การดำเนินการบวกแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่น

นายจักรพันธุ์ สุคนธราช

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2551

PARALLEL ADDITIVE OPERATION FOR FLEXIBLE INTERVAL REPRESENTATION SYSTEM

Mr.Jakrapan Sukontarach

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
Chulalongkorn University
Academic Year 2008

หัวข้อวิทยานิพนธ์	การดำเนินการบวกแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่น
โดย	นายจักรพันธุ์ สุคนธราช
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ
(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

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

.... กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.พิษณุ คนองชัยยศ)

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

จักรพันธุ์ สุคนธราช : การดำเนินการบวกแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่น. (PARALLEL ADDITIVE OPERATION FOR FLEXIBLE INTERVAL REPRESENTATION SYSTEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.อรรถสิทธิ์ สุรฤกษ์, 38 หน้า.

ในวงการงานวิจัยทางด้านเลขคณิตสำหรับคอมพิวเตอร์นั้น ปัญหาที่เราสนใจคือเรื่องความเร็วในคำนวณ โดยงานวิจัยเป็นจำนวนมากมุ่งเน้นไปที่ปัจจัยและเทคนิคที่สามารถทำให้การคำนวณมีความเร็วสูง แต่ในบางครั้งเราไม่สามารถคำนวณให้ได้คำตอบที่ถูกต้องเสมอไป ดังนั้นระบบแทนช่วงจึงได้ถูกเสนอขึ้นเพื่อแก้ปัญหาดังกล่าว เนื่องจากช่วงประกอบด้วยจำนวนสองจำนวน เราจึงสามารถรับประกันได้ว่าข้อมูลนำเข้าที่มีความคลาดเคลื่อนสามารถถูกเขียนให้อยู่ในรูปของช่วงได้ แต่อย่างไรก็ตามระบบแทนช่วงประสบปัญหาทั้งทางด้านความสิ้นเปลืองเนื้อที่และความล่าช้าในการคำนวณ ระบบแทนช่วงแบบยืดหยุ่นจึงถูกเสนอขึ้นเพื่อลดเวลาในการคำนวนและยังลดเนื้อที่ที่ใช้แทนช่วงให้น้อยลง โดยผลลัพธ์ทางทฤษฎีแสดงให้เห็นว่าระบบนี้สามารถลดเนื้อที่ที่ใช้ในการแทนช่วงให้น้อยลง 25 เปอร์เซ็นต์เมื่อเปรียบเทียบกับระบบแทนช่วงแบบมีเครื่องหมายดั้งเดิม แต่ข้อจำกัดของระบบแทนช่วงแบบยืดหยุ่นที่ไม่สามารถทำการคำนวณแบบขนานได้ ทำให้เกิดความล่าช้าในการคำนวณเมื่อขนาดของข้อมูลมีขนาดใหญ่ขึ้น

งานวิจัยนี้ มุ่งเน้นที่การทำให้ระบบแทนช่วงแบบยืดหยุ่นสามารถทำการคำนวณพื้นฐานทางเลขคณิตได้แก่การบวกและการลบแบบขนานได้ เนื่องจากมีช่วงบางช่วงมีรูปแบบแทนค่าเพียงรูปแบบเดียว การบวกและการลบแบบขนานจึงไม่สามารถทำได้ เราจึงปรับปรุงระบบแทนช่วงแบบยืดหยุ่นโดยการเพิ่มความซ้ำซ้อนของระบบ จากนั้นเราเสนออัลกอริทึมในการดำเนินการพื้นฐานทางเลขคณิต โดยเฉพาะการบวกและการลบแบบขนาน สำหรับระบบแทนช่วงแบบยืดหยุ่นพร้อมทั้งบทพิสูจน์

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต จักรพันธุ์ สุคนธราช
สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก Atthasit Surarerks
ปีการศึกษา 2551

5070238021 : MAJOR COMPUTER ENGINEERING
KEYWORDS : INTERVAL ARITHMETIC / REDUNDANT NUMBER REPRESENTATION SYSTEM / PARALLEL ALGORITHMS

JAKRAPAN SUKONTARACH : PARALLEL ADDITIVE OPERATION FOR FLEXIBLE INTERVAL REPRESENTATION SYSTEM. ADVISOR : ATHASIT SURARERKS, Ph.D., 38 pp.

A major problem in a domain of computer arithmetic concerns how computational time can be speeded up. Many researches focused on introducing high speed computing techniques. However, the computation may not always produce the exact value. Therefore, interval representation system is established to handle the problem. Since an interval is a pair of numbers, it is guaranteed that uncertainty in the input data can be represented in this system. However, the space used and computational time for interval arithmetic is very high. Flexible interval representation system is introduced in order to reduce space used and computational time. The theoretical result shows that space can be reduced up to twenty-five percents comparing with space used for the classical signed-digit interval representation system. So far parallel computation cannot be implemented in this sytem. Therefore, it takes much computational time when the data size becomes large.

This thesis focuses on addition and subtraction for flexible interval representation system in parallel manner. Since some intervals have a unique representation in the flexible interval representation system, addition and subtraction cannot be performed in parallel manner. We then modify flexible interval representation system in order to increase redundancy of the system. We propose parallel addition and subtraction algorithm for flexible interval representation system together with the proof.

Department: Computer Engineering Student's Signature ....
Field of Study: Computer Engineering Advisor's Signature ....
Academic Year: 2008

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความอนุเคราะห์ และความช่วยเหลืออย่างดียิ่งจาก ผศ.ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ให้คำปรึกษา ให้ข้อเสนอแนะที่เป็นประโยชน์อย่างยิ่งต่อการวิจัย และช่วยตรวจแก้ไขในส่วนที่บกพร่องต่างๆ ทำให้วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วง

ขอขอบพระคุณ ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ผู้ช่วยศาสตราจารย์ ดร.พิษณุ คนองชัยยศ ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง ประธานกรรมการและกรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น และขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ประสิทธิประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

กราบขอบพระคุณ บิดา มารดา ที่เป็นกำลังใจสำคัญตลอดมา ขอขอบคุณ เพื่อนๆ พี่ๆ และน้องๆ ทุกคน โดยเฉพาะสมาชิกห้องปฏิบัติการวิจัย ELITE ที่ได้ให้ความช่วยเหลือ เสนอแนะ และแก้ไขเอกสาร จนกระทั่งวิทยานิพนธ์สำเร็จด้วยดี

สุดท้ายนี้ ผู้วิจัยหวังเป็นอย่างยิ่งว่างานวิจัยนี้จะเป็นประโยชน์ต่อผู้ที่สนใจหรือผู้ที่เกี่ยวข้องทั่วไป และหากมีข้อผิดพลาดประการใด ผู้วิจัยขออภัยมา ณ ที่นี้ด้วย

สารบัญ

สารบัญตาราง

สารบัญภาพ

คำอธิบายสัญลักษณ์และคำย่อ

สัญกรณ์ทางคณิตศาสตร์

X	รูปแบบแทนจำนวน (number representation)
ρ	ค่าฐาน (base)
$\min()$	การเลือกค่าที่น้อยที่สุด (minimum)
$\max()$	การเลือกค่าที่มากที่สุด (maximum)
lower	ค่าต่ำสุดของช่วงที่ได้จากการคำนวณ
upper	ค่าสูงสุดของช่วงที่ได้จากการคำนวณ
x_i	ตัวเลขตั้งต้นในระบบแทนช่วงแบบยืดหยุ่น
y_i	ตัวเลขที่นำมาคำนวณกับตัวเลขตั้งต้นในระบบแทนช่วงแบบยืดหยุ่น
z_i	ตัวเลขที่เป็นผลลัพธ์ในระบบแทนช่วงแบบยืดหยุ่น
λ	ฟังก์ชันการแปลงระหว่างชุดตัวเลข (conversion mapping function)
σ_d	ฟังก์ชันการส่งผ่านตัวทด (carry-transfer function)
ε_d	ฟังก์ชันจับคู่ตัวเลข (digit mapping function)

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัญหาที่สำคัญอย่างหนึ่งในวงการวิจัยทางเลขคณิตสำหรับคอมพิวเตอร์คือการออกแบบการคำนวณให้มีความถูกต้องสูง แต่ในระบบคอมพิวเตอร์เราไม่สามารถแทนจำนวนของจำนวนจริงด้วยรูปแบบที่จำกัด (finite representation) ได้ โดยผลลัพธ์ของการคำนวณอาจมีความคลาดเคลื่อนเพราะอาจต้องทำการปัดเศษเลขทศนิยมในตำแหน่งที่หน่วยประมวลผลไม่สามารถแสดงได้ (round-off error) ทำให้ผลลัพธ์ที่ได้เกิดความคลาดเคลื่อน ซึ่งเมื่อนำผลลัพธ์ไปทำการคำนวณต่อ ค่าความคลาดเคลื่อนนี้จะปรากฏชัดมากขึ้นและทำให้เกิดค่าความคลาดเคลื่อนสะสม [1] นอกจากนี้ ความคลาดเคลื่อนอาจเกิดได้จากปัจจัยภายนอกเช่น ความผิดพลาดของเครื่องมือวัด ทำให้ข้อมูลนำเข้ามีความคลาดเคลื่อนไปด้วย ซึ่งในกรณีนี้เราไม่สามารถที่จะหาคำตอบที่ไม่มีความคลาดเคลื่อนได้ แนวทางในการแก้ปัญหาคือการจำกัดขอบเขตของการคำนวณว่า คำตอบที่ถูกต้องอยู่ในช่วงใด ระบบแทนจำนวนแบบช่วง (interval number representation system) จึงได้ถูกนำเสนอขึ้น โดยใช้วิธีการแสดงจำนวนต่างๆ ให้อยู่ในรูปของช่วงที่ครอบคลุมค่าที่ถูกต้องนั้นแทนรูปแบบแทนจำนวนที่มีข้อจำกัด อย่างไรก็ตาม ระบบแทนจำนวนแบบช่วงนั้นประสบปัญหาทั้งทางด้านความสิ้นเปลืองเนื้อที่และความล่าช้าในการคำนวณ จึงได้มีความพยายามคิดค้นวิธีที่จะแก้ปัญหาดังกล่าวโดยนำระบบจำนวนซ้ำซ้อน (redundant number system) เข้ามาใช้ ซึ่งสามารถเพิ่มความเร็วในการคำนวณให้สูงขึ้น แต่เนื่องจากระบบจำนวนซ้ำซ้อนมีการขยายชุดตัวเลข (digit set) เพิ่มขึ้นเกือบสองเท่า ทำให้เกิดผลเสียในเรื่องขนาดของเนื้อที่ (spaces) ที่ต้องใช้ในการแทนค่าตัวเลข (digit) ที่ต้องเพิ่มขึ้นเป็นสองเท่าตามไปด้วย สำหรับวิธีแก้ไขปัญหาดังกล่าว ในงานวิจัย [2] ได้เสนอระบบแทนช่วงในรูปแบบใหม่ที่เรียกว่า ระบบแทนช่วงแบบยืดหยุ่น (flexible interval representation system) ซึ่งสามารถลดเวลาในการดำเนินการพื้นฐานทางเลขคณิตบนช่วง และใช้เนื้อที่น้อยลง แต่ข้อจำกัดของระบบแทนช่วงแบบยืดหยุ่นที่ไม่สามารถทำการคำนวณแบบขนานได้ ทำให้เกิดความล่าช้าในการคำนวณเมื่อขนาดของข้อมูลมีขนาดใหญ่ขึ้น

งานวิจัยนี้ มุ่งเน้นที่การทำให้ระบบแทนช่วงแบบยืดหยุ่นสามารถทำการคำนวณได้รวดเร็วขึ้นโดยไม่เพิ่มขนาดในการจัดเก็บข้อมูล ดังนั้นในงานนี้จึงได้มีการปรับปรุงและนำเสนอระบบแทนช่วงแบบยืดหยุ่น (flexible interval representation system) ใหม่ให้สามารถทำการคำนวณแบบขนานได้พร้อมทั้งเสนออัลกอริทึมในการดำเนินการพื้นฐานทางเลขคณิต (fundamental arithmetic operations) โดยเฉพาะการบวกและการลบแบบขนาน สำหรับระบบแทนช่วงแบบยืดหยุ่น

1.2 วัตถุประสงค์ของงานวิจัย

เพื่อสร้างระบบแทนจำนวนแบบใหม่ที่สามารถลดขนาดของรูปแบบแทนช่วงลงและยังสามารถทำการคำนวณแบบขนานได้ พร้อมทั้งเสนออัลกอริทึมสำหรับการดำเนินการพื้นฐานทางเลขคณิต (fundamental arithmetic operations) สำหรับระบบแทนช่วงแบบยืดหยุ่น

1.3 ขอบเขตของงานวิจัย

1) ปรับปรุงระบบแทนช่วงแบบยืดหยุ่น และเสนออัลกอริทึมแบบขนานสำหรับการดำเนินการพื้นฐานทางเลขคณิตได้แก่การบวกและการลบในระบบนี้
2) เสนอรูปแบบทั่วไปสำหรับระบบแทนช่วงแบบยืดหยุ่น คือ ระบบแทนช่วงแบบยืดหยุ่นฐานใดๆ ที่มีค่ามากกว่าหรือเท่ากับ 2

1.4 ขั้นตอนและวิธีดำเนินงานวิจัย

1) ศึกษางานวิจัยทางด้านการคำนวณแบบช่วง
2) วิเคราะห์ปัญหาของงานวิจัยที่มีความสอดคล้องกับงานวิจัยที่สนใจ
3) ปรับปรุงระบบแทนช่วงแบบยืดหยุ่น
4) ออกแบบอัลกอริทึมสำหรับการดำเนินการพื้นฐานทางเลขคณิต
5) พิสูจน์อัลกอริทึมที่ได้ออกแบบไว้
6) จัดทำวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับจากงานวิจัย

1) ได้ระบบแทนช่วงแบบใหม่ที่สามารถลดขนาดของรูปแบบแทนช่วงลงเทียบกับระบบแทนช่วงด้วยตัวเลขแบบมีเครื่องหมายและทำการคำนวนแบบขนานได้
2) ได้รูปแบบทั่วไปสำหรับระบบแทนช่วงแบบยืดหยุ่น กล่าวคือ ระบบแทนช่วงแบบยืดหยุ่นฐานใดๆ ที่มีค่ามากกว่าหรือเท่ากับ 2

