การดำเนินการบวกแบบขนานบนระบบแทนจำนวนฐานคู่

นายวุฒิภัทร เฉลิมฉัตรวิเชียร

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2555

PARALLEL ADDITIVE OPERATION ON DOUBLE BASE NUMBER SYSTEM

Mr. Wutthipat Chalermchatwichien

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
ChulalongkornUniversity
Academic Year 2012

หัวข้อวิทยานิพนธ์	การดำเนินการบวกแบบขนานบนระบบแทนจำนวนฐานคู่
โดย	นายวุฒิภัทร เฉลิมฉัตรวิเชียร
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

……………………………………………คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

……………………………………………ประธานกรรมการ
(ศาสตราจารย์ ดร.ประภาสจงสถิตย์วัฒนา)

……………………………………………อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

……………………………………………กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง)

วุฒิภัทร เฉลิมฉัตรวิเชียร: การดำเนินการบวกแบบขนานบนระบบแทนจำนวนฐานคู่ (PARALLEL ADDITIVE OPERATION ON DOUBLE BASE NUMBER SYSTEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ.ดร.อรรถสิทธิ์ สุรฤกษ์, 37หน้า.

ระบบแทนจำนวนฐานคู่เป็นระบบแทนจำนวนทางเลือกนอกเหนือจากระบบแทนจำนวนฐานสองรูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่นั้นมีความคล้ายคลึงกันกับรูปแบบแทนจำนวนในระบบอื่นทั่วไปแต่อาศัยเลขฐานสองฐาน ได้แก่ ฐานสอง และ ฐานสาม แทนเลขฐานเดียว คุณสมบัติที่สำคัญสองประการของระบบแทนจำนวนฐานคู่คือ คุณสมบัติความซ้ำซ้อน และคุณสมบัติการกระจายตัวของบิทหนึ่งสูง ซึ่งคุณสมบัติความซ้ำซ้อนนั้นมีประโยชน์สำหรับการคำนวณเชิงเลขคณิตแบบขนาน ในงานวิจัยนี้เราสนใจการดำเนินการบวกแบบขนานบนระบบแทนจำนวนฐานคู่ โดยได้นำเสนออัลกอริทึมสำหรับการดำเนินการบวกแบบขนานบนระบบแทนจำนวนฐานคู่ พร้อมทั้งบทพิสูจน์ความถูกต้องของอัลกอริทึม นอกจากนั้นงานวิจัยชิ้นนี้ยังได้พัฒนาการทำงานจากอัลกอริทึมที่ทำงานกับตารางย่อยซึ่งถูกแบ่งออกจากตารางหลักที่มีขนาดเป็นสองให้เป็นอัลกอริทึมที่สามารถทำงานได้บนตารางย่อยขนาดใดๆที่มากกว่าหรือเท่ากับสองได้ซึ่งประสิทธิภาพเชิงเวลาที่ได้ เป็นประสิทธิภาพเชิงเวลาคงตัว

ภาควิชาวิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก

ปีการศึกษา2555

5570501421 : MAJOR COMPUTER ENGINEERING

KEYWORDS : DOUBLE BASE NUMBER SYSTEM / FUNDAMENTAL ARITHMETIC OPERATION / PARALLEL ADDITION

WUTTHIPAT CHALERMCHATWICHIEN: PARALLEL ADDITIVE OPERATION ON DOUBLE BASE NUMBER SYSTEM. ADVISOR: ASST.PROF.ATHASIT SURARERKS, Ph.D.,37pp.

Double base number system (DBNS) is an alternative number system besides the binary system. Its representation is similar to the radix number system together with two bases, usually be two and three. DBNS preserves the two important properties: redundancy and sparseness. The redundancy is the property accommodating with the parallelism. In this research, we are interested in parallel addition algorithm on DBNS. Our theoretical result shows that parallel addition in DBNS can be performed. An addition algorithm together with the proof of correctness is described in this paper. In general, we study the generalization form of DBNS addition algorithms in any sizes. The algorithm takes constant-time complexity.

Department : Computer Engineering Student's Signature

Field of Study : Computer Engineering Advisor's Signature

Academic Year : 2012

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์และความช่วยเหลืออย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์สุรฤกษ์ ซึ่งเป็นอาจารย์ที่ปรึกษาผู้ให้คำปรึกษา ข้อเสนอแนะที่เป็นประโยชน์อย่างยิ่งต่องานวิจัย ทั้งยังช่วยตรวจแก้ไขในส่วนที่ผิดพลาดต่างๆ ทำให้วิทยานิพนธ์ฉบับนี้ออกมาเป็นงานวิจัยที่สมบูรณ์ได้

ขอขอบพระคุณ ศาสตราจารย์ ดร.ประภาสจงสถิตย์วัฒนา และ ผู้ช่วยศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง ประธานกรรมการและกรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำและแนวทางในการแก้ไขปรับปรุงทั้งในด้านของเนื้อหาและรูปแบบการเขียนวิทยานิพนธ์ในส่วนที่ยังบกพร่องเพื่อให้งานมีความสมบูรณ์ที่สุด

ขอขอบพระคุณบัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ที่ได้ให้การสนับสนุน "ทุนอุดหนุนวิทยานิพนธ์สำหรับนิสิต" ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์จุฬาลงกรณ์มหาวิทยาลัย สำหรับ "ทุนอัจฉริยภาพคืนรัง" ซึ่งได้ให้ทุนการศึกษามาเป็นระยะเวลาหนึ่งปี ขอบพระคุณท่านอาจารย์และภาควิชาวิศวกรรมคอมพิวเตอร์ ที่ได้ให้โอกาส ความรู้ ประสบการณ์และการสนับสนุนตลอดระยะเวลาที่ข้าพเจ้าได้ศึกษาอยู่ ณ ที่แห่งนี้และทำวิจัย

ขอขอบคุณพี่ๆน้องๆสมาชิกห้องปฏิบัติการทางวิศวกรรมระบบนับได้เชิงทฤษฎี (Engineering Laboratory in Theoretical Enumerable System : ELITE)ที่ได้ให้ความช่วยเหลือ ข้อเสนอแนะ และให้คำปรึกษาในเรื่องต่างๆ จนกระทั่งวิทยานิพนธ์สำเร็จด้วยดี

สุดท้ายนี้ข้าพเจ้าขอขอบพระคุณคนที่สำคัญที่สุดในชีวิตคือ บิดาและมารดารวมถึงญาติพี่น้องของข้าพเจ้า ที่คอยเป็นกำลังใจและเฝ้าห่วงใยข้าพเจ้าอยู่เสมอมา และขอบคุณสำหรับกำลังใจความห่วงใยและเอาใจใส่อย่างดีจากคนสำคัญของข้าพเจ้า ทำให้ข้าพเจ้าสำเร็จการศึกษาในระดับชั้นปริญญาโทได้อย่างภาคภูมิใจ

สารบัญ

สารบัญรูป

หน้า

บทที่ 1

บทนำ

เนื้อหาในบทนี้นำเสนอที่มาและความสำคัญของปัญหาวัตถุประสงค์ขอบเขตข้อจำกัดขั้นตอนการศึกษาวิธีการดำเนินงานและประโยชน์ที่คาดว่าจะได้รับของวิทยานิพนธ์ฉบับนี้

1.1 ที่มาและความสำคัญของปัญหา

หากพิจารณาถึงระบบแทนจำนวนฐานสองที่ใช้ในสถาปัตยกรรมคอมพิวเตอร์ในปัจจุบันนั้นจะเห็นได้ว่าวิธีการหรืออัลกอริทึมในการเพิ่มประสิทธิภาพเชิงเวลาของระบบแทนจำนวนฐานสองนั้น ไม่สามารถทำงานเป็นแบบขนานได้โดยสมบูรณ์ซึ่งงานวิจัยส่วนใหญ่นั้นจะมุ่งเน้นไปที่การสร้างสถาปัตยกรรมคอมพิวเตอร์เพื่อใช้คำนวณตัวทดไว้ล่วงหน้า เพื่อจำกัดความยาวของสายการแพร่ของการทด (carry-propagation chain) ของการดำเนินการทางเลขคณิตพื้นฐาน

นอกเหนือจากนั้นหากพิจารณาระบบแทนจำนวนที่ใช้กันในปัจจุบัน(classical number system)ได้แก่ระบบแทนจำนวนฐานสิบแล้ว จะพบว่าคุณสมบัติและวิธีการคำนวณแบบเลขคณิตพื้นฐานนั้นมีความใกล้เคียงกับระบบแทนจำนวนฐานสองเป็นอย่างมากซึ่งไม่สามารถนำวิธีการคำนวณแบบขนานเข้ามาประยุกต์ใช้ได้สมบูรณ์เช่นกัน

จากปัจจัยต่างๆ ที่กล่าวมาข้างต้น งานวิจัยนี้จึงมีแนวความคิดที่จะประยุกต์การทำงานแบบขนานเข้ากับระบบที่ไม่ได้ใช้ในชีวิตประจำวัน (non-classical number system)ซึ่งได้แก่ระบบแทนจำนวนฐานคู่ โดยระบบแทนจำนวนฐานคู่นั้นมีรูปแบบการแทนจำนวนอยู่ในรูปแบบของตารางสองมิติ และเป็นระบบที่มีความซ้ำซ้อนซึ่งมีความเป็นไปได้ที่จะนำการคำนวณแบบขนานเข้ามาประยุกต์ใช้กับระบบแทนจำนวนฐานคู่นี้ งานวิจัยนี้จึงมองเห็นความเป็นไปได้ในการพัฒนาอัลกอริทึมสำหรับการดำเนินการทางเลขคณิตพื้นฐานบนระบบแทนจำนวนฐานคู่นี้ให้มีประสิทธิภาพเชิงเวลาที่ดีและหวังเป็นอย่างยิ่งว่างานวิจัยชิ้นนี้จะทำให้เกิดแรงบันดาลใจสำหรับนักวิจัยท่านอื่นๆ ในการพัฒนาประสิทธิภาพด้านอื่นๆ ของระบบแทนจำนวนฐานคู่ต่อไป

ระบบแทนจำนวนฐานคู่ได้ถูกนำเสนอครั้งแรกโดยดิมิทรอฟ จูเลียน และมิลเลอร์(V.S. Dimitrov,G.A.Jullien and W.C. Miller) [1] ในปีคริสตศักราช 1997โดยจุดประสงค์ที่ระบบ

แทนจำนวนนี้ได้ถูกคิดค้นและนำเสนอขึ้นมาก็คือ ความเป็นระบบแทนจำนวนที่มีความซ้ำซ้อนและมีการกระจายตัวของบิทหนึ่งค่อนข้างสูงเมื่อเทียบกับระบบแทนจำนวนฐานสอง ทำให้ระบบแทนจำนวนฐานคู่นี้มีความน่าสนใจในเชิงการพัฒนาการทำงานในระดับเลขคณิตพื้นฐานซึ่งมีแนวโน้มที่จะสามารถนำการคำนวณในรูปแบบขนานเข้ามาประยุกต์ใช้กับระบบแทนจำนวนฐานคู่นี้

เป็นที่ทราบกันดีว่าระบบแทนจำนวนฐานคู่เป็นระบบแทนจำนวนที่มีความซ้ำซ้อน ซึ่งภายหลังจากที่ระบบแทนจำนวนนี้ได้ถูกนำเสนอ ก็มีงานวิจัยเพื่อที่จะพยายามหารูปแบบแทนจำนวนของจำนวนเต็มใดๆ ในระบบแทนจำนวนฐานคู่นี้ ให้อยู่ในรูปแบบที่มีจำนวนบิทหนึ่งน้อยที่สุด โดยใช้ประสิทธิภาพเชิงเวลา ของอัลกอริทึมสำหรับการหารูปแบบนั้นให้มีประสิทธิภาพมากที่สุด [2-4]สามารถดูรายละเอียดในบทที่ 2ทฤษฎีและงานวิจัยที่เกี่ยวข้องสำหรับการคำนวณทางเลขคณิตพื้นฐานของระบบแทนจำนวนฐานคู่นี้ ยังไม่มีงานวิจัยที่เสนอแนวทางอัลกอริทึมในการดำเนินการทางเลขคณิตพื้นฐานที่มีประสิทธิภาพเชิงเวลาเป็นค่าคงที่ มีเพียงแต่การนำเสนอหลักการดำเนินการทางเลขคณิตพื้นฐานโดยอาศัยวิธีการของระบบแทนจำนวนฐานสองและวิธีการทางเลขคณิตพื้นฐานเข้ามาจัดการและประยุกต์ใช้ให้เข้ากับระบบแทนจำนวนฐานคู่นี้เท่านั้นซึ่งวิธีการที่ได้นั้นยังไม่มีประสิทธิภาพเชิงเวลาที่เป็นที่น่าพอใจ งานวิจัยนี้จึงมุ่งเน้นที่จะนำเสนอ อัลกอริทึมที่จะสามารถเพิ่มประสิทธิภาพเชิงเวลาในการคำนวณทางเลขคณิตพื้นฐาน เป็นหลัก

1.2 วัตถุประสงค์

เพื่อเสนออัลกอริทึมในการคำนวณแบบขนานของการดำเนินการทางเลขคณิตพื้นฐานบนระบบแทนจำนวนฐานคู่ ให้มีประสิทธิภาพเชิงเวลาเป็นค่าคงที่

1.3 ขอบเขตของวิทยานิพนธ์

1) ในงานวิจัยนี้เราสนใจการดำเนินการทางเลขคณิตพื้นฐานคือการบวกบนระบบแทนจำนวนฐานคู่
2) ระบบแทนจำนวนฐานคู่ที่สนใจในงานวิจัยนี้คือระบบแทนจำนวนฐานคู่ที่ประกอบด้วยฐานสอง และ ฐานสามเท่านั้น

1.4 ขั้นตอนการศึกษาและวิธีการดำเนินงาน

1) ศึกษางานวิจัยและบทความทางวิชาการที่เกี่ยวข้องกับระบบแทนจำนวนฐานคู่
2) ศึกษางานวิจัยและบทความทางวิชาการที่เกี่ยวข้องกับการคำนวณแบบขนาน
3) วิเคราะห์คุณสมบัติการทำงานของการคำนวณแบบขนานแต่ละประเภท ศึกษาคุณสมบัติของการคำนวณแบบขนาน และออกแบบการคำนวณให้เข้ากับการดำเนินการบวกบนระบบแทนจำนวนฐานคู่
4) พิสูจน์ความถูกต้องของอัลกอริทึมที่ได้ด้วยวิธีการทางคณิตศาสตร์
5) วิเคราะห์ประสิทธิภาพเชิงเวลาและข้อดีข้อเสียของอัลกอริทึมที่ได้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) ได้อัลกอริทึมในการดำเนินการเชิงเลขคณิตบวกบนระบบแทนจำนวนฐานคู่โดยอาศัยวิธีการแบบขนานซึ่งมีประสิทธิภาพเชิงเวลาเป็นค่าคงที่
2) ได้ผลลัพธ์ของการดำเนินการทางเลขคณิตพื้นฐานซึ่งสามารถนำไปต่อยอดกับการดำเนินการอื่นๆนอกจากการบวก

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวเรื่องดังต่อไปนี้

Chalermchatwichien, W., Surarerks, A., **“Parallel Additive Operation on Double-base Number System”.**Proceeding of the 17th International Annual Symposium on Computational Science and Engineering: ANSCSE17, Khonkaen, Thailand, March 27-29, 2013, Pages 99-100.

