บทที่ 5

การวิเคราะห์คำสั่งจากผลการทดสอบ

ในการศึกษาสมรรถนะของหน่วยประมวลผลโดยอาศัยตัวแบบจำลองการทำงาน ทำให้ได้ข้อมูลที่แสดงถึงปริมาณกลุ่มคำสั่งในโปรแกรมที่เข้าประมวลผล ซึ่งจะช่วยให้ผู้ออกแบบเข้าใจถึงคุณลักษณะของภาระงานที่หน่วยประมวลผลต้องกระทำการ และข้อมูลผลลัพธ์จากการจำลองการทำงานของหน่วยประมวลผลต้นแบบที่นำมาวิเคราะห์ คือปริมาณการใช้งานชุดคำสั่งในแง่ต่างๆ โดยนำเอาข้อมูลเหล่านี้มาวิเคราะห์กระแสข้อมูลคำสั่งเพื่อให้เป็นข้อมูลที่ผู้ออกแบบสามารถนำไปใช้ศึกษาและพิจารณาปรับปรุงการทำงานของระบบ หรือปรับเปลี่ยนการจัดสถาปัตยกรรมภายในระบบ เพื่อให้ได้ประสิทธิผลที่ดีขึ้น เช่น ถ้าคำสั่งใดไม่ได้ใช้งานเลย ควรพิจารณายกเลิก และให้พิจารณาปรับปรุงฮาร์ดแวร์ให้สามารถประมวลผลคำสั่งได้เร็วขึ้น โดยเฉพาะกลุ่มคำสั่งที่มีการใช้งานบ่อยครั้ง หรือโดยเฉพาะกับวิธีการอ้างตำแหน่งที่อยู่ที่มีอัตราส่วนการใช้งานมาก

การศึกษาวิเคราะห์ผลลัพธ์ที่ได้จากการนำโปรแกรมทดสอบสมรรถนะทั้ง 5 โปรแกรม เข้าปฏิบัติการภายในแบบจำลองนี้ ได้แยกพิจารณาวิเคราะห์ชุดคำสั่งจากการดำเนินการของตัวแบบจำลองการทำงานของหน่วยประมวลผลต้นแบบ เป็น 3 กรณีดังนี้ กรณีแรก จำแนกด้วยจำนวนตัวถูกกระทำการ กรณีที่สอง พิจารณาตามประเภทคำสั่ง และกรณีสุดท้าย เป็นการวิเคราะห์ชุดคำสั่งตามวิธีการอ้างตำแหน่ง

5.1 การวิเคราะห์ชุดคำสั่งตามจำนวนตัวถูกกระทำการ

ในการพิจารณารูปแบบคำสั่งของหน่วยประมวลผลต้นแบบที่กำหนดใช้ในงานวิจัยนี้สามารถแบ่งกลุ่มออกได้ด้วยจำนวนตัวถูกกระทำการที่ระบุใช้ในแต่ละคำสั่ง ซึ่งหน่วยประมวลผลต้นแบบนี้มีรูปแบบคำสั่ง 3 รูปแบบ คือรูปแบบคำสั่งที่ไม่มีตัวถูกกระทำการ รูปแบบคำสั่งที่มีตัวถูกกระทำการ 1 ตัวและรูปแบบคำสั่งที่มีตัวถูกกระทำการ 2 ตัว และจากการศึกษารูปแบบคำสั่ง พบว่า 1 คำสั่งมีความยาวไม่เกิน 6 ไบต์ โดยที่คำสั่งที่ไม่มีตัวถูกกระทำการ ใช้จำนวนไบต์เพียง 1- 2 ไบต์ ในขณะที่คำสั่งที่มีตัวถูกกระทำการ ใช้จำนวนไบต์ตั้งแต่ 1-6 ไบต์ และคำสั่งที่ไม่มีตัวถูกกระทำการ

ใช้เวลาประมวลผลคำสั่ง น้อยกว่าคำสั่งที่มีตัวถูกกระทำการ เนื่องจากไม่ต้องเสียเวลาในการคำนวณตำแหน่งที่อยู่ของข้อมูลที่ต้องการให้กับตัวถูกกระทำการ และในการเข้าถึงหน่วยความจำ