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นผลงานวิชาการในหัวข้อเรื่องดังต่อไปนี้

1) “Modified Arithmetic Algorithm for Flexible Interval Binary Representation System” โดย จักรพันธุ์ สุคนธราช พิภพ เทียนประภาสิทธิ์ และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ 12th Annual National Symposium on Computational Science and Engineering (ANSCSE12)

2) “การบวกและลบแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่น” โดย จักรพันธุ์ สุคนธราช และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ 12th National Computer Science and Engineering Conference (NCSEC2008)
3) “Parallel Addition and Subtraction for the Flexible Interval Representation System” โดย จักรพันธุ์ สุคนธราช และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ International Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-09)

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 ระบบจำนวน (number system)

ระบบจำนวน (ρ, D) ประกอบด้วยฐาน (base) ρ ซึ่ง ρ สามารถเป็นจำนวนจริงหรือจำนวนเชิงซ้อน โดยที่ $\|\rho\| > 1$ และชุดตัวเลขจำกัด (finite digit set) D โดยที่สมาชิกในชุดตัวเลขที่เรียกว่า ตัวเลข (digit) สามารถเป็นได้ทั้งจำนวนจริงและจำนวนเชิงซ้อน

รูปแบบแทนจำนวน X บนชุดตัวเลข D ภายใต้ฐาน ρ สามารถเขียนได้ดังนี้

$$X = (x_n x_{n-1} ... x_0 . x_{n-1} x_{n-2} ...)_\rho$$

โดยที่ $x_i \in D$ ซึ่ง $i \le n,\ n \in Z$

ค่าเชิงตัวเลข (numerical value) ของ X ในฐาน ρ สามารถเขียนได้ดังนี้

$$\|X\| = \sum_{i=-\infty}^{n} x_i \rho^n$$

ตัวอย่าง 2.1 *กำหนดให้ระบบแทนจำนวนประกอบด้วยเลขฐาน* $\rho = 5$ *และมีชุดตัวเลข* $D = \{0,1,2,3,4\}$ *จงเขียนค่าของ X =1000 ให้อยู่ในระบบจำนวนนี้*

วิธีทำ ค่าของ X =1000 สามารถเขียนให้อยู่ในระบบนี้ได้เป็น

$$1000 = (13000)_5 \qquad \square$$

2.2 ระบบจำนวนแบบมีเครื่องหมาย (signed digit number system)

ระบบจำนวนแบบมีเครื่องหมายนี้ได้ถูกเสนอขึ้นครั้งแรกโดย อวีเซียนีส (Avizienis) [4] ปี ค.ศ. 1961 จุดประสงค์ของการเสนอระบบจำนวนใหม่นี้เพื่อจำกัดการแพร่ของตัวทดในระหว่างการคำนวณ โดยตัวเลขใหม่ที่ถูกสร้างขึ้นสามารถมีเครื่องหมายกำกับได้ นอกจากนี้จำนวนตัวเลขในชุดตัวเลขจะมีมากกว่าจำนวนตัวเลขในระบบดั้งเดิม

ระบบจำนวนแบบมีเครื่องหมายของอวีเซียนีสนั้น กำหนดให้มีค่าฐาน $\rho \ge 3$ โดยชุดตัวเลขกำหนดให้เป็น

$$D = \{-a, -a+1, ..., -1, 0, 1, ..., a-1, a\}$$

โดยที่ a เป็นจำนวนเต็มใด ๆที่อยู่ในช่วงของ $\frac{1}{2} < a < \rho$ ในกรณีที่ฐาน $\rho = 2$ จะมีชุดตัวเลขเป็น $D = \{\bar{1}, 0, 1\}$

หมายเหตุ *ในการเขียนตัวเลขในชุดตัวเลขแบบมีเครื่องหมาย นิยมใช้สัญลักษณ์ $\bar{d}$ แทนตัวเลข $-d$*

ระบบจำนวนแบบมีเครื่องหมายนี้มีคุณสมบัติที่สำคัญประการหนึ่งคือ สมบัติซ้ำซ้อน (redundant property) ซึ่งหมายความว่า จำนวนบางจำนวนสามารถมีรูปแบบแทนจำนวน (number representation) แบบมีเครื่องหมายได้มากกว่าหนึ่งรูปแบบ

ตัวอย่าง 2.2 *กำหนดให้ระบบแทนจำนวนแบบมีเครื่องหมายประกอบด้วยเลขฐาน $\rho = 5$ และมีชุดตัวเลข $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ จงเขียนค่าของ X =1000 ให้อยู่ในระบบจำนวนนี้*

วิธีทำ ค่าของ X = 1000 สามารถเขียนให้อยู่ในระบบนี้ได้มากกว่าหนึ่งแบบคือ

$$1000 = (13000)_5 = \left(2\bar{2}000\right)_5$$ □

ต่อมาในปี ค.ศ.1990 พาร์ฮามี (Parhami) [5] ได้เสนอรูปแบบทั่วไปของระบบแทนจำนวนแบบมีเครื่องหมาย (generalized signed-digit: GSD) ซึ่งคุณสมบัติที่แตกต่างจากระบบเดิมคือ ชุดตัวเลขไม่จำเป็นต้องเป็นเซตที่สมมาตร โดยกำหนดเลขฐาน ρ เป็นจำนวนเต็มบวกที่ $\rho \geq 2$ ชุดตัวเลขสามารถกำหนดให้เป็น

$$D = \{-l, -l+1, -l+2, \ldots, \bar{1}, 0, 1, \ldots, m-1, m\}$$

โดยที่ $l \leq 0$ และ $m \geq 0$ โดยที่ $m - l + 1 > \rho$

ตัวอย่าง 2.3 *กำหนดให้ระบบแทนจำนวนแบบมีเครื่องหมายประกอบด้วยเลขฐาน $\rho = 5$ และมีชุดตัวเลข $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2\}$ จงเขียนค่าของ X =1000 ให้อยู่ในระบบจำนวนนี้*

วิธีทำ ค่าของ X = 1000 สามารถเขียนให้อยู่ในระบบนี้ได้เป็น

$$1000 = \left(2\bar{2}000\right)_5$$ □

ข้อดีของระบบแทนจำนวนซ้ำซ้อนคือมีความสามารถในการจำกัดการแพร่ของตัวทดในการคำนวณทางคณิตศาสตร์ไม่ให้แพร่ไปอย่างไม่จำกัดได้ ซึ่งต่างไปจากระบบแทน

จำนวนไม่ซ้ำซ้อน เพราะฉะนั้นระบบแทนจำนวนแบบมีเครื่องหมายจึงสามารถทำการคำนวณแบบขนานได้ สำหรับบางตัวดำเนินการ (operator) ทำให้เวลาที่ใช้ในการคำนวณคงที่และไม่ขึ้นกับขนาดความยาวของตัวถูกดำเนินการ (operand)

2.3 ระบบแทนช่วง (interval representation system)

ในระบบการคำนวณสำหรับจำนวนจริงนั้น ความผิดพลาดอาจเกิดขึ้นได้ในระหว่างการคำนวณ ทั้งนี้เพราะรูปแบบที่จำกัดของการแทนจำนวน (finite representation of number) แนวคิดในการแทนจำนวนโดยใช้ช่วงและการคำนวณในรูปแบบของช่วงนั้นเปรียบเสมือนเป็นการคำนวณเพื่อหาขอบเขต (bound) ที่เป็นไปได้ของคำตอบในกรณีที่จำนวนที่ต้องการนำมาคำนวณนั้นมีความคลาดเคลื่อนไปได้ในช่วงที่กำหนด โดยมีการกำหนดค่าต่ำสุด (x_l) และค่าสูงสุด (x_u) ของช่วง ผลลัพธ์ที่ได้จากการคำนวณสามารถรับประกันได้ว่าผลลัพธ์ที่ถูกต้องต้องปรากฏอยู่ในช่วงที่ได้จากการคำนวณอย่างแน่นอน

กำหนดให้ X เป็นค่าเชิงตัวเลขใดๆ ในระบบแทนช่วง รูปแบบการแทนช่วงของ X ในระบบแทนช่วงเป็นดังนี้

$$X = [x_l, x_u]$$

2.4 การดำเนินการพื้นฐานทางเลขคณิตของระบบแทนช่วง (fundamental arithmetic operations for interval representation system)

กำหนดให้ $X = [x_l, x_u]$ และ $Y = [y_l, y_u]$ เป็นช่วงในระบบแทนช่วง โดยที่การดำเนินการพื้นฐานทางเลขคณิตของระบบแทนช่วงเป็นการคำนวณที่เกิดจากการบวก ลบ คูณ และหารกันของค่าต่ำสุดและค่าสูงสุดของช่วง โดยสามารถศึกษาได้จาก [6-8] ซึ่งสรุปได้ดังนี้

$$X + Y = [x_l + y_l, x_u + y_u]$$

$$X - Y = [x_l - y_u, x_u - y_l]$$

$$\begin{aligned} X \times Y = [&\min(x_l \times y_l, x_l \times y_u, x_u \times y_l, x_u \times y_u), \\ &\max(x_l \times y_l, x_l \times y_u, x_u \times y_l, x_u \times y_u)] \end{aligned}$$

$$\begin{aligned} X \div Y = [&\min(x_l \div y_l, x_l \div y_u, x_u \div y_l, x_u \div y_u), \\ &\max(x_l \div y_l, x_l \div y_u, x_u \div y_l, x_u \div y_u)] \end{aligned}$$

สำหรับการหารนั้น ช่วงที่เป็นตัวหารจะต้องไม่ครอบคลุมศูนย์

ตัวอย่างที่ 2.4 *กำหนดให้ $X = [5.25, 7.1]$, $Y = [4.65, 8.2]$ จงแสดงการบวก ลบ คูณ และหาร ของช่วง X และ Y*

วิธีทำ การคำนวณพื้นฐานของตัวดำเนินการบวก ลบ คูณ และหารของช่วง X และ Y สามารถแสดงได้ดังนี้

$$X + Y = [9.9, 15.3]$$

$$X - Y = [-2.95, 2.45]$$

$$\begin{aligned} X \times Y &= [\min(24.41, 43.05, 33.02, 58.22), \\ &\quad \max(24.41, 43.05, 33.02, 58.22)] \\ &= [24.41, 58.22] \end{aligned}$$

$$\begin{aligned} X \div Y &= [\min(1.129, 0.64, 1.527, 0.866), \\ &\quad \max(1.129, 0.64, 1.527, 0.866)] \\ &= [0.64, 1.527] \end{aligned}$$

□

2.5 การแปลงชุดตัวเลข (digit set conversion)

หลักการของการแปลงชุดตัวเลขคือการแปลงจากชุดตัวเลขหนึ่งไปยังอีกชุดตัวเลขหนึ่ง ซึ่งมีเลขฐาน ρ เดียวกัน สำหรับงานวิจัยในเรื่องนี้สามารถศึกษาเพิ่มเติมได้จาก [9, 10] โดยกำหนดให้ D และ E เป็นชุดตัวเลขจำกัดที่ต่างกัน และ ρ เป็นเลขฐานที่สามารถเป็นได้ทั้งจำนวนจริงและจำนวนเชิงซ้อน การแปลงชุดตัวเลขในระบบเลขฐาน ρ จากชุดตัวเลข D ไปยังชุดตัวเลข E สามารถเขียนได้ดังนี้

$$\lambda : D \rightarrow E$$

โดยที่ $X \in D$ และ $\|\lambda(X)\| = \|X\|$

สำหรับปัญหาของการแปลงชุดตัวเลขนั้นได้ถูกนำไปใช้ในการอธิบายการดำเนินการพื้นฐานทางเลขคณิต เช่น การบวกสามารถเทียบได้กับการแปลงชุดตัวเลขหนึ่งไปยังอีกชุดตัวเลขหนึ่งที่มีขนาดแตกต่างกัน แต่มีเลขฐานเดียวกัน ในรูปแบบของ $D = \{d \in Z \mid 2a \le d \le 2b\}$ และ $E = \{e \in Z \mid a \le e \le b\}$ เป็นต้น

ตัวอย่างที่ 2.5 *การบวกกันของเลขสองจำนวนในระบบเลขฐานสอง X = 0110110 และ Y = 0100101 โดยวิธีการแปลงชุดตัวเลข*

วิธีทำ *การบวกกันของเลขสองจำนวนบนระบบเลขฐานสอง X = 0110110 และ Y = 0100101 สามารถพิจารณาเป็นการแปลงชุดตัวเลขบนระบบเลขฐานสองจากชุดตัวเลข D = {0, 1, 2} ไปยังชุดตัวเลข E = {0, 1} ได้ โดยการแปลงจาก 0210211 ไปเป็น 1011011 ได้ดังนี้*

$$
\begin{array}{ccccccccl}
0 & 1 & 1 & 0 & 1 & 1 & 0 & & X \\
0 & 1 & 0 & 0 & 1 & 0 & 1 & + & Y \\
\hline
0 & 2 & 1 & 0 & 2 & 1 & 1 & & X+Y \\
1 & 0 & 1 & 1 & 0 & 1 & 1 & & \\
\hline\hline
\end{array}
$$

□

2.5.1 การแปลงแบบขนาน (parallel conversion)