Chalermchatwichien, W., Surarerks, A., **“Parallel addition for double base number system”.**Proceeding of the 10th International Joint Conference on Computer Science and Software Engineering: JCSSE’13, Khonkaen, Thailand, May 30-31, 2013.

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ทฤษฎีและงานวิจัยที่เกี่ยวข้องของงานวิจัยนี้แบ่งออกเป็นสองส่วนได้แก่ส่วนที่หนึ่งกล่าวถึงบทนิยามและรูปแบบแทนจำนวนของระบบแทนจำนวนฐานคู่นอกจากนั้นยังได้กล่าวถึงอัลกอริทึมที่ใช้ในการหารูปแบบแทนจำนวนในระบบฐานคู่สำหรับจำนวนเต็มใดๆรวมทั้งประสิทธิภาพเชิงเวลาของอัลกอริทึมดังกล่าวส่วนที่สองกล่าวถึงการคำนวณเชิงเลขคณิตแบบขนานโดยอาศัยแนวคิดจากงานวิจัยอื่นๆที่ได้เสนอมาและพิจารณาเห็นว่ามีแนวโน้มที่จะสามารถนำมาสร้างแนวคิดที่จะใช้ให้เข้ากับการดำเนินการทางเลขคณิตพื้นฐานสำหรับระบบแทนจำนวนฐานคู่ได้อย่างมีประสิทธิภาพ

2.1 ระบบแทนจำนวนฐานคู่ (double base number system)

ระบบแทนจำนวนฐานคู่ ของ ดิมิทรอฟ จูเลียน และมิลเลอร์(V.S. Dimitrov,G.A. Jullien and W.C. Miller) [1] ที่ถูกเสนอขึ้นในปีคริสตศักราช 1997มีคุณสมบัติที่น่าสนใจสองประการได้แก่ ระบบแทนจำนวนฐานคู่เป็นระบบที่มีสมบัติความซ้ำซ้อน(redundant system) หมายความว่า มีจำนวนอย่างน้อยหนึ่งจำนวนที่สามารถหารูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่ได้มากกว่าหนึ่งรูปแบบดังนั้นการหาวิธีหรือคิดค้นอัลกอริทึมที่ใช้สำหรับหารูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่ของจำนวนเต็มใดๆ โดยให้มีจำนวนบิทของหนึ่งน้อยที่สุด ซึ่งเรียกว่ารูปแบบคาโนนิคัล (canonical form) [2]จึงเป็นสิ่งที่ท้าทายและมีงานวิจัยที่เกี่ยวข้องซึ่งได้เสนออัลกอริทึมดังกล่าวซึ่งจะได้กล่าวในลำดับถัดไปและคุณสมบัติที่สำคัญของระบบแทนจำนวนฐานคู่ประการสองคือการกระจายตัวของบิทหนึ่ง (sparseness)ซึ่งมีอัตราสูงเมื่อเทียบกับระบบแทนจำนวนอื่นนอกจากนี้มีงานวิจัยที่นำระบบแทนจำนวนฐานคู่ไปประยุกต์ใช้ในการคำนวณการเข้ารหัสด้วยการคูณเชิงสเกลาร์แบบวงรีในเอกสารเชิงวิชาการอีกหลายฉบับ [5-6]

นิยามที่2.1 *ให้ m แทนจำนวนเต็มใดๆ m จะสามารถเขียนให้อยู่ในรูป*แบบแทนจำนวนฐานคู่(double basenumber representation)*ได้ก็ต่อเมื่อ สามารถเขียน $m=\sum d_{i,j}2^i3^j$ โดยที่$d_{i,j} \in \{0, 1\}$ สำหรับจำนวนเต็มi,j ใดๆ*

หากต้องการเขียนรูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่ให้อยู่ในรูปแบบเชิงกายภาพที่สามารถมองเข้าใจได้ง่ายเรานิยมแทนรูปแบบแทนจำนวนฐานคู่หนึ่งๆด้วยตารางสองมิติ โดยให้หลักแต่ละหลักแทนกำลังของฐานสอง 2^i และแถวแต่ละแถวแทนกำลังของฐานสาม 3^j และหากสัมประสิทธิ์หรือตัวเลข (digit) $d_{i,j}$ ที่พจน์ 2^i3^j ใดๆมีค่าเท่ากับ 1 เราระบุที่ช่องของตารางนั้นด้วยเลขหนึ่ง(active cell) ดังรูปต่อไปนี้

	2^0	2^1	2^2	2^3
3^0	0	1	0	0
3^1	0	1	1	0
3^2	0	0	1	0
3^3	0	1	0	0

110

	2^0	2^1	2^2	2^3
3^0	1	0	0	1
3^1	0	0	0	1
3^2	0	0	0	0
3^3	1	0	0	0

60

	2^0	2^1	2^2	2^3
3^0	1	1	0	1
3^1	0	1	1	1
3^2	0	0	1	0
3^3	1	1	0	0

170

รูปที 2.1 รูปแสดงตัวอย่างตารางแทนจำนวนในระบบแทนจำนวนฐานคู่

รูปที่ 2.1 แสดงให้เห็นรูปแบบการแทนจำนวนฐานคู่ของเลข 110 60 และ 170 ตามลำดับ จะเห็นว่า จำนวนในระบบแทนจำนวนฐานคู่สามารถแสดงให้เข้าใจได้ง่ายขึ้นในรูปแบบของตารางสองมิติ เช่น $110 = 1\times3^3 2^1 + 1\times3^2 2^2 + 1\times3^1 2^2 + 1\times3^1 2^1 + 1\times3^0 2^1$ ซึ่งประกอบด้วยตัวเลข 1 จำนวน 5 หลัก ทั้งนี้ เรานิยมจะเรียกเซล (cell) ต่างๆ ในตารางที่มีคาตัวเลขเป็น 1 ว่า แอ๊คทีฟเซล (active cell)

นอกจากนี้ จำนวนบางจำนวนในระบบแทนจำนวนฐานคู่ สามารถเขียนได้มากกว่าหนึ่งแบบ เช่น 110 สามารถแจกแจงเป็นตัวเลขในระบบแทนจำนวนฐานคู่ได้อีกมากกว่าหนึ่งรูปแบบ คือ

$$
\begin{aligned}
110 &= 1\times3^3 2^1 + 1\times3^1 2^4 + 1\times3^1 2^1 + 1\times3^0 2^1 \\
&= 1\times3^3 2^1 + 1\times3^1 2^4 + 1\times3^0 2^3
\end{aligned}
$$

ซึ่งจะเห็นว่า รูปแบบใหม่นี้จะมีจำนวน แอ๊คทีฟเซลลดลงเป็นสี่เซล และสามเซล ตามลำดับ

ทั้งนี้ เป็นที่สังเกตได้ง่ายว่า ศูนย์ จะมีเพียงรูปแบบแทนจำนวนรูปแบบเดียวเท่านั้นที่สามารถหาได้ คือ ตารางที่ไม่มีแอ๊คทีฟเซลเลย หรืออีกนัยหนึ่งคือ ไม่มีตัวเลขหนึ่งปรากฏในรูปแบบแทนจำนวนเลย

อัลกอริทึมสำหรับหารูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่สำหรับจำนวนเต็มใดๆ [1] สามารถอธิบายได้ดังนี้

Algorithm: Find a double base number representation for x

Input: x a number

Output: d[i][j] where $x = \sum d[i][j] \times 3^i 2^j$

Begin

Generate $R_{MN}=\{r_{m,n} = 3^m 2^n : 0 \le m \le M, 0 \le n \le N\}$ where M = Floor($\log_3 x$)

N = Floor($\log_2 x$)

Sort: R_{MN} from max to min, element in R_{MN} denoted by R_i

For(k=max to min and x>0)**do**

If(R[k] ≤x)**then** d[m][n] =1

x =x-R[k];

else d[m][n] = 0

endif

enddo

End

ตัวอย่างที่2.2*กำหนดให้ จำนวนเต็ม x=23จะคำนวณรูปแบบแทนจำนวนฐานคู่ได้ดังนี้*

วิธีทำ คำนวณค่า M และ N ได้คือ M=Floor($\log_3 23$) = 2 และN = Floor($\log_2 23$) = 4 ดังนั้นจะได้R_{24} = {1,2,3,4,6,8,9,12,16,18,24,36,48,72,144} โดยจากอัลกอริทึมจะได้ว่า23= 18 + 4 + 1 ดังนั้น รูปแบบแทนจำนวนของ 23 คือ $1\times3^2 2^1 + 1\times3^0 2^2 + 1\times3^0 2^0$ □

โดยวิธีนี้ไม่ได้รับประกันว่า คำตอบที่ได้จะอยู่ในรูปแบบแทนจำนวนคาโนนิคัลเสมอไป แต่ก็เป็นอัลกอริทึมที่มีประสิทธิภาพสำหรับการหารูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่และทำงานได้รวดเร็ว

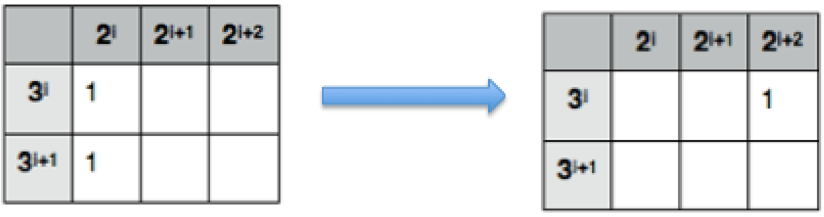
จากงานวิจัยของกิลเบอร์ก[2] ในปี ค.ศ. 2005 ได้มีการเสนออัลกอริทึมในการหารูปแบบแทนจำนวนฐานคู่ โดยใช้แนวทางของอัลกอริทึมเชิงละโมบ (greedy algorithm) ที่เรียกว่า มัลติพาธกรีดดีอัลกอริทึม(multipath-greedy algorithm) เป็นอัลกอริทึมที่ปรับปรุงมาจากอัลกอริทึมเดิมที่ถูก

เสนอไว้ โดยอัลกอริทึมนี้สามารถหาคำตอบได้มากกว่าหนึ่งคำตอบต่อการทำงานหนึ่งครั้งซึ่งทำให้มีแนวโน้มที่จะได้คำตอบเป็นรูปแบบคาโนนิคัลมากขึ้นซึ่งอัลกอริทึมนี้มีประสิทธิภาพเชิงเวลาเป็น $O(\log^2 x)$ เมื่อ x คือจำนวนเต็มใดๆที่ต้องการหารูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่

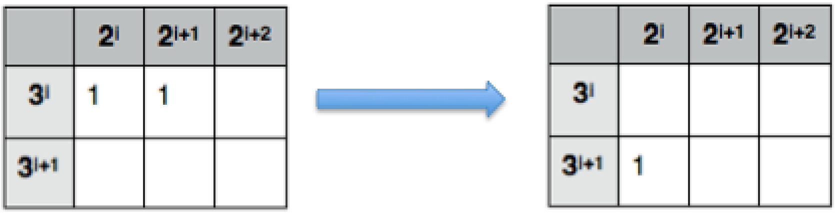
2.2 การดำเนินการทางเลขคณิตพื้นฐานบนระบบแทนจำนวนฐานคู่

สำหรับการคำนวณในระบบแทนจำนวนฐานคู่นี้ พิจารณาและออกแบบโดยอาศัยกฎการทดและกฎการลดรูปจากคุณสมบัติของระบบ โดยสามารถอธิบายหลักการพื้นฐานได้คือ

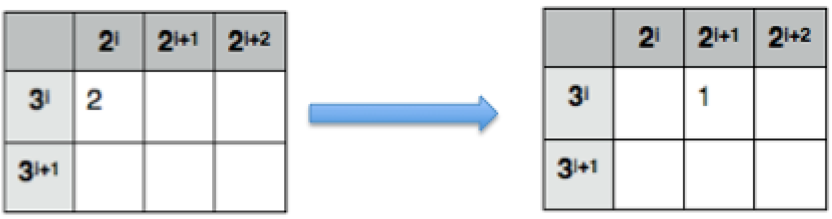
กฎการลดทอน (reduction rules) ซึ่งมีลักษณะสองลักษณะที่แตกต่างกันคือแบบที่หนึ่ง ในกรณีที่บิตของหนึ่งอยู่ติดกันในแนวแถว ดังแสดงในรูปที่ 2.2 (ก)ซึ่งสอดคล้องกับสมการ $1 \times 3^i 2^j + 1 \times 3^{i+1} 2^j = 1 \times 3^i 2^{j+2}$ และแบบที่สอง ในกรณีที่บิตของหนึ่งอยู่ติดกันในแนวหลักดังแสดงในรูปที่ 2.2(ข)ซึ่งสอดคล้องกับสมการ $1 \times 3^i 2^j + 1 \times 3^i 2^{j+1} = 1 \times 3^{i+1} 2^j$



(ก)



(ข)



(ค)

