อัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

นายศิษฎิวัชร์ เสริมสุขสกุลชัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2554

ON-LINE DIVISION ALGORITHM ON FLEXIBLE INTERVAL REPRESENTATION SYSTEM

Mr.Sitthivat Sermsuksakulchai

A Thesis Submitted in Partial Fulfillment of the Requirements

for the Degree of Master of Science Program in Computer Science

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2011

หัวข้อวิทยานิพนธ์	อัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น
โดย	นายศิษฏิวัชร์ เสริมสุขสกุลชัย
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

…………………………………………… ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ดร.เศรษฐา ปานงาม)

………………………………………….... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ดร.อรรถสิทธิ์ สุรฤกษ์)

…………………………………………… กรรมการภายนอกมหาวิทยาลัย
(ผู้ช่วยศาสตราจารย์ดร.อานนท์ รุ่งสว่าง)

ศิษฎิวัชร์ เสริมสุขสกุลชัย : อัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น. (ON-LINE DIVISION ALGORITHM ON FLEXIBLE INTERVAL REPRESENTATION SYSTEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.อรรถสิทธิ์ สุรฤกษ์, 29 หน้า.

ระบบจำนวนแทนช่วงได้ถูกเสนอขึ้นมาเพื่อจัดการกับข้อผิดพลาดที่เกิดขึ้นจากการปัดเศษในระหว่างการคำนวณหรือจากข้อมูลนำเข้า แต่ประสบปัญหาด้านความสิ้นเปลืองเนื้อที่และความล่าช้าในการคำนวณ หลังจากนั้นระบบแทนช่วงแบบยืดหยุ่นได้ถูกพัฒนาต่อจากระบบแทนช่วงซึ่งสามารถแสดงค่าของช่วงด้วยจำนวนเพียงชุดเดียวทำให้ใช้พื้นที่ที่ใช้ในการแทนช่วงน้อยลงส่งผลให้เวลาที่ใช้ในการคำนวณน้อยลงด้วย อย่างไรก็ตามระบบแทนช่วงแบบยืดหยุ่นไม่สามารถทำการคำนวณแบบขนานได้ทำให้เกิดความล่าช้าในการคำนวณข้อมูลที่มีขนาดใหญ่

งานวิจัยนี้มุ่งเน้นที่จะทำให้ระบบแทนช่วงแบบยืดหยุ่นสามารถทำการหารแบบเชื่อมตรงได้โดยการปรับปรุงระบบแทนช่วงแบบยืดหยุ่นให้อยู่ในรูปแบบบรรทัดฐานเพื่อให้สามารถทำการหารแบบเชื่อมตรงได้ ด้วยความหน่วงเท่ากับห้า จากนั้นเราเสนออัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นพร้อมทั้งบทพิสูจน์

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต ....

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก ....

ปีการศึกษา 2554

5371449521 : MAJOR COMPUTER SCIENCE

KEYWORDS : COMPUTER ARITHMETIC / ON-LINE ARITHMETIC / FLEXIBLE INTERVAL REPRESENTATION

SITTHIVAT SERMSUKSAKULCHAI : ON-LINE DIVISION ALGORITHM ON FLEXIBLE INTERVAL REPRESENTATION SYSTEM. ADVISOR : ASST.PROF. ATHASIT SURARERKS, 29 pp.

Interval arithmetic has been introduced in order to handle an round-off error problem in the computation model. But the space used and computational time for interval arithmetic is very high. Flexible interval representation system (FIRS) is one of the recently proposed number systems. Its representation can be expressed by one sequence of digits which computational time can be speed up. However, flexible interval representation system cannot be used in parallel computation. Therefore, it takes much computational time when the data size becomes large.

In this work, we are interested in an on-line division operation for this interval system. In detail, we proposed an on-line division algorithm in the FIRS with base two that the division can be performed with the delay five. The proof of correctness is demonstrated.

Department : Computer Engineering Student's Signature

Field of Study : Computer Science Advisor's Signature

Academic Year : 2011

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จเรียบร้อยได้ด้วยดีเพราะได้รับคำแนะนำ และ ให้คำปรึกษาจาก ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษาซึ่งเป็นผู้ชี้แนะแนวทางในการศึกษา และให้ข้อเสนอแนะที่เป็นประโยชน์ยิ่งต่อการวิจัย จนเกิดเป็นวิทยานิพนธ์ฉบับนี้ขึ้นนอกจากนี้ ขอขอบคุณคณะกรรมการสอบวิทยานิพนธ์ ซึ่งได้แก่ ผู้ช่วยศาสตราจารย์ ดร.เศรษฐา ปานงาม ผู้ซึ่งเป็นประธาน ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง ผู้ซึ่งเป็นกรรมการ ที่ได้สละเวลาอันมีค่ามาใช้ในการชี้แจงถึงข้อบกพร่อง รวมถึงแนวทางการแก้ไขและข้อแนะนำดีๆ ให้แก่ข้าพเจ้า

ขอขอบพระคุณ คุณพ่อ คุณแม่ รวมไปถึง พี่ๆ ที่ทำงาน และ เพื่อนๆ ที่คอยให้คำแนะนำ โดยเฉพาะสมาชิกห้องปฏิบัติการวิจัย ELITE เมื่อข้าพเจ้าพบปัญหาต่างๆ ในขณะทำงานวิจัยเล่มนี้ทำให้วิทยานิพนธ์เล่มนี้สำเร็จสมบูรณ์ไปด้วยดี และยิ่งไปกว่านั้นต้องขอขอบคุณต่อเจ้าหน้าที่ของภาควิศวกรรมศาสตร์คอมพิวเตอร์ทุกท่านที่อำนวยความสะดวกในทุกๆ เรื่องที่เกี่ยวกับการศึกษา

สุดท้ายนี้ ขอขอบพระคุณทุกท่านอีกทีที่คอยส่งเสริมและให้กำลังใจ รวมถึงคำปรึกษาที่ดีในการศึกษาและทำงานวิจัยให้ประสบความสำเร็จในจุฬาลงกรณ์มหาวิทยาลัยแห่งนี้

สารบัญ

สารบัญตาราง

หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในวงการวิจัยทางด้านเลขคณิตสำหรับคอมพิวเตอร์ (computer arithmetic) ซึ่งเป็นงานวิจัยด้านการออกแบบรูปแบบและวิธีการคำนวณทางคณิตศาสตร์สำหรับคอมพิวเตอร์โดยทั่วไปมีวัตถุประสงค์หลักคือ การเพิ่มประสิทธิภาพของการคำนวณในด้าน ความถูกต้อง และความเร็ว ปัจจัยหนึ่งที่มีผลต่อประสิทธิภาพคือรูปแบบแสดงจำนวน (number representation) และการออกแบบตัวแบบการคำนวณ (computational model) ซึ่งมีงานวิจัยมากมายที่นำเสนอรูปแบบต่าง ๆ โดยรูปแบบแทนจำนวนที่มีประสิทธิ์ภาพสูงรูปแบบหนึ่ง คือ ระบบรูปแบบแทนจำนวนแบบมีเครื่องหมายของอวีเซียนีส(Avizienis signed digit number representation system) ได้ถูกเสนอขึ้นโดยอวีเซียนีส (Avizienis) ในปี ค.ศ. 1961 [1]ระบบจำนวนนี้นำเสนอให้ใช้ตัวเลขแบบมีเครื่องหมาย (signed digit) ร่วมกับระบบจำนวนเลขฐาน (radix number system) จากงานวิจัยพบว่า การคำนวณของตัวดำเนินการพื้นฐานทางเลขคณิต(fundamental arithmetic operations) ในระบบนี้ การแพร่กระจายของตัวทดมีขอบเขตที่จำกัด และสามารถนำไปสู่วิธีการคำนวณที่มีความเร็วสูงขึ้นได้

ระบบแทนช่วง(interval representation system)[2] เป็นระบบจำนวนที่แสดงค่าของช่วงด้วยค่าสูงสุดและต่ำสุดระบบนี้สามารถนำมาใช้ในการแก้ปัญหาความคลาดเคลื่อนที่เกิดขึ้นจากการคำนวณได้ เช่นในบางครั้งการคำนวณอาจมีความจำเป็นต้องปัดเศษทิ้งไปในบางตำแหน่ง เป็นต้น นอกจากนี้การคำนวณบนจำนวนที่มีความคลาดเคลื่อนจากปัจจัยอื่น ๆ เช่น ข้อมูลที่ได้มาจากเครื่องมือวัดที่มีความคลาดเคลื่อนได้ เป็นต้น ระบบแทนช่วงนี้สามารถใช้ในการอธิบายผลการคำนวณบนค่าที่มีความคลาดเคลื่อนได้เช่นกันแต่ระบบแทนช่วงมีข้อเสียในด้านการคำนวณช้าและสิ้นเปลืองเนื้อที่ ในปี ค.ศ. 2006 ได้มีการเสนอระบบแทนช่วงแบบยืดหยุ่น(flexible interval representation system)โดย พิภพ[3-4] ซึ่งใช้แนวคิดของตัวเลขยืดหยุ่น (flexible digits) โดยสามารถแสดงค่าของช่วงด้วยลำดับของตัวเลขชุดเดียว ระบบนี้สามารถลดขนาดของเนื้อที่ที่ใช้ในการเก็บจำนวนเมื่อเทียบกับระบบแทนช่วงทั่วไป และยังสามารถนำไปสู่การลดเวลาที่ใช้ในการคำนวณให้น้อยลง

สำหรับงานวิจัยทางด้านตัวแบบการคำนวณนั้น แนวทางการคำนวณแบบขนาน (parallel computation) เป็นตัวแบบที่เป็นที่ยอมรับว่ามีประสิทธิภาพเชิงเวลาสูง แต่สิ้นเปลือง

ทรัพยากรมากเช่นเดียวกัน ในปี ค.ศ. 1977 ทราเวดีและเออเซโกวัก (Trivedi and Ercegovac)[5] ได้นำเสนอแนวทางการประมวลผลด้วยตัวแบบเชื่อมตรง (on-line computation) บนระบบรูปแบบแทนจำนวนแบบมีเครื่องหมายของอวีเซียนีสลักษณะของการคำนวณจะเริ่มทำการคำนวณจากหลักที่มีนัยสำคัญมากที่สุด (most significant digit) ไปยังหลักที่มีนัยสำคัญน้อยที่สุด (least significant digit)สิ่งสำคัญสำหรับระบบนี้คือการผลิตคำตอบสำหรับตัวเลขแรกหากไม่สามารถกระทำได้ทันทีและจำเป็นต้องใช้ตัวเลขของตัวถูกดำเนินการเป็นจำนวน δ ตัวเลข ตัวดำเนินการนี้เรียกว่าค่าความหน่วงเชื่อมตรง(on-line delay δ)