การแปลงชุดตัวเลขแบบขนานนั้นมีข้อดีอย่างหนึ่งคือ ความเร็วในการคำนวณที่สูง เนื่องจากสามารถทำการคำนวณไปพร้อมกันในทุก ๆหลักของตัวเลขที่เป็นข้อมูลนำเข้า โดยเฉพาะในกรณีข้อมูลนำเข้าที่นำมาคำนวณกันอยู่ในรูปแบบที่ไม่มีตัวทด (carry-free) ส่งผลให้เราสามารถหาผลลัพธ์ได้พร้อมกันในทุก ๆหลัก โดยใช้เวลาในการคำนวณคือ O(1) แต่ในกรณีที่ข้อมูลนำเข้าที่นำมาคำนวณกันก่อให้เกิดตัวทดขึ้น ผลลัพธ์ในแต่ละหลักจะไม่สามารถผลิตออกมาได้พร้อมกัน เนื่องจากเกิดผลกระทบจากสายการแพร่ของตัวทด (carry propagation chain) ส่งให้ผลให้ตัวเลขทางซ้ายอาจเกิดความเปลี่ยนแปลงขึ้นได้ ดังนั้นการคำนวณแบบขนานอาจจำเป็นต้องทำการคำนวณมากกว่าหนึ่งครั้ง ทั้งนี้ขึ้นอยู่กับขนาดและรูปแบบของข้อมูลนำเข้า

2.6 สถาปัตยกรรมแบบทำควบคู่กัน (on-the-fly architecture)

สถาปัตยกรรมการแปลงแบบทำควบคู่กันเป็นกระบวนการผลิตตัวเลขที่เป็นข้อมูลนำออกแบบขนาน โดยอาศัยแนวคิดของฟังก์ชันประกอบ (composite function) เข้ามาประยุกต์ใช้ งานวิจัยของเออเสกโกแวค (Ercegovac) และแลง (Lang) [11] ถือเป็นงานวิจัยแรกที่ได้นำวิธีการแปลงชุดตัวเลขโดยใช้สถาปัตยกรรมแบบทำควบคู่กันมาใช้ โดยแปลงจำนวนจากจำนวนซ้ำซ้อนแบบมีเครื่องหมายไปเป็นรูปแบบสัญนิยม (conventional representation) ต่อมาคอร์เนอรัป (Kornerup) [9] ได้เสนอสถาปัตยกรรมแบบทำควบคู่กันเพื่อแปลงชุดตัวเลขซ้ำซ้อนให้เป็นชุดตัวเลขไม่ซ้ำซ้อนบนเลขฐานเดียวกัน โดยชุดตัวเลขซ้ำซ้อนจะต้องเป็นระบบจำนวนซ้ำซ้อนแบบมีเครื่องหมายเพื่อลดการเกิดสายการแพร่ของตัวทด ข้อดีของสถาปัตยกรรมแบบทำควบคู่กันคือการคำนวณแบบขนาน ทำให้การคำนวณมีความเร็วสูง เพราะทุก ๆตัวเลขของข้อมูลนำเข้าสามารถคำนวณไปพร้อม ๆกันได้ ทั้งนี้ตัวเลขที่มีนัยสำคัญสูงสุดของผลลัพธ์ขึ้นอยู่กับตัวเลขที่มีนัยสำคัญต่ำสุดของข้อมูลนำเข้า ดังนั้นเวลาที่ใช้ในการส่งผ่านค่าของตัวทดจากตัวเลขทางขวาสุดไปยังตัวเลขทางซ้ายสุดคือ O(log n) เมื่อ n หมายถึงจำนวนตัวเลขของข้อมูล

นำเข้า ส่วนข้อเสียของสถาปัตยกรรมแบบทำควบคู่กันคือ การสิ้นเปลืองทรัพยากรที่ใช้ในขั้นตอนการแปลงเป็นจำนวนมาก เช่น เรจิสเตอร์ เป็นต้น ทั้งนี้ขึ้นอยู่กับขนาดของข้อมูลนำเข้า ข้อเสียอีกประการหนึ่งคือ เราจะต้องทราบขนาดของข้อมูลนำเข้าก่อน จึงจะทำการแปลงได้

ในการแปลงชุดตัวเลข $D \to E$ โดยที่ $X \in P(\rho, D)$ และ $Y \in P(\rho, E)$ โดย $\|X\| = \|Y\|$ และ $D \neq E$ จะมีนัยสำคัญดังนี้

กำหนดให้ชุดตัวเลข E เป็นชุดตัวเลขไม่ซ้ำซ้อน และชุดตัวเลข C เป็นชุดตัวเลขของตัวทด c เราสามารถเขียนตัวเลข d ใดๆ ที่อยู่ในชุดตัวเลข D ซึ่งเป็นชุดตัวเลขซ้ำซ้อนให้อยู่ในรูปของสมการของการแปลงจาก $P(\rho, D)$ ไปยัง $P(\rho, E)$ คือ $d = c\rho + e$ โดยที่ $e \in E$ และ $c \in C$ แต่ในการคำนวณตามความเป็นจริงเราจะต้องรวมตัวทดที่เข้ามา (incoming carry) เข้ากับ d ก่อน หลังจากนั้นจึงผลิตตัวทดที่ส่งออก (outgoing carry) และ e ออกไป เพราะฉะนั้นจะได้ความสัมพันธ์ของการแปลงระหว่างชุดตัวเลข λ (conversion mapping λ) เป็นฟังก์ชันการแปลงดังต่อไปนี้

$$\lambda : C \times D \to C \times E$$

เมื่อ C เป็นชุดตัวเลขของตัวทด สำหรับบาง (c, d) ใน $C \times D$ มี c' อยู่ใน C และ e อยู่ใน E ซึ่ง

$$\lambda : (c', d) \to (c, e)$$

โดยเราสามารถเขียนเป็นสมการได้ดังนี้

$$c' + d = c\rho + e$$

เมื่อ ρ เป็นเลขฐาน โดยเราสามารถเรียก c' และ c คือ ตัวทดนำเข้าและตัวทดนำออกตามลำดับในแต่ละตัวเลข d ที่เป็นข้อมูลนำเข้า เช่น ในฐาน $\rho = 2$ และ $D = \{0, 1\}$ กำหนดให้ตัวทดนำเข้า $c' = 1$ และตัวเลข $d = 1$ เราสามารถคำนวณได้ว่าตัวเลข e และตัวทดนำออก c จะมีค่า $e = 0$ และ $c = 1$ เป็นต้น กำหนดให้ D = {0, 1, 2}, E = {0, 1} และ ρ = 2 ฟังก์ชันการแปลงสามารถแสดงได้ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงฟังก์ชันการแปลงจาก D ไป E บนเลขฐานสอง

λ		D		
		0	1	2
C	0	00	01	10
	1	01	10	11

ผลลัพธ์ในแต่ละคู่ในตารางที่ 2.1 แทนค่า ce ซึ่ง (c, e) อยู่ใน $C \times E$ โดยการคำนวณในตารางจะเริ่มจากค่า $c = 0$ จะเห็นว่า ce ที่เป็นไปได้คือ 00, 01 และ 10 เมื่อ d คือ 0, 1 และ 2 ตามลำดับ ดังนั้นตัวทดนำออกที่เป็นไปได้คือ 0 และ 1 เราจึงต้องเพิ่มแถวที่ค่า $c = 1$ และผลลัพธ์ที่ได้คือ 01, 10 และ 11 จากค่า ce ทั้งหมดจะเห็นว่าตัวทดที่เกิดขึ้นคือ 0 และ 1 เท่านั้น ดังนั้นตารางนี้จึงสมบูรณ์แล้ว

จากแนวคิดฟังก์ชันการแปลงชุดตัวเลข เราสามารถเขียนฟังก์ชันของชุดตัวเลขของตัวทดได้เป็น $\{\sigma_d\}, \sigma_d : C \to C$ โดยที่ $d \in D$ เรียกว่า ฟังก์ชันการส่งผ่านตัวทด (carry-transfer function) ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$\forall c \in C : \sigma_d(c') = c \text{ โดย } \lambda(c', d) = (c, e)$$

โดยที่ σ_d เป็นฟังก์ชันที่อธิบายเกี่ยวกับการจับคู่ (mapping) ค่าของตัวทดที่เข้ามา c' ไปยังค่าของตัวทดที่ส่งออกไป c โดยผ่านตัวเลข d หนึ่งๆ เท่านั้น ซึ่งเราสามารถเขียนฟังก์ชันการจับคู่นี้ผ่าน d ทุกตัวที่อยู่ใน D ได้ดังนี้

$$\sigma_{d_i d_{i-1} \ldots d_j}(c) = \sigma_{d_i}(\sigma_{d_{i-1}}(\ldots \sigma_{d_j}(c) \ldots))$$

เราเรียกฟังก์ชันนี้ว่าเป็นฟังก์ชันประกอบ และจากฟังก์ชันนี้เองทำให้เราสามารถหาค่า c ใดๆ ได้ โดยกำหนดให้ c_0 เริ่มต้นมีค่าเท่ากับ 0

$$c_i = \sigma_{d_i d_{i-1} \ldots d_j}(0)$$

เมื่อเราสามารถหาฟังก์ชันที่ทำการสร้างตัวทดใดๆ ได้แล้ว ต่อไปเราก็สามารถหาฟังก์ชันในการหา e ในลำดับใดๆได้ด้วย ฟังก์ชันจับคู่ตัวเลข (digit mapping function) $\{\varepsilon_d\}, \varepsilon_d : C \to E$ โดยที่ $d \in D$ ซึ่งเราสามารถเขียนเป็นสมการได้ดังนี้

$$e_i = \varepsilon_{d_i}(\sigma_{d_i d_{i-1} \ldots d_j}(0))$$

ตัวอย่างที่ 2.6 *กำหนดให้* $X = 3\bar{2}112\bar{2}11$ *เป็นผลลัพธ์ที่ได้จากการนำจำนวน 3 จำนวนในฐาน* $\rho = 3$ *ซึ่งมีชุดตัวเลขคือ* $\{\bar{1},0,1\}$ *มาทำการบวกกันแบบขนาน จงทำการแปลงชุดตัวเลขด้วยสถาปัตยกรรมแบบควบคู่กันโดยที่* $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ *และ* $E = \{\bar{1}, 0, 1\}$

วิธีทำ กำหนดให้ $X = x_8 x_6 \ldots x_1$ เป็นข้อมูลนำเข้า โดยที่ $x_i \in D$ เมื่อ i เป็นจำนวนเต็มบวก ฟังก์ชันการแปลงจะทำการผลิตคู่ผลลัพธ์ได้แก่ ตัวทดนำออกและตัวเลขที่เป็นผลลัพธ์ดังสมการต่อไปนี้

$$c' + d = c\rho + e$$

สามารถเขียนสมการใหม่ได้เป็น

$$c_{i-1} + x_i = c_i \rho + y_i$$

โดยที่มีตัวทดนำเข้าเริ่มต้น $c_0 =$ 0 และ $y_i \in E$ ผลลัพธ์ที่ได้คือ $Y = 1\bar{1}1111111$ เมื่อ $\|X\| = \|Y\|$ ทั้งนี้การแปลงนั้นจะแปลงจาก $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ ไปเป็น $E = \{\bar{1}, 0, 1\}$ โดยมี ρ = 3 และ $C = \{\bar{1}, 0, 1\}$ เราสามารถหาค่า c และ e ในแต่ละหลักได้ดังนี้

ค่า c จากฟังก์ชันประกอบสามารถแสดงได้คือ

$$c_0 = 0$$
$$c_1 = \sigma_1(c_0) = \sigma_1(0) = 0$$
$$c_2 = \sigma_2\sigma_1(0) = \sigma_2(0) = 0$$
$$c_3 = \sigma_3\sigma_2\sigma_1(0) = \sigma_3(0) = \bar{1}$$
$$c_4 = \sigma_4\sigma_3\sigma_2\sigma_1(0) = \sigma_4(\bar{1}) = 0$$
$$c_5 = \sigma_5\sigma_4\sigma_3\sigma_2\sigma_1(0) = \sigma_5(0) = 0$$
$$c_6 = \sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_1(0) = \sigma_6(0) = 0$$
$$c_7 = \sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_1(0) = \sigma_7(0) = \bar{1}$$
$$c_8 = \sigma_8\sigma_7\sigma_6\sigma_5\sigma_4\sigma_3\sigma_2\sigma_1(0) = \sigma_8(\bar{1}) = 1$$

ค่า e จากฟังก์ชันจับคู่ตัวเลข

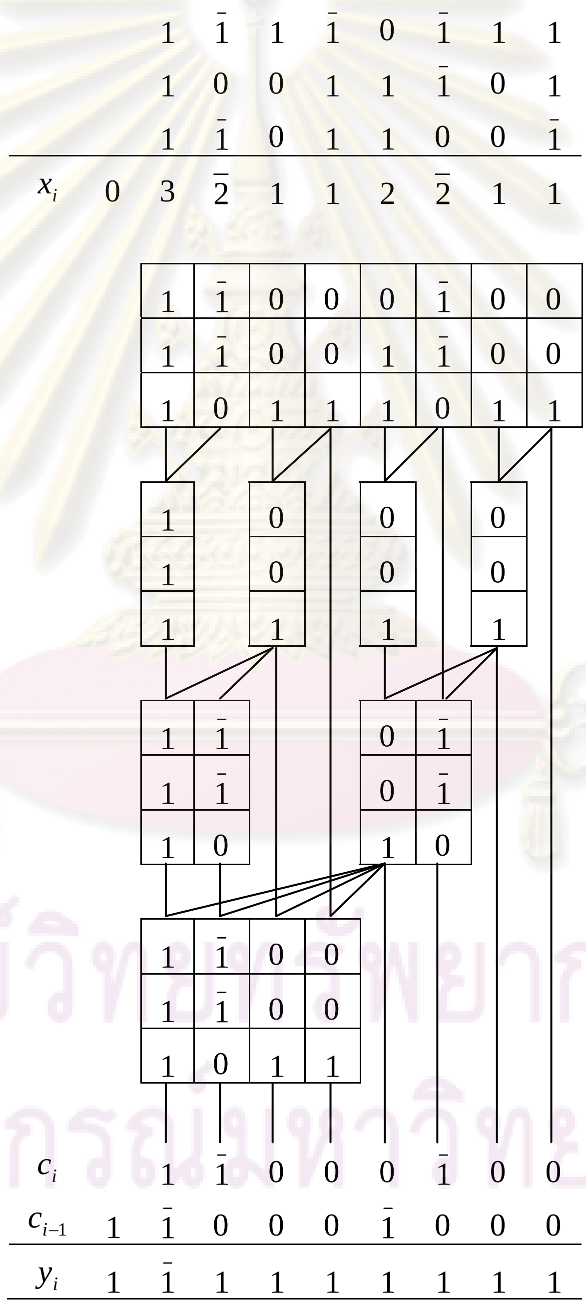
$$e_1 = \varepsilon_1(c_0) = \varepsilon_1(0) = 1 = y_1$$
$$e_2 = \varepsilon_2(c_1) = \varepsilon_2(0) = 1 = y_2$$
$$e_3 = \varepsilon_3(c_2) = \varepsilon_3(0) = 1 = y_3$$
$$e_4 = \varepsilon_4(c_3) = \varepsilon_4(\bar{1}) = 1 = y_4$$
$$e_5 = \varepsilon_5(c_4) = \varepsilon_5(0) = 1 = y_5$$
$$e_6 = \varepsilon_6(c_5) = \varepsilon_6(0) = 1 = y_6$$
$$e_7 = \varepsilon_7(c_6) = \varepsilon_7(0) = 1 = y_7$$
$$e_8 = \varepsilon_8(c_7) = \varepsilon_8(\bar{1}) = \bar{1} = y_8$$
$$e_9 = \varepsilon_9(c_8) = \varepsilon_9(1) = 1 = y_9$$