รูปที่ 2.2 รูปแสดงกฎการลดทอนสามประเภท

กฎการทด (carry propagation rules) ในกรณีที่มีการบวกกันแล้วทำให้ค่าประจำเซล มีค่าเท่ากับ 2 โดยอาศัยสมการ $2 \times 3^i 2^j = 1 \times 3^i 2^{j+1}$ ดังแสดงในรูปที่ 2.2(ค)

2.2.1 **กระทำทางเลขคณิตพื้นฐานบวก**

วิธีการพื้นฐานสำหรับการดำเนินการทางเลขคณิตพื้นฐานบวกจากตัวถูกกระทำสองตัว ตัวตั้งและตัวบวก (addend and adder)สามารถทำได้โดยง่ายโดยเราจะพิจารณาตัวถูกกระทำแต่ละตัวในรูปแบบของตารางสองมิติและนำตารางตัวถูกกระทำทั้งสองมาซ้อนทับกันและพิจารณาค่าของเซลล์ผลลัพธ์ที่ได้แบบเซลล์ต่อเซลล์โดยอาจมีสัญลักษณ์ชั่วคราวไว้สำหรับเก็บค่าในบางกรณีเช่นเซลล์หนึ่งเจอกับเซลล์หนึ่งซึ่งจะให้เซลล์ผลลัพธ์เก็บค่าสองไว้และทำการเปลี่ยนรูปให้อยู่ในรูปแบบของระบบแทนจำนวนฐานคู่ภายหลัง

2.2.2 การดำเนินการทางเลขคณิตพื้นฐานลบ

วิธีการพื้นฐานสำหรับการดำเนินการทางเลขคณิตพื้นฐานลบจากตัวตั้งและตัวลบสามารถทำได้โดยง่ายคล้ายกับการดำเนินการทางเลขคณิตพื้นฐานบวกคือการนำตารางของตัวตั้งและตัวลบมาซ้อนกันและพิจารณาค่าของเซลล์ผลลัพธ์จากค่าของเซลล์ที่ซ้อนทับกันแบบเซลล์ต่อเซลล์โดยจะต้องมีสัญลักษณ์ชั่วคราวในการเก็บค่าของเซลล์ผลลัพธ์เช่นกันเช่นเซลล์ตัวตั้งมีค่าเป็น0 และเซลล์ตัวลบมีค่าเป็น 1 จะได้เซลล์ผลลัพธ์มีค่าเป็น -1 หลังจากนั้นจึงเข้าสู่กระบวนการเปลี่ยนรูปเป็นรูปแบบของระบบแทนจำนวนฐานคู่ในภายหลังแต่การลบสามารถกระทำได้ก็ต่อเมื่อ ตัวตั้งต้องมีค่ามากกว่าตัวลบ ทั้งนี้เพราะระบบแทนจำนวนฐานคู่ไม่สามารถแสดงจำนวนเต็มลบได้

2.2.3 การดำเนินการทางเลขคณิตพื้นฐานคูณ

เนื่องจากการแทนจำนวนฐานคู่ด้วยตารางสองมิติเป็นการแทนโดยใช้จำนวนแทนอยู่ในรูปแบบของผลบวกของผลคูณ(sum of product) ดังนั้นในการดำเนินการทางเลขคณิตคูณนั้นเราจำเป็นต้องขยายขนาดของตารางเพิ่มขึ้นเป็นสองเท่าจากขนาดของตัวถูกกระทำเดิมและทำการคูณโดยมีลักษณะคล้ายการคูณแบบพหุนามคือทุกเซลล์ตารางที่มีค่าเป็น 1 มีการพบกันหมดและเพิ่มขนาดเลขชี้กำลัง iและ j จากผลบวกของเลขชี้กำลังของเซลล์แต่ละเซลล์ที่พบกันสำหรับฐานสองและฐานสามตามลำดับ

2.3 ระบบแทนจำนวนแบบมีเครื่องหมาย(signed digit number system)

ระบบแทนจำนวนแบบมีเครื่องหมาย (signed-digit number representation system) ได้ถูกเสนอขึ้นครั้งแรกในโดยอวิเซียนีส(Avizienis) [7] โดยใช้แนวคิดของการกำหนให้ ตัวเลข (digit) มีเครื่องหมายกำกับซึ่งทำให้ระบบรูปแบบแทนจำนวนนี้มีสมบัติ ความซ้ำซ้อน (redundancy property) และสามารถสร้างตัวดำเนินการทางคณิตศาสตร์แบบขนานได้ โดยการบวกแบบขนานได้ถูกพัฒนาขึ้นมาเป็นลำดับดังปรากฏใน [8] ดังนี้

นิยามที่2.3ระบบแทนจำนวนแบบมีเครื่องหมาย (signed-digit number representation system) *ประกอบด้วย เลขฐาน (base) βเป็นจำนวนเต็มโดยที่ $|\beta| > 1$ และ D ชุดของตัวเลข (digit set) เป็นเซตจำกัดของตัวเลขตั้งแต่จำนวนเต็ม a ถึง จำนวนเต็ม b โดยที่ $a < 0 < b$ เขียนได้ดังนี้*

$$D = \{ a, a\text{-}1, \ldots , 0. \ldots, b\text{-}1, b \}$$

โดยที่ $b - a + 1 > \beta$. รูปแบบแทนจำนวนของจำนวนเต็ม x ใดๆ สามารถเขียนในระบบนี้ได้ดังนี้

$$x = d_0\beta^0 + d_1\beta^1 + \ldots + d_n\beta^n$$

เมื่อ d_iเป็นสมาชิกใน D

นิยามที่ 2.4*ระบบแทนจำนวนแบบมีเครื่องหมาย เลขฐาน β จะถูกเรียกว่ามีสมบัติ* การแทนศูนย์แบบแข็งแรง(strong representation of zero property, SRZ) *ถ้ามีจำนวนเต็ม $b_k, b_{k-1}, \ldots, b_1, b_0, b_{-1}, \ldots, b_{-h}$สำหรับจำนวนเต็มที่ไม่เป็นลบ h และ k บางตัว โดยที่ βเป็นรากของพหุนาม*

$$S(X) = b_kX^k + b_{k-1}X^{k-1} + \ldots + b_1X + b_0 + b_{-1}X^{-1} + \ldots + b_{-h}X^{-h}$$

*และ$b_0 > 2 \sum_{i\in\{-h,\ldots,k\}\{0\}} |b_i|$ โดยพหุนาม S จะถูกเรียกว่า*พหุนามแข็งแรง (strong polynomial) *สำหรับ β*

เพื่อความสะดวกในการเรียกใช้สัญลักษณ์ กำหนดให้ $B = b_0$และ $M = \sum_{i\in\{-h,\ldots,k\}\{0\}} |b_i|$ เราจะสามารถเขียนอสมการใหม่ได้ดังนี้คือ$B > 2M$ และสมบัตินี้เป็นสมบัติที่ของระบบแทนจำนวนแบบมีเครื่องหมายที่สามารถทำการบวกแบบขนานได้

หากเราให้เลขฐานβ มีสมบัติการแทนศูนย์แบบแข็งแรง และให้เซตของตัวเลข Aเป็นเซตตัวเลขแบบสมมาตร$A = \{-a,\ldots,0,\ldots,a\}$ เมื่อ

$$a = \lceil (B-1)/2 \rceil + \lceil (B - 1)/2(B - 2M) \rceil \times M$$

เมื่อกำหนดให้ $c = \lceil (B-1)/2(B-2M) \rceil$ และ $a' = \lceil (B-1)/2 \rceil$ ดังนั้นจะได้ $a = a' + cM$ และกำหนดให้เซตตัวเลข $A^* = \{ -a',\ldots,0,\ldots,a'\} \subset A$ ซึ่งจะถูกเรียกว่าอักขระภายใน (inner alphabet) และการบวกแบบขนานสามารถทำได้โดยอัลกอริทึมต่อไปนี้

Algorithm: Parallel addition in SRZ signed-digit number representation system

Input: $x = \sum_{i=m}^{n} x_i \beta^i$ where $x_n, \ldots, x_m$ be digits in A^* and $m \le n$

$y = \sum_{i=m}^{n} y_i \beta^i$ where $y_n, \ldots, y_m$ be digits in A^* and $m \le n$

Output: $z = x + y = \sum_{i=m-h}^{n+k} z_i \beta^i$ where $z_{n+k}, \ldots, z_{m-h}$ be digits in A^* and $m \le n$

begin

for each i (done in parallel) **do**

$z_i := x_i + y_i$

find $q_i \in \{-c,\ldots,0,\ldots,c\}$ where $z_i - q_i B \in A^*$

$z_i := z_i - \sum_{j=-h}^{k} q_{i-j} b_j$

enddo

end

ตัวอย่างที่ 2.5 *พิจารณาระบบแทนจำนวนแบบมีเครื่องหมายสำหรับเลขฐาน $\beta = (1+\sqrt{5})/2$ และชุดของตัวเลข $A = \{ -5, -4, \ldots, -, \ldots, 4, 5 \}$ คำนวณการบวกแบบขนานของ $x = 25\bar{2}5\bar{5}003$ และ $y = 512\bar{2}5\bar{4}005$*

วิธีทำ พิจารณาค่าของเลขฐาน $\beta = (1+\sqrt{5})/2$ ซึ่งเป็นรากของสมการของพหุนามดีกรีสอง

$$X^2 = X + 1$$

รากอีกตัวหนึ่งได้แก่ $\beta' = (1-\sqrt{5})/2 = -1/\beta$ เนื่องจาก $\beta^4 + (\beta')^4 = 7$ ดังนั้น β เป็นรากของพหุนามแข็งแรง

$$S(X) = -X^4 + 7 - 1/X^4$$

จะได้ว่า $B = 7$ และ $M = 2$ ซึ่งมีผลทำให้ $c = 1$ และ $a' = 3$ และ $a = 5$ อักขระที่เราจะทำงานด้วยคือ $A = \{-5,\ldots,0,\ldots,5\}$ ซึ่งอักขระภายในคือ $A^* = \{-3,\ldots,0,\ldots,3\}$
จากที่กำหนดมาให้

$x = 2\beta^7 + 5\beta^6 - 2\beta^5 + 5\beta^4 - 5\beta^3 + 3$

$y = 5\beta^8 + \beta^7 + 2\beta^6 - 2\beta^5 + 5\beta^4 - 4\beta^3 + 5$

จะได้ zซึ่งเป็นผลบวกของ xและ yคือ

$$
\begin{array}{cccccccccccccccccc}
z \longrightarrow & & & & & 5 & 3 & 7 & \bar{4} & 10 & \bar{9} & 0 & 0 & 8 & & & & \\
 & 1 & 0 & 0 & 0 & \bar{7} & 0 & 0 & 0 & 1 & & & & & & & & \\
 & & & 1 & 0 & 0 & 0 & \bar{7} & 0 & 0 & 0 & 1 & & & & & & \\
 & & & & \bar{1} & 0 & 0 & 0 & 7 & 0 & 0 & 0 & \bar{1} & & & & & \\
 & & & & & 1 & 0 & 0 & 0 & \bar{7} & 0 & 0 & 0 & 1 & & & & \\
 & & & & & & \bar{1} & 0 & 0 & 0 & 7 & 0 & 0 & 0 & \bar{1} & & & \\
 & & & & & & & & & 1 & 0 & 0 & 0 & \bar{7} & 0 & 0 & 0 & 1 \\
z \longrightarrow & 1 & 0 & 1 & \bar{1} & \bar{1} & 2 & 0 & 3 & 5 & \bar{2} & 1 & \bar{1} & 2 & \bar{1} & 0 & 0 & 1
\end{array}
$$

ดังนั้น จะได้ $x+y = z = \beta^{12} + \beta^{10} - \beta^9 - \beta^8 + 2\beta^7 + 3\beta^5 + 5\beta^4 - 2\beta^3 + \beta^2 - \beta + 2 - \beta^{-1} + \beta^{-4}$ดังนั้นผลการบวกคือ 441.66252584 □

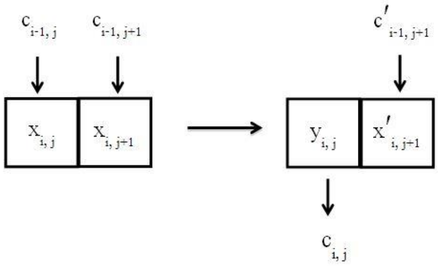
จะเห็นว่า อัลกอริทึมที่ได้ถูกเสนอมาข้างต้นนั้น เป็นการคำนวณแบบขนานโดยหาตัวเลขชุดหนึ่งซึ่งมาผลรวมเป็นศูนย์เพื่อมาลดทอนหลักที่มีอักขระเกินกว่าA*โดยพิจารณาโดยพิจารณาจากกรณีที่จะทำให้อักขระของผลลัพธ์ออกมาอยู่ในรูปของอักขระในเซตAเสมอซึ่งเราจะนำแนวคิดนี้ไปใช้กับงานวิจัยชิ้นนี้

2.4 งานวิจัยต่างๆที่เกี่ยวข้องกับระบบแทนจำนวนฐานคู่

ในงานวิจัยที่ผ่านมา ตัวดำเนินการบวกบนระบบแทนจำนวนฐานคู่นั้น นอกจากอัลกอริทึมที่ได้มีการเสนอแบบดั้งเดิม ของดิมิทรอฟ จูเลียน และมิลเลอร์[1] ซึ่งอาศัยกฎการลดทอนทั้งสามแบบแล้ว ยังได้มีงานวิจัย[9] ที่นำแนวคิดของการทำงานแบบ ออนเดอร์ฟราย มาพัฒนาเพื่อเพิ่มประสิทธิภาพเชิงเวลาของตัวดำเนินการบวก โดยงานวิจัยที่ได้ใช้วิธีการออนเดอร์ฟลายเข้ามาช่วยเพิ่มประสิทธิภาพเชิงเวลาในการทำงานได้นำเสนอการหาตัวทดออกสำหรับแต่ละแถวดังรูปที่2.3 โดยการเลื่อนกรอบการพิจารณาตามแนวแถวไปพร้อมๆกันในทุกๆคู่แถวที่ทำการคอมโพสิทและ

นำฟังก์ชันตัวทดออกที่ได้มาคอมโพสิทกันระหว่างแถว โดยเมื่อทำการคำนวณตัวทดล่วงหน้าแล้ว หลังจากนั้นจึงใช้กฎการลดทอนในรูปที่2.2 มาใช้เพื่อปรับให้ตารางแทนจำนวนของผลลัพธ์อยู่ในรูปแบบปกติ(normal form)