การคำนวณแบบเชื่อมตรงสำหรับระบบแทนช่วงแบบยืดหยุ่นได้มีการศึกษาแล้วในการดำเนินการบวก การลบ และการคูณ[6-7] ในงานวิจัยนี้งานวิจัยนี้มุ่งเน้นที่จะเสนออัลกอริทึมการหารแบบเชื่อมตรงสำหรับระบบแทนช่วงแบบยืดหยุ่นพร้อมทั้งพิสูจน์ความถูกต้องและหาค่าความหน่วงที่เหมาะสม

1.2 วัตถุประสงค์ของการวิจัย

เพื่อสร้างอัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น พร้อมทั้งหาค่าความหน่วงที่เหมาะสมกับระบบแทนช่วงแบบยืดหยุ่นนี้

1.3 ขอบเขตของการวิจัย

1. อัลกอริทึมการหารแบบเชื่อมตรงที่พัฒนาขึ้นสามารถนำไปใช้กับระบบแทนช่วงแบบยืดหยุ่นเท่านั้น
2. ค่าความหน่วงที่หาได้นี้สามารถนำไปใช้กับการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นเท่านั้น

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้อัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น
2. ได้ค่าความหน่วงที่เหมาะสมสำหรับการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

1.5 วิธีดำเนินการวิจัย

1. ศึกษางานวิจัยทางด้านการคำนวณแบบเชื่อมตรงและการคำนวณแบบช่วง

2. วิเคราะห์ปัญหาของงานวิจัยที่มีความสอดคล้องกับงานวิจัยที่สนใจ
3. ออกแบบอัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น
4. พิสูจน์อัลกอริทึมที่ได้ออกแบบไว้
5. สรุปผลการวิจัยและจัดทำรายงานวิทยานิพนธ์

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นผลงานวิชาการในหัวข้อเรื่องดังต่อไปนี้

1. "On-line Division for Flexible Number System" โดย ศิษฏิวัชร์ เสริมสุขสกุลชัย และ อรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ Proceeding of The 16th International Annual Symposium on Computational Science and Engineering (ANSCSE16)

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1.ระบบแทนช่วง(Interval representation system)

การแทนจำนวนในระบบคอมพิวเตอร์นั้นเป็นการแทนจำนวนในรูปแบบที่จำกัด (finite representation of number) ทำให้การคำนวณสำหรับจำนวนจริงนั้นเกิดความผิดพลาดที่เกิดจากการปัดเศษทศนิยมในตำแหน่งที่ไม่สามารถแสดงผลได้ในระหว่างการคำนวณ (round-off error)ผลลัพธ์ที่ได้จึงมีความคลาดเคลื่อนเกิดขึ้นและความคลาดเคลื่อนจะสะสมเพิ่มขึ้นเมื่อนำผลลัพธ์ไปคำนวณต่อ ระบบแทนช่วงใช้แนวคิดในการแสดงจำนวนต่างๆให้อยู่ในรูปของช่วงที่ครอบคลุมค่าที่ถูกต้องของจำนวนจริง

ระบบแทนช่วงที่นิยมทั่วไป จะเป็นระบบที่แทนช่วงใดๆ ด้วยจำนวนสองจำนวน คือ ค่าขอบเขตล่าง (lower bound) และขอบเขตบน (upper bound) ซึ่งหมายถึงเซตของจำนวนจริงทั้งหมดที่มีค่าตั้งแต่ค่าขอบเขตล่างจนถึงค่าขอบเขตบนสำหรับรูปแบบช่วง X ใดๆในระบบแทนช่วง รูปแบบการแทนช่วงของ Xสามารถเขียนได้ในรูปแบบของ

$$X = [x_L x_U],$$

โดยที่ x_Lและ x_U หมายถึงค่าขอบเขตล่าง และ ค่าขอบเขตบนตามลำดับ

การดำเนินการพื้นฐานทางเลขคณิตบนระบบแทนช่วง (fundamental arithmetic operations for interval representation system)อันได้แก่ การบวก การลบ การคูณ และการหาร สำหรับช่วง $X = [x_L, x_U]$ และ$Y = [y_L, y_U]$ สามารถสรุปได้ดังนี้

$$\begin{aligned}
X + Y &= [x_L + y_L, x_U + y_U] \\
X - Y &= [x_L - y_U, x_U - y_L] \\
X \times Y &= [\min(x_L \times y_L, x_L \times y_U, x_U \times y_L, x_U \times y_U), \\
&\quad \max(x_L \times y_L, x_L \times y_U, x_U \times y_L, x_U \times y_U)] \\
X \div Y &= [\min(x_L \div y_L, x_L \div y_U, x_U \div y_L, x_U \div y_U), \\
&\quad \max(x_L \div y_L, x_L \div y_U, x_U \div y_L, x_U \div y_U)]
\end{aligned}$$

สำหรับการหารนั้น ช่วงที่เป็นตัวหารจะต้องไม่ครอบคลุมศูนย์

ข้อสังเกตอย่างหนึ่งคือ การคำนวณในระบบแทนช่วงนั้น สิ้นเปลืองทรัพยากรที่ใช้ในการคำนวณ ทั้งนี้เนื่องจากว่า แต่ละช่วงแสดงด้วยจำนวนสองจำนวน การคำนวณทั้งหมดจึงมีภาระมากตามไปด้วย

ตัวอย่างที่ 2.1*กำหนดให้ X = [3.4, 5.1] , Y = [4.25, 7.1] จงแสดงการบวก ลบ คูณ และหารของช่วง X และ Y*

วิธีทำ การคำนวณพื้นฐานของตัวดำเนินการบวก ลบ คูณ และหารของช่วง X และ Y สามารถแสดงได้ดังนี้

X + Y = [7.65, 12.2]

X - Y = [-0.85, -2.0]

X × Y = [min(14.45, 24.14, 21.68, 36.21) , max(14.45, 24.14, 21.68, 36.21)]

= [14.45, 36.21]

X ÷ Y = [min(0.8, 0.48, 1.2, 0.72) , max(0.8, 0.48, 1.2, 0.72)]

= [0.48, 1.2] □

ระบบแทนช่วงแบบยืดหยุ่น (Flexible interval representation system)

ระบบแทนช่วงแบบยืดหยุ่นนำเสนอโดยพิภพ ดังปรากฏใน[3] ในงานวิจัยนี้มีแนวคิดคือการใช้ลำดับของตัวเลขเพียงชุดเดียวในการแสดงค่าของช่วงได้ ทั้งนี้เพื่อลดขนาดของรูปแบบแทนช่วง และเมื่อช่วงสามารถแทนได้ด้วยลำดับของตัวเลขชุดเดียว การคำนวณบนระบบดังกล่าว น่าจะมีประสิทธิภาพในเชิงเวลาสูงขึ้น ตัวอย่างเช่น การคูณและการหาร ซึ่งจะต้องทำการคูณหรือหารหารย่อยสี่ครั้งสำหรับการคำนวณของช่วงแต่ละคู่ ตัวเลขที่ใช้ในระบบนี้แต่ละตัวจะแสดงค่าขอบเขตล่างและของเขตบนไว้ในตัวเลขเพียงตัวเดียวเรียกว่า *ตัวเลขยืดหยุ่น (flexible digit)* โดยระบบแทนช่วงแบบยืดหยุ่นมีนิยามดังต่อไปนี้

นิยามที่ 2.1 ระบบแทนช่วงแบบยืดหยุ่น (flexible interval representation system) *ประกอบด้วยเลขฐานจำนวนเต็ม* β *และชุดตัวเลข (Digit set)* F *โดยที่* $\beta = 2$ *และ* $F = \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$ *ซึ่ง* $\gamma, \alpha, \bar{\gamma}, \bar{\alpha}$ *เรียกว่า* ตัวเลขยืดหยุ่น (flexible digit) *โดยที่* $\bar{\gamma}, \bar{\alpha}$ *เป็นตัวผกผันการบวกของ* γ *และ* α *ตามลำดับ*

โดยค่าขอบเขตล่างและค่าขอบเขตบนของตัวเลขยืดหยุ่นสามารถระบุค่าได้ดังแสดงไว้ในตารางที่ 2.1

ตารางที่ 2.1 ขอบเขตล่างและขอบเขตบนของตัวเลขยืดหยุ่น

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

นิยามที่ 2.2 รูปแบบแทนช่วง $X = \mathrm{x}_n\mathrm{x}_{n-1}\mathrm{x}_{n-2} \ldots \mathrm{x}_0$ *ในระบบแทนช่วงแบบยืดหยุ่น โดยที่* x_0, x_1, x_2, … ,x_n*เป็นตัวเลขในชุดตัวเลข* *F สามารถแสดงช่วง[A,B] ได้โดย*

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

สมบัติที่สำคัญของระบบแทนช่วงแบบยืดหยุ่นคือ ระบบนี้สามารถแสดงช่วงได้โดยใช้จำนวนเพียงจำนวนเดียว(ชุดตัวเลขชุดหนึ่ง) นอกจากนี้ระบบแทนช่วงแบบยืดหยุ่นมีคุณสมบัติเป็น ระบบแทนจำนวนซ้ำซ้อนด้วย ดังนิยามต่อไปนี้

นิยามที่ 2.3 ระบบแทนจำนวนซ้ำซ้อน (Redundant number representation system) *หมายถึง ระบบจำนวนที่มีอย่างน้อยหนึ่งจำนวนสามารถหารูปแบบแทนจำนวนได้มากกว่าหนึ่งรูปแบบ*

ตัวอย่างต่อไปนี้ เป็นตัวอย่างแสดงให้เห็นว่า ช่วงสามารถแสดงให้อยู่ในรูปแบบของ รูปแบบแทนช่วงแบบยืดหยุ่นได้ และมีคุณสมบัติเป็นระบบแทนจำนวนซ้ำซ้อนด้วย ดังตารางที่ 2.2

ตัวอย่างที่ 2.2 แสดงว่าสามารถ*หารูปแบบแทนช่วงแบบยืดหยุ่นของช่วง [3,19] ได้*

วิธีทำ[3,9] สามารถเขียนให้อยู่ในรูปแบบแทนช่วงแบบมีเครื่องหมายได้เป็น [11,1001]
เพราะว่า $\alpha\mathbf{0011}$แสดงเป็นรูปแบบแทนช่วงได้คือ [00011,10011]ซึ่งมีค่าเท่ากับ [3,19]
และ $\alpha\bar{\alpha}\mathbf{0011}$แสดงเป็นรูปแบบแทนช่วงได้คือ [000011,1$\bar{1}$0011]ซึ่งมีค่าเท่ากับ [3,19]
จะเห็นว่า [3,19] สามารถเขียนเป็นรูปแบบแทนช่วงแบบยืดหยุ่นได้มากกว่าหนึ่งรูปแบบดังนั้น รูปแบบแทนช่วงแบบยืดหยุ่นมีคุณสมบัติเป็นระบบแทนจำนวนซ้ำซ้อน □