เพราะฉะนั้น $X = 3\bar{2}112\bar{2}11$ โดยที่ $x_i \in D$ จะถูกแปลงได้เป็น $Y = 1\bar{1}1111111$ โดยที่ $y_i \in E$ สำหรับฟังก์ชันการแปลงชุดตัวเลขจาก $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ ไปยัง $E = \{\bar{1}, 0, 1\}$ ในฐาน $\rho = 3$ สามารถแสดงได้ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงฟังก์ชันการแปลงจาก $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ ไปยัง $E = \{\bar{1}, 0, 1\}$ ในฐาน $\rho = 3$

λ		D						
		$\bar{3}$	$\bar{2}$	$\bar{1}$	0	1	2	3
C	$\bar{1}$	$\bar{1}\bar{1}$	$\bar{1}0$	$\bar{1}1$	$0\bar{1}$	00	01	$1\bar{1}$
	0	$\bar{1}0$	$\bar{1}1$	$0\bar{1}$	00	01	$1\bar{1}$	10
	1	$\bar{1}1$	$0\bar{1}$	00	01	$1\bar{1}$	10	11

กระบวนการของการแปลงชุดตัวเลขโดยใช้สถาปัตยกรรมแบบทำควบคู่กันนั้น จะเริ่มต้นโดยการคำนวณในรอบที่หนึ่งนั้น จะนำกรณีที่เป็นไปได้ของตัวทดนำเข้า ได้แก่ $\bar{1}$, 0 และ 1 มาทำการคำนวณกับค่าของ x_i ทั้ง 8 ตัว ซึ่งจะได้ผลลัพธ์คือตัวทดนำออกของแต่ละ x_i หลังจากนั้นทำการจับคู่ทั้ง 8 หลักมาคำนวณหาผลลัพธ์ในรอบที่สอง โดยอาศัยแนวคิดของฟังก์ชันประกอบ ทำการจับคู่ผลลัพธ์ในรอบที่สอง เพื่อคำนวณหาผลลัพธ์ในรอบที่สาม ทำซ้ำกระบวนการเดิมจนถึงรอบที่สี่ สำหรับกระบวนการของการแปลงชุดตัวเลขโดยใช้สถาปัตยกรรมแบบทำควบคู่กันนั้น สามารถแสดงได้ดังรูปที่ 2.1



รูปที่ 2.1 แสดงตัวอย่างการแปลงชุดตัวเลขโดยใช้สถาปัตยกรรมแบบทำควบคู่กัน

จากรูปที่ 2.1 จะเห็นได้ว่าผลลัพธ์ของ y_i ในหลักใดๆ จะได้จากการพิจารณาตัวทดนำเข้า c_{i-1} และ x_i โดยเริ่มจากหลักที่ $x_1 = 1$ จะกำหนดให้ตัวทดนำเข้า $c_0 = 0$ ซึ่งจะได้ผลลัพธ์ $y_1 = 1$ และ $c_1 = 0$ หลังจากนั้นในหลักที่ $x_2 = 1$ และตัวทดนำเข้า $c_1 = 0$ จะได้ผลลัพธ์ $y_2 = 1$ และ $c_2 = 0$ วนซ้ำกระบวนการนี้ไปจนถึง y_9 □

บทที่ 3

ระบบแทนช่วงแบบยืดหยุ่น

จุดประสงค์หลักของงานวิจัยนี้ คือ การปรับปรุงและการนำเสนออัลกอริทึมการดำเนินการบวกแบบขนานสำหรับระบบแทนจำนวนแบบช่วงแบบใหม่ ที่มีชื่อเรียกว่า ระบบแทนช่วงแบบยืดหยุ่น (Flexible interval representation system, FIRS) โดยที่ระบบการแทนช่วงแบบยืดหยุ่นนี้สามารถลดเนื้อที่ที่ใช้ในการแสดงช่วงให้น้อยลงได้ เนื่องจากระบบนี้ใช้จำนวนเพียงจำนวนเดียวในการแสดงช่วง

สำหรับเนื้อหาในบทนี้จะกล่าวถึงแรงจูงใจในการเสนอนำเสนออัลกอริทึมการดำเนินการบวกแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่น การปรับปรุงระบบแทนช่วงแบบยืดหยุ่น ความสมบูรณ์ของระบบแทนช่วงแบบยืดหยุ่น ความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น การดำเนินการบวกแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่น

3.1 บทกล่าวนำ

ระบบแทนช่วงแบบยืดหยุ่นเป็นระบบจำนวนที่ถูกพัฒนามาจากระบบแทนช่วง ซึ่งระบบแทนช่วงเป็นระบบที่ถูกเสนอขึ้นมาเพื่อแก้ปัญหาที่เกิดขึ้นในระหว่างการคำนวณทางคณิตศาสตร์ ซึ่งในบางครั้งอาจมีความจำเป็นต้องปัดเศษทิ้งไปในบางตำแหน่งเพื่อลดเวลาที่ใช้ในการคำนวณ หรือในกรณีที่ระบบมีรูปแบบจำกัดของการแทนจำนวนทำให้ระบบไม่สามารถรองรับการคำนวณให้ครบทุกตัวเลขได้ ทำให้มีความจำเป็นต้องปัดเศษทิ้งในบางตำแหน่งเพื่อให้ระบบสามารถทำงานต่อไป อีกทั้งความผิดพลาดของข้อมูลอาจเกิดขึ้นได้จากความคลาดเคลื่อนของอุปกรณ์การวัด หรือจากข้อมูลนำเข้า ซึ่งปัญหาต่าง ๆเหล่านี้ ส่งผลให้คำตอบที่ได้จากการคำนวณมีความผิดพลาดคลาดเคลื่อน ดังนั้นวิธีการแก้ปัญหาในเรื่องนี้คือการแทนข้อมูลที่ได้มาด้วยช่วง ซึ่งประกอบด้วยค่าขอบเขตสองค่า ทำให้สามารถรับประกันได้ว่าคำตอบที่ถูกต้องจะต้องปรากฏอยู่ในช่วงที่ได้จากการคำนวณอย่างแน่นอน แต่ระบบแทนช่วงประสบปัญหาทางด้านความสิ้นเปลืองเนื้อที่และความล่าช้าในการคำนวณ

จากที่ได้กล่าวมาข้างต้น ระบบแทนช่วงแบบยืดหยุ่นจึงได้ถูกนำเสนอขึ้นโดยใช้แนวคิดของระบบแทนช่วงมาประยุกต์ใช้กับระบบจำนวนแบบมีเครื่องหมาย ทำให้ระบบแทนช่วงแบบยืดหยุ่นสามารถลดความสิ้นเปลืองของเนื้อที่ให้น้อยลง 25 เปอร์เซ็นต์เมื่อเปรียบเทียบกับระบบแทนช่วงแบบมีเครื่องหมายดั้งเดิม (classical signed digit interval representation system) ระบบนี้สามารถแสดงช่วงได้โดยใช้จำนวนเพียงจำนวนเดียว การดำเนินการพื้นฐาน

ทางเลขคณิต ได้แก่ บวก ลบ คูณ และหาร แบบลำดับสำหรับระบบแทนช่วงแบบยืดหยุ่นนี้ได้ถูกพิสูจน์แล้วว่าสามารถทำได้

เพื่อที่จะลดเวลาที่ใช้ในการคำนวณให้น้อยลง แนวคิดของการคำนวณแบบขนานได้ถูกนำมาประยุกต์ใช้กับระบบแทนช่วงแบบยืดหยุ่น โดยระบบแทนช่วงแบบยืดหยุ่นได้ถูกปรับปรุงและนำเสนอใหม่ให้มีความซ้ำซ้อนสูงขึ้นเพื่อที่สามารถรองรับการคำนวณแบบขนานได้

3.2 ระบบแทนช่วงแบบยืดหยุ่น (Flexible interval representation system)

ระบบแทนช่วงแบบยืดหยุ่นใช้แนวคิดของการแสดงค่าต่ำสุดหรือค่าสูงสุดไว้ในตัวเลขเพียงตัวเลขเดียว เรียกว่า *"ตัวเลขยืดหยุ่น"* กล่าวคือ จะมองตัวเลขแต่ละตัวในรูปของช่วง ทำให้ระบบนี้สามารถใช้จำนวนเพียงจำนวนเดียวในการแทนค่าช่วงได้ โดยระบบแทนช่วงแบบยืดหยุ่นได้ถูกนำเสนอใหม่โดยมีการเพิ่มตัวเลขยืดหยุ่นเข้าไปในระบบอีก 2 ตัว นั่นคือ $\overline{\alpha}$ และ $\overline{\gamma}$ ทำให้ระบบมีความซ้ำซ้อนมากขึ้น โดยระบบแทนช่วงแบบยืดหยุ่นมีดังนิยามต่อไปนี้

นิยามที่ 3.1 ระบบแทนช่วงแบบยืดหยุ่น*ประกอบด้วยฐาน* ρ *และชุดตัวเลข* D *โดยที่* $\rho = 2$ *และ* $D = \{\bar{1}, \gamma, 0, \alpha, 1, \overline{\alpha}, \overline{\gamma}\}$ *ซึ่ง* $\alpha, \gamma, \overline{\alpha}, \overline{\gamma}$ *เรียกว่า* ตัวเลขยืดหยุ่น (flexible digit) *โดยที่* $\overline{\alpha}$ เป็นตัวผกผันการบวกของ α *และ* $\overline{\gamma}$ *เป็นตัวผกผันการบวกของ* γ *โดยค่าขอบเขตบนและขอบเขตล่างของตัวเลขยืดหยุ่นสามารถแสดงได้ดังตารางที่ 3.1*

ตารางที่ 3.1 แสดงขอบเขตบนและขอบเขตล่างของตัวเลขยืดหยุ่น

ตัวเลขยืดหยุ่น	ขอบเขตล่าง	ขอบเขตบน
α	0	1
γ	$\bar{1}$	0
$\overline{\alpha}$	0	$\bar{1}$
$\overline{\gamma}$	1	0

นิยามที่ 3.2 *รูปแบบแทนช่วง* $X = x_n x_{n-1} x_{n-2} \ldots x_0$ *ใน*ระบบแทนช่วงแบบยืดหยุ่น *สามารถแสดงช่วง [A, B] ได้โดย*

$$A = \sum_{i=0}^{n} \min(x_i) \times 2^i \quad \text{และ} \quad B = \sum_{i=0}^{n} \max(x_i) \times 2^i$$

สมบัติที่สำคัญของระบบแทนช่วงแบบยืดหยุ่นคือ ระบบนี้สามารถแสดงช่วงได้โดยใช้จำนวนเพียงจำนวนเดียว (ชุดตัวเลขหนึ่งชุด) เนื่องจากระบบแทนช่วงแบบยืดหยุ่นมีความ

ซ้ำซ้อน จำนวนช่วงในระบบนี้จึงอาจมีรูปแบบแทนค่าได้มากกว่าหนึ่งรูปแบบ ดังตัวอย่างใน ตารางที่ 3.1

ตารางที่ 3.2 แสดงตัวอย่างของระบบจำนวนแบบยืดหยุ่น

ช่วง	รูปแบบแทนช่วงแบบยืดหยุ่น
[3, 11]	$\alpha 011$, $\alpha\overline{\alpha}011$
[-2, 2]	$\gamma\overline{\gamma}10$, $\alpha\overline{1}0$
[0.25, 1.25]	$\alpha.010$
[26, 26]	11010

3.3 ความสมบูรณ์ของระบบแทนช่วงแบบยืดหยุ่น (Completeness of Flexible Interval Representation System)

เนื่องจากการนำเสนอระบบจำนวนแบบใหม่นั้นต้องคำนึงถึงความสมบูรณ์ (completeness) ของระบบ โดยจะแสดงให้เห็นว่าช่วงใดๆ สามารถหารูปแบบแทนจำนวนในระบบแทนช่วงแบบยืดหยุ่นได้อย่างน้อยหนึ่งรูปแบบ ดังทฤษฎีบทที่ 3.1

ทฤษฎีบทที่ 3.1 *สำหรับจำนวนจริง x และ y ใดๆ ช่วง [x, y] สามารถมีรูปแบบแทนช่วง (interval representation) ในระบบแทนช่วงแบบยืดหยุ่นได้*

พิสูจน์

วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมสำหรับแปลง (conversion) ช่วงใดๆ ที่อยู่ในรูปของรูปแบบแทนช่วงแบบมีเครื่องหมายดั้งเดิมให้อยู่ในรูปของรูปแบบแทนช่วงแบบยืดหยุ่น

Algorithm : Conversion

input interval [*A*, *B*]

