคำนวณใดๆเป็นพิเศษ ค่าตำแหน่งที่อยู่ของตัวถูกกระทำการ อยู่ในเบสรีจิสเตอร์ เบสพอยน์เตอร์ หรือรีจิสเตอร์ตัวชี้

5) แบบฐานสัมพัทธ์ (Base Relative)

ค่าตำแหน่งที่อยู่ของตัวถูกกระทำการคือ ผลบวกของค่าขจัด (displacement) และค่าในเบสรีจิสเตอร์หรือเบสพอยน์เตอร์ โดยค่าขจัดอยู่ในไบต์ถัดไปจากคำสั่ง

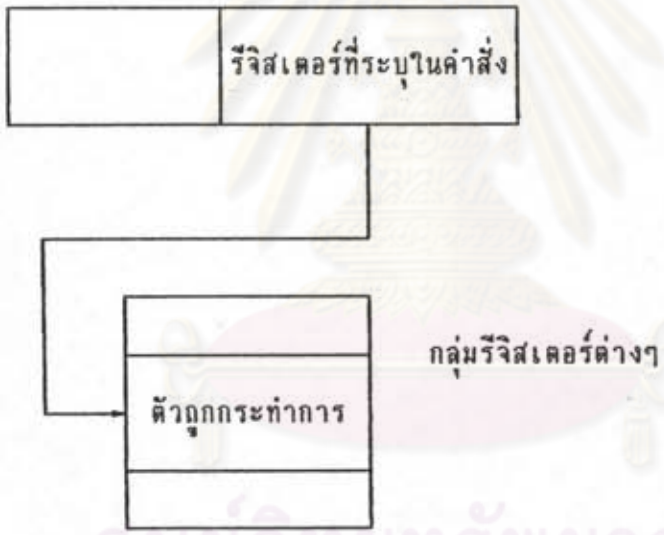
6) แบบดัชนีโดยตรง (Direct Indexed)

ค่าตำแหน่งที่อยู่ของตัวถูกกระทำการคือ ผลบวกของค่าขจัด และค่าในรีจิสเตอร์ตัวชี้ โดยที่ค่าขจัดอาจมีอยู่ในคำสั่งหรือไม่ก็ได้

7) แบบดัชนีฐาน (Base Indexed)

ค่าตำแหน่งที่อยู่ของตัวถูกกระทำการคือ ผลบวกของค่าในเบสรีจิสเตอร์ ค่าในรีจิสเตอร์ตัวชี้ และค่าขจัดซึ่งอาจจะมีหรือไม่มีก็ได้

<u>แบบรีจิสเตอร์</u>



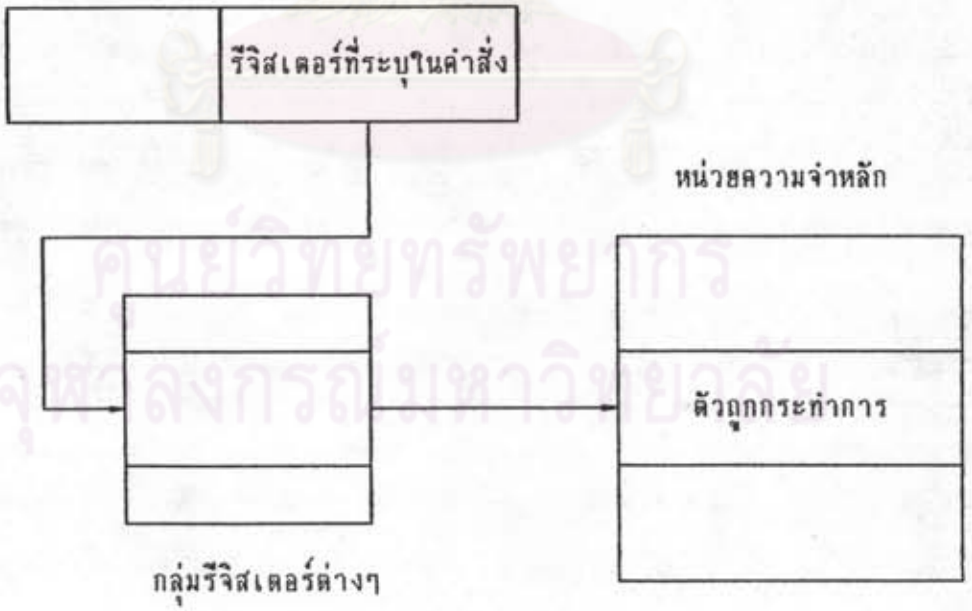
ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