ตารางที่ 2.2 ตัวอย่างรูปแบบแทนช่วงตัวเลขแบบยืดหยุ่น

ช่วง	ระบบแทนช่วงแบบยืดหยุ่น
$[3,19]$	$\alpha 0011, \alpha\bar{\alpha}0011$
$[-3,4]$	$\alpha\gamma\gamma$
$[0.625,1.625]$	$\alpha.11$
$[-15,-20]$	$\alpha\gamma\alpha 0\bar{\gamma}$

ในด้านประสิทธิภาพเชิงเนื้อที่ ระบบแทนช่วงแบบยืดหยุ่นเมื่อเทียบกับระบบแทนจำนวนแบบมีเครื่องหมายแบบดั้งเดิม (Classical signed-digit number system) แล้ว เราพิสูจน์ได้ว่า เนื้อที่สามารถลดดลงได้อย่างน้อย 25%[3]

สำหรับตัวดำเนินการพื้นฐานทางเลขคณิตสำหรับระบบแทนช่วงแบบยืดหยุ่นซึ่งประกอบไปด้วยการบวก การลบ การคูณและการหาร การดำเนินการพื้นฐานของระบบแทนช่วงแบบยืดหยุ่นไม่สามารถทำการคำนวณแบบทั่วไปได้ ต้องมีการประยุกต์ด้วยความรู้เลขคณิตของระบบแทนช่วงมาใช้ สามารถดำเนินการได้ดังตัวอย่างต่อไปนี้

ตัวอย่างที่ 2.3*การบวกของช่วง [1,7] กับ [3,11] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ ขั้นตอนแรก การแสดงช่วง [1,7] และ [3,11] ให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

[1,7] สามารถเขียนได้เป็น$0\alpha\alpha 1$

[3,11] สามารถเขียนได้เป็น$\alpha 011$

ขั้นตอนที่สอง ดำเนินการบวกบนระบบแทนช่วงแบบยืดหยุ่น ด้วยกฎการบวก

$0\,0\,\alpha\,\alpha\,1_{+}$ ↔ [1,7]

$\underline{0\,\alpha\,0\,1\,1}$ ↔ [3,11]

$\underline{1\,\gamma\gamma\,\alpha\,0}$ ↔ [4,18]

ผลลัพธ์ของการบวกในระบบแทนช่วงแบบยืดหยุ่น$1\gamma\gamma\alpha 0$สามารถให้คำตอบที่ถูกต้อง □

ตัวอย่างที่ 2.4*การลบของช่วง [1,7] กับ [3,11] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ ขั้นตอนแรก การแสดงช่วง [1,7] และ [3,11] ให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

[1,7] สามารถเขียนได้เป็น $0\alpha\alpha 1$

[3,11] สามารถเขียนได้เป็น$\alpha 011$

ขั้นตอนที่สอง แปลงให้อยู่ในรูปแบบแทนลบ

$0\alpha\alpha1$ สามารถเขียนได้เป็น $0\alpha\alpha1$

$\alpha011$ สามารถเขียนได้เป็น $\gamma0\overline{1}\overline{1}$

ขั้นตอนที่สาม ดำเนินการบวกบนระบบแทนช่วงแบบยืดหยุ่นด้วยกฎการบวก

$0\ \alpha\ \alpha\ 1$	$\leftrightarrow$	[1,7]
$\underline{\alpha\ 0\ 1\ 1}$	$\leftrightarrow$	[3,11]
$\underline{\gamma\ \alpha\ \gamma\ 0}$	$\leftrightarrow$	[-10,4]

ผลลัพธ์ของการลบในระบบแทนช่วงแบบยืดหยุ่น$\gamma\alpha\gamma0$ สามารถให้คำตอบที่ถูกต้อง □

ตัวอย่างที่ 2.5*การคูณของช่วง [1,7] กับ [3,11] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ ขั้นตอนการคูณของช่วง X = [1,7] กับ Y =[3,11] แสดงได้ดังขั้นตอนต่อไปนี้

ขั้นตอนแรก การแปลงช่วง [1,7] และ [3,11] ให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

[1,7] สามารถเขียนได้เป็น $0\alpha\alpha1$

[3,11] สามารถเขียนได้เป็น $\alpha011$

ขั้นตอนที่สอง ตรวจสอบเครื่องหมายของ X และ Y

เครื่องหมายของ X คือ positive

เครื่องหมายของ Y คือ positive

ขั้นตอนที่สาม จัดให้อยู่ในรูปแบบพร้อมคูณ

รูปแบบพร้อมคูณ ของ X คือ $0\alpha\alpha1$

รูปแบบพร้อมคูณ ของ Y คือ $\alpha011$

ขั้นตอนที่สี่ ดำเนินการคูณบนระบบแทนช่วงแบบยืดหยุ่น ด้วยกฎการคูณ

$0\alpha\alpha1 \times \alpha011 = \mathbf{100\alpha1\gamma1}$ = [3,77]

ผลลัพธ์ของการคูณในระบบแทนช่วงแบบยืดหยุ่น คือ$\mathbf{100\alpha1\gamma1}$ โดยมีค่าเชิงตัวเลขเท่ากับ [3,77] ดังนั้นผลลัพธ์ของการคูณในระบบแทนช่วงแบบยืดหยุ่นสามารถให้คำตอบที่ถูกต้อง

ตัวอย่างที่ 2.6 *การหารของช่วง [-72, 95] กับ [-22, 35] ในระบบแทนช่วงแบบยืดหยุ่น*

วิธีทำ ขั้นตอนการหารของช่วง X = [-72, 95] กับ Y =[-22, 35]แสดงได้ดังขั้นตอนต่อไปนี้

ขั้นตอนแรก การแปลงช่วง [-72, 95]และ [-22, 35]ให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

[-72, 95]สามารถเขียนได้เป็น $\alpha\gamma\bar{\alpha}0\gamma00\bar{\alpha}$

[-22, 35]สามารถเขียนได้เป็น $1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}$

ขั้นตอนที่สอง ตรวจสอบเครื่องหมายของ X และ Y

เครื่องหมายของ min_Xและmax_Xคือ negative , positive

เครื่องหมายของ min_Yและ max_Yคือ negative, negative

ขั้นตอนที่สาม จัดให้อยู่ในรูปแบบพร้อมคูณ

รูปแบบพร้อมคูณ ของ X คือ $\bar{\alpha}\bar{\gamma}\alpha 0\bar{\gamma}00\alpha$

รูปแบบพร้อมคูณ ของ Y คือ$1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}$

ขั้นตอนที่สี่ ดำเนินการหารบนระบบแทนช่วงแบบยืดหยุ่น ด้วยกฎการคูณ

$\bar{\alpha}\bar{\gamma}\alpha 0\bar{\gamma}00\alpha \times 1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha} = \alpha\bar{1}0\bar{\gamma}.\alpha\bar{1}$= [-3.25, 4.25]

ผลลัพธ์ของการหารในระบบแทนช่วงแบบยืดหยุ่นคือ $\alpha\bar{1}0\bar{\gamma}.\alpha\bar{1}$ โดยมีค่าเชิงตัวเลขเท่ากับ [-3.25, 4.25] ดังนั้นผลลัพธ์ของการคูณในระบบแทนช่วงแบบยืดหยุ่นสามารถให้คำตอบที่ถูกต้อง □

2.2.**การคำนวณแบบเชื่อมตรงบนระบบจำนวนแบบมีเครื่องหมาย** (On-line computational for signed-digit number system)

ระบบจำนวนแบบมีเครื่องหมาย (signed-digit number system) ได้ถูกนำเสนอโดย [1] ซึ่งใช้แนวคิดของการใช้ตัวเลขแบบมีเครื่องหมายในการแสดงจำนวนแต่ละจำนวน เพื่อประโยชน์ในการลดการแพร่กระจายของตัวทดระหว่างการคำนวณและทำให้การคำนวณสามารถทำได้ในระบบขนาน โดยได้มีการนำเสนอ นิยามของระบบนี้ไว้ดังนี้

นิยามที่ 2.4 ระบบจำนวนแบบมีเครื่องหมาย (Signed-digit number system) *ประกอบด้วยเลขฐาน βโดยที่β เป็นจำนวนเต็มที่มากกว่าหรือเท่ากับ 2 และชุดของตัวเลข (Digit set)* $F = \{a, a+1, \dots, -1, 0, 1, \dots, b-1, b\}$*โดยที่* $b - a + 1 < 2\beta$*เมื่อ a เป็นจำนวนเต็มลบ และ b เป็นจำนวนเต็มบวก*

ตัวอย่างที่ 2.7 *กำหนดให้ระบบแทนจำนวนแบบมีเครื่องหมายประกอบด้วยเลขฐาน* β*=5 และมีชุดตัวเลข D =* $\{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ *จงเขียนค่าของ X = 1010 ให้อยู่ในระบบจำนวนนี้*

วิธีทำค่าของ*X* = 1010 สามารถเขียนให้อยู่ในระบบนี้ได้เป็น1010 = $(2\bar{2}2\bar{2}0)_5$ □

ระบบจำนวนแบบมีเครื่องหมายนี้ ได้รับการพิสูจน์แล้วว่า ทุกจำนวนสามารถแสดงได้ในระบบนี้ และ ระบบนี้ยังมีคุณสมบัติที่สำคัญคือ เป็นระบบจำนวนซ้ำซ้อน ซึ่งคุณสมบัตินี้เองที่เป็นประโยชน์ต่อการคำนวณความเร็วสูง

จากนั้น ทราเวดีและเออเซโกวัก ใน [5] ได้นำเสนอการดำเนินการแบบเชื่อมตรงบนระบบจำนวนแบบมีเครื่องหมาย ซึ่งได้มีการพิสูจน์ว่า มีประสิทธิภาพเชิงเวลาไม่แตกต่างไปจากการคำนวณแบบขนาน ลักษณะการคำนวณในระบบนี้เป็นไปตามนิยามต่อไปนี้

นิยามที่ 2.5 การดำเนินการแบบเชื่อมตรงของตัวดำเนินการใดๆ *หมายถึง การดำเนินการในลักษณะลำดับ (serial operation) ทั้งในด้านการรับตัวถูกดำเนินการ (input operand) และการผลิตผลลัพธ์ (output result) เริ่มต้นจากตัวเลขที่มีนัยสำคัญสูงสุดไปยังตัวเลขที่มีนัยสำคัญต่ำสุดแบบตัวเลขทีละตัวเลข (digit-by-digit) โดยที่ตัวเลขแรกของผลลัพธ์ที่ผลิตได้มาจากการรับตัวเลขของตัวถูกดำเนินการ* δ *ตัวเลข ตัวดำเนินการนี้จะถูกเรียกว่ามี* ค่าความหน่วงเชื่อมตรง (on-line delay) *เท่ากับ* δ

ในงานนี้ จะขอกล่าวถึงเฉพาะ อัลกอริทึมการหารเท่านั้น

อัลกอริทึมการหารแบบเชื่อมตรงของระบบจำนวนแบบมีเครื่องหมาย

Algorithm 2.1 : On-line Division

ให้ N คือตัวตั้งหาร , D คือตัวหารและ Q คือผลหาร ซึ่ง

$$N = \sum_{i=1}^{\mathrm{m}} n_i 2^{-i} ,D = \sum_{i=1}^{\mathrm{m}} d_i 2^{-i} ,Q = \sum_{i=1}^{\mathrm{m}} d_i 2^{-i} , Q = \frac{N}{D}$$

Step 1 [Initial] : $P_0 = \sum_{i=1}^{\delta} n_i 2^{-i}$,

$D_0 = \sum_{i=1}^{\delta} d_i 2^{-i}$

Step 2 [Selection] : $q_{j+1} = Select(P_j)$

Step3 [Recursion] : $D_{j+1} = D_j + d_{j+\delta+1} \times 2^{-j-\delta-1}$

$P_{j+1} = 2 \times P_j + n_{j+\delta+1} \times r^{-\delta} - q_{j+1} \times D_{j+1}$
$-Q_j \times d_{j+\delta+1} r^{-\delta}$

ในอัลกอริทึม P_j คือเศษเหลือจากการหารในแต่ละรอบการทำงาน โดยที่เริ่มต้นการหารใช้ตัวเลขของตัวถูกดำเนินการเป็นจำนวน δ ตัวเลข ในแต่ละรอบนั้นจะผลิตคำตอบออกมา 1 จำนวนคือ q_{j+1} โดย จะพิจารณาจาก ช่วงของเศษเหลือ P_j ให้มีค่าเป็นช่วงที่แคบที่สุดเสมอ

ตัวอย่างที่ 2.8 *แสดงการหารแบบเชื่อมตรง ของ 95 ด้วย 22*

วิธีทำ จากโจทย์ N = 95 , D = 22

จะต้องหาค่าของQ= N/D

ขั้นตอนที่ 1 เขียนให้อยู่ในรูปแบบแทนจำนวนแบบมีเครื่องหมาย

N = 1011111

D = 10110

ขั้นตอนที่ 2 จัดให้เข้าเงื่อนไข N < D และ มีค่าอยู่ในช่วง [-1,1]

$N = 0.01011111 \times 2^8$

$D = 0.10110 \times 2^5$

ดำเนินการหารแบบเชื่อมตรงแสดงขั้นตอนดังตารางที่ 2.3

ตารางที่2.3 แสดงขั้นตอนการหารแบบเชื่อมตรง

j	D_j	P_j	q_{j+1}
0	0.1011	0. 0101	1
1	0.10110	0.101 + 0.0001 - 1x0.10110 - 0x1 = 0	0
2	0.101100	0 .000 + 0.0001 - 0x0.101100 - 0. 1x0 = 0.0001	0
3	0.1011000	0. 0010 + 0.0001 - 0x0.1011000 - 0. 10x 0 = 0. 011	1
4	0.10110000	0.11 + 0.0001 - 1x0.1011000 - 0.100x0 = 0.001	0

คำตอบที่ได้คือ 0.10010 เนื่องจาก

ผลลัพธ์ที่ได้จากอัลกอริทึมคือ 0.10010 ผลลัพธ์สุดท้ายหลังจากทำการจัดตำแหน่งทศนิยมแล้วจะได้

$$0.10010 \times 2^8 / 2^5 = 100.10$$

ค่าของช่วงในเลขฐานสิบของ100.10 คือ 4.5 ซึ่งเป็นผลลัพธ์ที่ถูกต้อง □

บทที่ 3

อัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

จุดประสงค์หลักของงานวิจัยนี้ คือ การนำเสนออัลกอริทึมการหารแบบเชื่อมตรง (on-line division algorithm) บนระบบแทนช่วงแบบยืดหยุ่น (flexible interval representation system, FIRS) พร้อมทั้งบทพิสูจน์ความถูกต้อง รวมถึงนำเสนอค่าความหน่วงเชื่อมตรงสำหรับการหารแบบเชื่อมตรงด้วย

เนื่องจากการหารของช่วงสองช่วงใดๆ มีข้อจำกัดบางประการคือในการหารของช่วงสองช่วงจะต้องการหารถึงสี่ครั้งแล้วนำผลลัพธ์จากการหารทั้งสี่มาทำการพิจารณาหาค่าสูงสุดและต่ำสุด ซึ่งถ้าเราทราบเครื่องหมายของทั้งสองช่วงแล้วเราสามารถพิจารณาดังตารางที่ 3.1 ซึ่งจะทำการหารเพียงครั้งเดียว ซึ่งทำให้ลดเวลาในการทำงานได้มาก และการหารแบบเชื่อมตรงนั้นจำเป็นต้องใช้ข้อมูลนำเข้ามีค่าอยู่ระหว่าง [-1,1] เพื่อเป็นการจำกัดค่าผลลัพธ์การหารในแต่ละรอบการทำงานเกิดการให้มีค่าอยู่ระหว่าง [-1,1] ตลอดจึงนำมาสู่การเสนอให้มีรูปแบบบรรทัดฐานและขอบเขตของตัวถูกดำเนินการให้อยู่ในรูป [-1,1]

ตารางที่ 3.1 กรณีเครื่องหมายและผลลัพธ์ของการหารที่เกิดขึ้นบนระบบแทนช่วง

ตัวถูกดำเนินการ				ผลลัพธ์			
x_L	x_U	y_L	y_U	z_L	z_U	z_L(Value)	z_U(Value)
$+$	$+$	$+$	$+$	$+$	$+$	$x_L \div y_U$	$x_U \div y_L$
$+$	$+$	$-$	$-$	$-$	$-$	$x_U \div y_U$	$x_L \div y_L$
$-$	$-$	$+$	$+$	$-$	$-$	$x_L \div y_L$	$x_U \div y_U$
$-$	$-$	$-$	$-$	$+$	$+$	$x_U \div y_L$	$x_L \div y_U$
$-$	$+$	$+$	$+$	$-$	$+$	$x_L \div y_L$	$x_U \div y_L$
$-$	$+$	$-$	$-$	$-$	$+$	$x_U \div y_U$	$x_L \div y_U$

3.1 รูปแบบบรรทัดฐาน สำหรับการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

การปรับปรุงรูปแบบแทนช่วงที่จะทำการหาร (ตัวถูกดำเนินการ) จะปรับปรุงให้อยู่ในรูปแบบบรรทัดฐาน (normal form) ก่อนที่จะทำการหาร ทั้งนี้ รูปแบบบรรทัดฐาน จะเป็นรูปแบบที่บอกถึงเครื่องหมายของ ขอบเขตบนและล่างของตัวถูกดำเนินการแต่ละตัว และเพื่อให้

ผลลัพธ์ของการหารแสดงอยู่ในรูปแบบบรรทัดฐาน ด้วยเช่นกัน จึงต้องกำหนดรูปแบบบรรทัดฐาน ดังต่อไปนี้

นิยามที่ 3.1 รูปแบบบรรทัดฐาน*ในระบบแทนช่วงแบบยืดหยุ่น* $X = 0.x_1x_2x_3\ldots x_n \times 2^r$ *คือรูปแบบแทนช่วงที่อยู่ในรูปของทศนิยม โดยมีค่าขอบเขตบน* $-1 < X_U < 1$ *และขอบเขตล่าง* $-1 < X_L < 1$ *และ สองตำแหน่งแรกของรูปแบบแทนช่วงต้องขึ้นต้นด้วย*

1	*สำหรับกรณีที่* $x_L > 0$ *และ* $x_U > 0$
$\alpha 0$ *หรือ* $\alpha\alpha$	*สำหรับกรณีที่* $x_L = 0$ *และ* $x_U > 0$
$\bar{1}$	*สำหรับกรณีที่* $x_L < 0$ *และ* $x_U < 0$
$\gamma 0$ *หรือ* $\gamma\gamma$	*สำหรับกรณีที่* $x_L < 0$ *และ* $x_U = 0$
$\alpha\bar{1}$, $\alpha\gamma$, $\gamma 1$ *หรือ* $\gamma\alpha$	*สำหรับกรณีที่* $x_L < 0$ *และ* $x_U > 0$

ในที่นี้ เราจะแสดงให้เห็นว่า ทุกช่วงที่สามารถหารูปแบบแทนช่วงแบบยืดหยุ่นได้ สามารถเขียนให้อยู่ในรูปแบบบรรทัดฐานได้เสมอ ดังกล่าวไว้ในทฤษฎีบทต่อไปนี้

ทฤษฎีบทที่ 3.1 *รูปแบบแทนช่วงใด ๆ* $X = [A, B]$ *เมื่อ A และ B เป็นจำนวนจริง สามารถเขียนให้อยู่ในรูปแบบแทนช่วงแบบบรรทัดฐาน* $X = 0.x_1x_2x_3\ldots x_n \times 2^r$ *ได้นั่นคือ* $-1 < X_L, X_U < 1$ *เมื่อ* X_L *และ* X_U *หมายถึงค่าขอบเขตล่างและค่าขอบเขตบนของ X ตามลำดับ*

พิสูจน์วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมสำหรับการแปลงช่วงใดๆที่อยู่ในรูปของแบบแทนช่วงของระบบเลขฐานสองแบบคิดเครื่องหมายให้อยู่ในรูปแบบบรรทัดฐาน

Algorithm 1: ConvertBiToNormalForm

Input: Interval[A,B]

$A = a_0a_1a_2\ldots a_n$, $B = b_0b_1b_2\ldots b_n$ where $a_i, b_i \in \{0,1\}$, a_0, b_0 is a signed bit

Output: Normal form $S = 0.s_0s_1s_2\ldots s_n$ where $s_i \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Begin

```
if(A > 0)      then Lower  = ToSignedDigit( Two's Complement (A) )
                    Upper  = ToSignedDigit( Two's Complement (B) )
                    s_0    = 1
                    s_1 s_2 ... s_n = ToFIRS(Lower,Upper)
else if(B < 0)      then  Similarly case A > 0 but s_0 = -1
else if ((A=0and B>0) or (A<0 and B=0) then
                    Lower  = ToSignedDigit( A )
                    Upper  = ToSignedDigit( B )
                    S      =ToFIRS(Lower,Upper)
else if (A<0 and B >0) then
        if(a_1 a_2 ... a_n < b_1 b_2 ... b_n) then
                    if(a_1=0) then  Lower  = Two's Complement (a_2 ... a_n)
                                    Upper  =  b_2 ... b_n
                                    s_0 s_1    = α1̄
                                    s_1 s_2 ... s_n = ToFIRS(Lower,Upper)
                    else    Lower  = 1̄ × (a_2 ... a_n)
                            Upper  = 1̄ × Two's Complement (b_2 ... b_n)
                            s_0 s_1    = αγ
                            s_2 ... s_n  = ToFIRS(Lower,Upper)
                    end if
        else
                    if(b_1=0) then  Lower = 1̄ × (a_2 ... a_n)
                            Upper  = 1̄ ×Two's Complement (b_2 ... b_n)
                            s_0 s_1    = γ1
                            s_2 ... s_n   = ToFIRS(Lower,Upper)
                    else    Lower  = Two's Complement (a_2 ... a_n)
                            Upper  =  b_2 ... b_n
                            s_0 s_1    = γα
                            s_1 s_2 ... s_n = ToFIRS(Lower,Upper)
                    end if
```

end if

end if

$S = 2^{-n} \times S$

End

หมายเหตุ ฟังก์ชัน ToSignedDigit เป็นการแปลงจำนวนเลขฐานสองแบบคิดเครื่องหมาย (Signed binary number) ให้เป็นจำนวนฐานสองแบบมีเครื่องหมาย (Signed digit number) และฟังก์ชัน ToFIRS เป็นการแปลงจำนวนเลขฐานสองแบบมีเครื่องหมายให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

พิสูจน์อัลกอริทึม การพิสูจน์อัลกอริทึมเราจะแสดงว่าผลลัพธ์ที่ได้จากอัลกอริทึมข้างต้นเป็นรูปแบบบรรทัดฐานสำหรับการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น เมื่อพิจารณาช่วง [A,B] กำหนดให้

$A = a_0a_1a_2\ldots a_n$ โดยที่ $a_i \in \{0,1\}$, a_0 is a signed bit นั่นคือ

$A = \mathrm{Sign}(a_0) \times (a_1 2^{n-1} + a_2 2^{n-2} + \ldots + a_n 2^0)$

$B = b_0b_1b_2\ldots b_n$ โดยที่ $b_i \in \{0,1\}$, b_0 is a signed bit นั่นคือ

$B = \mathrm{Sign}(b_0) \times (b_1 2^{n-1} + b_2 2^{n-2} + \ldots + b_n 2^0)$

ทำการแปลง A และ B ที่เป็นเลขฐานสองไปเป็นระบบแทนจำนวนแบบมีเครื่องหมาย และนำจำนวนทั้งสองไปแปลงให้อยู่ในรูปแบบบรรทัดฐาน โดยพิจารณาจากเครื่องหมายของช่วงซึ่งแบ่งออกเป็นห้ากรณี ดังนี้

กรณีที่ 1: A > 0 และ B > 0

ขั้นตอนแรก หาค่า Two's complement ของ A และ B

Two's complement(A) = $a'_0a'_1a'_2\ldots a'_n$ = Invert bits($a_0a_1a_2\ldots a_n$) + 1

Two's complement(B) = $b'_0b'_1b'_2\ldots b'_n$ = Invert bits($b_0b_1b_2\ldots b_n$) + 1

เพราะว่า A>0 และ B>0จะได้ว่า $a_0 = 0, b_0 = 0$ ดังนั้น $a'_0 = 1, b'_0 = 1$ และค่าที่เป็นไปได้ของ $a'_1, a'_2, \ldots, a'_n$ และ $b'_1, b'_2, \ldots b'_n$ คือ 0 หรือ 1

ขั้นตอนสอง เขียนให้อยู่ในรูปของ Signed Digit

เนื่องจาก $a'_0 = 1$, $b'_0 = 1$ สามารถเขียนให้อยู่ในรูปของ Signed digit ได้เป็น

$\overline{a'_1a'_2} \ldots \overline{a'_n}$ และ $\overline{b'_1b'_2} \ldots \overline{b'_n}$

ขั้นตอนสามเขียนให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

เนื่องจาก a'_1, a'_2 , … , a'_nและ b'_1 , b'_2 , … b'_nคือ 0 หรือ 1ดังนั้นค่าที่เป็นไปได้ของ $\overline{a'_1}, \overline{a'_2}, \dots, \overline{a'_n}$และ $\overline{b'_1}, \overline{b'_2}, \dots, \overline{b'_n}$คือ0 หรือ $\bar{1}$ดังนั้น รูปแบบแทนช่วงที่เป็นไปได้ของ $s_1,s_2,\dots,s_n$คือ$\bar{1}, \gamma, 0, \bar{\alpha}$

ดังนั้น s_0, $s_1,s_2,\dots,s_n \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้องและตำแหน่งแรกมีค่าเป็น 1

กรณีที่ 2 : A = 0 และ B > 0

ขั้นตอนแรก เขียนให้อยู่ในรูปของ Signed digit

เนื่องจาก a_0 = 0 และ b_0 = 0 สามารถเขียนให้อยู่ในรูปของ Signed digit ได้เป็น$a_1a_2\dots a_n$ และ$b_1b_2\dots b_n$

ขั้นตอนที่สอง เขียนให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

เนื่องจาก A = 0 ดังนั้น$a_1 = a_2 = \dots = a_n = 0$

และจาก B > 0 จะได้ว่าค่าที่เป็นไปได้ของ b_1 คือ 1 และb_2 , … , b_nคือ 0 หรือ 1

ดั้งนั้นรูปแบบแทนช่วงที่เป็นไปได้ของ s_1 คือα และ $s_2,\dots,s_n$คือ 0 หรือ α

ดังนั้น s_0, $s_1,s_2,\dots,s_n \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้องและสองตำแหน่งแรกมีค่าเป็น $\alpha 0$หรือ $\alpha\alpha$

กรณีที่ 3 : A < 0 และ B < 0

กรณีนี้จะคล้ายกับกรณีที่ 1 เพียงแต่สลับเครื่องหมาย จากบวกเป็นลบ จะได้ว่ารูปแบบแทนช่วงที่เป็นไปได้ของ $s_1,s_2,\dots,s_n$คือ$1, \bar{\gamma}, 0, \alpha$

ดังนั้น s_0, $s_1,s_2,\dots,s_n \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้องและตำแหน่งแรกมีค่า เป็น $\bar{1}$

กรณีที่ 4 : A < 0 และ B = 0

ในทำนองเดียวกันกับกรณีที่ 2 แต่สลับเครื่องหมายจากบวกเป็นลบและสลับค่ากันระหว่าง A กับ Bจะได้ว่า

B = 0 ดังนั้น$b_1 = b_2 = \dots = b_n = 0$

และจาก A < 0 จะได้ว่าค่าที่เป็นไปได้ของ a_1 คือ $\bar{1}$และa_2 , … , a_nคือ 0 หรือ $\bar{1}$

ดั้งนั้นรูปแบบแทนช่วงที่เป็นไปได้ของ s_1 คือγ และ $s_2,\dots,s_n$คือ 0 หรือ γ

ดังนั้น s_0, $s_1,s_2,\dots,s_n \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้องและสองตำแหน่งแรกมีค่าเป็น $\gamma 0$ หรือ $\gamma\gamma$

กรณีที่ 5 : A < 0 และ B > 0

ในกรณีนี้จะแบ่งการพิจารณาออกเป็นสองกรณีย่อยดังนี้

กรณีที่ 5.1 : $\|b_1 b_2 \ldots b_n\|$>$\|a_1 a_2 \ldots a_n\|$ ในกรณีนี้เราจะพิจารณาที่ b_1 = 1และจะแยกพิจารณาออกเป็นสองกรณีย่อยคือ a_1 = 0 และ a_1 = 1

กรณี 5.1.1: a_1 = 0กรณีนี้จะให้สองตำแหน่งแรกเป็น $\alpha\bar{1}$

ขั้นตอนแรกหาค่า Two's complement ของ $a_2 \ldots a_n$ซึ่งเขียนได้เป็น $a'_2 \ldots a'_n$

ขั้นตอนที่สอง เขียนให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

จะเห็นว่าค่าที่เป็นไปได้ของ a_2 , … , a_nคือ 0 หรือ 1 และb_2 , … , b_nคือ 0 หรือ 1

ดั้งนั้นรูปแบบแทนช่วงที่เป็นไปได้ของ $s_2,\ldots,s_n$คือ $1, \bar{\gamma}, 0, \alpha$

ดังนั้น s_0, $s_1, s_2, \ldots, s_n \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้องและสองตำแหน่งแรกมีค่าเป็น $\alpha\bar{1}$

กรณี 5.1.2: a_1 = 1กรณีนี้จะให้สองตำแหน่งแรกเป็น $\alpha\gamma$

ขั้นตอนแรกหาค่า Two's complement ของ $b_2 \ldots b_n$ ซึ่งเขียนได้เป็น $b'_2 \ldots b'_n$

หลังจากนั้นคูณ ด้วย $\bar{1}$จะได้เป็น$\overline{b'_1 b'_2} \ldots \overline{b'_n}$และ $\overline{a_1 a_2} \ldots \overline{a_n}$

ขั้นตอนที่สอง เขียนให้อยู่ในรูปแบบแทนช่วงแบบยืดหยุ่น

จะเห็นว่าค่าที่เป็นไปได้ของ $\overline{a_2}, \ldots, \overline{a_n}$คือ 0 หรือ$\bar{1}$และ$\overline{b'_2}, \ldots \overline{b'_n}$คือ 0 หรือ$\bar{1}$

ดั้งนั้นรูปแบบแทนช่วงที่เป็นไปได้ของ $s_2,\ldots,s_n$คือ$\bar{1}, \gamma, 0, \bar{\alpha}$

ดังนั้น s_0, $s_1, s_2, \ldots, s_n \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้องและสองตำแหน่งแรกมีค่าเป็น $\alpha\bar{1}$

กรณีที่ 5.2 : $\|\mathrm{a_1 a_2 \ldots a_n}\|$>$\|\mathrm{b_1 b_2 \ldots b_n}\|$ในกรณีนี้เราจะพิจารณาที่ a_1 = 1และจะแยกพิจารณาออกเป็นสองกรณีย่อยคือ b_1 = 0 และ b_1 = 1

กรณี 5.2.1: b_1 = 0กรณีนี้จะให้สองตำแหน่งแรกเป็น $\gamma 1$

ในกรณีนี้จะคล้ายกับกรณีที่ 5.1.1 โดยสลับค่ากันระหว่าง A กับ B

กรณี 5.2.2: b_1 = 0กรณีนี้จะให้สองตำแหน่งแรกเป็น $\gamma\alpha$

ในกรณีนี้จะคล้ายกับกรณีที่ 5.1.2 โดยสลับค่ากันระหว่าง A กับ B

จากกรณีที่ 5.1 และ 5.2 เราสามารถสรุปได้ว่า สองตำแหน่งแรกมีค่าเท่ากับ $\alpha\bar{1}, \alpha\gamma$, $\gamma 1$และ$\gamma\alpha$

หลังจากการพิจารณาเครื่องหมายในทุกกรณีแล้ว จะทำการคูณ ผลลัพธ์ที่ได้ด้วย 2^{-n} ทำให้ค่าอยู่ในช่วง [-1,1] ดังนั้น ผลลัพธ์ที่ได้จากอัลกอริทึมอยู่ในรูปแบบบรรทัดฐาน ■

ตัวอย่าง 3.1 *หารูปแบบบรรทัดฐานสำหรับการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นของ [10, 18]*

วิธีทำ ขั้นตอนการหารูปแบบบรรทัดฐาน

ขั้นตอนที่ 1 แสดง [10,18] ให้อยู่ในระบบจำนวนฐานสอง ได้

[001010, 010010] ให้ A = 001010และ B = 010010

ขั้นตอนที่ 2 หารูปแบบ two's complement

A' = 110110และ B' = 101110

ขั้นตอนที่ 3 แสดงให้อยู่ในระบบแทนจำนวนแบบมีเครื่องหมาย

A = $\bar{1}0\bar{1}\bar{1}0$ *และ* B = $0\bar{1}\bar{1}\bar{1}0$

ขั้นตอนที่4 แปลงเป็นระบบแทนช่วงแบบยืดหยุ่นทีละตำแหน่ง

$1\gamma\bar{\alpha}\bar{1}\bar{1}0$

ขั้นตอนสุดท้ายจัดให้มีค่าอยู่ในช่วง [-1,1]

$1\gamma\bar{\alpha}\bar{1}\bar{1}0 \times 2^{-6} = 0.1\gamma\bar{\alpha}\bar{1}\bar{1}0$

จะได้ว่ารูปแบบบรรทัดฐานสำหรับของ[10, 18] คือ $0.1\gamma\bar{\alpha}\bar{1}\bar{1}0$

ตรวจสอบค่ารูปแบบบรรทัดฐาน $0.1\gamma\bar{\alpha}\bar{1}\bar{1}0$

$0.1\gamma\bar{\alpha}\bar{1}\bar{1}0$เขียนให้อยู่ในรูปแบบแทนช่วงได้เป็น

$0.1\gamma\bar{\alpha}\bar{1}\bar{1}0 \times 2^{6} = 1\gamma\bar{\alpha}\bar{1}\bar{1}0 = [1\bar{1}0\bar{1}\bar{1}0 , 10\bar{1}\bar{1}\bar{1}0] = [10,18]$

3.2 การหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

การหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น ระบบแทนช่วงแบบยืดหยุ่นบางรูปแบบนั้นไม่สามารถทำการหารแบบเชื่อมตรงได้ เพราะในกระบวนการหารนั้นจะต้องทราบเครื่องหมายของช่วงจึงได้มีการปรับปรุงระบบแทนช่วงแบบยืดหยุ่นให้อยู่ในรูปแบบบรรทัดฐาน

ทฤษฎีบท 3.2 *กำหนดให้รูปแบบแทนช่วงแบบยืดหยุ่นสองจำนวนที่อยู่ในรูปแบบบรรทัดฐาน N และ D สามารถดำเนินการหารแบบเชื่อมตรงได้ด้วยค่าความหน่วงเชื่อมตรง 5*

พิสูจน์ วิธีการพิสูจน์ทฤษฎีบทนี้ จะเสนออัลกอริทึมการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

Algorithm 2: On-line division flexible interval representation system

Input: Normal Form $N = \sum_{i=1}^{m} n_i\beta^{-i}$ and $D = \sum_{i=1}^{m} d_i\beta^{-i}$

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

Output: Normal Form $Q = \sum_{i=1}^{m} q_i\beta^{-i}$ where $q_i \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Such that $Q = \frac{N}{D}$, β is the base and N < D

Begin

$P_0 = \sum_{i=1}^{\delta} n_i\beta^{-i}$

$D_0 = \sum_{i=1}^{\delta} d_i\beta^{-i}$

Sg = ChkSigned(P_0 , D_0)

$q_1 = Sg$

$j = 0$

While(j< m-1)do

D_{j+1} = Increase Rule(D_j,$d_{j+\delta+1} \times \beta^{-j-1}$)

P_{j+1}=Decrease Rule(Decrease Rule(Increase Rule$\left(\beta \times P_j, n_{j+\delta+1} \times \beta^{-\delta}\right)$

, Multiply Rule$\left(q_{j+1}, D_{j+1}, \mathrm{Sg}, d_1\right)$) ,

Multiply Rule($\sum_{i=1}^{j} q_i\beta^{-i}$, $d_{j+\delta+1}\beta^{-\delta}$, Sg , d_1))

q_{j+2} = Combine(SelectFIRS(Lower($\beta \times P_{j+1}$)), SelectFIRS(Upper($\beta \times P_{j+1}$)))

End do

End

Algorithm 3: ChkSigned(Check Signed of Division FIRS)

Input: Normal Form $A = \sum_{i=1}^{m} a_i\beta^{-i}$, $B = \sum_{i=1}^{m} b_i\beta^{-i}$

Where $a_i, b_i \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Output: *Sg* where *Sg* $\in \{1, \bar{1}, \alpha, \gamma\}$ [Signed of Division]

Begin

Asg ,*Bsg* is first digit not zero where *Asg* $\in \{1, \bar{1}, \alpha, \gamma\}$, *Bsg* $\in \{1, \bar{1}\}$

$i = 0$

While(i<m +1) do

if($a_i \neq \mathbf{0}$) then *Asg* = a_i ; exit while;end if end do

Bsg = b_1

if (*Asg* = 1) then if(*Bsg* = 1) then *Sg* = 1 else *Sg* = $\bar{1}$ endif

else if (*Asg* = $\bar{1}$) then if(*Bsg* = 1) then *Sg* = $\bar{1}$ else *Sg* = 1 endif

else if (*Asg* = α) then if(*Bsg* = 1) then *Sg* = α else *Sg* = γ endif

else if (*Asg* = γ) then if(*Bsg* = 1) then *Sg* = γ else *Sg* = α endif

end if

End

Algorithm 4: Multiply Rule

Input: a, b Where $a, b \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Sg Where $Sg \in \{1, \bar{1}, \alpha, \gamma\}$ Sg is Signed of Divider

DSg Where $DSg \in \{1, \bar{1}\}$ DSg is Signed of Division

Output: c where $c \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Begin

foreach(a in digit set of signed-FIRS)

foreach(b in digit set of signed-FIRS)

if ($Sg = 1$) then

if ($DSg = 1$) then c = Combine(min a × max b , max a × min b)

else c = Combine(max a × min b , min a × max b) end if

else if($Sg = \bar{1}$) then

if($DSg = 1$)) then c = Combine(min a x min b , max a x max b)

else c = Combine(max a × max b , min a × min b) end if

else

if($DSg = 1$) then c = Combine(min a × min b , max a × min b)

else c = Combine(max a × max b , min a × max b) end if

end if

end for

end for

End

Algorithm 5: Increase rule

Input: a, b Where $a, b \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Output: c where $c \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

```
Begin
foreach( a in digit set of signed-FIRS)
        foreach( b in digit set of signed-FIRS)
                c = Combine(min a + min b , max a + max b)
        end for
end for
End
```

Algorithm 6: Decrease rule

Input: a, b Where $a, b \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

Output: c where $c \in \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$

```
Begin
foreach( a in digit set of signed-FIRS)
        foreach( b in digit set of signed-FIRS)
                c = Combine(min a -min b , max a - max b)
        end for
end for
End
```

พิสูจน์อัลกอริทึมกำหนดให้ชุดตัวเลข $F = \{\bar{1}, \gamma, 0, \alpha, 1, \bar{\gamma}, \bar{\alpha}\}$ และ $N = \sum_{i=1}^{m} n_i \beta^{-i}$, $D = \sum_{i=1}^{m} d_i \beta^{-i}$, $Q = \sum_{i=1}^{m} q_i \beta^{-i}$โดยที่ $n_i, d_i, q_i \in F$,β คือเลขฐาน, δ คือค่าความหน่วงเชื่อมตรง

ต้องการหาผลหาร $Q = \frac{N}{D}$ โดยที่ $N<D$

ในการพิสูจน์จะแบ่งเป็นสามส่วน ในส่วนแรกจะเป็นการพิสูจน์หาค่าความหน่วงที่ใช้สำหรับการหารแบบเชื่อมตรงในระบบแทนช่วงแบบยืดหยุ่น และส่วนที่สองจะเป็นการพิสูจน์ให้เห็นว่าผลลัพธ์ของอัลกอริทึมเป็นผลลัพธ์ที่อยู่ในช่วงที่ถูกต้อง และส่วนที่สามจะเป็นการพิสูจน์ว่าผลลัพธ์ที่ได้จากอัลกอริทึมเป็นรูปแบบบรรทัดฐาน

ส่วนที่ 1: พิจารณาค่าความหน่วง

จากการ Recursive ของ Algorithm2 จะได้ว่า P_j สามารถเขียนให้อยู่ในรูปของ j ได้เป็น

$$P_j = \beta^j \left[\sum_{i=1}^{m} n_i \beta^{-i} - \left(\sum_{i=1}^{j} q_i \beta^{-i} \right) \left(\sum_{i=1}^{m} d_i \beta^{-i} \right) \right]$$

จาก [5] ด้วย β = 2 จะได้ว่า

$$-D + 2 \times -2^{-\delta} \leq 2P_j \leq -4 \times 2^{-\delta} \text{ และ}$$

$$2 \times -2^{-\delta} \leq 2P_j \leq D - 4 \times 2^{-\delta} \quad (1)$$

จากนิยามของรูปแบบบรรทัดฐานจะได้ว่าทุกรูปแบบในสองหลักแรกที่ขึ้นต้นในรูปแบบบรรทัดฐาน จะได้ว่า 1/4 < D < 1/2 ซึ่งในกรณีที่ D = 1/4 จะได้ว่า

$$2 \times -2^{-\delta} < 1/4 - 4 \times 2^{-\delta} \quad (2)$$

จาก(2) และค่าความหน่วงต้องมีค่าเป็นจำนวนเต็ม เราสามารถสรุปได้ว่าการหารแบบเชื่อมตรงบนฐานสองจะสามารถดำเนินการได้ด้วยค่าความหน่วงมีค่ามากกว่าหรือเท่ากับ 5

ส่วนที่ 2: พิจารณาผลลัพธ์อยู่ในช่วงที่ถูกต้อง

จาก(1) และค่าความหน่วง 5 จะได้ว่า SelectFIRSมีค่าดังนี้

$$\text{SelectFIRS }(2X) = \begin{cases} 1, 2X > \frac{1}{8} \\ -1, 2X < -\frac{1}{8} \\ 0, Otherwise \end{cases} \quad (3)$$

จาก Algorithm2 ผลลัพธ์จากการหารในแต่ละรอบการหาร q_j เขียนอยู่ในรูป

$$q_j = \text{Combine(SelectFIRS(Lower}(2P_j)\text{)), SelectFIRS(Upper}(2P_j)\text{)))} \quad (4)$$

จาก(3) และ(4) เราจะได้ว่า $q_j \in F$ ซึ่งแสดงว่าผลลัพธ์อยู่ในช่วงที่ถูกต้อง

ส่วนที่ 3: พิจารณาผลลัพธ์อยู่ในรูปแบบบรรทัดฐาน

เนื่องจาก ผลลัพธ์ในตำแหน่งแรก (q_1) เกิดจากการเลือกค่าของฟังก์ชัน ChkSigned ซึ่งผลลัพธ์ที่เป็นไปได้จากฟังก์ชันนี้คือ $1, \overline{1}$, α หรือ γ ในกรณีที่ q_1 = 1 หรือ 1 จะเห็นได้ว่าอยู่ในรูปแบบบรรทัดฐานแล้วในกรณีที่ q_1 = α หรือ γ จะต้องพิจารณาผลลัพธ์ในตำแหน่ง q_2 ต่อไป โดยจะแยกพิจารณาเป็นสองกรณีดังนี้

กรณีที่ 3.1: $q_1 = \alpha$

ในกรณีนี้เกิดจากตัวถูกดำเนินการ a_1 ขึ้นต้นด้วย α และ b_1 ขึ้นต้นด้วย 1 หรือ a_1 ขึ้นต้นด้วย γ และ b_1 ขึ้นต้นด้วย $\overline{1}$

กรณีที่ 3.1.1: $a_1 = \alpha$และ $b_1 = 1$

เนื่องจาก N เป็นรูปแบบบรรทัดฐาน ดังนั้นค่าที่เป็นไปได้ของ a_2คือγ , $\bar{1}$

จากเงื่อนไข N < D จะมีการ shift N ไปทางขวา 2 ตำแหน่ง ซึ่ง 5 ตำแหน่งแรกจะอยู่ในรูป $0.0a_1a_2a_3a_4$และ $0.b_1b_2b_3b_4b_5$ในขั้นตอนการหารรอบที่2 ซึ่งในรอบแรกผลิตคำตอบแรกได้ aจะได้ว่าเศษเหลือ$P_1 = 0.0a_1a_2a_3a_4 - \alpha \times 0.b_1b_2b_3b_4b_5 = 0.0\alpha a_2a_3a_4 - \alpha \times 0.1b_2b_3b_4b_5$

ดังนั้นค่าของเขตล่างของ P_1 = lower($0.0\alpha a_2a_3a_4$) – lower($\alpha \times 0.1b_2b_3b_4b_5$) ค่าเนื่องจาก N < D จะได้ว่า ค่าขอบเขตล่างที่เป็นไปได้ของ P_1จะมีค่าน้อยกว่าศูนย์ เพราะ lower($0.0\alpha a_2a_3a_4$) มีค่าน้อยกว่าศูนย์เนื่องจากอยู่ในช่วง ลบบวก และ lower($\alpha \times 0.1b_2b_3b_4b_5$)มีค่าเท่ากับศูนย์ จาก Multiply Rule

ส่วนค่าขอบเขตบนของ p_1จะมีค่าน้อยกว่าศูนย์ด้วย เพราะ upper($\alpha \times 0.1b_2b_3b_4b_5$)มีค่ามากกว่าศูนย์ จาก Multiply Rule

จาก q_2= Combine(SelectFIRS(Lower($2P_1$)), SelectFIRS(Upper($2P_1$))) ค่าที่เป็นไปได้ของ q_2 คือ 0 หรือ $\bar{1}$และผลลัพธ์ที่ได้อยู่ในช่วง [-1,1] ดังนั้นผลลัพธ์สองตำแหน่งแรกที่ได้จากกรณีนี้เป็น $\alpha 0$, $\alpha\gamma$หรือ $\alpha\bar{1}$ซึ่งอยู่ในรูปแบบบรรทัดฐาน

ในกรณีที่ 3.1.2: $a_1 = \gamma$และ $b_1 = \bar{1}$

เนื่องจาก N เป็นรูปแบบบรรทัดฐาน ดังนั้นค่าที่เป็นไปได้ของ a_2คือ α , 1

การพิสูจน์จะคล้ายกับกรณี3.1.1จะได้ว่า$q_1 = 0.0\gamma a_2a_3a_4 - 0.\bar{1}b_2b_3b_4b_5$ดังนั้นค่าของเขตล่างของ P_1 = lower($0.0\gamma a_2a_3a_4$) – lower($\alpha \times 0.\bar{1}b_2b_3b_4b_5$) ค่าเนื่องจาก N < D จะได้ว่า ค่าขอบเขตล่างที่เป็นไปได้ของP_1จะมีค่ามากกว่าศูนย์ เพราะ lower($0.0\gamma a_2a_3a_4$) มีค่าน้อยกว่าศูนย์เนื่องจากอยู่ในช่วง ลบบวก และ lower($\alpha \times 0.\bar{1}b_2b_3b_4b_5$) จาก Multiply Ruleจะได้ว่ามีค่าน้อยกว่าศูนย์

ส่วนค่าขอบเขตบนของ q_1จะมีค่ามากกว่าศูนย์ เพราะ upper($\alpha \times 0.\bar{1}b_2b_3b_4b_5$) มีค่าเท่ากับศูนย์

จาก q_2= Combine(SelectFIRS(Lower($2P_1$)), SelectFIRS(Upper($2P_1$))) ค่าที่เป็นไปได้ของ q_2 คือ 0 หรือ 1 และผลลัพธ์ที่ได้อยู่ในช่วง [-1,1] ดังนั้นผลลัพธ์สองตำแหน่งแรกที่ได้จากกรณีนี้เป็น $\gamma 0$, $\gamma\alpha$หรือ $\gamma 1$ซึ่งอยู่ในรูปแบบบรรทัดฐาน

กรณีที่ 3.2: $q_1 = \gamma$

ในกรณีนี้เกิดจากตัวถูกดำเนินการ a_1ขึ้นต้นด้วย γและ b_1 ขึ้นต้นด้วย 1 หรือ a_1ขึ้นต้นด้วย α และ b_1 ขึ้นต้นด้วย $\bar{1}$

กรณีที่ 3.2.1: $a_1=\gamma$และ b_1 = 1

เนื่องจาก N เป็นรูปแบบบรรทัดฐาน ดังนั้นค่าที่เป็นไปได้ของ a_2คือ α , 1

การพิสูจน์จะคล้ายกับกรณี3.1.1จะได้ว่าq_1= $0.0\gamma a_2a_3a_4$ - $0.1b_2b_3b_4b_5$ดังนั้นค่าของเขตล่างของ P_1 = lower($0.0\gamma a_2a_3a_4$) – lower($\gamma\times 0.1b_2b_3b_4b_5$) ค่าเนื่องจาก N < D จะได้ว่า ค่าขอบเขตล่างที่เป็นไปได้ของ P_1จะมีค่ามากกว่าศูนย์ เพราะ lower($0.0\gamma a_2a_3a_4$) มีค่าน้อยกว่าศูนย์เนื่องจากอยู่ในช่วง ลบบวก และ lower($\gamma\times 0.1b_2b_3b_4b_5$) จาก Multiply Ruleมีค่าน้อยกว่าศูนย์

ส่วนค่าขอบเขตบนของ q_1จะมีค่ามากกว่าศูนย์ เพราะ upper($\gamma\times 0.1b_2b_3b_4b_5$) มีค่าเท่ากับศูนย์

จาก q_2= Combine(SelectFIRS(Lower($2P_1$)), SelectFIRS(Upper($2P_1$))) ค่าที่เป็นไปได้ของ q_2 คือ 0 หรือ 1 และผลลัพธ์ที่ได้อยู่ในช่วง [-1,1] ดังนั้นผลลัพธ์สองตำแหน่งแรกที่ได้จากกรณีนี้เป็น $\gamma 0, \gamma\alpha$หรือ $\gamma 1$ซึ่งอยู่ในรูปแบบบรรทัดฐาน

กรณีที่ 3.2.2: $a_1=\alpha$ และ b_1 = $\bar{1}$

เนื่องจาก N เป็นรูปแบบบรรทัดฐาน ดังนั้นค่าที่เป็นไปได้ของ a_2คือ α , 1

การพิสูจน์จะคล้ายกับกรณี3.1.1จะได้ว่าq_1= $0.0\gamma a_2a_3a_4$ - $0.\bar{1}b_2b_3b_4b_5$ดังนั้นค่าของเขตล่างของ P_1 = lower($0.0\gamma a_2a_3a_4$) – lower($\gamma\times 0.\bar{1}b_2b_3b_4b_5$) ค่าเนื่องจาก N < D จะได้ว่า ค่าขอบเขตล่างที่เป็นไปได้ของ P_1จะมีค่าน้อยกว่าศูนย์ เพราะ lower($0.0\gamma a_2a_3a_4$) มีค่าน้อยกว่าศูนย์เนื่องจากอยู่ในช่วง ลบบวก และ lower($\gamma\times 0.\bar{1}b_2b_3b_4b_5$) จาก Multiply Ruleมีค่าเท่ากับศูนย์

ส่วนค่าขอบเขตบนของ P_1จะมีค่าน้อยกว่าศูนย์ เพราะ upper($\gamma\times 0.\bar{1}b_2b_3b_4b_5$) จาก Multiply Ruleมีค่ามากกว่าศูนย์

จาก q_2= Combine(SelectFIRS(Lower($2P_1$)), SelectFIRS(Upper($2P_1$))) ค่าที่เป็นไปได้ของ q_2 คือ 0 หรือ 1 และผลลัพธ์ที่ได้อยู่ในช่วง [-1,1] ดังนั้นผลลัพธ์สองตำแหน่งแรกที่ได้จากกรณีนี้เป็น $\alpha 0, \alpha\gamma$หรือ $\alpha\bar{1}$ซึ่งอยู่ในรูปแบบบรรทัดฐาน

จากทุกกรณีเราจะได้ว่าผลลัพธ์ที่ได้จากอัลกอริทึมจะอยู่ในรูปแบบบรรทัดฐาน ■

ตัวอย่างที่ 3.2*ให้ N = [-72, 95] =* $\alpha\gamma\bar{\alpha}0\gamma00\bar{\alpha}$ *และ D = [22, 35] =* $1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}$ *ซึ่งรูปแบบบรรทัดฐานของ N และ D คือ* $0.\alpha\gamma\bar{\alpha}0\gamma00\bar{\alpha}\times 2^8$*และ*$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}\times 2^7$*ตามลำดับหาผลลัพธ์จากการหารแบบเชื่อมตรงของ N ด้วย D ในระบบแทนช่วงแบบยืดหยุ่น*

ก่อนทำการหารแบบเชื่อมตรงจะต้องจัดรูปแบบบรรทัดฐาน N ให้เข้ากับเงื่อนไข $N<D$ โดยการ shift ไปทางขวาสองตำแหน่งได้ว่า $N = \mathbf{0.00\alpha\gamma\bar{\alpha}0\gamma00\bar{\alpha}} \times 2^{10}$ หลังจากนั้นจะเริ่มทำการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นตาม Algorithm2 จะแสดงดังตารางที่ 4.1

ตารางที่ 4.1ตัวอย่างการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

j	D_j	P_j	q_{j+1}
0	$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}$	$0.00\alpha\gamma\bar{\alpha}$	α
1	$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma$	$0.000\bar{1}\bar{\alpha}\alpha$	$\bar{1}$
2	$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}$	$0.0000\bar{\gamma}\alpha$	0
3	$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}0$	$0.000\bar{\gamma}\alpha$	$\bar{\gamma}$
4	$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}00$	$0.00\gamma10\bar{\gamma}$	α
5	$0.1\gamma\bar{\alpha}\bar{1}\bar{\alpha}\gamma\bar{\alpha}000$	$0.00\bar{1}\alpha\gamma$	$\bar{1}$

ผลลัพธ์ที่ได้จากอัลกอริทึมคือ $\alpha.\bar{1}0\bar{\gamma}\alpha\bar{1}$ ผลลัพธ์สุดท้ายหลังจากทำการจัดตำแหน่งทศนิยมแล้วจะได้

$$\alpha.\bar{1}0\bar{\gamma}\alpha\bar{1} \times 2^{10} / 2^{7} = \alpha\bar{1}0\bar{\gamma}.\alpha\bar{1}$$

ค่าของช่วงในเลขฐานสิบของ $\alpha\bar{1}0\bar{\gamma}.\alpha\bar{1}$ คือ $[0\bar{1}01.0\bar{1}, 1\bar{1}00.10]$ = [-3.25, 4.5] = [-72, 95] / [-22, 35] ซึ่งเป็นผลลัพธ์ที่ถูกต้อง

บทที่ 4

บทวิเคราะห์การหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่น

เนื้อหาในบทนี้กล่าวถึงการวิเคราะห์เกี่ยวกับระบบแทนช่วงแบบยืดหยุ่นและการหารแบบเชื่อมตรง โดยมีหัวข้อดังต่อไปนี้

4.1 เนื้อที่ที่ใช้ในการแสดงรูปแบบแทนช่วง

รูปแบบแทนช่วงแบบยืดหยุ่นที่ใช้สำหรับการหารแบบเชื่อมตรงคือ รูปแบบบรรทัดฐาน ซึ่งสามารถจำแนกเครื่องหมายของช่วงได้จากการพิจารณาตัวเลขสองบิตแรกของชุดตัวเลขนั้นใช้พื้นที่ในการแทนจำนวนเท่ากับรูปแบบแทนช่วงแบบยืดหยุ่นเดิมจึงมีประสิทธิภาพในเชิงพื้นที่ไม่ต่างกับรูปแบบแทนช่วงแบบยืดหยุ่นเดิม ส่วนจำนวนรูปแบบของการแสดงจำนวนของ รูปแบบบรรทัดฐานแม้ว่าจะมีจำนวนรูปแบบน้อยกว่ารูปแบบแทนช่วงแบบยืดหยุ่นเดิมแต่ก็สามารถแสดงได้ครอบคลุมทุกช่วง

4.2 เวลาที่ใช้ในการคำนวณ

เวลาที่ใช้ในการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นเท่ากันกับการหารแบบเชื่อมตรงดั้งเดิมซึ่งใช้เวลาเท่ากับ $O(n^2)$ จึงมีประสิทธิภาพเชิงเวลาไม่ต่างกัน ทั้งนี้การหารแบบเชื่อมตรงดั้งเดิมและการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นสามารถดำเนินการแบบท่อตรงได้ทำให้สามารถเพิ่มประสิทธิภาพในการคำนวณแบบลำดับได้ ข้อดีของการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นคือผลลัพธ์ที่ได้จากการคำนวณจะมีความละเอียดสูงกว่าการหารแบบเชื่อมตรงเดิมเพราะการหารแบบเชื่อมตรงเดิมสามารถดำเนินการกับตัวเลขที่มีความยาวจำกัด ทำให้มีการปัดเศษในบางตำแหน่งเพื่อให้สามารถทำงานต่อได้ ผลลัพธ์ที่ได้จึงมีความคลาดเคลื่อนเมื่อเทียบกับการหารแบบเชื่อมตรงบนระบบแทนช่วงแบบยืดหยุ่นที่สามารถดำเนินการกับตัวเลขที่มีความยาวไม่จำกัดได้ ดังนั้นผลลัพธ์ที่ได้จากการดำเนินการจึงมีความละเอียดสูงกว่าหากเราต้องการผลลัพธ์ที่มีความละเอียดเท่าใดก็สามารถหยุดการทำงานตามที่ต้องการ

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

งานวิจัยนี้ได้ปรับปรุงระบบแทนช่วงแบบยืดหยุ่นให้สามารถดำเนินการหารแบบเชื่อมตรงได้ซึ่งจะเรียกว่ารูปแบบบรรทัดฐาน

รูปแบบบรรทัดฐานนี้จะมีค่าของช่วงมากกว่า -1 และน้อยกว่า 1 นอกจากนั้นยังสามารถพิจารณาเครื่องหมายได้จากสองตำแหน่งแรกของชุดตัวเลขโดยจะจำแนกช่วงของข้อมูลออกเป็น 5 กรณีด้วยกันคือ ช่วงบวก-บวก เมื่อสองตำแหน่งแรกขึ้นต้นด้วย 1, ช่วงศูนย์-บวก เมื่อสองตำแหน่งแรกขึ้นต้นด้วย $\alpha 0$ หรือ $\alpha\alpha$, ช่วงลบ-ลบ เมื่อสองตำแหน่งแรกขึ้นต้นด้วย $\bar{1}$, ช่วงลบ-ศูนย์ เมื่อสองตำแหน่งแรกขึ้นต้นด้วย $\gamma 0$ หรือ $\gamma\gamma$, ช่วงลบบวก เมื่อสองตำแหน่งแรกขึ้นต้นด้วย $\alpha\bar{1}$, $\alpha\gamma$, $\gamma 1$ หรือ $\gamma\alpha$เราสามารถนำรูปแบบบรรทัดฐานนี้มาทำการหารแบบเชื่อมตรงได้โดยที่ค่าความหน่วงมากกว่าหรือเท่ากับห้า

5.2 ข้อเสนอแนะ

สำหรับในงานวิจัยนี้สิ่งที่จะเสนอแนะคือ การออกแบบให้รูปแบบบรรทัดฐานสำหรับระบบแทนช่วงแบบยืดหยุ่นนี้สามารถใช้กับการดำเนินการอื่นเช่น การหารากแบบเชื่อมตรง

รายการอ้างอิง

[1] Avizienis, A.A., Signed-Digit Number Representations for Fast Parallel Arithmetic. IRE Transactions of Electronic Computers, 10(1961) : 389-400.

[2] Kearfott, R.B., Interval Computations: Introduction, Uses, and Resources, Euromath Bulletin, 2(1996) : 95-112.

[3] Thienprapasith, P., Surarerks, A., Flexible interval representation system. Proceedings of the 10th Annual National Symposium on Computation Science and Engineering, (2006) : 327-332.

[4] Thienprapasith, P., Surarerks, A., A Flexible interval Rrepresentation System and Its Fundamental Arithmetic Operations. Proceedings of the 5th International Conference on Information Technology and Applications, (2008).

[5] Trivedi, K.S., Ercegovac, M.D., On-Line Algorithm for Division and Multiplication. IEE Transactions on Computer, (1977), 681-687.

[6] Sukontarach, J., Surarerks, A., Parallel Addition and Subtraction for the flexible interval representation system. Proceedings of the 12th International Nation Computer Science and Engineering Conference, (1998), 9-14.

[7] Taweepanyayote, P., Surarerks, A., On-line Multiplication on signed Flexible Interval Representation System. TENCON 2011 - 2011 IEEE Region 10 Conference, (2011), 1164-1169.

[8] Frougny, C., Surarerks, A., On-line multiplication in real and complex base, Proceedings of the 16th IEEE Symposium on Computer Arithmetic, (2003) : 212-219.

[9] Parhami , B., Generalized Signed-Digit Number Systems: A Unifying Framework for Redundant Number Representations,IEEE Transactions on Computers, 39(1990) : 89-98.

[10] Surarerks , A., Digit Set Conversion by On-Line Finite Automata, Bull. Belg. Math. Soc., (1993) :337-358.

ประวัติผู้เขียนวิทยานิพนธ์

นายศิษฏิวัชร์ เสริมสุขสกุลชัย เกิดเมื่อวันที่ 9 สิงหาคม 2527 เรียนจบการศึกษาระดับชั้นมัธยมศึกษาที่โรงเรียนวัดสุทธิวราราม กรุงเทพมหานคร และจบการศึกษาระดับปริญญาบัณฑิต สาขาคณิตศาสตร์ ภาควิชาคณิตศาสตร์ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2549 และศึกษาต่อในระดับปริญญามหาบัณฑิตที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีการศึกษา 2553