    A = $a_n a_{n-1} ... a_0$ where $a_i \in \{\overline{1}, 0, 1\}$

    B = $b_n b_{n-1} ... b_0$ where $b_i \in \{\overline{1}, 0, 1\}$

output *S* = $s_n s_{n-1} ... s_0$ where $s_i \in \{\overline{1}, \gamma, 0, \alpha, 1\}$

begin

    C = *B* – *A* where *C* is binary number

    $i \leftarrow 0$

    while (not-end-of-data) **do**

if $c_i = 1$ **then** $c_i \leftarrow \alpha$

$i \leftarrow i+1$

endif

enddo

$S \leftarrow C + A$

end

พิสูจน์อัลกอริทึม

เนื่องจากขนาดของช่วงซึ่งคำนวณได้จากการนำค่าสูงสุดมาลบด้วยค่าต่ำสุดนั้นสามารถแสดงได้โดย *C* ที่เป็นจำนวนไม่เป็นลบ ดังนั้นช่วง [0, *C*] ใดๆ สามารถแสดงได้โดยใช้ลำดับของตัวเลข 0 และ α เพียงลำดับเดียวได้ เพราะฉะนั้นรูปแบบแทนช่วง [*A*, *B*] ใดๆ สามารถแสดงได้โดยการนำค่าต่ำสุดของช่วง (*A*) มาทำการบวกเข้ากับขนาดของช่วง (*C*) ■

ตัวอย่างที่ 3.1 *หารูปแบบแทนช่วง [1, 5] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ ขั้นตอนการหารูปแบบแทนช่วงจะแสดงในตารางที่ 3.2

ตารางที่ 3.3 แสดงขั้นตอนการแปลงช่วง [1, 5] ให้มาอยู่ในระบบแทนช่วงแบบยืดหยุ่น

ขั้นตอนการแปลง	ผลลัพธ์
1. ช่วง [1, 5]	$[0001, 0101]$
2. รูปแบบแทนจำนวนของค่าต่ำสุดของช่วง	0001
3. รูปแบบแทนช่วงของขนาดของช่วง	$0\alpha00$
4. รูปแบบแทนช่วงของ [1, 5] แสดงได้โดยการนำผลลัพธ์ของขั้นตอนที่ 2 และ 3 มาบวกกัน	$0\alpha01$

□

3.4 ความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น (Redundancy on Flexible Interval Representation System)

ในหัวข้อนี้จะกล่าวถึงความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่นที่ได้รับปรับปรุงใหม่ โดยจะแสดงให้เห็นว่าช่วงใดๆ ในระบบแทนช่วงแบบยืดหยุ่นที่ไม่ใช่ช่วงศูนย์ มีรูปแบบการแทนจำนวนได้มากกว่าหนึ่งรูปแบบ ดังทฤษฎีบทที่ 3.2

ทฤษฎีบทที่ 3.2 *จำนวนช่วงใดๆ ในระแบบแทนช่วงแบบยืดหยุ่น ที่ไม่ใช่ช่วงศูนย์สามารถมีรูปแบบการแทนจำนวนได้มากกว่าหนึ่งรูปแบบ*

พิสูจน์

วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมสำหรับสร้าง (generate) รูปแบบแทนจำนวนที่แตกต่างแต่มีค่าในเชิงตัวเลขเท่ากับข้อมูลนำเข้า

Algorithm: GenFIRS

Input FIRS X

$X = X_n X_{n-1} \ldots X_0$

where $X_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

Output FIRS Z

$Z = Z_m Z_{m-1} \ldots Z_0$

where $Z_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

begin

i = n

while i >= 0 do

if X_i is not zero **then**

if $X_i = 1$

then $T_{i+1}\, T_i = 1\bar{1}$ **endif**

else if $X_i = \bar{1}$

then $T_{i+1}\, T_i = \bar{1}1$ **endif**

else if $X_i = \alpha$

then $T_{i+1}\, T_i = \alpha\ \bar{\alpha}$ **endif**

else if $X_i = \gamma$

then $T_{i+1}\, T_i = \gamma\ \bar{\gamma}$ **endif**

else if $X_i = \bar{\alpha}$

then $T_{i+1}\, T_i = \bar{\alpha}\ \alpha$ **endif**

else if $X_i = \bar{\gamma}$

then $T_{i+1}\, T_i = \bar{\gamma}\, \gamma$ **endif**

$X_i = 0$

$Z = X_n X_{n-1} \ldots X_1 + T_{i+1}\, T_i$

$i = -1$

else

$i = i-1$

endif

endwhile

end

พิสูจน์อัลกอริทึม

อัลกอริทึมทำการหาตัวเลขที่ไม่ใช่ศูนย์โดยเริ่มจากทางซ้ายสุดของข้อมูลนำเข้า แล้วแทนค่าตัวเลขนั้นด้วยตัวเลขสองตัวที่มีค่าเชิงตัวเลขเท่ากัน เช่นแทนค่า 1 ด้วย $1\bar{1}$, $\bar{1}$ ด้วย $\bar{1}1$ และอื่นๆ ดังนั้นอัลกอริทึมจะให้ผลลัพธ์ที่มีรูปแบบแทนจำนวนที่แตกต่างกับข้อมูลนำเข้าแต่มีค่าเชิงตัวเลขเท่ากัน ซึ่งหมายความว่าช่วงใดๆ ที่ไม่ใช่ช่วงศูนย์มีรูปแบบการแทนจำนวนเป็นอนันต์ ■

3.5 การบวกและการลบแบบขนานในระบบแทนช่วงแบบยืดหยุ่น

การดำเนินการพื้นฐานทางเลขคณิตของงานวิจัยนี้ จะมุ่งเน้นไปที่การบวกและการลบแบบขนาน เนื่องจากแก่นของระบบแทนช่วงแบบยืดหยุ่นเป็นการหารูปแบบแทนจำนวนแบบใหม่ของระบบแทนช่วง จึงต้องนำความรู้เรื่องเลขคณิตของระบบแทนช่วงมาประยุกต์ใช้กับระบบแทนช่วงแบบใหม่นี้ด้วย [12-15]

3.5.1 การบวกแบบขนานในระบบแทนช่วงแบบยืดหยุ่น

สำหรับแนวคิดของการบวกแบบขนานของระบบแทนช่วงแบบยืดหยุ่นคือ การทำการคำนวณแบบขนานของตัวเลขในแต่ละหลักโดยแยกการคำนวณของขอบเขตบนและขอบเขตล่างออกจากกัน ซึ่งค่าของตัวเลขในขอบเขตบนและขอบเขตล่างจะเป็น $\bar{1}$, 0 หรือ 1 ซึ่งเมื่อนำมารวมกันก็จะได้ตัวเลขยืดหยุ่น

เนื่องจากค่าของตัวเลขที่เป็นไปได้ของขอบเขตบนและขอบเขตล่างมี 3 ค่า เมื่อนำมารวมกันจึงมีรูปแบบของตัวเลขยืดหยุ่นที่เป็นไปได้ 9 รูปแบบ แต่ชุดตัวเลขในระบบแทนช่วงแบบยืดหยุ่นมีตัวเลขเพียง 7 ตัว และตัวเลขอีก 2 ตัวอาจเกิดขึ้นได้ในระหว่างการคำนวณ เราจึงกำหนดให้ตัวเลขสองตัวนั้นเป็นตัวเลขชั่วคราว กล่าวคือ

β = $[\bar{1}, 1] = 2\alpha + \bar{1}$,

$\bar{\beta}$ = $[1, \bar{1}] = 2\gamma + \bar{1}$.

ซึ่งตัวเลขชั่วคราวเหล่านี้จะต้องถูกกำจัดทิ้งในภายหลังเพื่อให้ได้ผลลัพธ์สุดท้ายของการบวกแบบขนานในระบบแทนช่วงแบบยืดหยุ่น

ทฤษฎีบทที่ 3.3 *การบวกแบบขนานของช่วงสองช่วงใด ๆ สามารถทำได้ในระบบแทนช่วงแบบยืดหยุ่น*

พิสูจน์

วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมสำหรับการบวกแบบขนาน (parallel addition) พร้อมทั้งบทพิสูจน์ โดยอัลกอริทึมจะถูกแบ่งออกเป็นสองส่วน ส่วนแรกจะทำการคำนวณแบบขนานแต่ผลลัพธ์อาจมีตัวเลขชั่วคราวนั่นคือ β และ $\bar{\beta}$ ซึ่งจะถูกกำจัดในอัลกอริทึมส่วนที่สอง

ส่วนที่ 1 : เราได้เสนออัลกอริทึมการแปลง (transformation algorithm) พร้อมทั้งบทพิสูจน์

Algorithm I :

Input FIRS X, Y

$X = X_n X_{n-1} \ldots X_0$

$Y = Y_n Y_{n-1} \ldots Y_0$

where $X_i\ Y_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

Output FIRS Z

$Z = Z_n Z_{n-1} \ldots Z_0$ where $Z_i \in D \cup \{\beta, \bar{\beta}\}$

begin

foreach *i*

$X_i = [a_i, b_i]$

$Y_i = [c_i, d_i]$

$X_i + Y_i = [a_i + c_i, b_i + d_i] = 2C_{i+1} + S_i$

$= 2[r_{i+1}, s_{i+1}] + [t_i, u_i]$

where

Case $a_i+c_i =$ -2 $\quad r_{i+1} = \bar{1}, t_i = 0$

Case $a_i+c_i =$ -1 $\quad r_{i+1} = \bar{1}, t_i = 1$ if $a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} < -2$

$\quad r_{i+1} = 0, t_i = \bar{1}$ if $a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} \geq -2$

Case $a_i+c_i =$ 0 $\quad r_{i+1} = 0, t_i = 0$

Case $a_i+c_i =$ 1 $\quad r_{i+1} = 0, t_i = 1$ if $a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} \leq 2$

$\quad r_{i+1} = 1, t_i = \bar{1}$ if $a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} > 2$

Case $a_i+c_i =$ 2 $\quad r_{i+1} = 1, t_i = 0$

Case $b_i+d_i =$ -2 $\quad s_{i+1} = \bar{1}, u_i = 0$

Case $b_i+d_i =$ -1 $\quad s_{i+1} = \bar{1}, u_i = 1$ if $b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} < -2$

$\quad s_{i+1} = 0, u_i = \bar{1}$ if $b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} \geq -2$

Case $b_i+d_i = 0$ $\quad s_{i+1} = 0, u_i = 0$

Case $b_i+d_i = 1$ $\quad s_{i+1} = 0, u_i = 1$ if $b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} \leq 2$

$\quad s_{i+1} = 1, u_i = \bar{1}$ if $b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} > 2$

Case $b_i+d_i = 2$ $\quad s_{i+1} = 1, u_i = 0$

$Z_i = C_i + S_i$

end

พิสูจน์อัลกอริทึม

สำหรับการพิสูจน์อัลกอริทึมนี้จะแบ่งออกเป็นสองส่วน

ส่วนที่ 1.1) พิสูจน์ความถูกต้อง (proof of correctness)

เราจะแสดงว่า $X+Y = Z$

เนื่องจาก $X+Y = \sum_{all\ i} 2^i X_i + \sum_{all\ i} 2^i Y_i$

$= \sum_{all\ i} 2^i (X_i + Y_i)$

$= \sum_{all\ i} 2^i (C_i + S_i)$

$= \sum_{all\ i} 2^i Z_i$

$= Z$

ส่วนที่ 1.2) พิสูจน์การให้เหตุผล (Proof of validation)

เราจะแสดงว่า $Z_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \overline{\alpha}, \overline{\gamma}\} \cup \{\beta, \overline{\beta}\}$

จากอัลกอริทึม, $Z_i = C_i + S_i$

$= [r_i, s_i] + [t_i, u_i]$

เราจะแสดงว่า $t_i + r_i \in \{\bar{1}, 0, 1\}$.

กรณีที่ 1 : $t_i = 0$

จากอัลกอริทึม, $r_i \in \{\bar{1},0,1\}$

ดังนั้น $t_i + r_i \in \{\bar{1},0,1\}$

กรณีที่ 2 : $t_i = 1$

จากอัลกอริทึม, $a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} < -2$ หรือ

$a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} \le 2$

ดังนั้น $r_i \in \{\bar{1},0\}$

$t_i + r_i \in \{0,1\}$

กรณีที่ 3 : $t_i = \bar{1}$

จากอัลกอริทึม, $a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} \ge -2$ หรือ

$a_{i-1}+c_{i-1}+a_{i-2}+c_{i-2} > 2$

ดังนั้น $r_i \in \{0,1\}$

$t_i + r_i \in \{\bar{1},0\}$

จากทั้งสามกรณี เราสามารถสรุปได้ว่า

$t_i + r_i \in \{\bar{1}, 0, 1\}$ (1)

ต่อไปเราจะแสดงว่า $u_i + s_i \in \{\bar{1}, 0, 1\}$ โดยใช้แนวคิดในการพิสูจน์เดียวกัน

กรณีที่ 1 : $u_i = 0$

จากอัลกอริทึม, $s_i \in \{\bar{1},0,1\}$

ดังนั้น $u_i + s_i \in \{\bar{1},0,1\}$

กรณีที่ 2 : $u_i = 1$

จากอัลกอริทึม, $b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} < -2$ หรือ

$b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} \le 2$

ดังนั้น $s_i \in \{\bar{1},0\}$

$u_i + s_i \in \{0,1\}$

กรณีที่ 3 : $u_i = \bar{1}$

จากอัลกอริทึม, $b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} \geq -2$ หรือ

$b_{i-1}+d_{i-1}+b_{i-2}+d_{i-2} > 2$

ดังนั้น $s_i \in \{0,1\}$

$u_i + s_i \in \{\bar{1},0\}$

จากทั้งสามกรณี เราสามารถสรุปได้ว่า

$$u_i + s_i \in \{\bar{1}, 0, 1\} \quad (2)$$

จาก (1) และ (2), เราสรุปได้ว่าขอบเขตบนและขอบเขตล่างของ $Z_i \in \{\bar{1}, 0, 1\}$ นั่นคือ $Z_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\} \cup \{\beta, \bar{\beta}\}$

ส่วนที่ 2 : การกำจัด β และ $\bar{\beta}$ (elimination of β and $\bar{\beta}$)