<u>แบบทันที</u>

รูปที่ 4.3 วิธีการอ้างตำแหน่ง

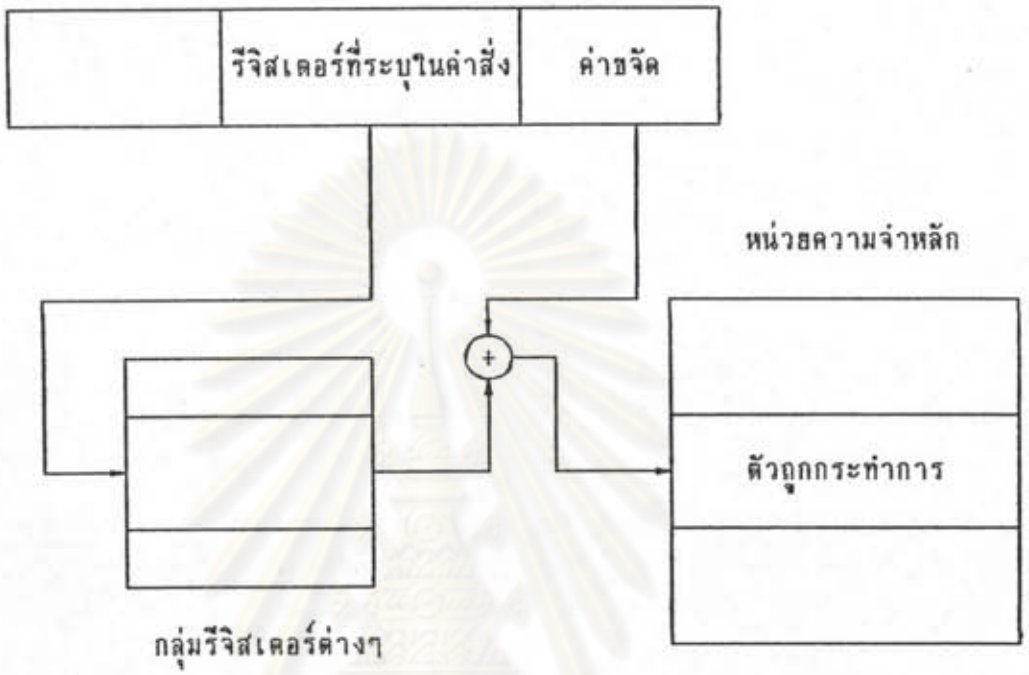
แบบโดยตรง

แบบรีจิสเตอร์โดยอ้อม

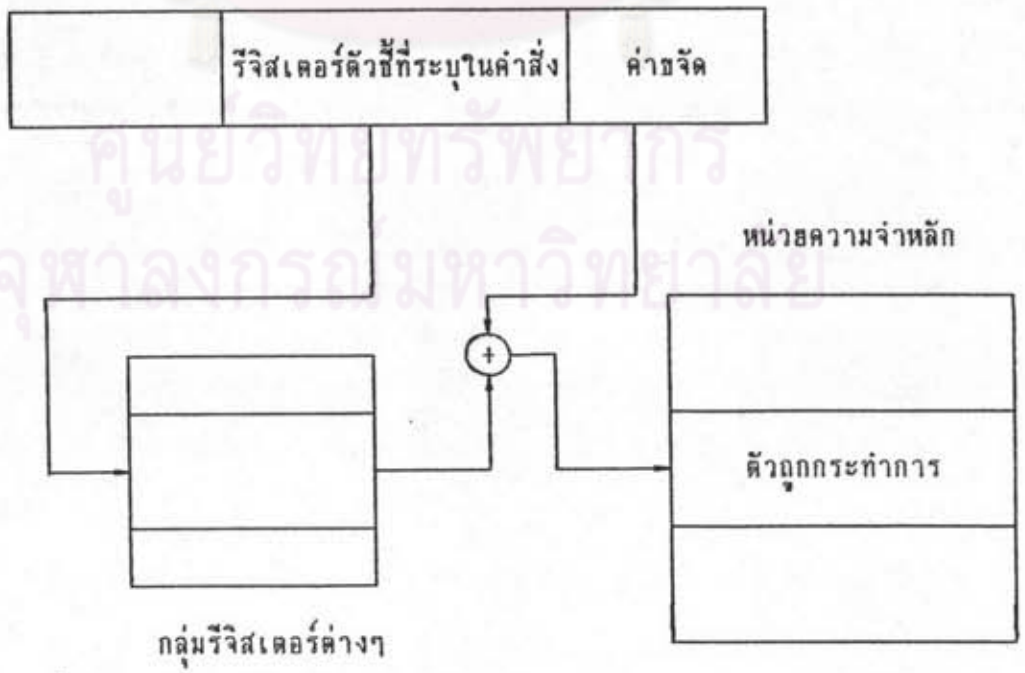


รูปที่ 4.3 วิธีการอ้างตำแหน่ง (ต่อ)

แบบฐานสัมพัทธ์



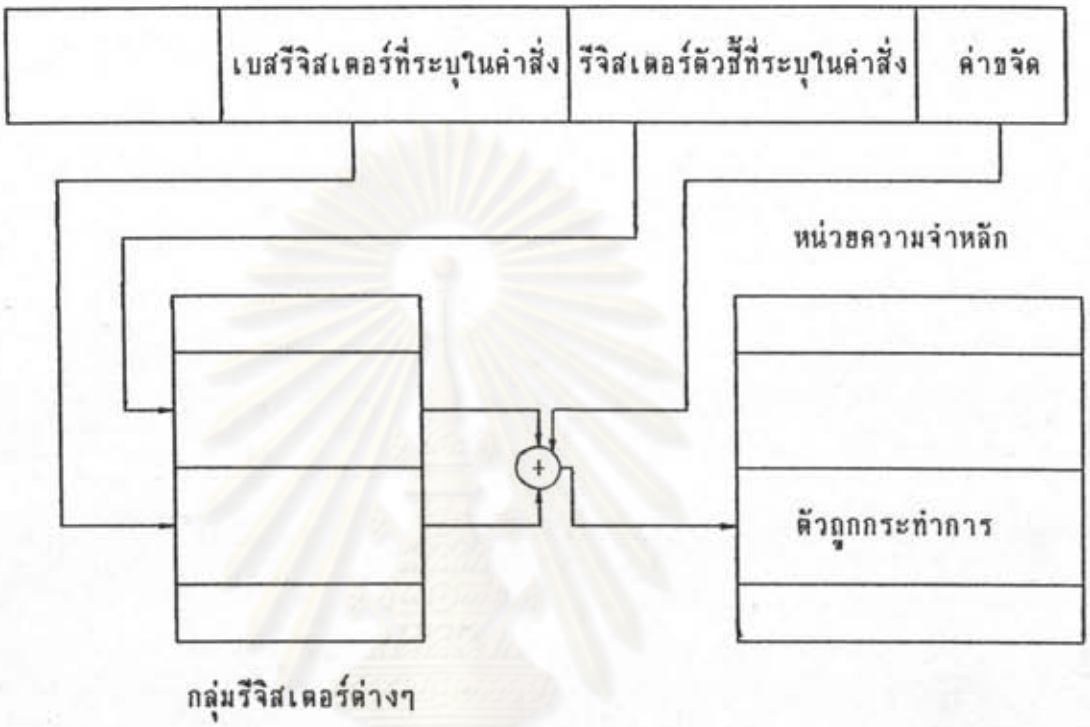
แบบดัชนีโดยตรง



รูปที่ 4.3 วิธีการอ้างตำแหน่ง (ต่อ)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

แบบดัชนีฐาน



รูปที่ 4.3 วิธีการอ้างตำแหน่ง (ต่อ)

4.1.4 ชุดคำสั่งสำหรับปฏิบัติการ

การดำเนินการภายในระบบคอมพิวเตอร์ จำเป็นอย่างยิ่งที่ต้องมีชุดคำสั่งเพื่อให้มีการทำงานเกิดขึ้นภายในหน่วยประมวลผล ซึ่งกลุ่มคำสั่งสำหรับหน่วยประมวลผลต้นแบบนี้แบ่งตามลักษณะการทำงานออกเป็น 8 กลุ่มคือ