นิยามที่2.5 รูปแบบตารางแทนจำนวนจะถูกเรียกเป็นรูปแบบปกติ(normal form) ก็ต่อเมื่อรูปแบบตารางแทนจำนวนนั้นๆเป็นรูปแบบตารางแทนจำนวนฐานคู่และไม่มีเซลล์หนึ่งติดกันในแนวแถว



รูปที่2.3 แสดงรูปแบบของข้อมูลขาเข้าและข้อมูลขาออกของอัลกอริทึมที่ใช้เทคนิคออนเดอร์ฟลาย

The on-the-fly addition algorithm:

input: $A = \sum a_{i,j} 2^i 3^j$ and $B = \sum b_{i,j} 2^i 3^j$

output: $Z = \sum z_{i,j} 2^i 3^j$ where $Z = A + B$

begin **for** $k = 1$ **to** $\lceil \lg n \rceil$ **step** 1 **do**

for all pairs of compositions performing in parallel

composition row r and s by computing $C_{i,r} \circ C_{i,s} (x_{i,s})$

for all $j = 1$ **to** n **do**

perform the carry function $c_{i,j} (x_{i,j} = a_{i,j} + b_{i,j})$

enddo

perform composition of all carry functions of two rows in parallel

$z_{i,j}$ = round($x_{i,j} + c_{i,j}$)

enddo

perform all cell additions in parallel

end

หมายเหตุ $C_{i,j}(x_{i,j}) = (c_{i-1,j} + 2c_{i-1,j+1} + x_{i,j} + 2x_{i+1,j} - y_{i,j} - 2y_{i,j+1} - 2c'_{i-1,j-1})/3.$

คำอธิบายอัลกอริทึม

ในแต่ละรอบของการทำงานออนเดอร์ฟลายเราจะนำฟังก์ชันสำหรับหาตัวทดออก $C_{i,j}(x_{i,j}) = (c_{i-1,j} + 2c_{i-1,j+1} + x_{i,j} + 2x_{i+1,j} - y_{i,j} - 2y_{i,j+1} - 2c'_{i-1,j-1})/3$ ของแต่ละแถวมารวมกันแบบออนเดอร์ฟลาย และเมื่อได้ฟังก์ชันการรวมกันของตัวทดออกระหว่างแถวแล้วจึงสร้างกรอบการพิจารณาตัวทดออก ซึ่งจะต้องทำการเลื่อนกรอบการพิจารณาไปทีละหนึ่งตำแหน่งเนื่องจากมีการเปลี่ยนแปลงข้อมูลขาเข้าสำหรับการคำนวณฟังก์ชันจึงต้องเริ่มดำเนินการจากซ้ายไปขวา และคำนวณหาตัวทดออกจากฟังก์ชันจนครบขนาดของคอลัมน์ n จากนั้นจึงวนกลับไปทำงานแบบออนเดอร์ฟลายใหม่จนครบ log(n) รอบ โดยขนาดของ ตัวถูกดำเนินการมีขนาด $n \times n$ ซึ่งจะเห็นได้ว่าอัลกอริทึมที่ได้จากการใช้เทคนิคออนเดอร์ฟลายเข้ามาช่วยในการบวกตัวถูกดำเนินการที่อยู่ในรูปแบบแทนจำนวนฐานคู่จะสามารถเพิ่มประสิทธิภาพเชิงเวลาจาก $O(n^2)$ เป็น $O(n\log n)$

บทที่ 3

การดำเนินการบวกแบบขนานบนระบบแทนจำนวนฐานคู่

ในบทนี้จะกล่าวถึงการออกแบบอัลกอริทึมการบวกแบบขนานสำหรับระบบแทนจำนวนฐานคู่ โดยนำแนวคิดของคุณสมบัติ การซ้ำซ้อน ของระบบแทนจำนวนมาช่วยในการลดความยาวของสายการทดจนสามารถจำกัดสายการทดให้เป็นค่าคงที่ได้ ทั้งนี้เพราะ จากงานวิจัยที่ผ่านมา [8] ทำให้สามารถสรุปได้ว่า หากสายการทดของระบบการคำนวณสามารถถูกจำกัดได้ด้วยค่าคงที่ค่าหนึ่ง การคำนวณแบบขนานสามารถกระทำได้ในตัวดำเนินการนั้น ทั้งนี้ในงานวิจัยที่ผ่านมา [9] ตัวดำเนินการบวกของระบบแทนจำนวนญานคู่สามารถกระทำได้ด้วยอัลกอริทึมที่มีประสิทธิภาพที่มีอัตราการเติบโตเชิงเวลาเป็น $\Theta(n \log n)$ เมื่อ n เป็นขนาดของรูปแบบแทนจำนวนฐานคู่

3.1 แนวคิดของงานวิจัย

เนื่องจากระบบแทนจำนวนฐานคู่ เป็นระบบแทนจำนวนที่พัฒนามาจากพื้นฐานของระบบแทนจำนวนเลขฐานทั่วไป โดยแนวคิดของระบบแทนจำนวนฐานคู่นั้นถูกออกแบบมาเพื่อให้มีคุณสมบัติการกระจายตัวของตัวเลขหนึ่ง หมายความว่า ปริมาณของตัวเลขหนึ่งในรูปแบบแทนจำนวนมีน้อยลงเมื่อเทียบกับขนาดหรือตัวเลขทั้งหมดของรูปแบบแทนจำนวน ทั้งนี้เพราะ ตัวเลขหนึ่ง มีผลต่อการเกิดสายการทดในตัวดำเนินการพื้นฐานทั่วไป ซึ่งแนวทางนี้มีความแตกต่างจากระบบแทนจำนวนแบบมีเครื่องหมายของอเวเซียนีส ที่อนุญาตให้มีตัวเลขที่มีเครื่องหมายเป็นตัวกำหนดความยาวของสายการทดได้ สิ่งที่มีเหมือนกันในทั้งสองแนวคิดของการจำกัดหรือลดขนาดของสายการทดก็คือ การทำให้ระบบแทนจำนวนมีสมบัติของความซ้ำซ้อน ดังนั้นในงานวิจัยนี้ จึงมุ่งเน้นที่ค่าความซ้ำซ้อนที่เกิดขึ้นในระบบแทนจำนวน

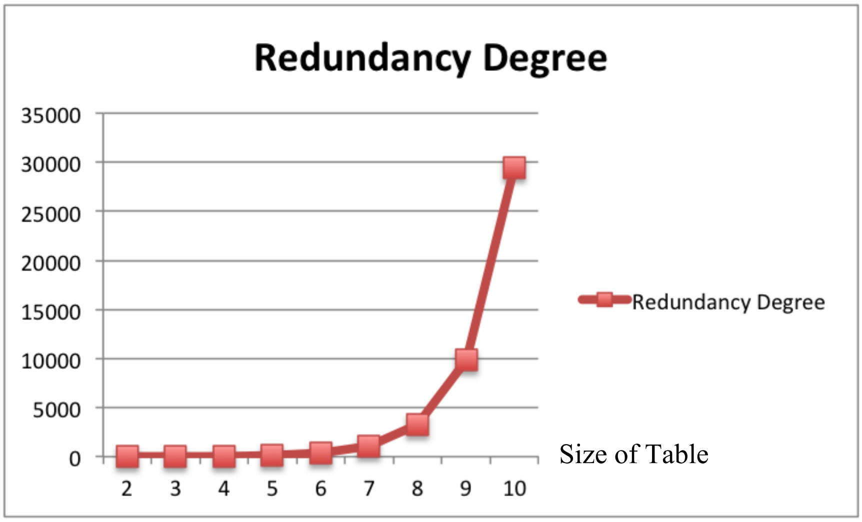
ค่าความซ้ำซ้อนของระบบแทนจำนวนใดๆ สามารถพิจารณาได้จาก ระดับความซ้ำซ้อน (redundancy degree) ซึ่งพิจารณาได้จากสัดส่วนของ จำนวนรูปแบบแทนจำนวนทั้งหมดที่เป็นไปได้เทียบกับจำนวนค่าเชิงตัวเลขที่เป็นไปได้เมื่อกำหนดขนาดของรูปแบบแทนจำนวนที่ค่าคงที่ค่าหนึ่ง

หากพิจารณา ค่าความซ้ำซ้อนที่เกิดขึ้นในระบบแทนจำนวนแบบมีเครื่องหมายของอเวเซียนีส จะพบว่า ค่าความซ้ำศ้อนจะมีค่ามากหรือน้อย สามารถปรับเปลี่ยนค่าได้โดยการ เพิ่มจำนวน

ตัวเลขในระบบ เช่น ระบบแทนจำนวนแบบมีเครื่องหมายบนฐาน 5 โดยใช้ชุดตัวเลข { -3, -2, -1, 0, 1, 2, 3} เมื่อเทียบกับระบบแทนจำนวนแบบมีเครื่องหมายบนเลขฐาน 5 แต่ใช้ชุดตัวเลขเป็น {-4, -3, -2, -1, 0, 1, 2, 3, 4 } จะเห็นว่า เมื่อขยายขนาดของชุดของตัวเลขจะทำให้ ระดับความซ้ำซ้อนมีค่าเพิ่มขึ้นได้ และระดับความซ้ำซ้อนนี่จะมีผลต่อความสามารถในการสร้างตัวดำเนินการแบบขนาน

ในงานวิจัยนี้ ระบบแทนจำนวนฐานคู่ บนเลขฐานสองและฐานสาม ชุดของตัวเลขที่ใช้ประกอบด้วย ตัวเลขศูนย์ และตัวเลขหนึ่งเท่านั้น ทั้งนี้ เนื่องจากการขยายขนาดของชุดตัวเลขของระบบแทนจำนวนฐานคู่จะมีผลอย่างมากต่อขนาดของรูปแบบแทนจำนวน เพราะจำนวนหลักที่ใช้ในระบบแทนจำนวนฐานคู่มีจำนวนสูงมากเมื่อเทียบกับรูปแบบแทนจำนวนแบบมีเครื่องหมาย และอีกประการหนึ่ง ระบบปทนจำนวนฐานคู่มีสมบัติความซ้ำซ้อนได้โดยไม่จำเป็นต้องขยายขนาดของชุดตัวเลข ดังนั้น งานวิจัยนี้จึงได้พิจารณาการขยายระดับความซ้ำซ้อนโดยวิธีการพิจารณา การจัดกลุ่มของหลักของตัวเลขในรูปแบบแทนจำนวนแทนการขยายขนาดของชุดตัวเลข

ในเบื้องต้น ผู้วิจัยได้ศึกษาค่าระดับความซ้ำซ้อนของระบบแทนจำนวนฐานคู่ และหากพิจารณากราฟความสัมพันธ์ระหว่างระดับความซ้ำซ้อนและขนาดของตารางแทนจำนวนบนระบบแทนจำนวนฐานคู่ จะได้กราฟที่มีความสัมพันธ์ดังปรากฎในรูปที่ 3.1 ทั้งนี้การพิจารณาขนาดของรูปแบบแทนจำนวนฐานคู่เพื่อความเหมาะสมกับการออกแบบการทำงาน ผู้วิจัยจึงสนใจเฉพาะกรณีที่ กลุ่มของตัวเลขที่อยู่ในรูปแบบกำลังสอง (square) หมายความว่า รูปแบบทานจำนวน หรือ ตารางสองมิติจะเป็นตารางแบบจตุรัสเสมอ ที่มีขนาดต่างๆ กัน



รูปที่ 3.1 กราฟแสดงความสัมพันธ์ระหว่างระดับความซ้ำซ้อน และขนาดของตารางแทนจำนวน

จะเห็นได้ว่าระดับความซ้ำซ้อนมีค่าสูงขึ้นมากเมื่อเทียบกับการขยายขนาดของกลุ่มของตัวเลขที่พิจารณา ดังนั้นในงานวิจัยนี้ จึงมุ่งเน้นที่จะสร้างตัวดำเนินการพื้นฐานการบวก บนระบบแทนช่วงฐานคู่แบบขนาน โดยการจัดกลุ่มของตัวเลขในรูปแบบตารางจตุรัส และสร้างตัวดำเนินการในระดับกลุ่มแทนการดำเนินการในระดับตัวเลขเดี่ยว

สำหรับอัลกอริทึมการบวกแบบขนานนั้น จะดำเนินการโดยอาศัยแนวทางพื้นฐานดังนี้

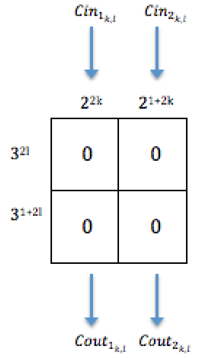
1. ทำการบวกตัวเลขของของทุกคู่พร้อมๆ กัน
2. พิจารณาจัดกลุ่มการคำนวณย่อย และดำเนินการบวก โดยพิจารณา จำนวนทดออกที่เหมาะสมสำหรับให้รองรับจำนวนทดเข้า
3 คำนวณคำตอบสุดท้ายเมื่อพิจารณาจำนวนทดเข้า

3.2 ตัวดำเนินการบวกแบบขนานสำหรับการจัดกลุ่มคำนวณขนาด 2 ×2

ในหัวข้อนี้ เราจะศึกษาความเป็นไปได้ในการออกแบบตัวดำเนินการบวกแบบขนานภายใต้การพิจารณาการคำนวณแบบเป็นกลุ่ม โดยพิจารณา ตารางตัวดำเนินการขนาด 2×2 โดยการบวกแบบขนานสามารถทำงานได้ตามทฤษฎีบทที่ 3.1 ต่อไปนี้

ทฤษฎีบทที่ 3.1 *การดำเนินการบวกบนระบบแทนจำนวนฐานคู่สามารถทำได้โดยใช้ประสิทธิภาพเชิงเวลาเป็นค่าคงที่ เมื่อมีการจัดกลุ่มการบกด้วยตารางขนาด 2 ×2*

พิสูจน์การพิสูจน์ทฤษฎีบทสามารถทำได้โดยการนำเสนออัลกอริทึมสำหรับการดำเนินการบวกบนระบบแทนจำนวนฐานคู่แบบขนาน และการพิสูจน์ว่าอัลกอริทึมการบวกนี้มีประสิทธิภาพเชิงเวลาเป็นค่าคงที่O(1) พร้อมทั้งบทพิสูจน์ความถูกต้องของอัลกอริทึมด้วย