ในการกำจัดตัวเลขชั่วคราว β และ $\bar{\beta}$ ออกจากผลลัพธ์ชั่วคราวจากอัลกอริทึม 1 มีแนวคิดคือการเปลี่ยนรูปแบบแทนจำนวนของ β และ $\bar{\beta}$ ให้เป็นรูปแบบแทนจำนวนอื่นในระบบแทนช่วงแบบยืดหยุ่น เนื่องจากรูปแบบแทนจำนวนในระบบนี้มีหลายรูปแบบ ดังนั้นเพื่อกำจัดความกำกวมในการเปลี่ยนรูปแบบเราจึงเลือกมาเพียงรูปแบบเดียวสำหรับ β และหนึ่งรูปแบบสำหรับ $\bar{\beta}$ กล่าวคือ

$$\beta = 2[0, 1] + [\bar{1},\bar{1}] = 2\alpha + \bar{1},$$

$$\bar{\beta} = 2[1, 0] + [\bar{1},\bar{1}] = 2\gamma + \bar{1}.$$

Algorithm II:

Input FIRS Z

$Z = Z_n Z_{n-1} \ldots Z_0$ where $Z_i \in D \cup \{\beta, \bar{\beta}\}$

Output FIRS F

$F = F_n F_{n-1} \ldots F_0$ where $F_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

begin

foreach i

$Z_i = [a_i, b_i]$

$= 2C_{i+1} + T_i$

$= 2[p_{i+1}, q_{i+1}] + [m_i, n_i]$

where

$p_{i+1} = a_i$, $m_i = -a_i$ if $a_i = a_{i-1}$

$p_{i+1} = 0$, $m_i = a_i$ if $a_i \neq a_{i-1}$

$q_{i+1} = b_i$, $n_i = -b_i$ if $b_i = b_{i-1}$

$q_{i+1} = 0$, $ni = b_i$ if $b_i \neq b_{i-1}$

$Y_i = [p_i, q_i] + [m_i, n_i]$

if (i mod 2 = 0) **then**

if $Y_i = \beta$ and $Y_{i-1} = \overline{\beta}$

then $Y_i = \alpha$, $Y_{i-1} = \bar{1}$ **endif**

if $Y_i = \overline{\beta}$ and $Y_{i-1} = \beta$

then $Y_i = \gamma$, $Y_{i-1} = \bar{1}$ **endif**

endif

if (i mod 2 = 1) **then**

if $Y_i = \beta$ and $Y_{i-1} = \overline{\beta}$

then $Y_i = \alpha$, $Y_{i-1} = \bar{1}$ **endif**

if $Y_i = \overline{\beta}$ and $Y_{i-1} = \beta$

then $Y_i = \gamma$, $Y_{i-1} = \bar{1}$ **endif**

endif

if $Y_i = \beta$ and $Y_{i-1} = \overline{\alpha}$

then $Y_i = \gamma$, $Y_{i-1} = \alpha$ **endif**

if $Y_i = \overline{\beta}$ and $Y_{i-1} = \gamma$

then $Y_i = \overline{\alpha}$, $Y_{i-1} = \overline{\gamma}$ **endif**

$Y_i = 2R_{i+1} + S_i$ (ใช้ตารางที่ 3)

$F_i = R_i + S_i$

end

ตารางที่ 3.4 แสดงกฎการแปลงเพื่อกำจัด β และ $\bar{\beta}$

Y_i	Y_{i-1}	R_{i+1}	S_i
α	$0, \bar{1}, \bar{\alpha}, \bar{\gamma}, \bar{\beta}$	0	α
	γ	α	$\bar{\alpha}$
γ	$0, 1, \bar{\alpha}, \bar{\gamma}, \bar{\beta}$	0	γ
	α	γ	$\bar{\gamma}$
$\bar{\alpha}$	$0, 1, \alpha, \gamma, \beta$	0	$\bar{\alpha}$
	$\bar{\gamma}$	$\bar{\alpha}$	α
$\bar{\gamma}$	$0, \bar{1}, \alpha, \gamma, \beta$	0	$\bar{\gamma}$
	$\bar{\alpha}$	$\bar{\gamma}$	γ
1	$0, \bar{1}, \gamma, \bar{\alpha}$	0	1
	$\bar{\gamma}$	1	$\bar{1}$
$\bar{1}$	$0, 1, \alpha, \bar{\gamma}$	0	$\bar{1}$
	$\bar{\alpha}$	$\bar{1}$	1
0	Any digit	0	0
β	Any digit	α	$\bar{1}$
$\bar{\beta}$	Any digit	$\bar{\gamma}$	$\bar{1}$

พิสูจน์อัลกอริทึม

สำหรับการพิสูจน์อัลกอริทึมนี้จะแบ่งออกเป็นสองส่วน

ส่วนที่ 1.1) พิสูจน์ความถูกต้อง (proof of correctness)

เราจะแสดงว่า $Z = F$

$$
\begin{aligned}
Z &= \sum_{all\ i} 2^i Z_i \\
&= \sum_{all\ i} 2^i (C_i + T_i) \\
&= \sum_{all\ i} 2^i Y_i \\
&= \sum_{all\ i} 2^i (R_i + S_i) \\
&= \sum_{all\ i} 2^i F_i \\
&= F
\end{aligned}
$$

ส่วนที่ 1.2) พิสูจน์การให้เหตุผล (Proof of validation)

เราจะแสดงว่า $F_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

จากอัลกอริทึม, $Y_i = 2R_{i+1} + S_i$, และ

$F_i = R_i + S_i$.

การพิสูจน์แบ่งได้เป็น 2 กรณี

กรณีที่ 1: $Y_{i-1} \in \{\beta, \bar{\beta}\}$

นั่นคือ $R_i = \alpha$ or γ

ดังนั้น จากตารางที่ 3 $R_i + S_i \in$ D

กรณีที่ 2: $Y_{i-1} \in$ D

นั่นคือ $R_i = 0$ or Y_{i-1}

2.1 $R_i = 0$, $S_i = Y_{i-1}$

ดังนั้น $R_i + S_i = Y_{i-1} \in$ D

2.2 $R_i = Y_{i-1}$, $S_i = -Y_{i-1}$

ดังนั้น $R_i + S_i = 0 \in$ D

จากทั้งสามกรณี เราสามารถสรุปได้ว่า $F_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$.

ดังนั้นเราสามารถสรุปได้ว่าการบวกแบบขนานสามารถทำได้ในระบบแทนช่วงแบบยืดหยุ่น ■

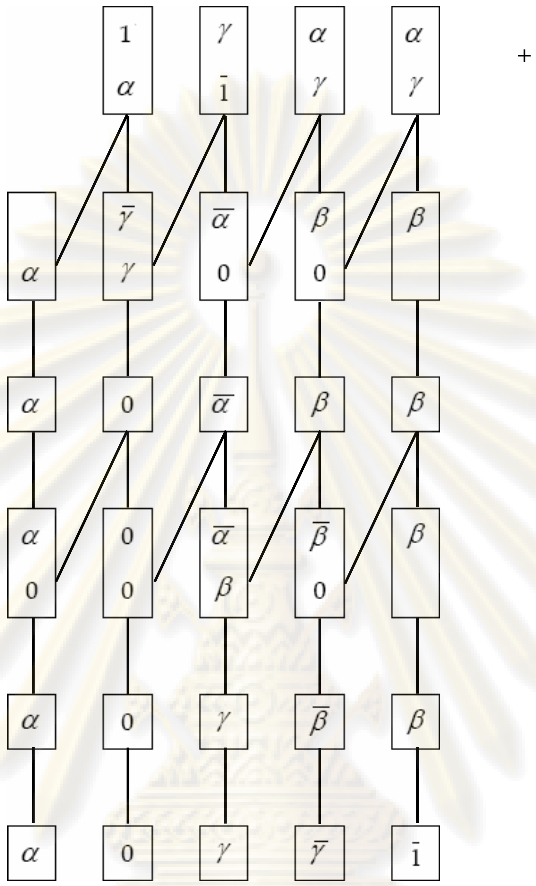
ตัวอย่างที่ 3.2 *การบวกแบบขนานของช่วง [4, 11] และ [-7, 4] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ สำหรับขั้นตอนการบวกนั้น เริ่มจากการแปลงช่วงให้อยู่ในระบบแทนช่วงแบบยืดหยุ่น นั่นคือ

[4, 11] $= 1\gamma\alpha\alpha$

[-7, 4] $= \alpha\bar{1}\gamma\gamma$

จากนั้นทำการคำนวณแบบขนานโดยอัลกอริทึม 1 และใช้อัลกอริทึม 2 ในการกำจัดตัวเลขชั่วคราวตามลำดับ



รูปที่ 3.1 แสดงการบวกแบบขนานในระบบแทนช่วงแบบยืดหยุ่น

การบวกแบบขนานในระบบแทนช่วงแบบยืดหยุ่นให้ผลลัพธ์ที่ถูกต้อง นั่นคือ $\alpha 0 \gamma \bar{\gamma} \bar{1}$ = [-3, 15] □

3.5.2 การลบแบบขนานในระบบแทนช่วงแบบยืดหยุ่น

สำหรับแนวคิดการลบแบบขนานในระบบแทนช่วงแบบยืดหยุ่นนั้นสามารถคำนวณได้โดยการนำช่วงที่เป็นตัวตั้งมาทำการบวกกับรูปแบบแทนลบ (negative representation) ของตัวที่นำมาลบ ซึ่งรูปแบบแทนลบสามารถหาได้จากการเปลี่ยนตัวเลข $\bar{1}$ เป็น 1, γ เป็น α, α เป็น γ, $\overline{\alpha}$ เป็น $\bar{\gamma}$, $\bar{\gamma}$ เป็น $\overline{\alpha}$ และ 1 เป็น $\bar{1}$

กระบวนการลบในระบบแทนช่วงแบบยืดหยุ่น สามารถทำได้ด้วยการใช้อัลกอริทึมการลบและอัลกอริทึมการบวกแบบขนานในระบบแทนช่วงแบบยืดหยุ่น ดังทฤษฎีบทที่ 3.3

ทฤษฎีบทที่ 3.4 *การลบแบบขนานสำหรับช่วงสองช่วงใดๆในระบบแทนช่วงแบบยืดหยุ่นสามารถคำนวณได้*

พิสูจน์

วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมการลบสำหรับระบบแทนช่วงแบบยืดหยุ่น

Algorithm : Subtraction

input FIR *X*, *Y*

$X = x_n x_{n-1} \ldots x_0$ where $x_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

$Y = y_n y_{n-1} \ldots y_0$ where $y_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

output FIR *Z*

$Z = z_n z_{n-1} \ldots z_1$ where $z_i \in \{\bar{1}, \alpha, 0, \gamma, 1, \bar{\alpha}, \bar{\gamma}\}$

begin

foreach *i*

if $y_i = \bar{1}$ **then** $y_i \leftarrow 1$

else if $y_i = 1$ **then** $y_i \leftarrow \bar{1}$

else if $y_i \leftarrow \alpha$ **then** $y_i \leftarrow \gamma$

else if $y_i \leftarrow \gamma$ **then** $y_i \leftarrow \alpha$

else if $y_i \leftarrow \alpha$ **then** $y_i \leftarrow \gamma$

else if $y_i \leftarrow \gamma$ **then** $y_i \leftarrow \alpha$

endif

Parallel Addition (*X*, *Y*)

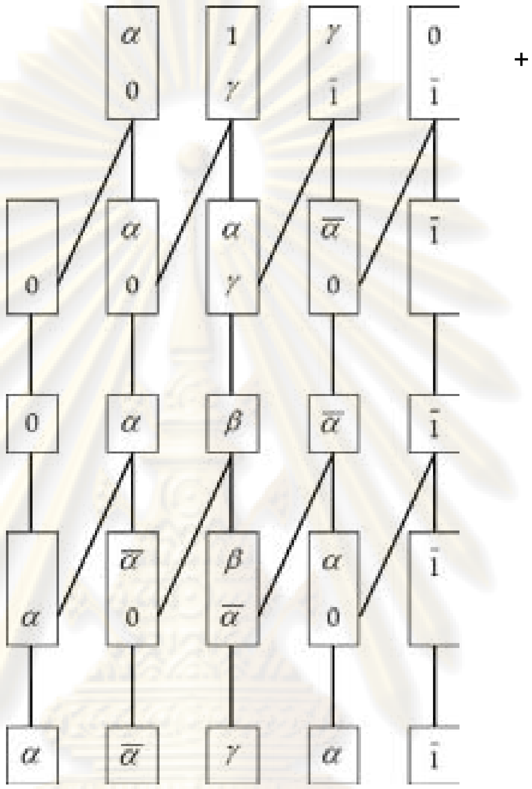
end

พิสูจน์อัลกอริทึม

เนื่องจากเราทราบว่า $X - Y = [x_l - y_u, x_u - y_l]$ หรือ *X* − *Y* = *X* + (-1)**Y* ซึ่ง (-1)*$\bar{1}$ เป็น 1, (-1)*γ เป็น α, (-1)*α เป็น γ, (-1)*$\bar{\alpha}$ เป็น $\bar{\gamma}$, (-1)*$\bar{\gamma}$ เป็น $\bar{\alpha}$ และ (-1)*1 เป็น $\bar{1}$ ดังนั้นเมื่อทำการหารูปแบบแทนลบแล้ว จึงนำมาทำการบวกแบบขนานกับตัวตั้งได้ซึ่งวิธีการพิสูจน์อัลกอริทึมการลบนั้น สามารถทำการพิสูจน์เช่นเดียวกับอัลกอริทึมการบวก ■

ตัวอย่างที่ 3.3 *การลบของช่วง [2, 12] โดย [3, 7] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ สำหรับขั้นตอนการลบนั้นเริ่มจากการหารูปแบบแทนช่วงของ [2, 12] และ [3, 7] ในระบบแทนช่วงแบบยืดหยุ่นคือ $\alpha 1 \gamma 0$ และ $0 \alpha 11$ ตามลำดับ รูปแบบแทนลบของตัวที่จะนำมาลบได้จากการแปลง $0\alpha 11$ เป็น $0\gamma\bar{1}\bar{1}$ จากนั้นสามารถทำการบวกแบบขนานระหว่าง $01\gamma\alpha$ และ $0\gamma\bar{1}\bar{1}$ ซึ่งผลลัพธ์ของการบวกแบบขนานจะเป็นคำตอบของการลบ $\alpha 1 \gamma 0$ ด้วย $0\alpha 11$