โปรแกรมที่ใช้ในการทดสอบ	จำนวนตัวถูกกระทำการ		
	0	1	2
Dhrystone	27.01	33.47	39.52
Fibonacci	5.75	58.15	36.10
Netflow	0.55	29.88	69.57
Quicksort	0.29	35.00	64.71
Spanfo	6.64	29.32	64.04

ตารางที่ 5.1 แสดงอัตราส่วน (ร้อยละ) คำสั่งของแต่ละโปรแกรม โดยแยกตามจำนวนตัวถูกกระทำการที่ระบุไว้ในคำสั่ง

จากตารางข้างบนนี้ อาจวิเคราะห์ได้ว่า การใช้งานรูปแบบคำสั่งต่างๆในแต่ละโปรแกรม ยกเว้น Fibonacci มีลักษณะการกระจายการใช้งานรูปแบบคำสั่งที่คล้ายคลึงกัน คือ มีการใช้งานรูปแบบคำสั่งที่มีตัวถูกกระทำการ 2 ตัวด้วยอัตราส่วนสูงสุด ใช้รูปแบบคำสั่งที่มีตัวถูกกระทำการ 1 ตัวด้วยอัตราส่วนที่รองลงมาและใช้งานรูปแบบคำสั่งที่ไม่มีตัวถูกกระทำการด้วยอัตราส่วนน้อยสุด โดยที่พบว่ากลุ่มโปรแกรมที่ใช้โครงสร้างข้อมูลแบบแถวลำดับ ซึ่งได้แก่ โปรแกรม Netflow Quicksort และ Spanfo ใช้รูปแบบคำสั่งที่มีตัวถูกกระทำการ 2 ตัวเป็นส่วนใหญ่ และกลุ่มโปรแกรมที่มีลักษณะการทำงานแบบการเวียนเกิดภายในโปรแกรม ซึ่งได้แก่ โปรแกรม Fibonacci และ Quicksort มีอัตราส่วนการใช้งานรูปแบบคำสั่งที่มีตัวถูกกระทำการ 1 ตัวสูงกว่าโปรแกรมอื่นๆ

เมื่อพิจารณาแต่ละโปรแกรม พบว่า Dhrystone มีการกระจายการใช้งานแต่ละรูปแบบคำสั่งด้วยอัตราส่วนที่แตกต่างกันไม่มาก เป็นเพราะผู้ที่พัฒนาโปรแกรมมีจุดประสงค์ที่จะให้มีการใช้งานรูปแบบคำสั่งเกิดขึ้นโดยไม่เน้นไปทางด้านใดโดยเฉพาะ (Weicker,1984) สำหรับ Fibonacci นั้นมีอัตราส่วนการใช้งานรูปแบบคำสั่งที่ไม่มีตัวถูกกระทำการ คิดเป็นร้อยละ 5.75 และให้อัตราส่วนการใช้งานรูปแบบคำสั่งที่มีตัวถูกกระทำการ 1 ตัวเป็น 1.6 เท่าของการใช้งานรูปแบบคำสั่งที่มีตัวถูกกระทำการ 2 ตัว เนื่องจากลักษณะโปรแกรมที่มีการทำงานแบบเวียนเกิดทำให้มีการเรียกตัวเองทำงานซ้ำๆ ระบบจึงต้องจัดการเก็บสถานะต่างๆที่จำเป็นสำหรับการทำงานของหน่วยประมวลผลเมื่อกลับจากการทำงานในแต่ละรอบได้อย่างถูกต้อง และการจัดเก็บสถานะเหล่านั้นได้ถูกกำหนดให้เก็บลงบนสแตกซึ่งเป็นเนื้อที่หน่วยความจำส่วนหนึ่ง ดังนั้นการทำงานแบบเวียนเกิดจึงเป็นผลให้มีการเคลื่อนย้ายข้อมูลบนสแตกบ่อยครั้ง และคำสั่งที่กระทำการบนสแตกโดยตรง เช่นคำสั่ง push หรือ pop จะระบุตัวถูกกระทำการในคำสั่งเพียงตัวเดียวโดยถือว่าตัวถูกกระทำการอีกตัวหนึ่งนั้นคือ ตำแหน่งบนสุดของสแตก จึงไม่จำเป็นต้องระบุไว้ในคำสั่งอีก สำหรับในกลุ่มโปรแกรม Netflow Quicksort และ Spanfo จะมุ่งเน้นรูปแบบคำสั่งที่มีตัวถูกกระทำการ 2 ตัวมากกว่ารูปแบบคำสั่งที่มีตัวถูกกระทำการ 1 ตัวโดยเฉลี่ยประมาณ 2 เท่า ทั้งนี้น่าจะเป็นผลจากลักษณะของโปรแกรมที่มีการใช้โครงสร้างข้อมูลแบบแถวลำดับ ซึ่งเมื่อต้องกระทำการใดๆกับข้อมูลเหล่านี้ที่เรียงลำดับต่อเนื่องกันในหน่วยความจำ จำเป็นต้องระบุตำแหน่งของข้อมูลที่เป็นตัวถูกกระทำการไว้ในคำสั่งให้ชัดเจน และยังพบว่า Quicksort มีการใช้งานรูปแบบคำสั่งที่มีตัวถูกกระทำการ 1 ตัวมากกว่า Netflow และ Spanfo ซึ่งน่าจะเป็นเพราะภายในโปรแกรมมีลักษณะการทำงานแบบเวียนเกิดด้วย