รูปที่3.2 รูปแสดงทิศทางการส่งผ่านตัวทด และข้อมูลขาเข้า และขาออก

แนวทางในการออกแบบอัลกอริทึมคือการพิจารณา การบวกของกลุ่มย่อยขนาด 2 ×2 และทั้งนี้เนื่องจากการคำนวณการทดของระบบแทนจำนวนฐานคู่สามารถทดได้สองทิศทาง โดยมีความแตกต่างกัน คือ การทดในแนวแถวจะเป็นการทดบนเลขฐานสอง แต่การทดในแนวหลักจะเป็นการทดบนเลขฐานสาม เนื่องจากการทดบนเลขฐานสามสามารถทดค่าได้มากกว่าเลขฐานสอง ดังนั้นจึงเลือกใช้ดการทดในแนวหลักดังรูปที่ 3.2 และการคำนวณสามารถทำงานได้ดังอัลกอริทึมต่อไปนี้

Algorithm: Parallel addition in double base number system using 2 × 2 sub-table

Input: Two operands *A* and *B* with $2c \times 2c$

$A=\sum_{i=0,j=0}^{2c-1} a_{i,j} 2^i 3^j$ where $a_{i,j} \in \{0, 1\}$

$B=\sum_{i=0,j=0}^{2c-1} b_{i,j} 2^i 3^j$ where $b_{i,j} \in \{0, 1\}$

Output: The result *Z* with $2c \times 2c$

$Z=\sum_{i=0,j=0}^{2c-1} z_{i,j} 2^i 3^j$ where $z_{i,j} \in \{0, 1\}$

Begin

for all corresponding cells (done in parallel manner)

for all $0 \leq i, j < 2c$ **do**

$x_{i,j} = a_{i,j} + b_{i,j}$ **enddo**

enddo

for each 2 × 2sub-table

for all $0 \leq k,l < c$ **do**

$Sum_{k,l} = \sum_{i=0,j=0}^{1} x_{i,j} 2^i 3^j$

$Cout_{2_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^1 3^2} \right\rfloor$

$Cout_{1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^0 3^2} \right\rfloor - 2Cout_{2_{k,l}}$ **enddo**

for all $0 \leq k,l < c$ **do**

$Cin_{k,l} = Cout_{1_{k,l}} \times 2^0 3^0 + Cout_{2_{k,l}} \times 2^1 3^0$

$Sum_{k,l} = \sum_{i=0,j=0}^{1} x_{i,j} 2^i 3^j$

$Cout_{k,l} = 2^0 3^2 Cout_{1_{k,l}} + 2^1 3^2 Cout_{2_{k,l}}$

$Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$ **enddo**

End

การทำงานของอัลกอริทึม ประกอบด้วย สี่ ขั้นตอนการทำงานดังนี้

ขั้นตอนที่ 1

ดำเนินการบวกตัวตั้ง A และตัวบวกB ซึ่งอยู่ในรูปแบบตารางสองมิติเข้าด้วยกันโดยอาศัยหลักวิธีการซ้อนทับกันของรูปแบบแทนจำนวน โดยที่ตารางผลบวกนั้นสามารถอาศัยกฎการบวกดังนี้คือ เซลล์ศูนย์ซ้อนทับเซลล์ศูนย์ได้เซลล์ศูนย์ เซลล์หนึ่งซ้อนทับเซลล์ศูนย์ได้เซลล์หนึ่ง เซลล์ศูนย์ซ้อนทับเซลล์หนึ่งได้เซลล์หนึ่ง และเซลล์หนึ่งซ้อนทับเซลล์หนึ่งได้เซลล์สอง

ขั้นตอนที่ 2

แบ่งตารางผลรวมที่ได้จากขั้นตอนที่ 1 ออกเป็นตารางย่อยๆ ตารางละ 2×2 และทำการคำนวณ ตัวทดออก ของแต่ละตารางย่อย แบบขนาน โดยอาศัยสมการต่างๆ ดังนี้

สำหรับทุกค่าของ $0 \leq k,\mathrm{l} < c$

$$Sum_{k,l} = \sum_{\mathrm{i}=0,\mathrm{j}=0}^{1} x_{i,j} 2^{\mathrm{i}} 3^{\mathrm{j}}$$
$$Cout_{2_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^1 3^2} \right\rfloor$$
$$Cout_{1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^0 3^2} \right\rfloor - 2Cout_{2_{k,l}}$$

โดยที่

$Sum_{k,l}$คือตารางย่อยของผลลัพธ์ Z เป็นตารางย่อยๆขนาด 2×2

$Cout_{2_{k,l}}$คือค่าของตัวทดออกของตารางย่อยในหลักที่สอง ซึ่งมีค่าได้เพียง 0 หรือ 1 เท่านั้น

$Cout_{1_{k,l}}$คือค่าของตัวทดออกของตารางย่อยในหลักที่หนึ่ง ซึ่งมีค่าได้เพียง 0 หรือ 1 เท่านั้น

ขั้นตอนที่ 3

เป็นขั้นตอนการส่งผ่านตัวทดจากตารางย่อยตารางหนึ่งไปสู่ตารางย่อยอีกตารางหนึ่งซึ่งมีรูปแบบคือตัวทดออกจากตารางย่อยด้านบนเป็นตัวทดเข้าของตารางย่อยที่อยู่ด้านล่างตามลำดับ ตามสมการ ดังนี้

$$Cin_{k,l} = Cout_{1_{k,l}} \times 2^0 3^0 + Cout_{2_{k,l}} \times 2^1 3^0$$
$$Sum_{k,l} = \sum_{i=0,j=0}^{1} x_{i,j} 2^i 3^j$$
$$Cout_{k,l} = 2^0 3^2 Cout_{1_{k,l}} + 2^1 3^2 Cout_{2_{k,l}}$$

$Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$ โดยที่ $Result_{k,l}$คือค่าของตารางผลลัพธ์ย่อยแต่ละตาราง

ขั้นตอนที่4

ทำการสร้างตารางผลลัพธ์ย่อยที่ได้เข้ากับรูปแบบแทนจำนวนในรูปแบบตาราง 2 มิติ หลังจากนั้น จึงนำตารางผลลัพธ์ย่อยที่ได้มาประกอบกันขึ้นเป็นตารางผลลัพธ์

การพิสูจน์ความถูกต้องของอัลกอริทึม

จะสังเกตเห็นว่าแนวการทำงานหลักของอัลกอริทึมคือ ทุกๆตารางย่อยขนาด 2×2 ทำงานพร้อมกัน โดยการพิสูจน์ความถูกต้องจะแบ่งการพิจารณาออกเป็นสามส่วนดังนี้คือ

1) การพิจารณาตัวทดออก ว่าอยู่ในขอบเขตที่กำหนดได้เสมอ
2) ผลบวกที่หักตัวทดออกแล้วสามารถรองรับตัวทดเข้าได้เสมอ
3) ผลการบวกได้คำตอบที่ถูกต้องเสมอ

1) การพิจารณาตัวทดออกนั้น จากขั้นตอนที่สองจะพบว่า $0 \leq Sum_{k,l} \leq 24$ จาก

$Cout_{2_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^1 3^2} \right\rfloor$ ดังนั้น จะได้ว่า $0 \leq Cout_{2_{k,l}} \leq 1$ และ

$Cout_{1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^0 3^2} \right\rfloor - 2\mathrm{Cout}_{2_{k,l}}$ ดังนั้น จะได้ว่า $0 \leq Cout_{1_{k,l}} \leq 1$

นั่นหมายความว่า ตัวทดออกทั้งสองตัว เป็นได้เฉพาะ 0 หรือ 1 เท่านั้น และจะมีเพียงตัวใดตัวหนึ่งที่มีค่าเป็น 1 เท่านั้น

2) จากค่าของตัวทดออกทั้งสองที่เป็นได้เฉพาะ 0 หรือ 1 และมีเพียงตัวใดตัวหนึ่งที่มีค่าเป็น 1 เท่านั้นจะมีผลทำให้

$$0 \leq Sum_{k,l} - 2^1 3^2 Cout_{2_{k,l}} - 2^0 3^2 Cout_{1_{k,l}} \leq 8 \text{ เสมอ}$$

นั่นหมายความว่า ผลลัพท์จากการหักตัวทดออกไป จะมีค่าไม่เกินไปกว่า 8 แน่นอน

หากพิจารณา ตัวทดเข้าที่มีได้มากสุดคือทดที่ตำแหน่ง $Cout_{2_{k-1,l}}$ ดังนั้น

$$\begin{aligned} Result_{k,l} &= Sum_{k,l} + Cin_{k,l} - Cout_{k,l} \\ &= Sum_{k,l} - 2^1 3^2 Cout_{2_{k,l}} - 2^0 3^2 Cout_{1_{k,l}} + Cin_{k,l} \\ &\leq 8 + Cin_{k,l} \\ &= 8 + 2^0 3^0 Cout_{1_{k-1,l}} + 2^1 3^0 Cout_{2_{k-1,l}} \\ &\leq 8 + 2 = 10 \end{aligned}$$

และ

$$\begin{aligned} Result_{k,l} &= Sum_{k,l} + Cin_{k,l} - Cout_{k,l} \\ &= Sum_{k,l} - Cout_{k,l} + Cin_{k,l} \\ &\geq 0 + Cin_{k,l} \\ &\geq 0 \end{aligned}$$

ดังนั้น เราสามารถสรุปได้ว่า $0 \leq Result_{k,l} \leq 10$ และสามารถเขียนให้อยู่ในรูปแบบแทนจำนวนฐานคู่ในตารางขนาด 2×2 ได้เสมอ

3) การพิสูจน์ให้เห็นว่า คำตอบที่ได้เป็นผลบวก สามารถแสดงได้ดังนี้

พิจารณาค่าของตารางแทนจำนวน *Sum* ที่ได้จากการนำตัวถูกดำเนินการสองตัวขนาด 2c×2c มาซ้อนทับกันดังนี้

$$
\begin{aligned}
\sum_{i=0,j=0}^{2c-1} a_{i,j}2^i3^j + \sum_{i=0,j=0}^{2c-1} b_{i,j}2^i3^j &= Sum_{0,0} + Sum_{1,0} + Sum_{2,0} + \ldots + Sum_{c-1,0} \\
&\quad + Sum_{0,1} + Sum_{1,1} + Sum_{2,1} + \ldots + Sum_{c-1,1} + \ldots \\
&\quad + Sum_{c-1,c-1} \\
&= (Sum_{0,0} + Cin_{0,0} - Cout_{0,0}) + 2^0 3^2 (Sum_{1,0} + Cin_{1,0} \\
&\quad - Cout_{1,0}) + \ldots + 2^0 3^{2c} Cin_{c,0} + \ldots \\
&\quad + 2^{2(c-1)} 3^{2(c-1)} (Sum_{c-1,c-1} + Cin_{c-1,c-1} \\
&\quad - Cout_{c-1,c-1}) + 2^{2c} 3^{2(c-1)} Cin_{c,c-1} \\
&= (Sum_{0,0} + 0 - Cout_{0,0}) + 2^0 3^2 (Sum_{1,0} + Cin_{1,0} \\
&\quad - Cout_{1,0}) + \ldots + 2^0 3^{2c} Cin_{c,0} + \ldots \\
&\quad + 2^{2(c-1)} 3^{2(c-1)} (Sum_{c-1,c-1} + Cin_{c-1,c-1} \\
&\quad - Cout_{c-1,c-1}) + 2^{2(c-1)} 3^{2c} Cin_{c,c-1} \\
&= Result_{0,0} + 2^0 3^1 Result_{1,0} + \ldots \\
&\quad + 2^{2(c-1)} 3^{2(c-1)} Result_{c-1,c-1} + \sum_{k=0,l=0}^{c} Cin_{c,l} \\
&= Result + \sum_{l=0}^{c-1} 2^{2l} 3^{2c} Cin_{c,l} \\
&= \sum_{i=0,j=0}^{2c-1} z_{i,j} 2^i 3^j + \sum_{l=0}^{c-1} 2^{2l} 3^{2c} Cin_{c,l}
\end{aligned}
$$

ตัวอย่างที่ 3.2 *การบวกกันของ 177 และ 64 ในระบบแทนจำนวนฐานคู่*

วิธีทำ เราสามารถเขียนจำนวน 177 และ 64 ในรูปแบบแทนจำนวนฐานคู่ในตารางขนาด 4 × 4 ได้ดังรูปที่ 3.3

	2^0	2^1	2^2	2^3
3^0	1	0	1	1
3^1	0	0	0	1
3^2	0	0	0	0
3^3	1	0	0	0

64

	2^0	2^1	2^2	2^3
3^0	1	1	1	1
3^1	1	1	1	1
3^2	0	0	1	0
3^3	1	1	0	0

177

รูปที่3.3 แสดงรูปแบบแทนจำนวนของตัวถูกดำเนินการ 64 และ 177

จากขั้นตอนแรกคือการนำตัวถูกดำเนินการที่อยู่ในรูปแบบตารางสองมิติด้านบนมาซ้อนทับกัน จะได้ตารางผลบวก $x_{i,j}$ ดังรูปที่ 3.4 ต่อไปนี้

	2^0	2^1	2^2	2^3
3^0	2	1	2	2
3^1	1	1	1	2
3^2	0	0	1	0
3^3	2	1	0	0

รูปที่3.4 แสดงรูปแบบแทนจำนวนผลบวกที่ได้จากการซ้อนทับกันของตัวถูกดำเนินการ

หลังจากนั้นทำการแบ่งตารางผลบวกที่ได้ออกเป็นตารางย่อย ขนาด 2×2 จะได้ผลลัพธ์ดังรูปที่ 3.5

2	1
1	1

$Sum_{0,0}$

2	2
1	2

$Sum_{0,1}$

0	0
2	1

$Sum_{1,0}$

1	0
0	0

$Sum_{1,1}$

รูปที่3.5 แสดงตารางย่อยขนาด 2×2 ทั้งหมดที่ได้จากการแบ่งตาราง *Sum*

จากนั้นทำการคำนวณหาค่าตัวทดออกจากสมการ

$$Cout_{2_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^1 3^2} \right\rfloor$$

$$Cout_{1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^0 3^2} \right\rfloor - 2Cout_{2_{k,l}}$$