รูปที่ 3.2 แสดงการลบแบบขนานในระบบแทนช่วงแบบยืดหยุ่น

การลบแบบขนานในระบบแทนช่วงแบบยืดหยุ่นให้ผลลัพธ์ที่ถูกต้อง นั่นคือ $\alpha\bar{\alpha}\gamma\alpha\bar{1}$ = [-5, 9] □

3.6 ระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป

สำหรับระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปที่สนใจในหัวข้อนี้ จะพิจารณาเฉพาะฐานที่อยู่ในรูป 2^n เมื่อ n เป็นจำนวนเต็มบวก ซึ่งลักษณะของระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปนั้น จะมีการเพิ่มตัวเลขที่เป็นจำนวนเต็มและตัวเลขยืดหยุ่นในชุดตัวเลข D ให้มีจำนวนมากขึ้นเพื่อรองรับการแสดงช่วงต่างๆ ดังนิยามที่ 3.3 และ 3.4

นิยามที่ 3.3 ระบบแทนช่วงแบบยืดหยุ่นใน*รูปทั่วไปประกอบด้วยฐาน ρ และชุดตัวเลข D โดยที่ ρ เป็นจำนวนเต็มใดๆที่ $\rho = 2^n$ เมื่อ $n \in$ จำนวนเต็มบวก และชุดตัวเลข D ประกอบด้วย*

ตัวเลขที่เป็นจำนวนเต็มและตัวเลขยืดหยุ่น ในส่วนของตัวเลขที่อยู่ในรูปแบบของจำนวนเต็ม e จะมีจำนวน ρ+1 ตัว โดย e จะเป็นจำนวนเต็มทั้งหมดที่อยู่ระหว่าง -ρ/2 ถึง ρ/2 ($\{e \in \mathrm{Z} \mid -\rho/2 \le e \le \rho/2\}$) ในส่วนของตัวเลขยืดหยุ่น จะมีจำนวน $\rho^2+\rho-2$ ตัว ขนาดของตัวเลขยืดหยุ่นจะมีค่าตั้งแต่ 1 ถึง $\rho-1$ กำหนดให้ i เป็นขนาดของตัวเลขยืดหยุ่นซึ่งสามารถคำนวณได้จากการนำค่าสูงสุดของตัวเลขยืดหยุ่นมาลบด้วยค่าต่ำสุดของตัวเลขยืดหยุ่น โดยในแต่ละขนาดของตัวเลขยืดหยุ่น จะมีจำนวนตัวเลขยืดหยุ่น 2(ρ+1-i) ตัว*

นิยามที่ 3.4 *รูปแบบแทนช่วง $X = x_n x_{n-1} x_{n-2} \ldots x_0$ ในระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป สามารถแสดงช่วง [a , b] ได้โดย*

$$a=\sum_{i=0}^{n}\min(x_i)\times\rho^i \quad \text{และ} \quad b=\sum_{i=0}^{n}\max(x_i)\times\rho^i$$

ตัวอย่างที่ 3.4 *จงหาระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป สำหรับ ρ = 4*

วิธีทำ วิธีการสร้างระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป สำหรับ ρ = 4 นั้น จะแบ่งการสร้างชุดตัวเลข D ออกเป็น 2 กรณี คือ ตัวเลขที่เป็นจำนวนเต็มและตัวเลขยืดหยุ่น

กรณีที่ 1
ตัวเลขที่เป็นจำนวนเต็มใน D $(e \in \mathrm{Z} \mid -2 \le e \le 2)$ คือ $\bar{2}, \bar{1}, 0, 1, 2$

กรณีที่ 2
ตัวเลขยืดหยุ่นใน D จะมีขนาดของตัวเลขยืดหยุ่นไม่เกิน 3 นั่นคือ มีตัวเลขยืดหยุ่นที่มีขนาดเท่ากับ 1, 2 และ 3 ซึ่งจำนวนตัวเลขยืดหยุ่นในแต่ละขนาดจะมีจำนวนเป็น 4,3,2 ตัวตามลำดับรวมกับตัวผกผันการบวกอีกเท่าตัว(เกิดจากการสลับค่าสูงสุดและต่ำสุดในแต่ละตัวเลขยืดหยุ่น)

ถ้ากำหนดให้สัญลักษณ์ $\mu_{x,y}$ แทนตัวเลขยืดหยุ่นของช่วง [*x*, *y*]

เราได้ว่า ในช่วงขนาด 1 จะมีตัวเลขยืดหยุ่นจำนวน 4 ตัว คือ $\mu_{\bar{2},\bar{1}}$, $\mu_{\bar{1},0}$, $\mu_{0,1}$ และ $\mu_{1,2}$ และในช่วงขนาด 2 จะมีตัวเลขยืดหยุ่นจำนวน 3 ตัวคือ $\mu_{\bar{2},0}$, $\mu_{\bar{1},1}$ และ $\mu_{0,2}$ และในช่วงขนาด 3 จะมีตัวเลขยืดหยุ่นจำนวน 2 ตัวคือ $\mu_{\bar{2},1}$ และ $\mu_{\bar{1},2}$ รวมทั้งตัวผกผันการบวกขนาด 1 ซึ่งมีตัวเลขยืดหยุ่นจำนวน 4 ตัวคือ $\mu_{\bar{1},\bar{2}}$, $\mu_{0,\bar{1}}$, $\mu_{1,0}$ และ $\mu_{2,1}$ และตัวผกผันการบวกขนาด 2 ซึ่งมีตัวเลขยืดหยุ่นจำนวน 3 ตัวคือ $\mu_{0,\bar{2}}$, $\mu_{1,\bar{1}}$ และ $\mu_{2,0}$ และตัวผกผันการบวกขนาด 3 ซึ่งมีตัวเลขยืดหยุ่นจำนวน 2 ตัวคือ $\mu_{1,\bar{2}}$ และ $\mu_{2,\bar{1}}$

ดังนั้นในระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปที่มี ρ = 4 นั้น จะมีตัวเลขทั้งหมด 23 ตัว และชุดตัวเลขในระบบนี้คือ $D = \{\bar{2}, \bar{1}, 0, 1, 2, \mu_{\bar{2},\bar{1}}, \mu_{\bar{1},0}, \mu_{0,1}, \mu_{1,2}, \mu_{\bar{2},0}, \mu_{\bar{1},1}, \mu_{0,2}, \mu_{\bar{2},1},$

$\mu_{\bar{1},2}, \mu_{\bar{1},\bar{2}}, \mu_{0,\bar{1}}, \mu_{1,0}, \mu_{2,1}, \mu_{0,\bar{2}}, \mu_{1,\bar{1}}, \mu_{2,0}, \mu_{1,\bar{2}}, \mu_{2,\bar{1}}\}$ □

อีกสิ่งหนึ่งที่เราต้องคำนึงถึงคือเรื่องของความสมบูรณ์ (completeness) ของระบบ ซึ่งความสมบูรณ์ของระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปสามารถแสดงได้ดังทฤษฎีบทที่ 4.1

ทฤษฎีบทที่ 3.5 *ทุกช่วง [x, y] ใดๆ สามารถมีรูปแบบแทนช่วง ในระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปได้*

พิสูจน์

วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมสำหรับแปลง (conversion) ช่วงใดๆ ให้อยู่ในรูปของรูปแบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป

Algorithm : Conversion

input interval [*A*, *B*]

$A = a_n a_{n-1} \ldots a_0$ where $a_i \in \{-\rho/2, \ldots, \rho/2\}$

$B = b_n b_{n-1} \ldots b_0$ where $b_i \in \{-\rho/2, \ldots, \rho/2\}$

output $S = s_n s_{n-1} \ldots s_0$ where $s_i \in \{e_0, \ldots, e_n, f_0, \ldots, f_n\}$ with $\{e \in \mathrm{Z} \mid -\rho/2 \le e \le \rho/2\}$

and f_i are flexible digits

begin

$C = B - A$ where $c_i \in C$ with $c_i \in \{-\rho/2, \ldots, \rho/2\}$

$C = c_n c_{n-1} \ldots c_0$

where $c_i \leftarrow$ flexible digit that represents $[0, k]$

$S \leftarrow C + A$

end

พิสูจน์อัลกอริทึม

เนื่องจากขนาดของช่วงซึ่งคำนวณได้จากการนำค่าสูงสุดมาลบด้วยค่าต่ำสุดนั้นสามารถแสดงได้โดย *C* ซึ่งเป็นจำนวนไม่เป็นลบ ดังนั้นช่วง [0, *C*] ใดๆ สามารถแสดงได้โดยใช้ลำดับของตัวเลข 0 และตัวเลขยืดหยุ่นในแต่ละขนาดเพียงลำดับเดียวได้ เพราะฉะนั้นรูปแบบแทนช่วง [*A*, *B*] ใดๆ สามารถแสดงได้โดยการนำค่าต่ำสุดของช่วง (*A*) มาทำการบวกเข้ากับขนาดของช่วง (*C*) ■

สำหรับการดำเนินการพื้นฐานทางเลขคณิต ได้แก่ การบวก ลบ ของระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปนั้นสามารถใช้หลักการเดียวกับ การคำนวณในระบบแทนช่วงแบบยืดหยุ่นในฐาน 2 แต่อัลกอริทึมที่ถูกใช้ในการคำนวณต้องถูกพัฒนาให้เป็นในรูปแบบทั่วไปเพื่อที่จะสามารถรองรับตัวเลขยืดหยุ่นที่เพิ่มขึ้นตามฐานไปด้วย

ตัวอย่างที่ 3.5 *หารูปแบบแทนช่วง [1, 9] ในระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไปที่ ρ = 4*

วิธีทำ ขั้นตอนการหารูปแบบแทนช่วงจะแสดงในตารางที่ 4.1 โดยใช้สัญลักษณ์ $\mu_{x,y}$ แทนตัวเลขยืดหยุ่นของช่วง [*x*, *y*]

ตารางที่ 3.5 แสดงขั้นตอนการแปลงช่วง [1, 9] ให้อยู่ในระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป

ขั้นตอนการแปลง	ผลลัพธ์
1. ช่วง [1, 9]	$[0001, 0021]$
2. รูปแบบแทนจำนวนของค่าต่ำสุดของช่วง	0001
3. รูปแบบแทนช่วงของขนาดของช่วง	$00\mu_{0,2}0$
4. รูปแบบแทนช่วงของ [1, 9] สามารถแสดงได้โดยการนำผลลัพธ์ของขั้นตอนที่ 2 และ 3 มาบวกกัน	$00\mu_{0,2}1$

□

3.7 สรุป

ในบทนี้เราได้ปรับปรุงพร้อมทั้งนำเสนอระบบแทนช่วงแบบยืดหยุ่นใหม่ให้สามารถทำการคำนวณแบบขนานได้ ซึ่งระบบแทนช่วงแบบยืดหยุ่นนี้สามารถลดเนื้อที่ที่ใช้ในการแสดงช่วงให้น้อยลงได้ เนื่องจากระบบนี้สามารถใช้จำนวนเพียงจำนวนเดียวในการแสดงช่วง เราได้เพิ่มตัวเลขเข้าไปในระบบอีกสองตัวเพื่อเพิ่มความซ้ำซ้อนของระบบให้มากขึ้น ซึ่งการเพิ่มตัวเลขเข้าไปในระบบนี้ไม่ส่งผลกระทบต่อการเพิ่มเนื้อที่ในการแสดงตัวเลขในช่วง อีกทั้งในบทนี้ยังได้เสนอระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป ความสมบูรณ์และความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น พร้อมทั้งอัลกอริทึมสำหรับการบวกแบบขนานและการลบแบบขนานสำหรับระบบแทนช่วงแบบยืดหยุ่นอีกด้วย

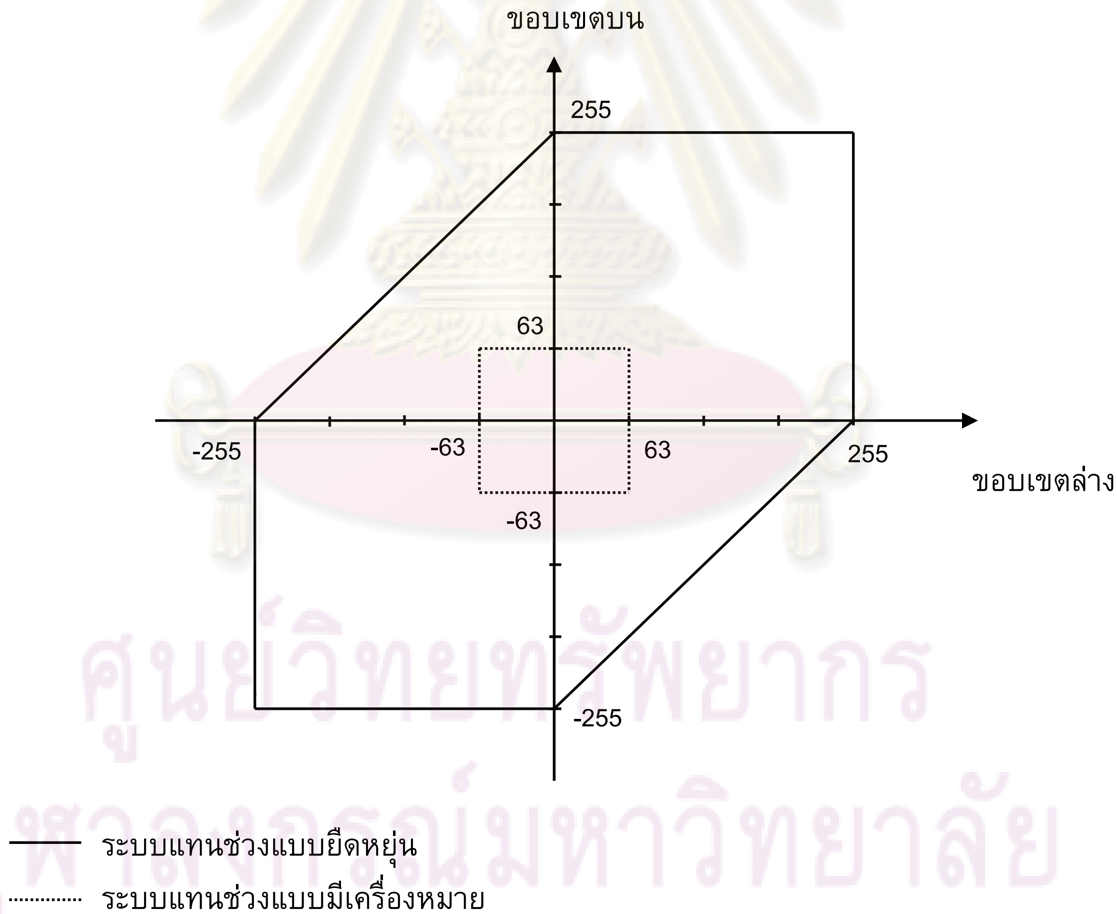
บทที่ 4

บทวิเคราะห์ระบบแทนช่วงแบบยืดหยุ่น

เนื้อหาในบทนี้กล่าวถึงบทวิเคราะห์เกี่ยวกับระบบแทนช่วงแบบยืดหยุ่นเทียบกับระบบแทนช่วงแบบมีเครื่องหมายแบบดั้งเดิม โดยมีหัวข้อดังต่อไปนี้

4.1 การเปรียบเทียบเนื้อที่ที่ใช้ในการแสดงช่วง

เนื้อที่ที่ถูกใช้ในรูปแบบการแทนช่วงในระบบแทนช่วงแบบยืดหยุ่นนั้นมีประสิทธิภาพมากกว่าระบบแทนช่วงแบบมีเครื่องหมาย ซึ่งภาพด้านล่างแสดงให้เห็นถึงการเปรียบเทียบช่วงที่สามารถแสดงได้ในระบบแทนช่วงแบบยืดหยุ่นและระบบแทนช่วงแบบมีเครื่องหมายในพื้นที่ 24 บิต



รูปที่ 4.1 แสดงการเปรียบเทียบช่วงที่สามารถแสดงได้ในระบบแทนช่วงแบบยืดหยุ่นและระบบแทนช่วงแบบมีเครื่องหมายในพื้นที่ 24 บิต

4.2 การเปรียบเทียบความซ้ำซ้อนของระบบ

การเปรียบเทียบในหัวข้อนี้จะใช้ระดับความซ้ำซ้อน (redundancy degree) เป็นตัววัด ซึ่งระบบใดมีความซ้ำซ้อนสูงก็จะมีค่าระดับความซ้ำซ้อนสูงตามไปด้วย โดย ถ้าให้ n เป็นความยาวของบิตของข้อมูลที่สนใจ ระดับความซ้ำซ้อนหาได้จาก

ระดับความซ้ำซ้อน = 2^n / จำนวนที่เป็นไปได้ในระบบนั้น ๆ ที่ n บิต

4.2.1 ระดับความซ้ำซ้อนของระบบแทนช่วงแบบมีเครื่องหมาย

สำหรับระบบแทนช่วงแบบมีเครื่องหมายทุกหนึ่งหลักของช่วง ประกอบไปด้วยสี่บิต ทุกสองบิตแทนค่าตัวเลขหนึ่งตัว ดังนั้นข้อมูลความยาว n บิตจะมีตัวเลขทั้งหมด n/2 หลัก ซึ่งแนวคิดในการนับรูปแบบคือ เราจะไม่นับรูปแบบที่มีตัวเลขที่เป็นตัวผกผันของการบวกอยู่ด้วยกัน (เช่น 1 อยู่กับ $\bar{1}$) เพราะจะเป็นรูปแบบที่ซ้ำซ้อนกับบางจำนวน (เช่น $1\bar{1}$ มีค่าเท่ากับ 01) ดังนั้นเราจะแยกการนับออกเป็น 2 กรณีคือ นับรูปแบบที่มี 0 และ 1 เท่านั้นซึ่งมี $2^{n/2}$ รูปแบบ และรูปแบบที่มี $\bar{1}$ และ 0 เท่านั้นซึ่งมี $2^{n/2}$ รูปแบบ และลบจำนวนที่ซ้ำกันคือ 00…0 ออก 1 ตัว ดังนั้น

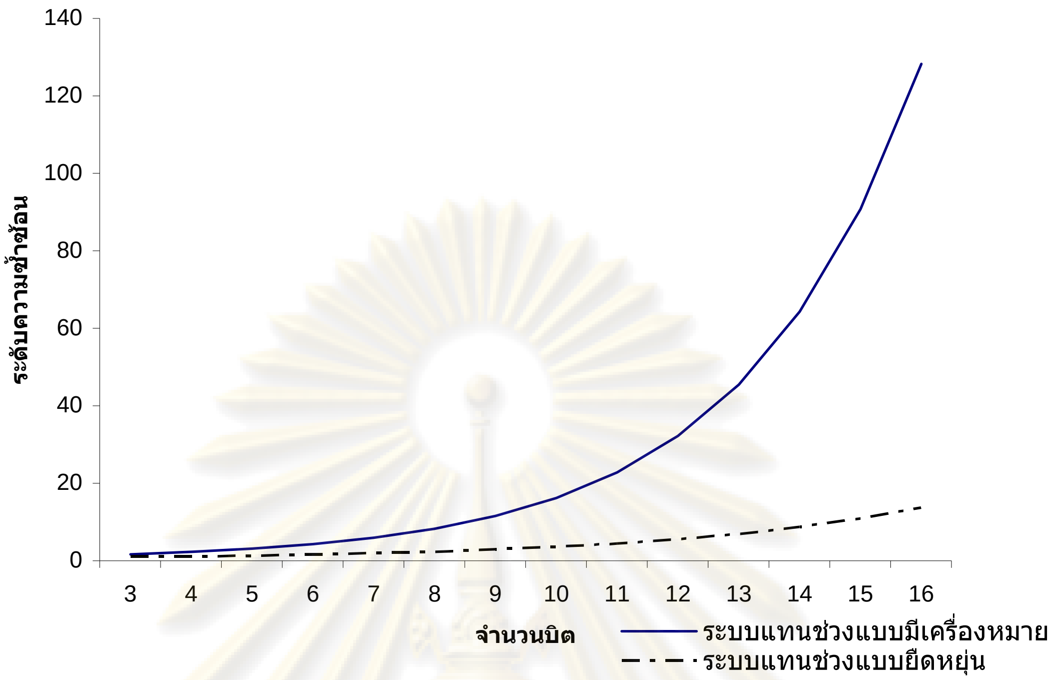
ระดับความซ้ำซ้อนของระบบแทนช่วงแบบมีเครื่องหมาย = $2^n / (2^{n/2+1}-1)$

4.2.2 ระดับความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น

สำหรับระบบแทนช่วงแบบยืดหยุ่นทุกหนึ่งหลักของช่วง ประกอบไปด้วยสามบิต ทุกสามบิตแทนค่าตัวเลขหนึ่งตัว ดังนั้นข้อมูลความยาว n บิต จะมีตัวเลขทั้งหมด n/3 หลัก สำหรับแนวคิดในการนับรูปแบบของระบบแทนช่วงแบบยืดหยุ่นคือ นับช่วงที่เป็นไปได้ทั้งหมดที่อยู่ใน $[-(2^{n/3}-1), 2^{n/3}-1]$ ซึ่งขนาดของช่วงที่เป็นไปได้จะต้องไม่เกิน $2^{n/3}-1$ อีกด้วย ดังนั้น

ระดับความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น = $2^n / [3*(2^{n/3}-1)^2 + 4*(2^{n/3}-1)]$

ซึ่งเมื่อพิจารณาระดับความซ้ำซ้อนที่จำนวนบิตเท่ากัน รูปที่ 4.2 แสดงให้เห็นถึงระดับความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น เปรียบเทียบกับระดับความซ้ำซ้อนของระบบแทนช่วงแบบมีเครื่องหมาย



รูปที่ 4.2 แสดงการเปรียบเทียบระดับความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่นและระบบแทนช่วงแบบมีเครื่องหมาย

จากรูปจะเห็นได้ว่าเมื่อพิจารณาที่จำนวนบิตเท่ากัน ระดับความซ้ำซ้อนของระบบแทนช่วงแบบมีเครื่องหมายมีค่าสูงกว่าระดับความซ้ำซ้อนของระบบแทนช่วงแบบยืดหยุ่น และความแตกต่างจะเห็นได้ชัดเจนขึ้นเมื่อจำนวนบิตมีค่ามากขึ้นไปเรื่อยๆ นั่นหมายความว่าระบบแทนช่วงแบบยืดหยุ่น มีระดับความซ้ำซ้อนน้อยกว่าทั้งที่ยังสามารถทำการคำนวณแบบขนานได้เช่นเดียวกัน ทำให้สามารถสรุปได้ว่าระบบแทนช่วงแบบยืดหยุ่น ใช้เนื้อที่น้อยกว่าระบบแทนช่วงแบบมีเครื่องหมาย

บทที่ 5

สรุปผลงานวิจัยและข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้ปรับปรุงและนำเสนอระบบแทนช่วงแบบยืดหยุ่นให้สามารถทำการคำนวณแบบขนานได้ ซึ่งระบบแทนช่วงแบบยืดหยุ่นนี้สามารถใช้จำนวนเพียงจำนวนเดียวในการแสดงช่วง เราได้เพิ่มตัวเลขเข้าไปในระบบอีกสองตัวเพื่อเพิ่มความซ้ำซ้อนของระบบให้มากขึ้น การเพิ่มตัวเลขเข้าไปในระบบไม่ส่งผลกระทบต่อการเพิ่มเนื้อที่ในการแสดงตัวเลขในช่วง นั่นคือยังคงใช้ 3 บิตเช่นเดิมในการแสดงช่วง และความซ้ำซ้อนที่เพิ่มขึ้นนี้ส่งผลให้ระบบสามารถทำการคำนวณแบบขนานได้

นอกจากนี้ เรายังได้นำเสนอระบบแทนช่วงแบบยืดหยุ่นในรูปทั่วไป และอัลกอริทึมสำหรับการดำเนินการพื้นฐานทางเลขคณิตแบบขนาน ได้แก่ การบวกและการลบแบบขนานในระบบแทนช่วงแบบยืดหยุ่นอีกด้วย

5.2 ข้อเสนอแนะ

สำหรับในงานวิจัยนี้สิ่งที่ต้องการจะเสนอแนะคือ การออกแบบการดำเนินการพื้นฐานทางเลขคณิตอื่นๆได้แก่ การคูณแบบขนานและการหารในระบบแทนช่วงแบบยืดหยุ่น ซึ่งเมื่อการออกแบบการดำเนินการพื้นฐานทางเลขคณิตอื่น ๆ สมบูรณ์ จะส่งผลให้ความเร็วในการคำนวณโดยรวมของระบบลดลงเมื่อนำไปใช้งานจริง

รายการอ้างอิง

[1] Skeel, Robert. Roundoff Error and the Patriot Missile. *SIAM News* 25, 11 (July 1992).

[2] Brian Hayes. A Lucid Interval. *American Scientist* 91, 6 (2003): 484-488.

[3] Shettar R., Banakar R.M, and Nataraj, P.S.V.. Design and Implementation of Interval Arithmetic Algorithms. *in. Proc. International Conference on Industrial and Information Systems*, (August 2006): 328-331.

[4] A. Avizienis. Signed-digit number representation for fast parallel arithmetic. *IRE Trans. Comput*. 10 (1961): 389-400.

[5] B. Parhami. Generalized Signed-Digit Number Systems: A Unifying Framework for Redundant Number Representations. *IEEE Trans. Comput.* 39, 1 (January 1990): 89-98.

[6] M. Daumas, G. Melquiond, and C. Muñoz. Guaranteed Proofs Using Interval Arithmetic. *in Proc. International Symposium on Computer Aritmetic*, (June 2005): 188-195.

[7] Moore, Ramon E.. Methods and Applications of Interval Analysis. *Society for Industrial and Applied Mathematics*, Philadelphia, (1979).

[8] R. B. Keafott. Interval computations: Introduction, uses, and resources. *Euromath Bulletin* 2(1) (1996): 95-112.

[9] P. Kornerup. Digit-Set Conversions: Generalizations and Applications. *IEEE Transactions on Computers* 43 (1994): 622-629.

[10] B. Phillips and N. Burgess. Minimal Weight Digit Set Conversions. *IEEE Transactions on Computers* 53 (2004): 666-677.

[11] M. D. Ercegovac and T. Lang. On-the-fly Conversion of Redundant into Conventional Representations. *IEEE Transactions on Computers* (1985): 895-897.

[12] Pipop Thienprapasith. *Flexible Interval Representation System*. Master's Thesis, Department of Computer Engineering, Graduate School, Chulalongkorn University, 2008.

[13] J. Sukontarach, P. Thienprapasith and A. Surarerks. Modified Arithmetic Algorithm for Flexible Interval Binary Representation System. *Proceedings of the 12th Annual National Symposium on Computational Science and Engineering (ANSCSE12)*, (2008).

[14] J. Sukontarach and A. Surarerks. Parallel Addition and Subtraction for the Flexible Interval Representation System. *Proceedings of the 12th National Computer Science and Engineering Conference (NCSEC2008)*, (2008).

[15] J. Sukontarach and A. Surarerks. Parallel Addition and Subtraction for the Flexible Interval Representation System. *International Conference on Theoretical and Mathematical Foundations of Computer Science (TMFCS-09)*, (2009).

ประวัติผู้เขียนวิทยานิพนธ์

นายจักรพันธุ์ สุคนธราช เกิดเมื่อวันที่ 18 สิงหาคม พ.ศ. 2528 เรียนจบการศึกษาระดับมัธยมศึกษาตอนต้นจากโรงเรียนอัสสัมชัญ สมุทรปราการและตอนปลายจากโรงเรียนเตรียมอุดมศึกษา จังหวัดกรุงเทพมหานคร เข้ารับการศึกษาต่อในระดับปริญญาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ จนสำเร็จการศึกษาในปีการศึกษา 2549