5.2 การวิเคราะห์ชุดคำสั่งตามประเภทคำสั่ง

ประเภทชุดคำสั่งที่กำหนดใช้ในการประมวลผลคำสั่งภายในระบบ มีความแตกต่างกันไปแล้วแต่ผู้ออกแบบ แต่โดยทั่วไปจำเป็นต้องมีชุดคำสั่งที่เป็นคำสั่งพื้นฐานในการทำงานกลุ่มหนึ่ง ได้แก่ คำสั่งเคลื่อนย้ายข้อมูล คำสั่งคำนวณทางคณิตศาสตร์ และคำสั่งย้ายการควบคุม ดังที่ได้กล่าวถึงชุดคำสั่งสำหรับปฏิบัติการของหน่วยประมวลผลต้นแบบในหัวข้อ 4.1.4 ซึ่งมีทั้งหมด 8 กลุ่ม แต่ในการวิเคราะห์กลุ่มคำสั่งที่ใช้งานบนหน่วยประมวลผลต้นแบบนี้ พิจารณาเพียง 7 กลุ่ม โดยที่ไม่พิจารณากลุ่มคำสั่งนำข้อมูลเข้า/ออก เนื่องจากตัวแบบจำลองที่พัฒนาขึ้นนี้ จำลองเฉพาะในส่วนการทำงานระหว่างหน่วยประมวลผลต้นแบบและหน่วยความจำ

ในการศึกษาการใช้งานชุดคำสั่งโดยแยกตามประเภทคำสั่งนี้ ได้แสดงข้อมูลการใช้งานชุดคำสั่งด้วยค่าอัตราส่วนร้อยละ โดยที่ข้อมูลในตารางที่ 5.2 แสดงให้เห็นอัตราส่วนการใช้งานแต่ละประเภทคำสั่งภายในแต่ละโปรแกรม และตารางที่ 5.3 แสดงอัตราส่วนการใช้งานคำสั่งแต่ละประเภทโดยเฉลี่ยจากทุกโปรแกรม เราอาจสรุปได้ว่ากลุ่มโปรแกรมที่ใช้โครงสร้างข้อมูลแบบแถวลำดับจำเป็นต้องมีการคำนวณหาตำแหน่งข้อมูลที่ต้องการภายในโครงสร้าง ซึ่งต้องใช้คำสั่งคำนวณทางคณิตศาสตร์เพื่อให้สามารถกระทำการกับข้อมูลในตำแหน่งที่ต้องการได้ จึงทำให้มีอัตราส่วนการใช้งานคำสั่งประเภทนี้สูง และกลุ่มโปรแกรมที่มีการทำงานแบบการเวียนเกิดมีการใช้งานประเภทคำสั่งย้ายการควบคุมมากกว่าโปรแกรมอื่นๆด้วย

ชนิดคำสั่ง โปรแกรม	คำสั่งเคลื่อนย้ายข้อมูล	คำสั่งคำนวณคณิตศาสตร์	คำสั่งปฏิบัติการกับบิต	คำสั่งปฏิบัติการตัวอักษร	คำสั่งย้ายการควบคุม	คำสั่งนำหน้า	คำสั่งควบคุมตัวประมวลผล
Dhrystone	45.25	15.86	2.78	22.05	12.43	1.04	0.59
Fibonacci	55.31	12.94	6.15	0.50	24.92	0.13	0.05
Netflow	44.41	34.52	10.55	0.09	10.23	0.15	0.05
Quicksort	35.38	27.09	10.05	0.03	27.44	0.01	0.003
Spanfo	41.48	26.61	14.55	4.62	12.71	0.02	0.001

ตารางที่ 5.2 แสดงอัตราส่วน (ร้อยละ) การใช้งานชุดคำสั่งของแต่ละโปรแกรม โดยแยกตามประเภทคำสั่ง

ประเภทชุดคำสั่ง	อัตราส่วนโดยเฉลี่ย (ร้อยละ)
เคลื่อนย้ายข้อมูล	44.36 ± 7.24
คำนวณทางคณิตศาสตร์	23.40 ± 8.86
ย้ายการควบคุม	17.55 ± 7.99
ปฏิบัติการกับบิต	8.82 ± 4.50
ปฏิบัติการกับตัวอักษร	5.46 ± 9.47
นำหน้า	0.27 ± 0.44
ควบคุมตัวประมวลผล	0.14 ± 0.25

ตารางที่ 5.3 แสดงอัตราส่วนโดยเฉลี่ยในการใช้งานประเภทคำสั่งโดยเรียงลำดับตามตัวเลขจากมากไปน้อย

จากข้อมูลในตารางทั้ง 2 นี้เราอาจกล่าวโดยรวมได้ว่า ทุกโปรแกรมใช้งานประเภทคำสั่งเคลื่อนย้ายข้อมูลซึ่งเป็นคำสั่งพื้นฐานในการทำงานของหน่วยประมวลผล ด้วยอัตราส่วนที่สูงกว่าประเภทคำสั่งอื่นๆ คือมีอัตราส่วนการใช้งานโดยเฉลี่ยร้อยละ 44.36 อัตราส่วนที่สูงเป็นอันดับสองคือ อัตราส่วนการใช้งานประเภทคำสั่งคำนวณทางคณิตศาสตร์ มีค่าโดยเฉลี่ยร้อยละ 23.40 ซึ่งนับว่ากลุ่มโปรแกรม Netflow Quicksort และ Spanfo ให้อัตราส่วนที่สูงกว่าอีก 2 โปรแกรม (ดูตารางที่ 5.2) อาจเป็นเพราะโครงสร้างข้อมูลที่ใช้เป็นแบบแถวลำดับ และเมื่ออ้างถึงข้อมูลในโปรแกรม จึงจำเป็นต้องคำนวณลำดับหรือตำแหน่งข้อมูลเป็นพิเศษ สำหรับคำสั่งย้ายการควบคุม เป็นกลุ่มคำสั่งที่ใช้งานในอัตราส่วนโดยเฉลี่ยสูงเป็นอันดับสาม คือร้อยละ 17.55 โดยที่โปรแกรม Fibonacci และ Quicksort ให้อัตราส่วนการใช้งานเฉลี่ยสูงกว่าอีก 3 โปรแกรม ประมาณ 2 เท่า (ดูตารางที่ 5.2) ทั้งนี้อาจเนื่องมาจากภายในโปรแกรมทั้ง

2 นี้มีจุดเด่นตรงที่ใช้หลักการทำงานแบบการเวียนเกิดที่เป็นผลให้มีการย้ายตำแหน่งในการปฏิบัติการโปรแกรมบ่อยครั้ง

สำหรับการนิจารณาการใช้งานกลุ่มคำสั่งที่เหลือคือ กลุ่มคำสั่งปฏิบัติการกับบิต มีอัตราส่วนการใช้งานโดยเฉลี่ยร้อยละ 8.82 กลุ่มคำสั่งปฏิบัติการกับตัวอักษร มีอัตราส่วนการใช้งานโดยเฉลี่ยร้อยละ 5.46 โดยที่มีข้อสังเกตว่าโปรแกรม Dhrystone ใช้งานประเภทคำสั่งปฏิบัติการตัวอักษรด้วยอัตราส่วนที่สูงกว่าโปรแกรมอื่นๆ โดยเฉลี่ยประมาณร้อยละ 16.6 เนื่องจากข้อมูลที่ใช้ปฏิบัติการในโปรแกรมโดยส่วนมากจะเป็นชนิดตัวอักษร ในขณะที่โปรแกรมอื่นๆใช้ข้อมูลชนิดตัวเลขเป็นส่วนมาก และสุดท้ายคือกลุ่มคำสั่งนำหน้า และกลุ่มคำสั่งควบคุมหน่วยประมวลผล มีอัตราส่วนการใช้งานโดยเฉลี่ยน้อยมาก คือร้อยละ 0.27 และ 0.14 ตามลำดับ ทั้งนี้อาจเนื่องมาจากการที่กลุ่มคำสั่งนำหน้าใช้งานได้เฉพาะกับชุดคำสั่งที่ปฏิบัติการกับตัวอักษรเท่านั้น ซึ่งจากข้อมูลในตารางที่ 5.3 ได้แสดงให้เห็นว่าการใช้งานกลุ่มคำสั่งปฏิบัติการกับตัวอักษรนั้น มีอัตราส่วนน้อยอยู่แล้ว จึงทำให้มีอัตราส่วนการใช้งานคำสั่งประเภทนี้น้อยไปด้วย และสำหรับกลุ่มคำสั่งควบคุมหน่วยประมวลผล ซึ่งเป็นชุดคำสั่งที่ตอบสนองการทำงานเฉพาะบางอย่างโดยเฉพาะการรับ/ส่งข้อมูลผ่านทางอุปกรณ์นำข้อมูลเข้า/ออก แต่แบบจำลองที่พัฒนาขึ้นนี้ไม่ได้จำลองการทำงานของหน่วยประมวลผลต้นแบบให้ครอบคลุมไปถึงส่วนนี้ ดังนั้นคำสั่งในกลุ่มนี้จึงมีอัตราส่วนการใช้งานน้อยเช่นเดียวกัน

5.3 การวิเคราะห์ชุดคำสั่งตามวิธีการอ้างตำแหน่ง

จากการศึกษารูปแบบคำสั่งที่กำหนดใช้ในหน่วยประมวลผลต้นแบบ และวิธีการอ้างตำแหน่งที่หน่วยประมวลผลต้นแบบใช้ในการเข้าถึงตัวถูกกระทำการโดยระบุไว้ในคำสั่ง ดังที่ได้อธิบายในหัวข้อ 4.1.3 เราสามารถจัดกลุ่มวิธีการอ้างตำแหน่งที่อยู่ได้เป็น 3 กลุ่มคือ กลุ่มที่ไม่มีการเข้าถึงหน่วยความจำ ได้แก่แบบรีจิสเตอร์ และแบบทันที กลุ่มที่เข้าถึงหน่วยความจำครั้งเดียว ได้แก่แบบโดยตรง และแบบรีจิสเตอร์โดยอ้อม และกลุ่มที่เข้าถึงหน่วยความจำมากกว่า 1 ครั้ง ได้แก่แบบฐานสัมพัทธ์ แบบดัชนีโดยตรง และแบบดัชนีฐาน

วิธีการอ้างตำแหน่งที่อยู่ / โปรแกรม	จำนวนครั้งที่เข้าถึงหน่วยความจำ						
	0		1		มากกว่า 1		
	รีจิสเตอร์	ทันที	โดยตรง	รีจิสเตอร์โดยอ้อม	ฐานสัมพัทธ์	ดัชนีโดยตรง	ดัชนีฐาน
Dhrystone	35.90	10.40	9.74	4.26	11.12	0.72	0.86
Fibonacci	68.35	10.37	10.02	0.37	5.13	0.0	0.01
Netflow	56.54	10.65	22.83	0.42	9.02	0.0	0.01
Quicksort	44.43	10.70	25.59	0.02	18.97	0.0	0.0
Spanfo	42.91	3.68	27.59	5.01	14.16	0.0	0.01

ตารางที่ 5.4 แสดงอัตราส่วน (ร้อยละ) คำสั่งของแต่ละโปรแกรมโดยแยกตามวิธีการอ้างตำแหน่ง

เมื่อนำข้อมูลที่ได้มาแสดงอัตราส่วนการใช้งานคำสั่งในแต่ละโปรแกรมโดยแยกตามวิธีการอ้างตำแหน่งและจำนวนครั้งในการเข้าถึงหน่วยความจำหลัก ดังตารางที่ 5.4 จะพบว่ากลุ่มที่ไม่มีการเข้าถึงหน่วยความจำหลักมีการใช้งานด้วยอัตราส่วนที่สูงกว่ากลุ่มที่ต้องเข้าถึงหน่วยความจำหลักประมาณ 1.8 เท่า และพบว่าทุกโปรแกรมใช้วิธีการอ้างตำแหน่งแบบรีจิสเตอร์สูงกว่าวิธีการอื่นๆ อาจเป็นการแสดงให้เห็นว่าการทำงานภายในหน่วยประมวลผลจะใช้วิธีการอ้างตำแหน่งข้อมูลที่มีรูปแบบการอ้างในคำสั่งได้ง่ายและเป็นวิธีการที่สามารถกระทำการได้เร็ว

จำนวนครั้งการเข้าถึงหน่วยความจำ	วิธีการอ้างตำแหน่งที่อยู่	อัตราส่วนโดยเฉลี่ย (ร้อยละ)
0	แบบรีจิสเตอร์	49.63
1	แบบโดยตรง	19.15
มากกว่า 1	แบบฐานสัมพัทธ์	11.68
0	แบบทันที	9.16
1	แบบรีจิสเตอร์โดยอ้อม	2.02
มากกว่า 1	แบบดัชนีฐาน	0.18
มากกว่า 1	แบบดัชนีโดยตรง	0.14

ตารางที่ 5.5 แสดงอัตราส่วนโดยเฉลี่ยในการใช้งานวิธีการอ้างตำแหน่งเรียงลำดับตามตัวเลขจากมากไปน้อย

จากข้อมูลในตารางที่ 5.5 แสดงให้เห็นว่า มีการใช้วิธีการอ้างตำแหน่งที่อยู่แบบรีจิสเตอร์ด้วยอัตราส่วนการใช้งานสูงสุด โดยเฉลี่ยร้อยละ 49.63 นั่นแสดงว่า หน่วยประมวลผลเรียกใช้รีจิสเตอร์ในการทำงานบ่อยครั้ง รองลงมาคืออัตราส่วนการใช้งานวิธีการอ้างตำแหน่งแบบโดยตรง ซึ่งเป็นวิธีการอ้างตำแหน่งข้อมูลในหน่วยความจำที่มีรูปแบบการกำหนดที่ง่าย มีค่าโดยเฉลี่ยร้อยละ 19.15 โดยที่พบว่ากลุ่มโปรแกรมซึ่งได้แก่ Netflow และ Spanfo จะใช้วิธีการอ้างตำแหน่งแบบนี้ด้วยอัตราส่วนที่มากกว่าการใช้งานในโปรแกรม Dhrystone ประมาณ 2.6 เท่า (ดูตารางที่ 5.4) อาจมีสาเหตุมาจากการใช้โครงสร้างข้อมูลแบบแถวลำดับทั้ง 1 มิติและ 2 มิติ ภายในโปรแกรม Netflow และ Spanfo และเช่นเดียวกันกับโปรแกรม Quicksort ซึ่งใช้วิธีการอ้างตำแหน่งแบบนี้ด้วยอัตราส่วนที่มากกว่าการใช้งานใน

โปรแกรม Fibonacci ประมาณ 2.6 เท่าด้วย (ดูตารางที่ 5.4) แต่สาเหตุนั้นอาจเกิดจากในส่วนของโปรแกรม Quicksort ที่มีการทำงานแบบเวียนเกิดนั้น มีปริมาณคำสั่งมากกว่า Fibonacci จึงทำให้หน่วยประมวลผลต้องใช้คำสั่งการย้ายการควบคุมโดยระบุตัวถูกกระทำการด้วยวิธีการอ้างตำแหน่งแบบโดยตรง ซึ่งเป็นวิธีการอ้างตำแหน่งข้อมูลในหน่วยความจำได้ในช่วงตำแหน่งที่กว้างมากกว่าวิธีการอ้างแบบรีจิสเตอร์และแบบทันที และสำหรับวิธีการอ้างตำแหน่งแบบฐานสัมพัทธ์ มีอัตราส่วนการใช้งานโดยเฉลี่ยร้อยละ 11.68 พบว่าโปรแกรม Fibonacci มีการใช้งานวิธีการอ้างตำแหน่งแบบนี้น้อยกว่าโปรแกรมอื่นๆ ประมาณ 2.6 เท่า ซึ่งน่าจะเป็นเพราะภายในโปรแกรม Fibonacci ไม่มีการใช้โครงสร้างข้อมูลแบบแถวลำดับ ในขณะที่โปรแกรมอื่นๆ มีการใช้โครงสร้างข้อมูลรูปแบบดังกล่าวซึ่งลักษณะข้อมูลมีการเรียงลำดับต่อเนื่องกันในหน่วยความจำ

5.4 สรุป

ผลจากการวิเคราะห์ทำให้ทราบถึงจากลักษณะเด่นที่สามารถพบได้ในทุกโปรแกรม ซึ่งได้แก่ การใช้งานรูปแบบคำสั่งที่มีตัวถูกกระทำการในอัตราส่วนที่สูง แสดงว่าจำเป็นต้องมีวิธีการอ้างตำแหน่งที่อยู่เพื่ออ้างถึงข้อมูลสำหรับใช้ในการปฏิบัติการคำสั่ง ซึ่งวิธีการอ้างตำแหน่งที่โปรแกรมใช้งานบ่อยครั้งและมีประโยชน์ต่อการทำงานของหน่วยประมวลผล ได้แก่วิธีการอ้างตำแหน่งที่หน่วยประมวลผลสามารถทำงานได้เร็ว เช่นแบบรีจิสเตอร์ และวิธีการอ้างตำแหน่งที่มีการคำนวณเพื่อเข้าถึงหน่วยความจำได้ง่าย เช่นแบบโดยตรง และแบบฐานสัมพัทธ์ สำหรับกลุ่มประเภทคำสั่งที่มีอัตราส่วนการใช้งานมาก จะเป็นกลุ่มคำสั่งพื้นฐานสำหรับการทำงานของเครื่องคอมพิวเตอร์ ซึ่งได้แก่การเคลื่อนย้ายข้อมูล การคำนวณทางคณิตศาสตร์ และการย้ายการควบคุม

ลักษณะเหล่านี้อาจนำไปใช้เป็นแนวทางในการพัฒนาสมรรถนะของหน่วยประมวลผลต้นแบบได้โดยออกแบบสถาปัตยกรรมภายในระบบให้มีประสิทธิภาพมากยิ่งขึ้นและทำงานได้เร็วขึ้น อันได้แก่ กำหนดชุดคำสั่งใช้งานเท่าที่จำเป็น ปรับวิธีการอ้างตำแหน่งที่อยู่ให้ง่ายขึ้นและลดจำนวนวิธีการให้น้อยลง กำหนดให้มีจำนวนรีจิสเตอร์ช่วยในการทำงานมากขึ้น และเพิ่มความเร็วในการทำงานของหน่วยความจำให้เร็วขึ้น แต่อีกแนวทางหนึ่งที่เราสามารถพัฒนาหน่วยประมวลผลให้เพิ่มประสิทธิภาพได้โดยที่ยังคงรูปแบบโครงสร้างสถาปัตยกรรมเดิม เช่น การประมวลผลคำสั่งแบบสายท่อ การดึงข้อมูลไว้ล่วงหน้า และการใช้หน่วยความจำแคช ซึ่งในงานวิจัยนี้ได้ทำการศึกษาและทดสอบตามแนวทางนี้ดังจะได้แสดงรายละเอียดในบทถัดไป