จะได้ผลลัพธ์ดังนี้คือ

$$Sum_{0,0} = 13, Cout_{2_{0,0}} = 0, Cout_{1_{0,0}} = 1$$

$$Sum_{0,1} = 21, Cout_{2_{0,1}} = 1, Cout_{1_{0,1}} = 0$$

$$Sum_{1,0} = 12, Cout_{2_{1,0}} = 0, Cout_{1_{1,0}} = 1$$

$$Sum_{1,1} = 1, Cout_{2_{1,1}} = 0, Cout_{1_{1,1}} = 0$$

ขั้นตอนต่อไปเป็นขั้นตอนการส่งผ่านตัวทดซึ่งเราจะนำตัวทดออกที่ได้จากตารางย่อยทางด้านบนมาเป็นตัวทดเข้าของตารางทางด้านล่าง ซึ่งเป็นการทำงานแบบขนาน

จากสมการ $Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$, $Cin_{k,l} = Cout_{1_{k-1,l}} \times 2^0 3^0 + Cout_{2_{k-1,l}} \times 2^1 3^0$ และ $Cout_{k,l} = 2^0 3^2 Cout_{1_{k,l}} + 2^1 3^2 Cout_{2_{k,l}}$

จะได้ค่าของตาราง Result ย่อยดังต่อไปนี้คือ

$$Result_{0,0} = 4$$
$$Result_{0,1} = 3$$
$$Result_{1,0} = 4$$
$$Result_{1,1} = 3$$

ขั้นตอนสุดท้ายคือการนำค่าของตาราง *Result* ย่อยแต่ละตารางมาสร้างเป็นรูปแบบแทนจำนวนตาราง 2 มิติ ซึ่งจะได้รูปแบบแทนจำนวนของตารางผลลัพธ์ดังนี้คือ

	2^0	2^1	2^2	2^3
3^0	1	0	0	0
3^1	1	0	1	0
3^2	1	0	0	0
3^3	1	0	1	0

รูปที่3.6 แสดงตาราง *Result* ที่ได้จากการดำเนินการบวกแบบขนานของตัวถูกดำเนินการ 177 และ 64

จากตาราง *Result* ที่ได้เมื่อนำมาคำนวณแล้วจะได้ $2^0 3^0 + 2^0 3^1 + 2^2 3^1 + 2^0 3^2 + 2^0 3^3 + 2^2 3^3 = 160$ และเมื่อนำมารวมกับ ตัวทดส่วนเกินซึ่งเกิดจาก $Cout_{1_{1,0}}$ ซึ่งมีค่า $2^0 3^4 = 81$ จะได้ 241 ซึ่งมีค่าเท่ากับผลรวมของตัวถูกดำเนินการ 177 + 64 □

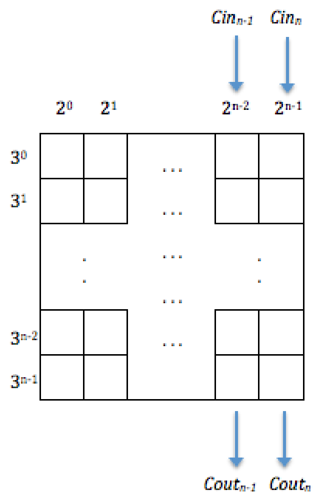
3.3 ตัวดำเนินการบวกแบบขนานสำหรับการจัดกลุ่มคำนวณขนาด n×n โดยที่ n $\geq$ 2

ในหัวข้อนี้เราจะทำการศึกษาความเป็นไปได้ของการดำเนินการบวกบนระบบแทนจำนวนฐานคู่แบบขนานโดยการพิจารณาตารางกลุ่มการคำนวณซึ่งมีขนาดในรูปแบบทั่วไป ซึ่งสามารถอธิบายโดยทฤษฎีบทที่ 3.3 ดังต่อไปนี้

ทฤษฎีบทที่ 3.3 *การดำเนินการบวกบนระบบแทนจำนวนฐานคู่สามารถทำได้โดยใช้ประสิทธิภาพเชิงเวลาเป็นค่าคงที่ เมื่อมีการจัดกลุ่มการบกด้วยตารางขนาด* $n \times n$ โดยที่ $n \geq 2$

พิสูจน์สามารถพิสูจน์ได้โดยการนำเสนออัลกอริทึมสำหรับการดำเนินการบวกบนระบบแทนจำนวนฐานคู่ ซึ่งมีประสิทธิภาพเชิงเวลาเป็น O(1) พร้อมทั้งบทพิสูจน์ความถูกต้องของอัลกอริทึมเมื่อ $n \geq 2$ดังต่อไปนี้

ในที่นี้จะกล่าวถึงอัลกอริทึมการดำเนินการบวกแบบขนานโดยมีขนาดของตารางย่อยอยู่ในรูปทั่วไปซึ่งมีขนาดมากกว่าหรือเท่ากับสอง ซึ่งจะสังเกตเห็นว่าการส่งผ่านตัวทดนั้นจะมีการส่งผ่านตัวทดเพียงที่คอลัมน์สุดท้ายและรองสุดท้ายของตารางย่อยเท่านั้น โดยจะกล่าวถึงบทพิสูจน์ของความถูกต้องของผลลัพธ์ที่ได้จากอัลกอริทึม และบทพิสูจน์ของสมการต่างๆโดยอาศัยวิธีการอุปนัยทางคณิตศาสตร์(Mathematic induction) เพื่อพิสูจน์ว่าสำหรับ n ใดๆที่มากกว่าหรือเท่ากับสองแล้ว จะทำให้อัลกอริทึมที่ได้ ได้คำตอบที่ถูกต้อง



รูปที่3.7 แสดงข้อมูลขาเข้า ข้อมูลขาออก และทิศทางการส่งผ่านตัวทดเมื่อขนาดตารางมีค่ามากกว่าหรือเท่ากับสอง

Algorithm: Parallel addition in double base number system using n×n sub-table

Input: Two operands *A* and *B* with n*c*×n*c*

$A=\sum_{i=0,j=0}^{nc-1} a_{i,j} 2^i 3^j$ where $a_{i,j} \in \{0, 1\}$

$B=\sum_{i=0,j=0}^{nc-1} b_{i,j} 2^i 3^j$ where $b_{i,j} \in \{0, 1\}$

Output: The result Z with $nc \times nc$

$Z = \sum_{i=0,j=0}^{nc-1} z_{i,j} 2^i 3^j$ where $z_{i,j} \in \{0, 1\}$

Begin

for all corresponding cells (done in parallel manner)

for all $0 \leq i, j < nc$ **do**

$x_{i,j} = a_{i,j} + b_{i,j}$

enddo

enddo

for each $n \times n$ sub-table

for all $0 \leq k,l < c$ **do**

$Sum_{k,l} = \sum_{i=0,j=0}^{n-1} x_{i,j} 2^i 3^j$

$Cout_{n_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^{n-1}3^n} \right\rfloor$

$Cout_{n-1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^{n-2}3^n} \right\rfloor - 2Cout_{n_{k,l}}$

enddo

forall $0 \leq k,l < c$ **do**

$Cin_{k,l} = Cout_{n-1_{k-1,l}} \times 2^{n-2}3^0 + Cout_{n_{k-1,l}} \times 2^{n-1}3^0$

$Sum_{k,l} = \sum_{i=0,j=0}^{n-1} x_{i,j} 2^i 3^j$

$Cout_{k,l} = 2^{n-2}3^n Cout_{n-1_{k,l}} + 2^{n-1}3^n Cout_{n_{k,l}}$

$Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$

enddo

End

ข้อมูลขาเข้า: $A = \sum_{i=0,j=0}^{cn-1} a_{i,j} 2^i 3^j$ และ

$B = \sum_{i=0,j=0}^{cn-1} b_{i,j} 2^i 3^j$ และ $a_{i,j}, b_{i,j} \in \{0, 1\}$

ข้อมูลขาออก: $Z = \sum_{i=0,j=0}^{cn-1} z_{i,j} 2^i 3^j$ และ $z_{i,j} \in \{0, 1\}$

หมายเหตุ: ข้อมูลขาเข้า และข้อมูลขาออกอยู่ในรูปแบบตารางสองมิติขนาด nc×nc เมื่อ c เป็นจำนวนเต็มบวกใดๆ และ c คือขนาดของตารางย่อย

i,jเป็นคู่อันดับเมื่อเทียบกับตารางขนาด nc×nc

k,lเป็นคู่อันดับของตารางย่อยเมื่อเทียบกับตารางขนาด nc×nc

สำหรับตัวทดเข้าและตัวทดออกของตารางย่อยแต่ละตารางเราจะกำหนดให้มีตัวทดเข้าและตัวทดออกที่ตำแหน่งคอลัมน์ที่ n และ n-1 เท่านั้น

การทำงานของอัลกอริทึมประกอบด้วยสี่ขั้นตอนการทำงานดังต่อไปนี้

ขั้นตอนที่ 1

กระทำการบวกตารางแทนจำนวนของตัวถูกดำเนินการเข้าด้วยกัน จะได้ $x_{i,j} = a_{i,j} + b_{i,j}$ โดยที่ $x_{i,j}$ เป็นค่าของเซลล์แต่ละเซลล์ของตาราง *Sum* โดยอาศัยหลักการบวกเดียวกันกับอัลกอริทึมในหัวข้อ 3.1

ขั้นตอนที่ 2

ทำการคำนวณค่า $Cout_{n_{k,l}}$ และ $Cout_{n-1_{k,l}}$ ของแต่ละตารางย่อยขนาด n×n แบบขนานดังสมการต่อไปนี้

$$Sum_{k,l} = \sum_{i=0,j=0}^{n-1} x_{i,j} 2^i 3^j$$

$$Cout_{n_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^{n-1}3^n} \right\rfloor$$

$$Cout_{n-1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^{n-2}3^n} \right\rfloor - 2Cout_{n_{k,l}}$$

ขั้นตอนที่ 3

เป็นขั้นตอนการส่งผ่านตัวทดโดยจะส่งผ่านตัวทดสำหรับคอลัมน์สุดท้ายของตารางย่อยและคอลัมน์รองสุดท้ายเท่านั้น

$$Cin_{k,l} = Cout_{n-1_{k-1,l}} \times 2^{n-2}3^0 + Cout_{n_{k-1,l}} \times 2^{n-1}3^0$$

$$Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$$

โดยที่ $Cout_{k,l} = 2^{n-2}3^n Cout_{n-1_{k,l}} + 2^{n-1}3^n Cout_{n_{k,l}}$

ขั้นตอนที่ 4

นำค่า *Result* ของตารางย่อยแต่ละตารางมาสร้างเป็นรูปแบบแทนจำนวนแบบตาราง และนำตารางย่อยๆเหล่านั้นมาประกอบกันเป็นตารางผลลัพธ์สมบูรณ์

บทพิสูจน์ความถูกต้องของอัลกอริทึม

การพิสูจน์ให้เห็นว่า คำตอบที่ได้เป็นผลบวก สามารถแสดงได้ดังนี้

พิจารณาค่าของตารางแทนจำนวน *Sum* ที่ได้จากการนำตัวถูกดำเนินการสองตัวขนาด nc×nc มาซ้อนทับกันดังนี้

$$\sum_{\mathrm{i=0,j=0}}^{\mathrm{nc-1}} a_{i,j} 2^{\mathrm{i}} 3^{\mathrm{j}} + \sum_{\mathrm{i=0,j=0}}^{\mathrm{nc-1}} b_{i,j} 2^{\mathrm{i}} 3^{\mathrm{j}} = Sum_{0,0} + Sum_{1,0} + Sum_{2,0} + \ldots + Sum_{c-1,0} + Sum_{0,1} + Sum_{1,1} + Sum_{2,1} + \ldots + Sum_{c-1,1} + \ldots + Sum_{c-1,c-1}$$

$$= (Sum_{0,0} + Cin_{0,0} - Cout_{0,0}) + 2^0 3^n (Sum_{1,0} + Cin_{1,0} - Cout_{1,0}) + \ldots + 2^0 3^{nc} Cin_{c,0} + \ldots + 2^{n(c-1)} 3^{n(c-1)} (Sum_{c-1,c-1} + Cin_{c-1,c-1} - Cout_{c-1,c-1}) + 2^{nc} 3^{n(c-1)} Cin_{c,c-1}$$

$$= (Sum_{0,0} + 0 - Cout_{0,0}) + 2^0 3^n (Sum_{1,0} + Cin_{1,0} - Cout_{1,0}) + \ldots + 2^0 3^{nc} Cin_{c,0} + \ldots + 2^{n(c-1)} 3^{n(c-1)} (Sum_{c-1,c-1} + Cin_{c-1,c-1} - Cout_{c-1,c-1}) + 2^{n(c-1)} 3^{nc} Cin_{c,c-1}$$

$$= Result_{0,0} + 2^0 3^n Result_{1,0} + \ldots + 2^{n(c-1)} 3^{n(c-1)} Result_{c-1,c-1} + \sum_{k=0,l=0}^{c-1} 2^{nl} 3^{nc} Cin_{c,l}$$

$$= Result + \sum_{l=0}^{c-1} 2^{nl} 3^{nc} Cin_{c,l}$$

$$= \sum_{i=0,j=0}^{nc-1} z_{i,j} 2^i 3^j + \sum_{l=0}^{c-1} 2^{nl} 3^{nc} Cin_{c,l}$$

จะเห็นได้ว่าหากตารางย่อยในแถวล่างสุดมีการส่งตัวทดออกจะทำให้เกิดปัญหาการเกิดตัวทดส่วนเกิน(overflow)ขึ้น

บทพิสูจน์ความถูกต้องของสมการและค่าของตัวแปรต่างๆที่อยู่ภายในอัลกอริทึม

เพื่อความง่ายต่อความเข้าใจ กำหนดให้สัญลักษณ์ $[1]_n$ แทนค่าของตารางแทนจำนวนที่ประกอบด้วยเซลล์หนึ่งทั้งหมด และ $[2]_n$ แทนค่าของตารางแทนจำนวนที่ประกอบด้วยเซลล์สองทั้งหมด โดยค่าต่างๆสามารถเขียนให้อยู่ในรูปประพจน์ทางคณิตศาสตร์ได้ดังนี้