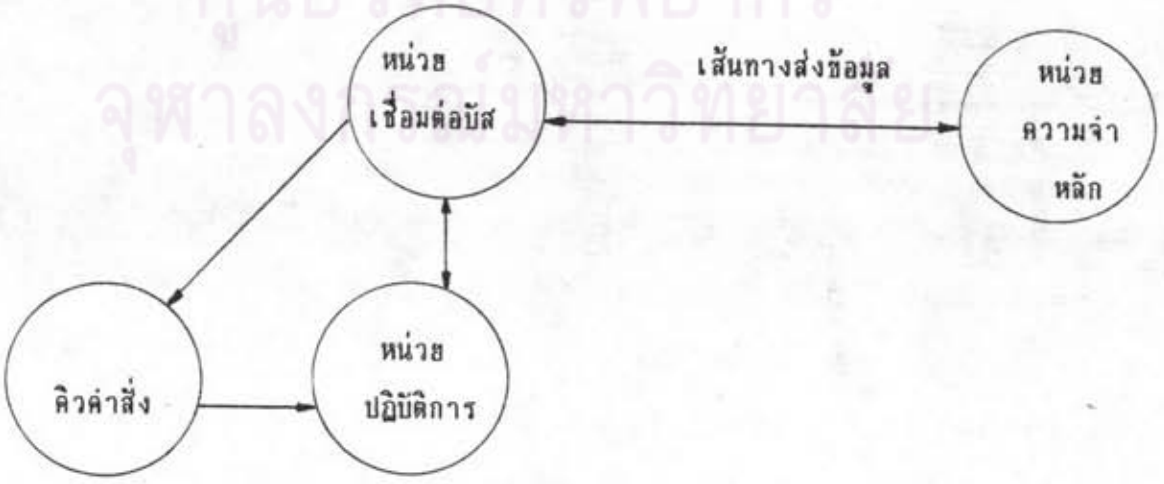
1) คำสั่งเคลื่อนย้ายข้อมูล
2) คำสั่งการคำนวณทางคณิตศาสตร์
3) คำสั่งปฏิบัติการกับบิต
4) คำสั่งปฏิบัติการกับกลุ่มตัวอักษร
5) คำสั่งโอกย้ายการควบคุม
6) คำสั่งนำหน้า (prefix)
7) คำสั่งเกี่ยวกับหน่วยข้อมูลเข้า/ออก
8) คำสั่งควบคุมหน่วยประมวลผล

สำหรับรายละเอียดเกี่ยวกับชุดคำสั่งในการปฏิบัติการได้มีการศึกษาและกำหนดตามเอกสาร (Rector and Alexy,1980) และงานวิจัยนี้ไม่พิจารณากลุ่มคำสั่งที่เกี่ยวกับหน่วยข้อมูลเข้า/ออก เนื่องจากตัวแบบจำลองที่พัฒนาขึ้นมาใช้ในการศึกษาสมรรถนะของหน่วยประมวลผลต้นแบบนี้มีการจำลองการทำงานเฉพาะส่วนของหน่วยประมวลผลต้นแบบที่เกี่ยวข้องกับหน่วยความจำ

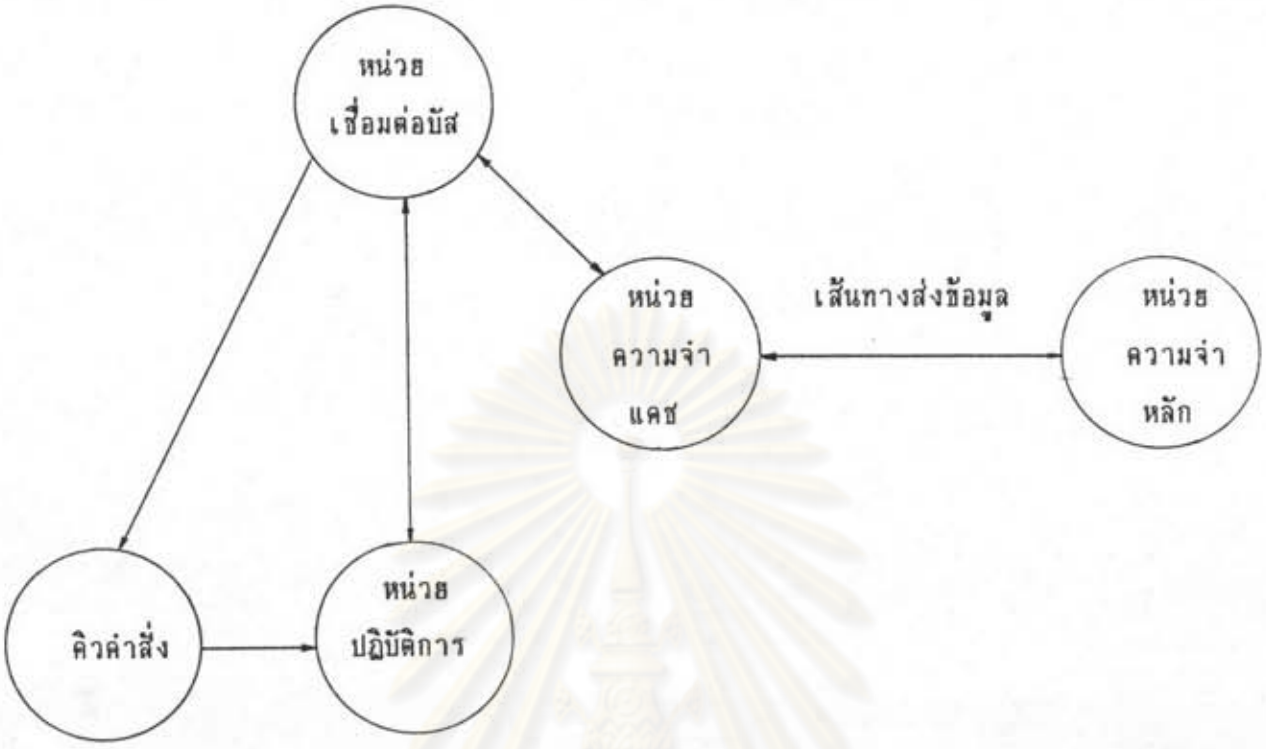
4.2 แบบจำลองการทำงานของหน่วยประมวลผลต้นแบบ

เมื่อพิจารณาส่วนประกอบต่างๆ รูปแบบการทำงาน ตลอดจนการติดต่อระหว่างกันของส่วนประกอบภายในหน่วยประมวลผลต้นแบบ ดังที่ได้กล่าวในหัวข้อ 4.1 ทำให้สามารถออกแบบและพัฒนาโปรแกรมเพื่อใช้เป็นแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบ โดยกำหนดขึ้นมาในรูปแบบกรรมวิธีที่มีการติดต่อระหว่างกัน และดำเนินการได้พร้อมกันภายในแบบจำลอง ซึ่งต้องอาศัยสิ่งอำนวยความสะดวกที่จัดไว้ให้ในตัวจำลองอิงกรรมวิธี np

การดำเนินการกับตัวแบบจำลองในการศึกษาสมรรถนะของหน่วยประมวลผลนี้ ผู้ออกแบบสามารถกำหนดรายละเอียดบางอย่าง เช่น กำหนดขนาดคิวคำสั่งเป็น 4 ไบต์ 6 ไบต์ หรือ 8 ไบต์ หรือกำหนดให้มีหน่วยความจำแคชในระบบ เป็นต้น ทำให้การปฏิบัติการคำสั่งภายในหน่วยประมวลผลเกิดขึ้นภายใต้ข้อกำหนดรายละเอียดที่ผู้ออกแบบได้กำหนด ทั้งนี้เพื่อให้มีการศึกษาเปรียบเทียบสมรรถนะของหน่วยประมวลผลในกรณีศึกษาต่างๆกันได้ โดยไม่จำเป็นต้องพัฒนาตัวแบบจำลองขึ้นใหม่ ดังนั้นแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบที่ใช้ในการศึกษาสมรรถนะของหน่วยประมวลผล ในกรณีไม่มีหน่วยความจำแคช และกรณีกำหนดมีหน่วยความจำแคชในระบบ จึงสามารถแสดงได้ดังรูปที่ 4.4 และรูปที่ 4.5



รูปที่ 4.4 แสดงส่วนประกอบการทำงานภายในแบบจำลอง กรณีไม่มีหน่วยความจำแคช



รูปที่ 4.5 แสดงส่วนประกอบการทำงานภายในแบบจำลอง กรณีกำหนดให้มีหน่วยความจำแคช

โครงสร้างของตัวแบบจำลองที่พัฒนาขึ้นนี้ ประกอบด้วยกรรมวิธีหลักคือ กรรมวิธี np ซึ่งต้องทำการสร้างกรรมวิธีย่อยๆให้เกิดขึ้นในระบบ โดยที่กรรมวิธีย่อยหนึ่งๆแทนองค์ประกอบหนึ่งๆภายในหน่วยประมวลผลต้นแบบซึ่งได้แก่ กรรมวิธีหน่วยเชื่อมต่อบัส กรรมวิธีหน่วยปฏิบัติการ และกรรมวิธีหน่วยควบคุมความจำแคช หลังจากนั้นแต่ละกรรมวิธีสามารถดำเนินการไปได้พร้อมกัน โดยที่กรรมวิธีเหล่านี้ได้รับการกำหนดแนวทางให้ดำเนินการตามสถานะที่ได้รับและกำหนดเงื่อนไขที่ต้องประมวลผลเพื่อให้มีการปฏิบัติการคำสั่งแต่ละคำสั่งที่เข้าไปได้อย่างถูกต้อง และในการดำเนินการไปพร้อมกันของแต่ละกรรมวิธี จำเป็นต้องมีกระบวนการควบคุมและจัดการการติดต่อสื่อสารระหว่างกรรมวิธีต่างๆที่ปฏิบัติการแบบขนานกันให้เป็นไปอย่างได้จังหวะกัน ด้วยการกำหนดเหตุการณ์ การรับและส่งข่าวสารให้แก่กัน รวมทั้งการกำหนดเวลาเข้าใช้ทรัพยากรของระบบ ซึ่งได้แก่ คิวคำสั่ง บัส และหน่วยความจำแคช

ในการดำเนินการของแต่ละกรรมวิธีที่เกิดขึ้นภายในตัวแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบนี้ สามารถอธิบายรายละเอียดได้ดังนี้

4.2.1 กรรมวิธี np

เป็นกรรมวิธีสำหรับเริ่มต้นการทำงานของตัวแบบจำลอง และใช้กำหนดข้อกำหนดรายละเอียดและค่าเริ่มต้นของระบบ ซึ่งผู้ออกแบบสามารถกำหนดข้อกำหนดรายละเอียดบางประการได้เช่น ขนาดของคิวคำสั่ง การกำหนดให้มีหน่วยความจำแคชในตัวแบบจำลอง ขนาดหน่วยความจำแคช และขนาดเส้นทางส่งผ่านข้อมูล ซึ่งข้อกำหนดเหล่านี้ทำให้ได้รูปลักษณะระบบสำหรับใช้ศึกษาในกรณีที่แตกต่างกัน โดยนำเอาข้อมูลเหล่านี้ไปเป็นค่าเริ่มต้นในการทำงานของตัวแบบจำลองต่อไป ได้แก่กำหนดทรัพยากรของระบบเช่น บัสข้อมูล คิวคำสั่ง และหน่วยความจำแคช กรรมวิธี np ทำการสร้างกรรมวิธีย่อยในระบบ เช่น กรรมวิธีหน่วยเชื่อมต่อบัส กรรมวิธีหน่วยปฏิบัติการ และถ้ามีการกำหนดให้มีหน่วยความจำแคชในระบบ กรรมวิธี np จะสร้างกรรมวิธีหน่วยควบคุมความจำแคชด้วย หลังจากกระทำการเหล่านี้แล้ว กรรมวิธี np หยุดรอจนกระทั่งเกิดเหตุการณ์ชื่อ end_program ในระบบแล้วจึงดำเนินการต่อไปได้ โดยเหตุการณ์นี้เป็นเหตุการณ์ที่คำสั่งเข้าปฏิบัติการนั้นมีความหมายให้โปรแกรมหยุดทำงาน และเมื่อเสร็จสิ้นการจำลอง จึงพิมพ์รายงานผลลัพธ์ที่ได้ทำการรวบรวมข้อมูลสถิติไว้ในระหว่างการดำเนินการของตัวแบบจำลอง

การทำงานของกรรมวิธี np นี้สามารถแสดงในรูปรหัสเทียม (pseudocode) ได้ดังนี้

1. กำหนดขนาดเส้นทางส่งข้อมูล (หน่วยเป็น บิต) ขนาดคิวคำสั่ง (หน่วยเป็น ไบต์) และจำนวนไบต์ที่ส่งผ่านเส้นทางส่งข้อมูล
2. ระบุความต้องการเปลี่ยนแปลงรูปลักษณะระบบ
3. ถ้าต้องการเปลี่ยนแปลง
   - 3.1 ระบุขนาดคิวคำสั่ง ขนาดเส้นทางส่งข้อมูล จำนวนไบต์ส่งผ่านเส้นทางส่งข้อมูล
   - 3.2 ระบุให้มีหน่วยความจำแคชหรือไม่
   - 3.3 ถ้ากำหนดให้มีหน่วยความจำแคช
     - 3.3.1 ระบุขนาดหน่วยความจำแคช (หน่วยเป็น กิโลไบต์)
     - 3.3.2 กำหนดจำนวนครั้งการเข้าถึงหน่วยความจำหลักต่อ 1 ครั้งของการสังเกตค่าอัตราส่วนการพลาด
     - 3.3.3 กำหนดจำนวนไบต์ในการถ่ายเทข้อมูลระหว่างหน่วยความจำแคชและหน่วยความจำหลัก
     - 3.3.4 กำหนดจำนวนรอบสัญญาณนาฬิกาสำหรับแต่ละรอบการเข้าถึงข้อมูลในหน่วยความจำแคชเท่ากับ 4 รอบสัญญาณนาฬิกา

4. เริ่มต้นกรรมวิธีหลัก mp
5. กำหนดค่าเริ่มต้นให้กับหน่วยประมวลผลต้นแบบ
6. นำโปรแกรมเข้ามาในหน่วยความจำหลัก
7. ถ้ากำหนดให้มีหน่วยความจำแคช
   7.1 สร้างกรรมวิธีหน่วยควบคุมความจำแคช
8. สร้างกรรมวิธีหน่วยปฏิบัติการ
9. สร้างกรรมวิธีหน่วยเชื่อมต่อบัส
10. รอเหตุการณ์ที่คำสั่งมีความหมายให้โปรแกรมหยุดทำงานให้เกิดขึ้นในระบบ
11. พิมพ์รายงานผลลัพธ์จากการจำลอง
12. จบการทำงาน

4.2.2 <u>กรรมวิธีหน่วยเชื่อมต่อบัส</u>

เนื่องจากการเข้าถึงข้อมูลในหน่วยความจำของหน่วยประมวลผลต้นแบบ จะต้องกระทำโดยกรรมวิธีนี้เท่านั้น ซึ่งความต้องการใช้เส้นทางส่งผ่านข้อมูลอาจใช้เพื่อการดึงคำสั่งมาเก็บในคิวคำสั่ง หรืออ่าน/บันทึกข้อมูลให้กับกรรมวิธีหน่วยปฏิบัติการ ดังนั้นจึงต้องมีการติดต่อกันระหว่างกรรมวิธีหน่วยเชื่อมต่อบัสและกรรมวิธีหน่วยปฏิบัติการ โดยถ้าระบบกำหนดสถานะของบัสให้เป็นการดึงคำสั่ง กรรมวิธีหน่วยเชื่อมต่อบัสจะทำหน้าที่ดึงคำสั่งจากโปรแกรมที่อยู่ในหน่วยความจำหลักตามเงื่อนไขการดึงคำสั่งเข้ามาเก็บในคิวคำสั่งดังที่อธิบายในหัวข้อ 4.1.2 โดยที่การดึงคำสั่งในแต่ละรอบการดึงคำสั่ง จะต้องมีการเข้าใช้บัสซึ่งเป็นทรัพยากรของระบบเป็นเวลา 4 รอบสัญญาณนาฬิกา (clock cycles) และปรับจำนวนที่ว่างในคิวคำสั่งให้ลดลง ถ้าพบว่าสถานะเดิมของคิวคำสั่งนั้นไม่มีคำสั่งเลย กรรมวิธีหน่วยเชื่อมต่อบัสจะกำหนดเหตุการณ์ชื่อ q ให้เกิดขึ้นในระบบแต่ถ้าพบว่ามีที่ว่างในคิวคำสั่งไม่เพียงพอ ระบบจะเปลี่ยนแปลงสถานะของบัสให้เป็นว่าง พร้อมกับกรรมวิธีหน่วยเชื่อมต่อบัสจะหยุดชั่วคราวจนกว่าจะเกิดเหตุการณ์ชื่อ room ขึ้นในระบบ ซึ่งเป็นเหตุการณ์ที่หมายถึงในคิวคำสั่งมีที่ว่างมากพอแล้ว กรรมวิธีนี้จึงกระทำการดึงคำสั่งต่อไปได้ แต่ถ้าสถานะของบัสเป็นการอ่านหรือบันทึกข้อมูล กรรมวิธีนี้จะกระทำการตามที่กำหนดพร้อมกับเข้าใช้บัสเป็นระยะเวลาหนึ่ง แล้วจึงส่งข่าวสารกลับไปยังกรรมวิธีหน่วยปฏิบัติการหลังจากที่ได้กระทำการตามต้องการให้แล้ว และสถานะของบัสจะถูกกำหนดให้เป็นการดึงคำสั่งอีกครั้ง

การทำงานภายในกรรมวิธีหน่วยเชื่อมต่อบัสนี้สามารถแสดงในรูปรหัสเทียม ได้ดังนี้

1. เริ่มต้นกรรมวิธีหน่วยเชื่อมต่อบัส
2. while (TRUE)
    2.1 ถ้าสถานะของบัสเป็นสัญญาณหยุด
        2.1.1 จบการทำงาน
    2.2 ถ้าสถานะของบัสเป็นการติงคำสั่ง
        2.2.1 ถ้ามีที่ว่างในคิวคำสั่งไม่พอ (จำนวนที่ว่างน้อยกว่าจำนวนไบต์ที่ส่งผ่านเส้นทางส่งข้อมูล)
            2.2.1.1 ถ้าสถานะของบัสเป็นติงคำสั่ง
                1) กำหนดสถานะของบัสเป็นว่าง
                2) รอให้มีที่ว่างในคิวคำสั่ง
        2.2.2 ถ้ามีที่ว่างในคิวคำสั่งเพียงพอ
            2.2.2.1 เข้าใช้บัส
            2.2.2.2 ถ้ากำหนดให้มีหน่วยความจำแคช
                1) ส่งข่าวสารไปยังกรรมวิธีหน่วยควบคุมความจำแคช
                2) รอรับข่าวสารจากกรรมวิธีหน่วยควบคุมความจำแคช
            2.2.2.3 ปรับค่าเวลาของระบบ
            2.2.2.4 เลิกใช้บัส
            2.2.2.5 ถ้าไม่มีคำสั่งอยู่ในคิวคำสั่ง
                1) กำหนดให้เกิดเหตุการณ์แสดงว่ามีคำสั่งอยู่ในคิวคำสั่งแล้ว
            2.2.2.6 ปรับจำนวนที่ว่างในคิวคำสั่งให้ลดลง
    2.3 ถ้าสถานะของบัสเป็นสัญญาณการอ่าน
        2.3.1 เข้าใช้บัส
        2.3.2 ถ้ากำหนดให้มีหน่วยความจำแคช
            2.3.2.1 ส่งข่าวสารไปยังหน่วยควบคุมความจำแคช
            2.3.2.2 รอรับข่าวสารจากกรรมวิธีหน่วยควบคุมความจำแคช

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

2.3.3 ปรับค่าเวลาของระบบ

2.3.4 เลิกใช้บัส

2.3.5 ส่งข่าวสารไปยังหน่วยปฏิบัติการ

2.3.6 ถ้าไม่มีคิวคำสั่งในระบบ

2.3.6.1 กำหนดให้สถานะของบัสเป็นว่าง

2.3.6.2 รอให้มีที่ว่างในคิวคำสั่ง

2.3.7 ถ้ามีคิวคำสั่งในระบบ

2.3.7.1 กำหนดให้สถานะของบัสเป็นการดึงคำสั่ง

2.4 ถ้าสถานะของบัสเป็นสัญญาณการบันทึก

2.4.1 เข้าใช้บัส

2.4.2 ถ้ากำหนดให้มีหน่วยความจำแคช

2.4.2.1 ส่งข่าวสารไปยังหน่วยควบคุมความจำแคช

2.4.2.2 รอรับข่าวสารจากกรรมวิธีหน่วยควบคุมความจำแคช

2.4.3 ปรับค่าเวลาของระบบ

2.4.4 เลิกใช้บัส

2.4.5 ส่งข่าวสารไปยังหน่วยปฏิบัติการ

2.4.6 ถ้าไม่มีคิวคำสั่งในระบบ

2.4.6.1 กำหนดให้สถานะของบัสเป็นว่าง

2.4.6.2 รอให้มีที่ว่างในคิวคำสั่ง

2.4.7 ถ้ามีคิวคำสั่งในระบบ

2.4.7.1 กำหนดให้สถานะของบัสเป็นการดึงคำสั่ง

2.5 ถ้าสถานะของบัสเป็นว่าง

2.5.1 กำหนดให้สถานะของบัสเป็นการดึงคำสั่ง

3. จบการทำงาน

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

4.2.3 กรรมวิธีหน่วยปฏิบัติการ

การปฏิบัติการตามคำสั่งที่เข้ามาจะเกิดขึ้นภายในกรรมวิธีนี้โดยดึงคำสั่งมาจากคิวคำสั่งพร้อมกับปรับจำนวนที่ว่างในคิวคำสั่งให้เพิ่มขึ้น และถ้าพบว่าสถานะของบัสเป็นว่างอยู่ กรรมวิธีหน่วยปฏิบัติการจะกำหนดให้เกิดเหตุการณ์ชื่อ room ในระบบ แต่ถ้าไม่มีข้อมูลในคิวคำสั่งสำหรับการปฏิบัติการ กรรมวิธีหน่วยปฏิบัติการจำเป็นต้องหยุดรอจนกว่าจะมีเหตุการณ์ที่แสดงว่ามีคำสั่งเข้ามาในคิวคำสั่งแล้ว กรรมวิธีนี้จึงสามารถปฏิบัติการต่อไป ซึ่งในแบบจำลองนี้ได้กำหนดเหตุการณ์ชื่อ q เป็นเหตุการณ์ที่บ่งบอกว่ามีคำสั่งในคิวคำสั่งแล้วและเมื่อกรรมวิธีหน่วยปฏิบัติการต้องการข้อมูลในหน่วยความจำมาใช้ปฏิบัติการในคำสั่ง กรรมวิธีนี้จะทำการเปลี่ยนแปลงสถานะของบัสให้เป็นการอ่านหรือบันทึกข้อมูล แล้วหยุดรอรับข่าวสาร ซึ่งกรรมวิธีนี้จะหยุดรอจนกว่าจะได้รับข่าวสารจากกรรมวิธีหน่วยเชื่อมต่อบัสว่าดำเนินการเสร็จแล้ว จึงเริ่มปฏิบัติการในลำดับต่อไปโดยกรรมวิธีคำสั่งจะเข้าใช้ทรัพยากรระบบซึ่งกำหนดชื่อ EU_fac ด้วยช่วงระยะเวลาหนึ่ง แล้วกรรมวิธีคำสั่งนั้นจึงออกจากระบบไป และถ้าคำสั่งที่กรรมวิธีคำสั่งนั้นได้รับเป็นคำสั่งให้โปรแกรมหยุดปฏิบัติการ กรรมวิธีหน่วยปฏิบัติการต้องติดต่อกับกรรมวิธี np เพื่อหยุดการจำลอง โดยกำหนดเหตุการณ์ชื่อ end_program ให้เกิดขึ้นในระบบ ซึ่งตรงนี้ทำให้ระบบส่งการควบคุมกลับไปยังกรรมวิธี np เพื่อรายงานผลลัพธ์ต่อไป

การทำงานภายในกรรมวิธีนี้สามารถแสดงด้วยรหัสเทียมได้ดังนี้

1. เริ่มต้นกรรมวิธีหน่วยปฏิบัติการ
2. while (TRUE)
   - 2.1 เข้าใช้หน่วยปฏิบัติการ
   - 2.2 ดึงข้อมูลจากคิวคำสั่งด้วยขั้นตอนวิธีดังนี้
     - 2.2.1 ถ้าไม่มีคำสั่งในคิวคำสั่ง
       - 2.2.1.1 ถ้าสถานะของบัสไม่ใช่สัญญาณหยุดและว่าง
         - 1) กำหนดให้เกิดเหตุการณ์ที่แสดงว่ามีที่ว่างเพียงพอในคิวคำสั่ง
       - 2.2.1.2 รอให้เกิดเหตุการณ์ที่แสดงว่ามีคำสั่งในคิวคำสั่งแล้ว
     - 2.2.2 เมื่อมีคำสั่งในคิวคำสั่ง ปรับที่ว่างในคิวคำสั่งให้เพิ่มขึ้น
     - 2.2.3 ถ้ามีที่ว่างในคิวคำสั่งเพียงพอ
       - 2.2.3.1 กำหนดให้เกิดเหตุการณ์ที่แสดงว่ามีที่ว่างเพียงพอในคิวคำสั่ง

2.3 ถ้าต้องการอ่านข้อมูล

2.3.1 กำหนดสถานะของบัสเป็นสัญญาณการอ่าน

2.3.2 รอรับข่าวสารจากหน่วยเชื่อมต่อบัส

2.4 ถ้าต้องการบันทึกข้อมูล

2.4.1 กำหนดสถานะของบัสเป็นสัญญาณการบันทึก

2.4.2 รอรับข่าวสารจากหน่วยเชื่อมต่อบัส

2.5 ปรับค่าเวลาของระบบ

2.6 เลิกใช้หน่วยปฏิบัติการด้วยคำสั่ง detach()

2.7 ถ้าเป็นคำสั่งให้หยุดปฏิบัติการโปรแกรม

2.7.1 กำหนดสถานะของบัสเป็นสัญญาณหยุด

2.7.2 กำหนดให้เกิดเหตุการณ์ที่แสดงว่าคำสั่งที่เข้ามามีความหมายให้โปรแกรมหยุดการทำงาน

2.8 จบการทำงาน

4.2.4 <u>กรรมวิธีหน่วยควบคุมความจำแคช</u>

กรรมวิธีนี้เกิดขึ้นในระบบก็ต่อเมื่อผู้ออกแบบระบุให้มีหน่วยความจำแคชในระบบ และเริ่มต้นการทำงานก็ต่อเมื่อได้รับข่าวสารจากกรรมวิธีหน่วยเชื่อมต่อบัส โดยที่เมื่อกรรมวิธีหน่วยเชื่อมต่อบัสต้องการข้อมูลก็จะทำการส่งข่าวสารไปยังกรรมวิธีหน่วยควบคุมความจำแคช ซึ่งในระหว่างนี้ กรรมวิธีหน่วยเชื่อมต่อบัสจะต้องหยุดรอจนกว่ากรรมวิธีหน่วยควบคุมความจำแคชส่งข่าวสารให้ทราบว่าได้ส่งข้อมูลมาแล้ว จึงปฏิบัติการกับข้อมูลนั้นตามต้องการได้ต่อไป สำหรับการทำงานของกรรมวิธีหน่วยควบคุมความจำแคชจะเกิดขึ้นเมื่อได้รับข่าวสารซึ่งเป็นการขอข้อมูลจากกรรมวิธีหน่วยเชื่อมต่อบัส โดยเริ่มค้นหาข้อมูลในหน่วยความจำแคชก่อน ถ้าไม่พบข้อมูลที่ต้องการ จึงให้มีการถ่ายเทข้อมูลที่ต้องการนั้นจากหน่วยความจำหลักมายังหน่วยความจำแคช และเมื่อได้ข้อมูลที่ต้องการแล้วจึงส่งข่าวสารไปให้กรรมวิธีหน่วยเชื่อมต่อบัสรับทราบ หลังจากนั้นกรรมวิธีหน่วยควบคุมความจำแคชก็หยุดรอจนกว่าจะได้รับการร้องขอข้อมูลอีก แต่ถ้าเป็นการบันทึกข้อมูล จำเป็นต้องมีการเข้าถึงหน่วยความจำแคชทุกครั้ง เนื่องจากกำหนดใช้วิธีการปรับปรุงข้อมูลแบบ write through โดยศึกษารายละเอียดเรื่องนี้ได้จาก (Hwang and Briggs,1988) (Smith,1982) และ (Stallings,1987) หรือดูคำอธิบายได้ในบทที่ 7

การทำงานภายในกรรมวิธีหน่วยควบคุมความจำแคชนี้สามารถแสดงด้วยรหัสเทียมได้ดังนี้

1. เริ่มต้นกรรมวิธีหน่วยควบคุมความจำแคช
2. while (TRUE)
    2.1 รอรับข่าวสารจากกรรมวิธีหน่วยเชื่อมต่อบัส
    2.2 ถ้าค้นหาแล้วพบข้อมูลในหน่วยความจำแคชในทันที
        2.2.1 ปรับปรุงค่าสถิติการพบ
    2.3 ถ้าไม่พบข้อมูลที่ต้องการในหน่วยความจำแคช
        2.3.2 ปรับปรุงค่าสถิติการพลาด
    2.4 ถ้าไม่พบข้อมูลที่ต้องการในหน่วยความจำแคชหรือเป็นการบันทึกข้อมูล
        2.4.1 เข้าใช้หน่วยความจำแคช
        2.4.2 ปรับค่าเวลาของระบบ
        2.4.3 เลิกใช้หน่วยความจำแคช
    2.5 ส่งข่าวสารไปยังกรรมวิธีหน่วยเชื่อมต่อบัส
3. จบการทำงาน

4.2.5 <u>รายงานผลลัพธ์</u>

หลังจากที่เกิดเหตุการณ์สิ้นสุดการจำลอง กรรมวิธี np จึงเริ่มพิมพ์รายงานผลลัพธ์ ซึ่งเป็นส่วนสำคัญที่ผู้ออกแบบต้องการสำหรับแสดงให้เห็นถึงสมรรถนะในการทำงานของหน่วยประมวลผลต้นแบบและนำไปใช้เป็นข้อมูลในการศึกษาวิเคราะห์พฤติกรรมที่สนใจ โดยที่ค่าข้อมูลสถิติเหล่านี้ได้จากการรวบรวมระหว่างการดำเนินการกับตัวแบบจำลองซึ่งเป็นผลจากการทำงานของตัวจำลองอิงกรรมวิธี np และบางส่วนเกิดขึ้นตามความต้องการใช้งานเอง

นอกจากนี้ในการทำงานของตัวแบบจำลอง จะต้องอาศัยข้อมูลที่จำเป็นในการปฏิบัติการแต่ละคำสั่งเช่น เวลา ชนิดคำสั่ง เพื่อให้ระบบทำงานได้อย่างถูกต้อง ดังแสดงไว้ในภาคผนวก ค โดยใช้ชุดคำสั่งและรายละเอียดเรื่องเวลานี้จากการศึกษาตามเอกสาร (Rector and Alexy, 1980) และข้อมูลซึ่งเป็นกระแสคำสั่งสำหรับเข้าไปดำเนินการใช้ทรัพยากรภายในหน่วยประมวลผลต้นแบบจะเป็นโปรแกรมภาษาเครื่อง ซึ่งได้จากการทำงานของตัวแปลภาษา Turbo C โดยใช้โปรแกรมสำหรับการวัดเปรียบเทียบสมรรถนะที่พัฒนาด้วยภาษา C เป็นแฟ้มข้อมูลเข้า

4.3 โปรแกรมสำหรับการวัดเปรียบเทียบสมรรถนะ (Benchmark Program)

ในการทดสอบการทำงานของหน่วยประมวลผลต้นแบบจำเป็นต้องอาศัยตัวอย่างโปรแกรมซึ่งเข้ากันได้กับข้อกำหนดรายละเอียดทางด้านสถาปัตยกรรมของหน่วยประมวลผลต้นแบบ และเป็นที่ยอมรับโดยทั่วไป สำหรับงานวิจัยนี้ได้นำเอาโปรแกรมดังกล่าวมาใช้ 5 โปรแกรมในการทดสอบและศึกษาการทำงานของหน่วยประมวลผลต้นแบบ โปรแกรมเหล่านี้มักได้รับการอ้างอิงและนำไปใช้ในงานวิจัยต่างๆ ดังนั้นผู้สนใจสามารถค้นหาได้จาก (Wiecker,1984) และ (Watcharawittayakul,1988) โปรแกรมทั้ง 5 นี้ได้แก่

1) Dhrystone

เป็นโปรแกรมที่มีผู้พัฒนาได้ทำการสังเคราะห์ขึ้นโดยให้มีการใช้งานคำสั่งประเภทต่างๆ หรือให้มีการใช้รูปแบบคำสั่งกระจายไปโดยไม่เน้นไปเฉพาะด้านใดด้านหนึ่ง

2) Fibonacci

เป็นโปรแกรมสำหรับการคำนวณตัวเลขในลำดับไฟโบแนคคี โดยที่ภายในโปรแกรมมีลักษณะการทำงานแบบเวียนเกิด (recursive) คือเรียกตัวเองให้ทำงานซ้ำๆ แต่มีเงื่อนไขในการหยุดทำงานเพื่อคำนวณหาผลลัพธ์ในแต่ละรอบการทำงาน โดยที่การทำงานรอบนั้นต้องอาศัยผลลัพธ์จากการทำงานของรอบที่ผ่านมา

3) Netflow

โปรแกรมนี้เป็นตัวอย่างปัญหาทางด้านการวิจัยดำเนินงาน ซึ่งใช้ในการหาเส้นทางในเครือข่ายที่ให้ผลสูงสุด การหาคำตอบสามารถแทนรูปแบบปัญหาได้เป็นกราฟแบบมีทิศทาง (directed graph) โดยที่การแทนข้อมูลกราฟในเครื่องคอมพิวเตอร์นี้ ใช้โครงสร้างข้อมูลแบบแถวลำดับ 2 มิติหรือที่เรียกว่าเมตริกซ์ (matrix)

4) Quicksort

เป็นโปรแกรมที่ใช้ขั้นตอนวิธีการเรียงลำดับข้อมูลวิธีการหนึ่ง ซึ่งมีลักษณะการทำงานในโปรแกรมเป็นแบบเวียนเกิด และต้องเกี่ยวข้องกับข้อมูลตัวเลข 1,000 จำนวน

5) Spanfo

เป็นโปรแกรมสำหรับหาจำนวนองค์ประกอบ (component) ภายในกราฟ ซึ่งมีเส้นทางเชื่อมโหนด (node) ได้ถึงกัน ในลักษณะของทรี (tree) หรือกลุ่มทรี โดยกำหนดแทนข้อมูลกราฟด้วยโครงสร้างข้อมูลแบบแถวลำดับ 2 มิติ

เมื่อพิจารณาลักษณะของโปรแกรมเหล่านี้ สามารถแบ่งเป็น 3 กลุ่ม กลุ่มแรก ได้แก่ โปรแกรม Fibonacci และ Quicksort ซึ่งมีจุดเด่นในเรื่องของการทำงานแบบเวียนเกิด กลุ่มที่ 2 ได้แก่ โปรแกรม Netflow และ Spanfo ซึ่งเน้นการทำงานกับโครงสร้างข้อมูลแบบแถวลำดับ (array) และเป็นตัวอย่างที่ดีสำหรับขั้นตอนวิธีเครือข่าย (network algorithm) สำหรับกลุ่มสุดท้าย คือ โปรแกรม Dhrystone เป็นโปรแกรมที่สังเคราะห์ขึ้นเพื่อให้ศึกษาการทำงานของชุดคำสั่งได้ครบทุกรูปแบบ

โปรแกรมทั้ง 5 โปรแกรมนี้ พัฒนาขึ้นด้วยภาษา C และใช้ตัวแปลภาษา Turbo C เวอร์ชัน 1.0 ในการสร้างแฟ้มข้อมูลสำหรับปฏิบัติการ (executable file) ซึ่งแฟ้มข้อมูลเหล่านี้จะนำมาใช้เป็นกระแสข้อมูลคำสั่งในการเข้าดำเนินการบนตัวแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบ และเนื่องจากตัวแบบที่พัฒนานี้ไม่ได้ทำการจำลองหน่วยนำข้อมูลเข้า/ออก ดังนั้นในการพัฒนาโปรแกรมตัวอย่างจึงต้องไม่มีประโยคคำสั่งใดๆที่เกี่ยวข้องกับหน่วยนำข้อมูลเข้า/ออก

4.4 ผลลัพธ์จากการดำเนินการของตัวแบบจำลอง

ในการดำเนินการกับตัวแบบจำลองเพื่อให้ได้ผลลัพธ์สำหรับใช้ในการศึกษาและวิเคราะห์นี้ ผู้วิจัยได้ทำการจำลองโดยอาศัยโปรแกรมสำหรับการวัดเปรียบเทียบสมรรถนะทั้ง 5 โปรแกรม แล้วรวบรวมผลลัพธ์จากรายงานที่ได้ออกมาเป็นข้อมูลสำหรับใช้ในการศึกษาวิเคราะห์สมรรถนะของหน่วยประมวลผลต้นแบบ ซึ่งข้อมูลเหล่านี้ได้แก่ ข้อมูลการใช้งานชุดคำสั่งโดยแบ่งตามจำนวนตัวถูกกระทำการ กลุ่มคำสั่ง และวิธีการอ้างตำแหน่งที่อยู่ ข้อมูลผลลัพธ์นี้นำไปใช้ในการวิเคราะห์กระแสข้อมูลคำสั่ง ซึ่งจะได้กล่าวในบทที่ 5 และในการศึกษาผลกระทบและวัดเปรียบเทียบในกรณีศึกษาที่แตกต่างกัน พิจารณาจากเวลาที่ใช้ในการดำเนินการกับตัวแบบจำลองเป็นข้อมูลหลัก ดังจะได้กล่าวโดยละเอียดในบทที่ 6 และ 7 ต่อไป