$[1]_n \quad = (2^n - 1)(3^n - 1)/2$

$[2]_n \quad = (2^n - 1)(3^n - 1)$

$Cout_n \quad = 2^{n-1}3^n$

$Cout_{n-1} = 2^{n-2}3^n$

$Cout_{n-2} = 2^{n-3}3^n$

$Cin_n \quad = 2^{n-1}$

$Cin_{n-1} \quad = 2^{n-2}$

หมายเหตุ ตัวแปรแต่ละตัวนั้นเป็นตัวแปรที่ใช้แทนสำหรับตารางย่อย n×n ของตารางที่ k,l ใดๆ

สมการที่ 1 $Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$

ฟังก์ชันที่ 1 $Cout_{n_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^{n-1}3^n} \right\rfloor$

ฟังก์ชันที่ 2 $Cout_{n-1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^{n-2}3^n} \right\rfloor - 2Cout_{n_{k,l}}$

เราต้องทำการพิสูจน์ว่าสมการทั้งหมดที่ถูกใช้ในอัลกอริทึมสำหรับการบวกแบบขนานบนระบบแทนจำนวนฐานคู่ในรูปแบบทั่วไปมีความถูกต้องโดย ไม่ขึ้นอยู่กับขนาดของตารางย่อย n×n โดยมีตัวทดเข้าคือ Cin_nหรือ Cin_{n-1} *หรือทั้งคู่ และ มีตัวทดออกคือ* $Cout_n$หรือ $Cout_n$หรือทั้งคู่ เราต้องทำการพิสูจน์ต่างๆดังต่อไปนี้

กรณีที่1 $0 \leq Sum_{k,l} \leq 2^{n-2}3^n - 1$ ซึ่งไม่มีตัวทดออก

ต้องพิสูจน์ว่า ถึงแม้ว่าจะมีตัวทดเข้าทั้ง Cin_n*และ* Cin_{n-1}แล้วตารางย่อยแต่ละตารางยังอยู่ในรูปแบบแทนจำนวนฐานคู่อยู่ โดย$Cout_{n-1}$ *และ* $Cout_n$*จากฟังก์ชันที่* 1 *และ* 2*มีค่าเป็น* 0 *เมื่อ* $Sum_{k,l}$มีค่าตั้งแต่ 0 จนถึง $2^{n-2}3^n - 1$ ต้องทำการพิสูจน์ดังต่อไปนี้

$Cin_n + Cin_{n-1} + 2^{n-2}3^n - 1 \leq [1]_n;$

$2^{n-1} + 2^{n-2} + 2^{n-2}3^n - 1 \leq (2^n - 1)(3^n - 1)/2$

$2^{n-2}(3 + 3^n) - 1 \leq (2^{n-1} - 0.5)(3^n - 1)$

ขั้นตอนฐาน กรณีเมื่อ n=2 จะได้

$2^{0}(3 + 3^{2}) – 1 \leq (2^{1}– 0.5) (3^{2}– 1)$

$11 \leq 12$ ซึ่งเป็นจริง

ขั้นตอนอุปนัย โดยแทน n ด้วย n+1

$2^{n-1}(3 + 3^{n+1}) – 1 \leq (2^{n}– 0.5) (3^{n+1}– 1)$

$2^{n-1}3 + 2^{n-1}3^{n+1}– 1 \leq 2^{n}3^{n+1}– 2^{n}– (0.5)3^{n+1}+ 0.5$

$2^{n-1}3 + 2^{n-1}3^{n+1}– 1 \leq 2^{n}3^{n+1}– (2)3^{n+1} < 2^{n}3^{n+1}– 2^{n}– (0.5)3^{n+1}+ 0.5 < (2^{n}– 0.5) (3^{n+1}– 1)$

จะเห็นได้ว่าอสมการข้างต้นเป็นจริง

กรณีที่2 $2^{n-2}3^{n} \leq Sum_{k,l} \leq 2^{n-1}3^{n} – 1$ ซึ่งมีตัวทดออกเพียงตัวเดียวคือ $Cout_{n-1}$

และหาก $Cout_{n-1}$ *จากฟังก์ชันที่2 มีค่าเท่ากับ* 1 *และ* $Cout_{n}$จากฟังก์ชันที่ 1มีค่าเท่ากับ 0 เมื่อ $Sum_{k,l}$มีค่ามากกว่าหรือเท่ากับ $2^{n-2}3^{n}$ แต่มีค่าไม่ถึง $2^{n-1}3^{n}$ เราจะต้องทำการพิสูจน์ว่าอสมการสองอสมการดังต่อไปนี้เป็นจริง

ขั้นแรก เราต้องทำการพิสูน์ว่า ค่าตัวทดออก $Cout_{n-1}$ มีค่าเพียงพอที่จะทำให้ตารางย่อยในช่วงนี้มีค่าอยู่ในรูปแบบตารางแทนจำนวนฐานคู่

$(Cout_{n} – 1) – Cout_{n-1}+ Cin_{n} + Cin_{n-1} \leq [1]_{n}$;

$2^{n-1}3^{n} – 1– 2^{n-2}3^{n} + 2^{n-1} + 2^{n-2} \leq (2^{n} – 1)(3^{n}– 1)/2$

$2^{n}3^{n}– 2 – 2^{n-1}3^{n} + 2^{n} + 2^{n-1} \leq 2^{n}3^{n}– 2^{n}– 3^{n} + 1$

$2^{n-1}3^{n}– 3^{n} – 2^{n+1}– 2^{n-1} + 3 \geq 0$

ขั้นตอนฐาน กรณีเมื่อ n=2 จะได้

$2^{1}3^{2}– 3^{2} – 2^{3}– 2^{1}+ 3 = 2 \geq 0$ ซึ่งเป็นจริง

ขั้นตอนอุปนัย โดยแทน n ด้วย n+1

$2^{n}3^{n+1}– 3^{n+1} – 2^{n+2}– 2^{n}+ 3 \geq 0$

$3^{n+1}(2^{n} – 1)\geq 3^{n+1}(2^{n} – 1)– (3)2^{n} \geq 0$

$2^{n}3^{n+1}– 3^{n+1} – 2^{n+2}– 2^{n}+ 3 \geq 0$

จะเห็นได้ว่าอสมการข้างต้นเป็นจริง

ค่าของ $Cout_{n-1}$จะต้องมีค่าน้อยกว่าหรือเท่ากับ $[1]_{n}$ เพื่อที่จะทำให้ค่าของตัวทดออกมีค่ามากกว่าค่าของตารางที่สนใจอยู่ ในกรณีที่ $Sum_{k,l}$มีค่ามากกว่า $[1]_{n}$ แต่ไม่ถึง $2^{n-1}3^{n}$

$Cout_{n-1} \leq [1]_n$ โดยสามารถพิสูจน์ได้ดังนี้คือ

$2^{n-2}3^n \leq (2^n - 1)(3^n - 1)/2$

$2^{n-1}3^n \leq 2^n3^n - 2^n - 3^n + 1$

$2^{n-1}3^n - 2^n - 3^n + 1 \geq 0$

ขั้นตอนฐาน กรณีเมื่อ n=2 จะได้

$2^1 3^2 - 2^2 - 3^2 + 1 = 6 \geq 0$ซึ่งเป็นจริง

ขั้นตอนอุปนัย โดยแทน n ด้วย n+1

$2^n 3^{n+1} - 2^{n+1} - 3^{n+1} + 1 \geq 0$

$2^n 3^{n+1} - 3^{n+1} = 3^{n+1}(2^n - 1) \geq 3^{n+1}(2^n - 1) - 2^{n+1} + 1 \geq 0$

จะเห็นได้ว่าอสมการข้างต้นเป็นจริง

กรณีที่3 $2^{n-1}3^n \leq Sum_{k,l} \leq [2]_n$ ซึ่งอาจมีตัวทดออกเพียงตัวเดียวคือ $Cout_n$ หรือมีตัวทดออกทั้งสองตัว คือ $Cout_n$ และ $Cout_{n-1}$ก็ได้ แต่เราจะพิสูจน์โดยคิดเสมือนว่ามีคัวทดออก $Cout_n$เพียงตัวเดียวก็สามารถทำให้รูปแบบของตารางย่อยอยู่ในรูปแบบแทนจำนวนฐานคู่ได้

ค่าของ $Cout_n$จากฟังก์ชันที่ 1 จะมีค่าเป็น 1 เมื่อ $Sum_{k,l}$มีค่ามากกว่าหรือเท่ากับ $2^{n-1}3^n$ ในกรณีที่ต้องการพิสูจน์ว่า ตารางย่อย $Result_{k,l}$อยู่ในรูปแบบของตารางแทนจำนวนในระบบแทนจำนวนฐานคู่ เราต้องทำการพิสูจน์ว่าอสมการสองอสมการดังต่อไปนี้เป็นจริงในกรณีที่ ตัวทดออกมีเพียงตัวเดียวคือ $Cout_n$และค่า $Sum_{k,l}$มีค่าสูงสุดเราต้องทำการพิสูจน์ว่า ค่าตัวทดออก Cout_nมีค่าเพียงพอที่จะทำให้ ตารางย่อย $Result_{k,l}$อยู่ในรูปแบบตารางแทนจำนวนฐานคู่

$[2]_n - Cout_n + Cin_n + Cin_{n-1} \leq [1]_n$โดยสามารถพิสูจน์ได้ดังนี้คือ

$(2^n - 1)(3^n - 1) - 2^{n-1}3^n + 2^{n-1} + 2^{n-2} \leq (2^n - 1)(3^n - 1)/2$

$2^{n-1}3^n - 2^{n-1} - 2^{n-2} \geq 2^{n-1}3^n - 2^{n-1} - 0.5(3^n) + 0.5$

$2^{n-1} - 3^n + 1 \leq 0$

ขั้นตอนฐาน กรณีเมื่อ n=2 จะได้

$2^1 - 3^2 + 1 = -6 \leq 0$ซึ่งเป็นจริง

ขั้นตอนอุปนัย โดยแทน n ด้วย n+1

$2^n - 3^{n+1} + 1 \leq 0$

จะเห็นได้ว่าอสมการข้างต้นเป็นจริง

ค่าของ $Cout_n$ ต้องมีค่าน้อยกว่าหรือเท่ากับ $2^{n-1}3^n$ เพื่อที่จะทำให้ตัวทดออกมีค่าไม่มากกว่าค่าของตารางย่อย $Sum_{k,l}$ ที่มีอยู่

$Cout_n \leq 2^{n-1}3^n$ จะต้องพิสูจน์ว่า

$2^{n-1}3^n \leq 2^{n-1}3^n$

จะเห็นว่าอสมการข้างต้นเป็นจริง

จากฟังก์ชันที่ 1 เราต้องทำการพิสูจน์ว่าค่าของ $Cout_n$ ในฟังก์ชันจะต้องมีค่าเท่ากับ 0 หรือ 1 ค่าใดค่าหนึ่งเท่านั้น

$[2]_n / Cout_n < 2$ โดยสามารถพิสูจน์ได้ดังนี้คือ

$(2^n - 1)(3^n - 1) / (2^{n-1}3^n) < 2$

$(2^n3^n - 2^n - 3^n + 1) / (2^{n-1}3^n) < 2$

$2 - 2 / 3^n - 1 / 2^{n-1} + 1 / (2^{n-1}3^n) < 2$

ขั้นตอนฐาน กรณีเมื่อ n=2 จะได้

$2 - 2 / 3^2 - 1 / 2^1 + 1 / (2^13^2) = 2 - 2/9 - 1/2 + 1/18 = 12/9 < 2$ ซึ่งเป็นจริง

ขั้นตอนอุปนัย โดยแทน n ด้วย n+1

$2 - 2 / 3^{n+1} - 1 / 2^n + 1 / (2^n3^{n+1}) < 2$

$2 - (2^{n+1} + 3^{n+1} - 1)/(2^n3^{n+1}) < 2$

จะเห็นว่าอสมการข้างต้นเป็นจริง

จากฟังก์ชันที่ 2 เมื่อค่าของตาราง $Sum_{k,l}$ มีค่ามากที่สุด ค่าของ $\left\lfloor \frac{\mathrm{Sum}_{k,l}}{2^{n-2}3^n} \right\rfloor$ จากฟังก์ชันที่ 2 ต้องมีค่าน้อยกว่า 4 เพื่อ กำหนดให้ค่า ของ $Cout_{n-1}$ มีค่าเป็น 0 หรือ 1 เท่านั้น

$[2]_n / Cout_{n-1} < 4$ โดยสามารถพิสูจน์ได้ดังนี้คือ

$(2^n - 1)(3^n - 1) / (2^{n-1}3^n) < 4$

$(2^n3^n - 2^n - 3^n + 1) / (2^{n-2}3^n) < 4$

$4 - 4 / 3^n - 1 / 2^{n-2} + 1 / (2^{n-2}3^n) < 4$

ขั้นตอนฐาน กรณีเมื่อ n=2 จะได้

$4 - 4/3^2 - 1/2^0 + 1/(2^0 3^2) = 4 - 4/9 - 1 + 1/9 = 8/3 < 4$ ซึ่งเป็นจริง

ขั้นตอนอุปนัย โดยแทน n ด้วย n+1

$4 - 4/3^{n+1} - 1/2^{n-1} + 1/(2^{n-1}3^{n+1}) < 4$

$4 - (2^{n+1} + 3^{n+1} - 1)/(2^{n-1}3^{n+1}) < 4$

จะเห็นว่าอสมการข้างต้นเป็นจริง

หากเราพิจารณาที่ฟังก์ชัน 1 และ 2 แล้ว เราจะพบว่า ถ้า $Sum_{k,l}$ มีค่ามากกว่าหรือเท่ากับ $3Cout_{n-1}$ จะทำให้เกิดตัวทดออกทั้ง $Cout_n$ และ $Cout_{n-1}$ ดังนั้นเราต้องทำการพิสูจน์ว่า เมื่อมีการทดออกสำหรับตัวทดออกทั้งคู่แล้ว จะต้องมีค่าทดออกน้อยกว่าหรือเท่ากับ $3Cout_{n-1}$ เพื่อที่จะทำให้ค่าที่ทดออกมีค่าไม่เกินค่าของ $Sum_{k,l}$

$Cout_n + Cout_{n-1} \leq 3Cout_{n-1}$

$2^{n-1}3^n + 2^{n-2}3^n \leq 3(2^{n-2}3^n)$

$2^{n-2}3^{n+1} \leq 2^{n-2}3^{n+1}$

จะเห็นได้ว่าอสมการข้างต้นเป็นจริง ■

ตัวอย่างที่ 3.4 *การบวกกันของ 10845 และ 19284 ในระบบแทนจำนวนฐานคู่*

วิธีทำ เราสามารถเขียนจำนวน 10845 และ 19284 ในรูปแบบแทนจำนวนฐานคู่ในตารางขนาด 6× 6 ได้ดังรูปที่ 3.8

	2^0	2^1	2^2	2^3	2^4	2^5
3^0	1	0	0	1	0	0
3^1	0	1	1	0	1	1
3^2	0	1	1	0	1	0
3^3	0	1	1	1	0	1
3^4	1	1	1	0	0	1
3^5	1	0	0	1	1	0

10845

	2^0	2^1	2^2	2^3	2^4	2^5
3^0	1	0	1	0	1	0
3^1	0	1	1	1	0	1
3^2	1	1	0	0	1	0
3^3	0	0	1	1	1	1
3^4	1	1	1	0	1	1
3^5	1	0	1	0	1	1

19284

รูปที่3.8 แสดงรูปแบบแทนจำนวนของตัวถูกดำเนินการ 10845 และ 19284

จากขั้นตอนแรกคือการนำตัวถูกดำเนินการที่อยู่ในรูปแบบตารางสองมิติด้านบนมาซ้อนทับกัน จะได้ตารางผลบวก $x_{i,j}$ ดังรูปที่ 3.9 ต่อไปนี้

	2^0	2^1	2^2	2^3	2^4	2^5
3^0	2	0	1	1	1	0
3^1	0	2	2	1	1	2
3^2	1	2	1	0	2	0
3^3	0	1	2	2	1	2
3^4	2	2	2	0	1	2
3^5	2	0	1	1	2	1

รูปที่3.9 แสดงรูปแบบแทนจำนวนผลบวกที่ได้จากการซ้อนทับกันของตัวถูกดำเนินการ

หลังจากนั้นทำการแบ่งตารางผลบวกที่ได้ออกเป็นตารางย่อย ขนาด 3×3 จะได้ผลลัพธ์ดังรูปที่ 3.10

2	0	1
0	2	2
1	2	1

123

1	1	0
1	1	2
0	2	0

72

0	1	2
2	2	2
2	0	1

106

2	1	2
0	1	2
1	2	1

123

รูปที่3.10 แสดงตารางย่อยขนาด 3×3 ทั้งหมดที่ได้จากการแบ่งตาราง *Sum*

จากนั้นทำการคำนวณหาค่าตัวทดออกจากสมการ

$$Cout_{2_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^2 3^3} \right\rfloor$$

$$Cout_{1_{k,l}} = \left\lfloor \frac{Sum_{k,l}}{2^1 3^3} \right\rfloor - 2Cout_{2_{k,l}}$$

จะได้ผลลัพธ์ดังนี้คือ

$$Sum_{0,0} = 123, Cout_{2_{0,0}} = 1, Cout_{1_{0,0}} = 0$$
$$Sum_{0,1} = 72, Cout_{2_{0,1}} = 0, Cout_{1_{0,1}} = 1$$
$$Sum_{1,0} = 106, Cout_{2_{1,0}} = 0, Cout_{1_{1,0}} = 1$$
$$Sum_{1,1} = 123, Cout_{2_{1,1}} = 1, Cout_{1_{1,1}} = 0$$

ขั้นตอนต่อไปเป็นขั้นตอนการส่งผ่านตัวทดซึ่งเราจะนำตัวทดออกที่ได้จากตารางย่อยทางด้านบนมาเป็นตัวทดเข้าของตารางทางด้านล่าง ซึ่งเป็นการทำงานแบบขนาน

จากสมการ $Result_{k,l} = Sum_{k,l} + Cin_{k,l} - Cout_{k,l}$, $Cin_{k,l} = Cout_{n-1_{k-1,l}} \times 2^{n-2}3^0 + Cout_{n_{k-1,l}} \times 2^{n-1}3^0$ และ $Cout_{k,l} = 2^{n-2}3^n Cout_{n-1_{k,l}} + 2^{n-1}3^n Cout_{n_{k,l}}$

จะได้ค่าของตาราง *Result* ย่อยดังต่อไปนี้คือ

$$Result_{0,0} = 15$$
$$Result_{0,1} = 18$$
$$Result_{1,0} = 56$$
$$Result_{1,1} = 17$$

ขั้นตอนสุดท้ายคือการนำค่าของตาราง *Result* ย่อยแต่ละตารางมาสร้างเป็นรูปแบบแทนจำนวนตาราง 2 มิติ ซึ่งจะได้รูปแบบแทนจำนวนของตารางผลลัพธ์ดังนี้คือ

	2^0	2^1	2^2	2^3	2^4	2^5
3^0	0	1	0	0	0	0
3^1	0	0	1	0	0	0
3^2	0	0	0	0	1	0
3^3	0	1	0	1	0	1
3^4	0	0	0	0	0	1
3^5	0	1	1	0	0	0

รูปที่3.11 แสดงตาราง *Result* ที่ได้จากการดำเนินการบวกแบบขนานของตัวถูกดำเนินการ 10845 และ 19284

จากตาราง *Result* ที่ได้เมื่อนำมาคำนวณแล้วจะได้ $2^1 3^0 + 2^2 3^1 + 2^4 3^2 + 2^1 3^3 + 2^3 3^3 + 2^5 3^3 + 2^5 3^4 + 2^1 3^5 + 2^2 3^5 = 5343$ และเมื่อนำมารวมกับ ตัวทดส่วนเกินซึ่งเกิดจาก $Cout_{1_{1,0}}$ ซึ่งมีค่า $2^1 3^6 = 1458$ จะและ $Cout_{2_{1,1}}$ ซึ่งมีค่า $2^5 3^6 = 23328$ จะได้ ตัวทดส่วนเกินคือ 24786 เมื่อรวมกับผลลัพธ์ 5243 จะได้ 30129 เท่ากับผลรวมของตัวถูกดำเนินการ 10845 + 19284 □

บทที่ 4

สรุปผลการวิจัยและข้อเสนอแนะ

4.1 สรุปผลการวิจัย

การดำเนินการบวกบนระบบแทนจำนวนฐานคู่นั้นสามารถกระทำได้โดยมีประสิทธิภาพเชิงเวลาสูงที่สุดคือ O(1) ซึ่งใช้เวลาคงตัว ตามที่ได้นำเสนออัลกอริทึม และบทพิสูจน์ความถูกต้องของอัลกอริทึมไปแล้วนั้นจะเห็นว่าอัลกอริทึมที่ได้มีความถูกต้องสมบูรณ์และสามารถสร้างเป็นทฤษฎีบทได้ โดยคุณสมบัติที่สำคัญสำหรับการได้มาซึ่งกระบวนการทำงานแบบขนานที่สามารถทำงานโดยใช้ประสิทธิภาพเชิงเวลาคงตัวนั้น คือคุณสมบัติความซ้ำซ้อนของระบบแทนจำนวนฐานคู่ ซึ่งระบบแทนจำนวนฐานสอง หรือ ฐานสิบ ที่ใช้กันโดยทั่วไปนั้น ไม่มีคุณสมบัติดังกล่าว ดังนั้นจึงไม่สามารถจำกัดสายการทดให้ไม่ขึ้นอยู่กับขนาดของจำนวนได้ การเพิ่มประสิทธิภาพเชิงเวลาจึงสามารถทำได้แค่เพียงการนำเทคนิคต่างๆเข้ามาประยุกต์ใช้ เช่น การใช้เทคนิคออนเดอร์ฟลาย แต่ก็ยังทำให้เกิดสายการทดอยู่ดีแต่สำหรับระบบแทนจำนวนฐานคู่นั้นจากที่ได้นำเสนอไปในบทที่ 3 จะเห็นได้ว่ายิ่งขนาดของตารางแทนจำนวนมีขนาดใหญ่ขึ้นก็ทำให้ระดับความซ้ำซ้อนของระบบแทนจำนวนมีค่าสูงขึ้นตามไปด้วย ซึ่งทำให้อัลกอริทึมที่ได้นำเสนอในตอนแรกคือการแบ่งตารางย่อยออกเป็น ตารางขนาด 2×2 สามารถขยายขอบเขตการทำงานออกไปได้อีกคือไม่จำกัดขนาดของตารางย่อยไว้ที่ 2×2 เท่านั้น แต่ยังสามารถสร้างอัลกอริทึมซึ่งอยู่ในรูปทั่วไป สำหรับตารางย่อยขนาด $n\times n$ โดยที่ n มีค่ามากกว่าหรือเท่ากับสองอีกด้วย

อย่างไรก็ตามระบบแทนจำนวนฐานคู่ยังคงมีปัญหาที่ยังไม่สามารถหาคำตอบได้ในปัจจุบัน เช่น การหารูปแบบแทนจำนวนในระบบแทนจำนวนฐานคู่ให้มีจำนวนบิทหนึ่งน้อยที่สุด หรือการที่จะตอบว่ารูปแบบแทนจำนวนที่ให้มามีจำนวนบิทหนึ่งน้อยที่สุดหรือไม่ ซึ่งปัญหาเหล่านี้ เป็นปัญหาที่อยู่ในกลุ่มเอ็นพีฮาร์ด(NP-hard problem) ซึ่งปัญหาเหล่านี้มีผลเชื่อมโยงถึงการออกแบบสถาปัตยกรรมคอมพิวเตอร์สำหรับการทำงานกับระบบแทนจำนวนฐานคู่โดยตรง

4.2 ข้อเสนอแนะสำหรับงานวิจัยในอนาคต

จากงานวิจัยนี้เป็นงานวิจัยเสนออัลกอริทึมที่ใช้ในคำนวณการดำเนินการบวกบนระบบแทนจำนวนฐานคู่แบบขนานซึ่งพบว่ายังสามารถทำงานวิจัยในการพัฒนาและปรับปรุงตามหัวข้อดังต่อไปนี้ได้

1) พัฒนาอัลกอริทึมการดำเนินการทางเลขคณิตพื้นฐานอื่นๆ ได้แก่การลบ(ซึ่งต้องขยายขอบเขตการแทนรูปแบบแทนจำนวนให้อยู่ในรูปจำนวนเต็มลบได้) การคูณ การหาร (ซึ่งต้องขยายขอบเขตค่าประจำหลักของแต่ละเซลล์ให้สามารถแทนจำนวนทศนิยมได้โดยเพิ่มเลขชี้กำลังให้เป็นจำนวนลบ) ซึ่งสามารถนำอัลกอริทึมที่ได้นำเสนอในงานวิจัยนี้ไปเป็นส่วนหนึ่งของงานต่อไปได้
2) การแก้ปัญหาตัวทดส่วนเกิน(overflow) โดยผู้ทำการวิจัยสามารถคิดและนำเสนออัลกอริทึมสำหรับจะดการกับตัวทดส่วนเกินโดยให้มีประสิทธิภาพเชิงเวลาคงตัวเช่นเดิมซึ่งจะเป็นการปรับปรุงให้งานวิจัยชิ้นนี้สมบูรณ์ยิ่งขึ้น
3) การพัฒนาด้านการออกแบบและพัฒนาสถาปัตยกรรมทางคอมพิวเตอร์เพื่อให้สามารถทำงานได้ตามอัลกอริทึมที่ได้เสนอในงานวิจัยชิ้นนี้

รายการอ้างอิง

[1] Dimitrov V.S., JullienG.A., Miller W.C, *Theory and Applications for Double-Base number System*,IEEE Transactions on Computers, Vol.**46**, 1997, pp. 44-51.

[2] Gilbert G., and Pierre Langlois J. M., *Multipath Greedy Algorithm for Canonical Representation of Numbers in the Double Base Number System*, IEEE Transactions on Computers, Vol.**49**, 2005, pp. 39-42

[3] Dimitrov V.S., Graham A. Jullien, William C. Miller, *Theory and Applicationsof the Double Base Number System*, IEEE Transactions on Computers, Vol.**48**, 1999, pp. 1098-1106

[4] Ming Li Kunpeng Wang, *New Greedy Algorithm for Double-Base Chains*, Fifth International Conference on Information Assurance and Security, 2009, pp. 445-450.

[5] Wong K.W., Edward C. W., Cheng L. M., and LiaoX., Fast Elliptic Scalar Multiplication using New Double-base Chain and Point Halving,Department Electronic Engineering, City University of Hong Kong, 400044, 2006

[6] DocheCh. and Imbert L., Extened Double-Base Number System with applications to Elliptic Curve Cryptography, Department of Computing Macquarie University, Australia and LaboratoiredInformatiquede Robotique et de Microelectronique de Montpellier CNRS UMR 5506, 2006

[7] Avizienis A., *Signed-digit numbcrrcpresentations for fast parallclarithmctic,* IRE Transactions on Elecironic Computers, Vol. EC-IO, no.**3**, 1961, pp. 389-400.

[8] Frougny Ch.,Pelantová E., SvobodováM.,*Parallel addition in non-standard numeration systems*, Theoretical Computer Science Vol.**412**, 2011, pp. 5714–5727.

[9] Boonchanawiwat P, Chalermchatwichien W., and Surarerks A., On-The-Fly Addition for Double-Base Number System, The 16th International Annual Symposium on. Computational Science and Engineering (ANSCSE 16), May 2012.

ประวัติผู้เขียนวิทยานิพนธ์

นายวุฒิภัทร เฉลิมฉัตรวิเชียร เกิดเมื่อวันที่ 12เมษายน พ.ศ. 2531จบการศึกษาระดับมัธยมตอนต้นและตอนปลายจากโรงเรียนวัดสุทธิวรารามเขตบางรัก จังหวัดกรุงเทพมหานคร เข้ารับการศึกษาต่อในระดับปริญญาบัณฑิต คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย จนสำเร็จการศึกษาในปี พ.ศ. 2551 และศึกษาต่อในระดับปริญญามหาบัณฑิตที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย