อัลกอริทึมการลบและการหารสำหรับระบบจำนวนฐานคู่

นายเอกพล มลฑลจุลเกศ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2552

SUBTRACTION AND DIVISION ALGORITHM FOR DOUBLE BASE NUMBER SYSTEM

MR.EKAPHON MONTHONJULAKET

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 2009

หัวข้อวิทยานิพนธ์ อัลกอริทึมการลบและการหารสำหรับระบบจำนวนฐานคู่

โดย นายเอกพล มลฑลจุลเกศ

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.... คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.บุญสม เลิศหิรัญวงศ์)

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา)

.... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

.... กรรมการภายนอกมหาวิทยาลัย

(ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง)

เอกพล มลฑลจุลเกศ : อัลกอริทึมการลบและการหารสำหรับระบบจำนวนฐานคู่. (SUBTRACTION AND DIVISION ALGORITHM FOR DOUBLE BASE NUMBER SYSTEM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.อรรถสิทธิ์ สุรฤกษ์, 47หน้า.

ระบบจำนวนมีบทบาทสำคัญต่อความเร็วในการคำนวณทางเลขคณิตในระบบคอมพิวเตอร์ ดังนั้นจึงมีการออกแบบระบบจำนวนชนิดใหม่เป็นจำนวนมาก ซึ่งระบบจำนวนฐานคู่เป็นระบบจำนวนระบบหนึ่งที่ถูกคิดค้นขึ้นมาเพื่อเพิ่มประสิทธิภาพในการคำนวณ ระบบจำนวนฐานคู่เป็นระบบที่ใช้แสดงจำนวนเต็มบวก โดยใช้ฐานสองฐาน คือ 2 และ 3 จุดเด่นของระบบจำนวนนี้คือความซ้ำซ้อนสูง และการกระจายตัวสูง มีงานวิจัยหลายงานที่ได้นำเสนอเกี่ยวกับตัวปฏิบัติการพื้นฐานของระบบจำนวนนี้ ซึ่งได้นำเสนอกระบวนการบวก และกระบวนการคูณเท่านั้น งานวิจัยนี้จึงนำเสนออัลกอริทึมการลบ และอัลกอริทึมการหารสำหรับระบบจำนวนฐานคู่ที่มีการทำงานเชิงกำหนด เทคนิคการสลับบิตและการบวกถูกนำมาใช้ในอัลกอริทึมการลบ ซึ่งผลลัพธ์ของการลบจะอยู่ในรูปแบบแทนจำนวนในระบบจำนวนฐานคู่ กระบวนการลบมีความซับซ้อนเชิงเวลาไม่เกินเวลาโพลีโนเมียล ส่วนกระบวนการหารนั้นจะสร้างจากกระบวนการบวก กระบวนการคูณที่มีอยู่ก่อนแล้ว และกระบวนการลบที่สร้างขึ้นมาใหม่ พร้อมทั้งพิสูจน์ความถูกต้องของอัลกอริทึมทั้งสอง

ภาควิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต
สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออ.ที่ปรึกษาวิทยานิพนธ์หลัก
ปีการศึกษา 2552

5071461721 : MAJOR COMPUTER SCIENCE
KEYWORDS : DOUBLE BASE NUMBER SYSTEM / ARITHMETIC OPERATIONS / COMPUTER ARITHMETIC / SUBTRACTION / DIVISION

EKAPHON MONTHONJULAKET : SUBTRACTION AND DIVISION ALGORITHM FOR DOUBLE BASE NUMBER SYSTEM. ADVISOR : ASSOCIATE PROFESSOR ATHASIT SURARERKS, 47 pp.

The number system plays an important role in computer arithmetic especially on the speed of computation. Several number systems have been introduced for that reason. A double base number system has been introduced for improving the performance of arithmetic. This system can represent only non-negative numbers by using two and three as the bases. The redundancy is the prominent point of this number system. Fundamental arithmetic operations such as addition and multiplication are the majority of research in this system. Our research is focused on an implementation for subtraction and division operations. The finite state algorithm is introduced for subtraction and division operations. One-complement technique and addition technique are applied to our algorithm to accomplish the subtraction. Theoretical results show that the proposed subtraction can be realized for double base number system. The time complexity of subtraction algorithm is polynomial on the size of the operands. For division, addition technique, multiplication technique and a new subtraction technique are applied into this algorithm to accomplish the division. The proof of algorithm is also provided in our research.

Department : Computer Engineering Student's Signature Ekaphon Monthonjulaket
Field of Study : Computer Science Advisor's Signature Athasit Surarerks
Academic Year : 2009

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยความช่วยเหลืออย่างดียิ่งจาก ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ให้คำปรึกษา ให้ข้อเสนอแนะที่เป็นประโยชน์อย่างยิ่งต่อการวิจัย ช่วยตรวจแก้ไขในส่วนที่บกพร่องต่างๆ และช่วยติดตามดูแลงานวิจัยอย่างใกล้ชิดมาโดยตลอด ขอขอบพระคุณอาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ เป็นอย่างสูงที่ช่วยดูแลงานวิจัยจนบรรลุผลสำเร็จเป็นอย่างดี

ขอขอบพระคุณในความเอื้อเฟื้อของ ศาสตราจารย์ ดร.ประภาส จงสถิตย์วัฒนา ผู้ช่วยศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง ประธานกรรมการและกรรมการสอบวิทยานิพนธ์ ที่ได้กรุณาให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น และขอขอบพระคุณคณาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัยทุกท่านที่ประสิทธิประสาทความรู้อันมีค่ายิ่งแก่ผู้วิจัย

กราบขอบพระคุณ บิดา มารดา ที่มีความเมตตา กรุณา และเป็นผู้คอยให้กำลังใจตลอดมา ขอขอบคุณ พี่ๆ เพื่อนๆ และน้องๆ ทุกคน โดยเฉพาะสมาชิกห้องปฏิบัติการวิจัยทางทฤษฎีวิศวกรรมระบบเชิงนับได้ (ELITE) ที่ได้ให้ความช่วยเหลือและแก้ไขเอกสาร จนกระทั่งวิทยานิพนธ์สำเร็จเป็นรูปเล่ม และขอขอบคุณแรงสนับสนุนและกำลังใจทุกท่านที่มิได้กล่าวนามไว้ ณ ที่นี้

สุดท้ายนี้ ผู้วิจัยหวังเป็นอย่างยิ่งว่างานวิจัยนี้จะเป็นประโยชน์ต่อผู้ที่สนใจหรือผู้ที่เกี่ยวข้องในระบบจำนวนและการคำนวณเลขคณิตในระบบคอมพิวเตอร์ และหากมีข้อผิดพลาดประการใด ผู้วิจัยขออภัยมา ณ ที่นี้ด้วย

สารบัญ

หน้า

สารบัญตาราง

ตารางที่ หน้า

สารบัญภาพ

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบจำนวนต่างๆสำหรับระบบคอมพิวเตอร์นั้น ถูกออกแบบมาเพื่อรองรับความต้องการทางด้านการคำนวณที่หลากหลาย ซึ่งงานวิจัยทางด้านเลขคณิตคอมพิวเตอร์นั้นมีวัตถุประสงค์เพื่อทำให้การคำนวณมีประสิทธิภาพสูงขึ้น เช่น รองรับการคำนวณที่มีความซับซ้อนสูง เพิ่มความรวดเร็วในการคำนวณ ความถูกต้องในการคำนวณ และการประหยัดพลังงาน เป็นต้น โดยที่แต่ละระบบจำนวนมีจุดเด่นและจุดด้อยแตกต่างกัน เช่น ระบบจำนวนลอการิทึม (logarithm number system) ที่ถูกนำเสนอโดย ดิมิทรอฟ (V. Dimitrov) [1] ในปี ค.ศ.1998 เป็นระบบจำนวนที่สามารถทำการคูณและหารได้อย่างรวดเร็วเนื่องจากคุณสมบัติของลอการิทึม แต่ระบบจำนวนนี้จำเป็นต้องใช้การสร้างตารางเรียกดูค่า ซึ่งเป็นตารางขนาดใหญ่ส่งผลให้เป็นการสิ้นเปลืองเนื้อที่ ต่อมาในปี ค.ศ.2007 มีงานวิจัยของพิชาญและอรรถสิทธิ์ [2] ได้พัฒนาระบบจำนวนลอกอริทึมให้สามารถทำการคูณและหารได้โดยไม่ต้องใช้ตารางเรียกดูค่า แต่ระบบจำนวนดังกล่าวจำเป็นต้องทำการประมาณค่าเพื่อใช้ในการคำนวณ ทำให้ค่าความถูกต้องและความแม่นยำลดน้อยลง นอกจากนี้ยังมีระบบจำนวนที่น่าสนใจอีกระบบคือ ระบบจำนวนฐานคู่ (double base number system) ที่ถูกนำเสนอโดย ดิมิทรอฟ [3] ในปี ค.ศ.1999 เป็นระบบจำนวนที่สามารถทำการบวกและคูณได้รวดเร็ว อีกทั้งยังสามารถทำการคำนวณแบบขนานได้ โดยใช้คุณสมบัติของตารางสองมิติในการแสดงจำนวน โดยมีเลขฐานสองจำนวนที่ต่างกัน คือ ฐานสองและฐานสาม ซึ่งคำตอบที่ได้จากการคำนวณในระบบจำนวนฐานคู่นี้มีความถูกต้องและแม่นยำ จุดเด่นที่สำคัญอีกประการหนึ่งของระบบจำนวนฐานคู่คือการนำระบบจำนวนซ้ำซ้อน (redundant number system) เข้ามาประยุกต์ใช้ ซึ่งระบบจำนวนซ้ำซ้อนมีสมบัติที่สำคัญประการหนึ่งคือ สมบัติซ้ำซ้อน (redundant property) หมายถึง จำนวนอย่างน้อยหนึ่งจำนวน สามารถมีรูปแบบแทนจำนวนได้มากกว่าหนึ่งรูปแบบ จากปัญหาต่างๆของระบบจำนวนฐานคู่ทำให้มีงานวิจัย [4,5] ที่นำเสนอวิธีการในการหารูปแบบแทนจำนวนที่มีขนาดเล็กที่สุดเพื่อให้สามารถดำเนินการในการคำนวณได้รวดเร็วขึ้น

เนื่องจากการดำเนินการทางเลขคณิตคอมพิวเตอร์ของระบบจำนวนฐานคู่ยังไม่สมบรูณ์ ซึ่งมีเพียงงานวิจัย [6-8] ที่ได้กล่าวถึงการสร้างตัวดำเนินการบวก และคูณเท่านั้น เพราะในระบบจำนวนฐานคู่นั้นสามารถแสดงจำนวนได้เฉพาะจำนวนเต็มที่ไม่เป็นลบ ทำให้การดำเนินการลบไม่

สามารถทำได้ในทุกกรณี การลบจะทำได้ก็ต่อเมื่อค่าของตัวตั้งมีค่ามากกว่าค่าของตัวลบ แต่การที่จะรู้ว่าค่าใดมากกว่าหรือน้อยกว่ากันนั้นจำเป็นต้องอาศัยการเปรียบเทียบ ซึ่งในระบบนี้ยังไม่มีการกล่าวถึงเรื่องการเปรียบเทียบค่าของจำนวนที่ถูกแสดงด้วยระบบจำนวนฐานคู่มาก่อน อีกทั้งการดำเนินการลบต้องคำนึงถึงกรณีต่างๆ ที่สามารถเกิดขึ้นได้จากความหลากหลายของรูปแบบแทนจำนวน

ดังนั้นงานวิจัยนี้จึงได้สนใจการนำเสนอกระบวนการสร้างตัวดำเนินการลบและการหารที่ให้ผลเป็นจำนวนบวก และมีลำดับการทำงานที่แน่นอน โดยมุ่งเน้นวิธีการสร้างกระบวนการลบและการหารในระบบจำนวนฐานคู่สองและสาม หรืออัลกอริทึมการลบเชิงกำหนดสำหรับระบบจำนวนฐานคู่สองและสาม และอัลกอริทึมการหารเชิงกำหนดสำหรับระบบจำนวนฐานคู่สองและสาม

1.2 วัตถุประสงค์ของการวิจัย

งานวิจัยนี้จะออกแบบกระบวนการในการคำนวณพื้นฐาน คือการลบและการหารที่ให้ผลเป็นจำนวนบวกของระบบจำนวนฐานคู่ ที่เป็นกระบวนการเชิงกำหนด โดยเสนอเป็นอัลกอริทึมของการลบสำหรับระบบเลขฐานคู่ และอัลกอริทึมของการหารสำหรับระบบเลขฐานคู่

1.3 ขอบเขตของการวิจัย

1. ข้อมูลนำเข้าสำหรับการคำนวณต้องเป็นรูปแบบแทนจำนวนพร้อมบวกทั้งสองจำนวน
2. อัลกอริทึมการลบรองรับการลบที่ให้ผลลัพธ์ที่สามารถแสดงได้ในรูปแบบแทนจำนวนเสมอ กล่าวคือค่าเชิงตัวเลขของผลลัพธ์จะไม่เป็นจำนวนลบ
3. ขนาดของผลลัพธ์จากการลบจะมีขนาดไม่เกิน $(N+3) \times M$ เมื่อ N คือจำนวนหลักของข้อมูลตัวตั้งและตัวลบ และ M คือจำนวนแถวของข้อมูลตัวตั้งและตัวลบ
4. ความซับซ้อนเชิงเวลา (time complexity) ของการลบจะต้องไม่เกินเวลาพหุนาม (polynomial time)

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1. เพิ่มความสามารถของระบบจำนวนฐานคู่ให้สามารถดำเนินการลบได้
2. ทำให้ระบบจำนวนฐานคู่มีการใช้งานได้แพร่หลายมากขึ้น

1.5 วิธีดำเนินการวิจัย

1. ศึกษาสมบัติของระบบจำนวนฐานคู่ และงานวิจัยที่เกี่ยวข้อง
2. วิเคราะห์ปัญหาของงานวิจัยที่มีความสอดคล้องกับงานวิจัยที่สนใจ
3. ออกแบบกระบวนการลบของระบบจำนวนฐานคู่ที่มีฐานเป็นสอง และสาม พร้อมทั้งพิสูจน์การทำงาน และวิเคราะห์ความซับซ้อนเชิงเวลา
4. พิสูจน์อัลกอริทึมที่ได้ออกแบบไว้
5. สรุปผล และเรียบเรียงวิทยานิพนธ์

1.6 ผลงานที่ตีพิมพ์จากวิทยานิพนธ์

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้รับการตีพิมพ์เป็นบทความทางวิชาการในหัวข้อเรื่องดังต่อไปนี้

"Subtraction algorithm for Double Base Number System" โดย เอกพล มลฑลจุลเกศ และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ 13th International ANnual Symposium on Computational Science and Engineering (ANSCSE13) ณ มหาวิทยาลัยเกษตรศาสตร์ จังหวัดกรุงเทพ ประเทศไทย ระหว่างวันที่ 25 – 27 มีนาคม พ.ศ. 2552

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 **ระบบจำนวนเลขฐาน** (base number system)

ระบบจำนวนเลขฐาน ประกอบด้วยสองส่วนคือ เลขฐาน β โดยเลขฐาน β เป็นได้ทั้งจำนวนจริงหรือจำนวนเชิงซ้อน และ D เป็นเซตจำกัดของตัวเลขที่สามารถเป็นได้ทั้งจำนวนจริงหรือจำนวนเชิงซ้อน

กำหนดให้ X เป็นจำนวนใดๆ X สามารถแสดงได้ในระบบจำนวนเลขฐาน β ในรูปแบบ

$$X = (x_n x_{n-1} \dots x_0 x_{-1}\, x_{-2} \dots)_\beta$$

ซึ่ง $x_k \in D$ โดยที่ $k \le n,\ n \in Z$

โดยค่าเชิงตัวเลขของ X ฐาน β สามารถเขียนให้อยู่ในรูป

$$\|X\| = \sum_{k=n}^{-\infty} x_k \beta^l$$

โดยที่ $l \in Z$

โดยค่าเชิงตัวเลขทั้งหมดที่สามารถแสดงได้ จะเขียนให้อยู่ในรูปของเซต $P[\beta, D]$ ได้ดังนี้

$$P_n^m[\beta, D] = \{X = (x_n x_{n-1} \cdots x_{m+1} x_m)_\beta \mid x_k \in D, m \le k \le n\}$$
$$P_n\ [\beta, D] = \{X = (x_n x_{n-1} \cdots x_0 x_{-1} x_{-2} \cdots)_\beta \mid x_k \in D, k \le n\}$$

โดยที่ $P_n^m[\beta, D]$และ $P_n[\beta, D]$ เท่ากับ เซตจำกัดและเซตไม่จำกัด ตามลำดับ

ระบบเลขฐานจำนวนเต็มโดยทั่วไปแล้วนิยมให้ $D = \{0, 1, \cdots, |\beta| - 1\}$ ซึ่ง D จะถูกเรียกว่าชุดตัวเลขแบบบัญญัติ (canonical digit-set)

2.2 **ระบบจำนวนซ้ำซ้อน** (Redundant Number System)

ระบบจำนวนซ้ำซ้อน คือ ระบบจำนวนที่มีรูปแบบแทนจำนวนของค่าเชิงตัวเลข (numerical value) ของ X หลายรูปแบบ เช่น กำหนดให้ β เป็นเลขฐาน โดยที่ β เป็นจำนวนเต็มที่ $\beta \ge 2$ และกำหนดให้ D เป็นชุดตัวเลขซึ่งอธิบายได้ด้วย $\{d \in Z \mid \alpha_1 \le d \le \alpha_2\}$ โดย $-\alpha_1 + \alpha_2 \le \beta$ ยกตัวอย่างเช่น ชุดตัวเลข $D = \{\bar{3}, \bar{2}, \bar{1}, 0, 1, 2, 3\}$ บนเลขฐาน $\beta = 6$ และค่าเชิงตัวเลข $X = 51$ สามารถแสดงรูปแบบได้ดังนี้

$$(1\ 2\ 3)_6 = 1\times 6^2 + 2\times 6^1 + 3\times 6^0$$

$$(1\ 3\ \bar{3})_6 = 1\times 6^2 + 3\times 6^1 + (-3)\times 6^0$$

$$(2\ \bar{3}\ \bar{3})_6 = 2\times 6^2 + (-3)\times 6^1 + (-3)\times 6^0$$

จากตัวอย่างข้างต้นเมื่อ α_1 และ α_2 เป็นไปตามเงื่อนไขดังกล่าว ระบบจำนวนจะมีคุณสมบัติของความซ้ำซ้อน เนื่องจากสามารถแสดงค่าของตัวเลข 51 ได้มากกว่าหนึ่งรูปแบบ

โดยสามารถเขียนอธิบายรูปแบบแทนจำนวนทั้งหมดของ x ได้ดังนี้

$$V_{(\beta,D)}(x) = \{P \in P[\beta, D] \mid \|P\| = x\}$$

โดยที่ $P[\beta, D] = \{P = (d_j d_{j-1} \cdots)_\beta \mid d_j \in D\}$, $\|P\| = \sum_{j=1}^{n} d_j \beta^k$ และ $k \in Z$

ดังนั้น สามารถกล่าวได้ว่าระบบจำนวนเต็มบวกใดๆ มีคุณสมบัติ

สมบูรณ์ (complete for radix β)	ก็ต่อเมื่อ $\forall_j \in Z: \left|V_{(\beta,D)}(j)\right| \geq 1$
กึ่งสมบูรณ์ (semi-complete for radix β)	ก็ต่อเมื่อ $\forall_j \in N: \left|V_{(\beta,D)}(j)\right| \geq 1$
ซ้ำซ้อน (redundant for radix β)	ก็ต่อเมื่อ $\exists_j \in Z: \left|V_{(\beta,D)}(j)\right| > 1$
ไม่ซ้ำซ้อน (non-redundant for radix β)	ก็ต่อเมื่อ $\forall_j \in Z: \left|V_{(\beta,D)}(j)\right| \leq 1$

2.3 **ระบบจำนวนฐานคู่** (double-base number system)

ระบบจำนวนฐานคู่ถูกคิดค้นโดยดิมิทรอฟ (Dimitrov) [3] ในปี ค.ศ. 1999 เป็นระบบจำนวนที่ประกอบด้วยเลขฐานจำนวนสองฐาน คือฐานสอง และฐานสาม ดังนิยามต่อไปนี้

นิยาม 1: ระบบจำนวนฐานคู่ *ประกอบด้วยฐานสองและฐานสาม และชุดตัวเลข* $D = \{0,1\}$ *โดยรูปแบบแทนจำนวน (number representation)* X *อยู่ในรูปของ*

$$X = \sum_{i,j} d_{i,j} 2^i 3^j \text{ เมื่อ } d_{i,j} \in D$$

ค่าเชิงตัวเลขของ X *เขียนแทนด้วย* $\|X\|$ *ซึ่งมีค่าเท่ากับ*

$$\|X\| = \sum_{i,j} d_{i,j} 2^i 3^j,\ d_{i,j} \in D$$

ระบบจำนวนฐานคู่นี้สามารถแสดงเป็นตารางสองมิติโดยแกน x จะเป็นแกนของเลขฐานสอง และ y จะเป็นแกนของเลขฐานสาม ดังรูปที่ 2.1 โดยที่ช่องสีดำจะเป็นตำแหน่งที่ $d_{i,j}$ มีค่าเป็น 1 ช่องสีขาวจะเป็นตำแหน่งที่ $d_{i,j}$ มีค่าเป็น 0

รูปที่ 2.1 ตารางสองมิติแสดงค่าของ 31

จากรูปที่ 2.1 เป็นการแสดงค่าของ $31 = 2^0 3^0 + 2^2 3^1 + 2^1 3^2$ ในรูปแบบตารางสองมิติ และเนื่องจากระบบฐานคู่เป็นระบบจำนวนแบบซ้ำซ้อน ดังนั้นรูปแบบแทนจำนวนของค่า 31 จะมีมากกว่า 1 รูปแบบในระบบจำนวนฐานคู่ (Double-Base number representation: DBNR) เช่น

$$\begin{aligned} 31 &= 2^0 3^0 + 2^0 3^1 + 2^0 3^2 + 2^1 3^2 \\ &= 2^0 3^0 + 2^0 3^1 + 2^0 3^3 \\ &= 2^0 3^0 + 2^1 3^1 + 2^3 3^1 \end{aligned}$$

จากความซ้ำซ้อนดังกล่าว ทำให้สังเกตได้ว่ารูปแบบแทนจำนวนที่แตกต่างกันจะมีจำนวนของเลขหนึ่ง ($d_{i,j} = 1$) ที่ไม่เท่ากัน

นิยาม 2 *รูปแบบแทนจำนวนที่มีจำนวนของตัวเลขหนึ่งน้อยที่สุดจะถูกเรียกว่า* รูปแบบคาโนนิคอล *(canonical representation)*

รูปแบบคาโนนิคคอลมีข้อดีคือ สามารถคำนวณได้เร็วในการคำนวณพื้นฐาน เนื่องจากมีโอกาสน้อยที่จะการเกิดการทดหรือการยืม แต่กระบวนการในการหารูปแบบคาโนนิคคอลมีความซับซ้อน ดังนั้นจึงมีงานวิจัย [4] ที่ถูกเสนอโดย เบอร์เต (V. Berthé) ในปี ค.ศ.2004 ได้เสนอวิธีในการหารูปแบบแทนจำนวนที่มีจำนวนของตัวเลขหนึ่งที่ใกล้เคียงกับจำนวนของตัวเลขหนึ่งที่น้อยที่สุด โดยจะเรียกรูปแบบแทนค่าที่ใช้อัลกอริทึมเชิงละโมบ (greedy algorithm) ในการหา ว่า รูปแบบใกล้คาโนนิคคอล (near-canonical representation) ซึ่งความซับซ้อนของอัลกอริทึมเป็น $O(\log x \,/\, (\log\log x))$

ต่อมาในปี ค.ศ.2005 จิลเบอร์ และแลงโลสิ (G. Gilbert and J. M. P. Langlois) [5] ได้ปรับปรุงวิธีในการหารูปแบบคาโนนิคคอลซึ่งจากเดิมนั้นจะใช้อัลกอริทึมเชิงละโมบในการหา โดยจะหาได้เพียงรูปแบบเดียว แต่รูปแบบคาโนนิคคอลในระบบฐานคู่นั้นอาจมีมากกว่าหนึ่งรูปแบบได้ โดยจิลเบอร์และแลงโลสิได้เสนออัลกอริทึมเชิงละโมบแบบหลายเส้นทาง (mutipath greedy algorithm) ขึ้นเพื่อใช้หารูปแบบคาโนนิคคอล โดยสามารถหารูปแบบคาโนนิคคอลได้มากกว่าหนึ่งรูปแบบ

2.4 การบวกของระบบจำนวนฐานคู่ (addition for double-base number system)

งานวิจัยของดิมิทรอฟ (V. Dimitrov) [3] ได้กล่าวถึงแนวคิดพื้นฐานสำหรับกระบวนการบวกของระบบฐานคู่ไว้ดังนี้ ก่อนที่จะทำการบวกตัวตั้งและตัวบวกต้องจัดให้อยู่ในรูปแบบพร้อมบวกก่อน (addition ready DBNS: ARDBNR) โดยให้คำนิยามรูปแบบพร้อมบวกดังนี้

นิยาม 3 รูปแบบพร้อมบวก (ARDBNR) คือ *รูปแบบแทนจำนวนในระบบจำนวนฐานคู่ที่ไม่มีตำแหน่งที่มีค่าอยู่ติดกันทั้งในแนวนอนและแนวตั้ง*

งานวิจัยดังกล่าวได้เสนอการเปลี่ยนรูปแบบแทนจำนวนใดๆ ในระบบจำนวนฐานคู่ให้อยู่ในรูปแบบพร้อมบวกโดยใช้กฎพื้นฐานเพียงสองกฎดังนี้

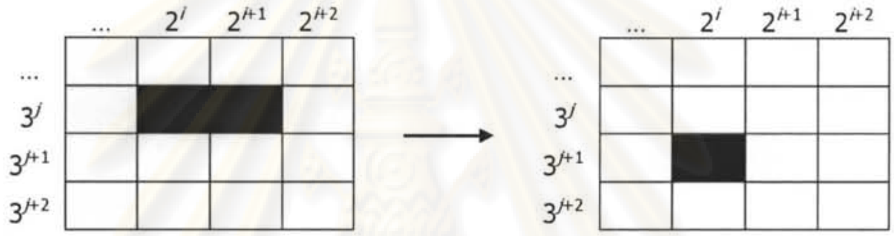
กฎที่ 1 กฎการลดแถว (row reduction)

$$2^i3^j + 2^{i+1}3^j = 2^i3^{j+1}$$

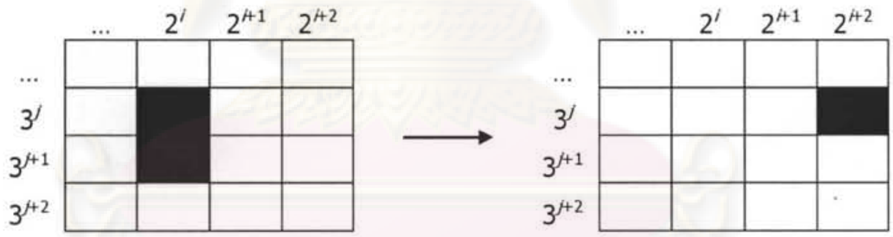
กฎที่ 2 กฎการลดหลัก (column reduction)

$$2^i3^j + 2^i3^{j+1} = 2^{i+2}3^j$$

จากกฎสองข้อดังกล่าวสามารถแสดงให้เข้าใจง่ายได้ดังรูปที่ 2.2 และ 2.3 ดังนี้



รูปที่ 2.2 กฎการลดแถว



รูปที่ 2.3 กฎการลดหลัก

กระบวนการบวกสามารถทำได้โดยนำตารางแสดงค่าตัวตั้งและตารางแสดงค่าตัวบวกที่อยู่ในรูปแบบพร้อมบวกมาซ้อนทับกัน โดยถ้าตำแหน่งที่มีค่าของตัวตั้งและตัวบวกถูกซ้อนทับกัน (collision active cells) จะทำให้เกิดสายการทดในหลักถัดไปของแถวเดียวกัน โดยมีการอธิบายแนวความคิดดังกล่าวด้วยตัวปฏิบัติการทางตรรกศาสตร์ดังนี้

กำหนดให้ $I_x(i,j)$ เป็นค่าในตำแหน่งที่ (i,j) ในตารางสองมิติที่ใช้ในการแสดงค่าของ x ให้ $I_y(i,j)$ เป็นค่าในตำแหน่งที่ (i,j) ในตารางสองมิติที่ใช้ในการแสดงค่าของ y และให้ $I_z(i,j)$ เป็นค่าในตำแหน่งที่ (i,j) ในตารางสองมิติที่ใช้ในการแสดงค่าของ z

ดังนั้นแต่ละตำแหน่งในตารางสองมิติที่ใช้ในการแสดงค่าของ $z = x + y$ สามารถแสดงได้ด้วยกฎสองข้อดังนี้

กฎที่ 1 $I_z(i + 1,j) = I_x(i,j)$ AND $I_y(i,j)$

กฎที่ 2 $I_z(i,j) = I_x(i,j)$ XOR $I_y(i,j)$

จากกฎที่ 1 ส่งผลให้เกิดสายการทดที่จะเกิดขึ้นในหลักถัดไปแต่ด้วยเงื่อนไขของค่าที่จะนำมาบวกกันอยู่ในรูปแบบพร้อมบวกทำให้เกิดสายการทดไม่เกินหนึ่งขั้นเสมอ ส่วนกฎที่ 2 ไม่ทำให้เกิดสายการทด

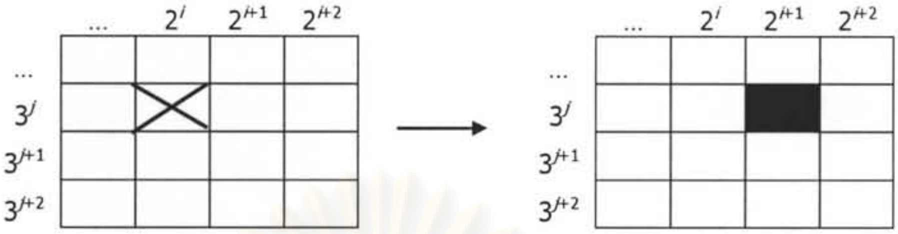
อย่างไรก็ตามผลลัพธ์หลังจากการบวกนั้นอาจไม่อยู่ในรูปแบบพร้อมบวก ดังนั้นจึงต้องมีการใช้กฎการลดแถวและลดหลัก ซึ่งแสดงด้วยรูปแบบตรรกศาสตร์ได้ดังนี้

กฎที่ 3 $I_z(i,j + 1) = I_z(i,j)$ AND $I_z(i + 1, j)$

กฎที่ 4 $I_z(i +2,j) = I_z(i,j)$ AND $I_z(i, j + 1)$

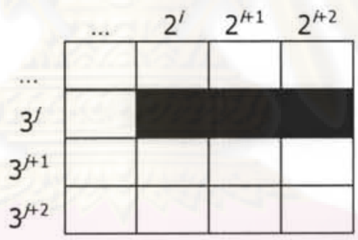
แนวคิดดังกล่าวเป็นเพียงแนวคิดพื้นฐานในการอธิบายว่ากระบวนการบวกสามารถทำได้ในระบบจำนวนฐานคู่ แต่แนวคิดดังกล่าวยังไม่ได้เสนอวิธีการนำกฎต่างๆ มาใช้สร้างเป็นขั้นตอนการบวกซึ่งมีผลต่อเวลาที่ใช้ในขั้นตอนการลดรูปเพื่อให้อยู่ในรูปแบบพร้อมบวก และยังไม่ได้คำนึงถึงกรณีพิเศษต่างๆ ที่สามารถเกิดขึ้นได้จากสายการทดที่เกิดขึ้น

ต่อมาในปี ค.ศ.2005 ได้มีงานวิจัยของพานคาลา (Pankaala) [6] ได้นำเสนอแนวความคิดเพิ่มเติมในกระบวนการบวก โดยได้เสนอว่าการบวกเลขในระบบจำนวนฐานคู่นั้นสามารถทำได้โดยการนำตารางของตัวตั้งและตัวบวกที่อยู่ในรูปแบบพร้อมบวกมาซ้อนทับกันโดยผลลัพธ์ที่ได้จะอยู่ในรูปแบบพร้อมบวกด้วยนั้นจำเป็นต้องพิจารณาถึงรูปแบบที่เป็นไปได้ในการซ้อนทับกัน โดยการซ้อนทับกันของตำแหน่งที่มีค่าจะทำให้เกิดสายการทดไปทางขวาเท่านั้น ดังรูปที่ 2.4 โดยตำแหน่งที่มีค่า $2'3'$ ซึ่งเกิดการซ้อนทับกันจะแสดงด้วยเครื่องหมายกากบาท ดังนั้นเมื่อพิจารณาแล้วพบว่าตำแหน่งของหลักที่ติดกันจะไม่มีผลต่อสายการทด ดังนั้นงานวิจัยนี้จึงได้ให้ความหมายของรูปแบบพร้อมบวกใหม่ (weak addition ready DBNS: WAR) คือ รูปแบบที่ไม่มีตำแหน่งที่มีค่าซ้อนทับกัน และไม่มีตำแหน่งที่มีค่าอยู่ติดกันในแถวเดียวกัน โดยลดความเข้มงวดของกฎลงและให้อนุญาตให้รูปแบบพร้อมบวกสามารถเกิดตำแหน่งที่มีค่าติดกันในแนวหลักได้ จะทำให้การออกแบบวงจรของการบวกมีความง่ายขึ้น



รูปที่ 2.4 การเกิดสายการทดของกรณีการซ้อนทับกันของตำแหน่งที่มีค่า

จากการซ้อนทับกันของตารางสามารถทำให้เกิดเหตุการณ์ตำแหน่งที่มีค่าติดกันมากกว่าสองช่องในแถวเดียวกันได้เช่นการเกิดช่องที่มีค่าสามช่องติดกันในแนวแถวเดียวกันดังรูปที่ 2.5 ซึ่งการเกิดเหตุการณ์เช่นนี้จำเป็นต้องพิจารณาเป็นพิเศษในการปรับเปลี่ยนรูปแบบของผลลัพธ์ให้อยู่ในรูปแบบพร้อมบวก เนื่องจากว่าถ้ามีการนำกฎการลดแถวมาใช้แล้วอาจจะเกิดความผิดพลาดได้ เช่น อาจจะเกิดการนำกฎการลดแถวมาใช้ซ้ำซ้อนกันได้ โดยใช้สำหรับช่อง 2^i3^j กับ $2^{i+1}3^j$ และ $2^{i+1}3^j$ กับ $2^{i+2}3^j$ จะพบว่าช่อง $2^{i+1}3^j$ ถูกใช้ถึงสองครั้งทำให้ผลลัพธ์เกิดความผิดพลาดขึ้น ดังนั้นในการออกแบบหรือพัฒนาวงจรสำหรับการบวกจำเป็นต้องทราบค่าของตำแหน่งมากกว่าสองตำแหน่งจึงจะตัดสินใจในการใช้กฎการลดแถว



รูปที่ 2.5 ตารางสองมิติกรณีพิเศษที่มีค่าติดกันสามช่อง

ดังนั้นการลดรูปให้อยู่ในรูปแบบพร้อมบวกนั้นจะใช้เพียงสองกฎดังนี้

กฎที่ 1 กฎการลดช่องที่ซ้อนทับ (collision reduction rule)

$$2^i3^j + 2^i3^j = 2^{i+1}3^j$$

กฎที่ 2 กฎการลดแถว (row reduction rule)

$$2^i3^j + 2^{i+1}3^j = 2^i3^{j+1}$$

ต่อมาในปีเดียวกัน อิบราฮิม (Ibrahim) [7] ได้ทำการเสนอวงจรแอนะล็อกสำหรับการบวกระบบจำนวนฐานคู่โดยใช้ข่ายงานประสาทระดับเทียม (cellular neural networks) โดยมีออกแบบเป็นสองขั้นตอนคือ ขั้นตอนการรวมค่าและขั้นตอนการจัดรูปให้อยู่ในรูปแบบพร้อมบวก

ขั้นตอนการรวมค่าสามารถทำได้โดยใช้การซ้อนทับกันของตารางสองมิติเหมือนดังที่ได้กล่าวไว้แล้วในงานวิจัยก่อนหน้านี้ ซึ่งขั้นตอนนี้จะใช้เวลาในการทำงานเป็นค่าคงที่ค่าหนึ่ง ส่วนขั้นตอนการจัดรูปแบบให้อยู่ในรูปแบบพร้อมบวกนั้นทำได้โดยการใช้กฎการลดแถวและกฎการทดที่เกิดจากการซ้อนทับกันของช่องที่มีค่าเหมือนกับงานวิจัยของพานคาลา [6] แต่งานวิจัยนี้มีการกำหนดลำดับการตัดสินใจในการนำกฎของการลดแถวมาใช้ว่าจะใช้กฎการลดแถวที่จุดใดก่อนหลัง ซึ่งสามารถลดปัญหาในกรณีที่เกิดช่องที่มีค่าติดกันมากกว่าสองช่องในแถวเดียวกันได้ โดยลำดับการตัดสินใจของการใช้กฎการลดแถวมีดังนี้คือ

1. $2^{i+1}3^{j+1} + 2^{i+2}3^{j+1} = 2^{i+1}3^{j+2}$
2. $2^{i}3^{j+1} + 2^{i+1}3^{j+1} = 2^{i}3^{j+2}$
3. $2^{i+1}3^{j} + 2^{i+2}3^{j} = 2^{i+1}3^{j+1}$
4. $2^{i}3^{j} + 2^{i+1}3^{j} = 2^{i}3^{j+1}$

เมื่อพิจารณาจากกฎการตัดสินใจดังกล่าวจะพบว่า ผู้วิจัยให้ความสำคัญกับคู่อันดับที่มีความสำคัญสูงไปยังคู่อันดับที่มีความสำคัญต่ำ หรือกล่าวให้เข้าใจง่ายขึ้นคือการพิจารณาการใช้กฎการลดแถวจะพิจารณาจากขวาไปซ้าย โดยพิจาณาจากด้านล่างขึ้นบน จากการใช้กฎการตัดสินใจดังกล่าวทำให้คู่อันดับที่มีความสำคัญต่ำกว่าจะถูกพิจารณาการใช้กฎการลดแถวทีหลังเสมอ ทำให้การนำผลลัพธ์บางส่วนไปใช้ก่อนที่จะทำการบวกเสร็จสิ้นทำได้ยาก

ดังนั้นในปีค.ศ. 2006 ได้มีงานวิจัยของเกรียงยุทธและอรรถสิทธิ์ [8] ได้เสนออัลกอริทึมการบวกสำหรับระบบจำนวนฐานคู่ ซึ่งมีฐานเป็นสอง และจำนวนเต็มใดๆ อีกหนึ่งจำนวน ซึ่งให้ผลลัพธ์อยู่ในฐานเดียวกันในรูปแบบพร้อมบวก โดยมีความซับซ้อนเชิงเวลาสำหรับอัลกอริทึมการบวกเป็น O (*m*) โดย *m* คือขนาดของรูปแบบแสดงค่า

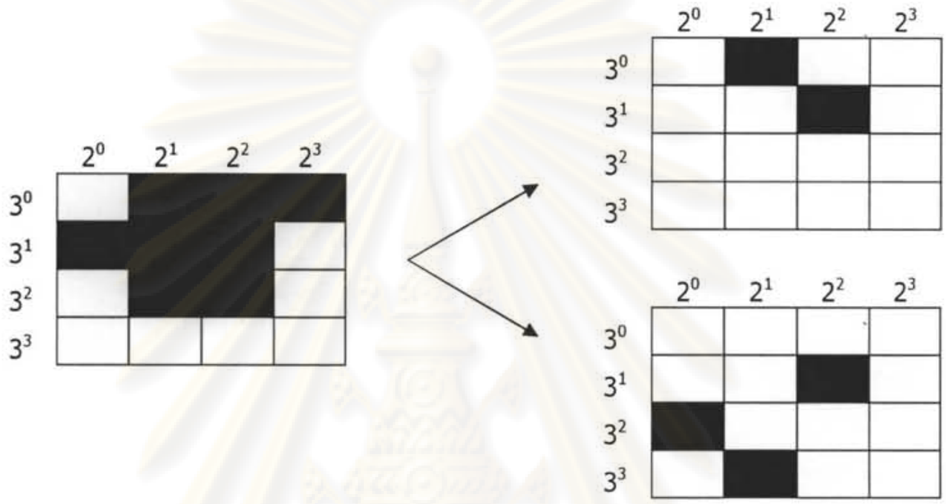
อัลกอริทึมการบวกมีการนำอัลกอริทึมแยกตารางมาช่วยในขั้นตอนของการบวก โดยอัลกอริทึมการแยกตารางใช้ในการกำจัดตำแหน่งที่มีค่าติดกันออกไปเป็นสองตารางที่อยู่ในรูปแบบพร้อมบวกอีกครั้ง หลังจากนั้นนำสองตารางมาทำการบวกโดยการซ้อนทับกันอีก ทำให้ผลลัพธ์สุดท้ายอยู่ในรูปแบบพร้อมบวกหนึ่งตารางโดยใช้เวลาในการทำงานเท่ากับจำนวนแถวของรูปแบบแสดงค่า

อัลกอริทึมการแยกตารางมีการกำหนดลำดับการใช้กฎการลดแถวที่แน่นอนเพื่อลดความกำกวม โดยให้ความสำคัญของหลักคู่ก่อนหลักคี่ดังนี้

1. $2^{2i}3^{j} + 2^{2i+1}3^{j} = 2^{2i}3^{j+1}$
2. $2^{2i+1}3^{j} + 2^{2i+2}3^{j} = 2^{2i+1}3^{j}$

จากอัลกอริทึมดังกล่าวสามารถแสดงผลของการทำงานได้ดังรูปที่ 2.6

เมื่อนำผลลัพธ์จากอัลกอริทึมแยกตารางมารวมกันด้วยการซ้อนทับกันจะได้ว่า แถวที่มีค่าประจำแถวน้อยที่สุดจะอยู่ในรูปแบบพร้อมบวกเสมอ ทำให้สามารถนำแถวที่มีค่าประจำแถวน้อยที่สุดไปเป็นผลลัพธ์บางส่วนของการบวกได้ทันที จึงสามารถนำผลลัพธ์ดังกล่าวไปใช้งานได้โดยที่ไม่จำเป็นต้องรอให้กระบวนการบวกทำงานจนเสร็จสิ้น



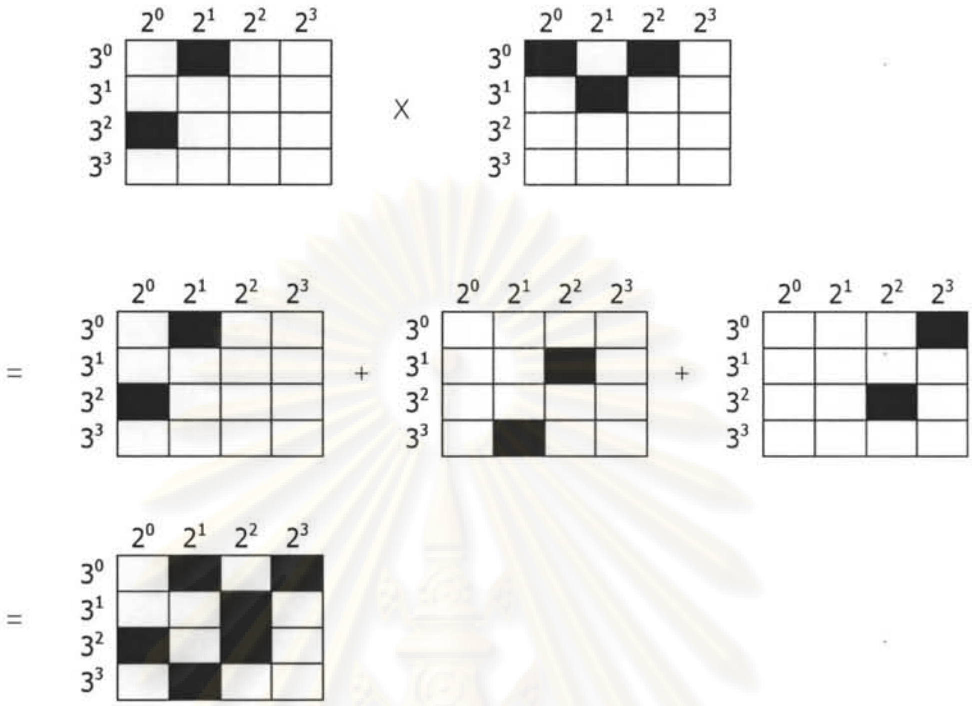
รูปที่ 2.6 การทำงานของอัลกอริทึมแยกตาราง

2.5 การคูณของระบบจำนวนฐานคู่ (multiplication for double-base number system)

งานวิจัยของดิมิทรอฟ (V. Dimitrov) [3] ได้กล่าวถึงแนวคิดพื้นฐานสำหรับกระบวนการคูณของระบบฐานคู่ไว้โดยอาศัยกระบวนการบวกเป็นพื้นฐานในการดำเนินการ การคูณคือการกระจายตัวตั้งไปยังทุกตำแหน่งของตัวคูณ แล้วนำผลจากการกระจายตัวตั้งทุกตัวรวมเข้าด้วยกัน ซึ่งสามารถเขียนในรูปแบบสมการได้ดังนี้

$$A \times B = A \times \sum_{i,j} b_{i,j} 2^i 3^j = \sum_{i,j} A b_{i,j} 2^i 3^j$$

จากสมการข้างต้น การคูณเป็นการรวมกันของตัวตั้งกับ $b_{i,j}$ ใดๆ โดยถ้ามองเป็นรูปตารางจะเห็นว่าตัวตั้งจะถูกเลื่อนตำแหน่ง เพิ่มขึ้น i ตำแหน่งในแนวนอนและ j ตำแหน่งในแนวตั้ง แล้วจึงนำผลของการเลื่อนตำแหน่งทั้งหมดมารวมกัน โดยสามารถแสดงเป็นภาพได้ดังรูป 2.7



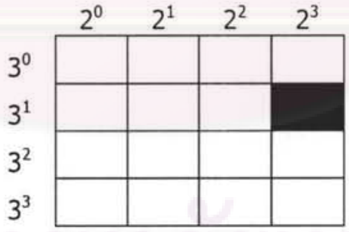
รูปที่ 2.7 การคูณของ 11 และ 11 ได้ผลลัพธ์เป็น 121

บทที่ 3

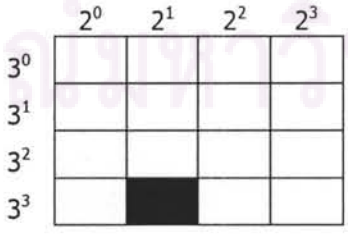
อัลกอริทึมการลบสำหรับจำนวนฐานคู่

จากงานวิจัยที่เกี่ยวข้องในหัวข้อที่ผ่านมาแสดงได้ให้เห็นว่าในระบบจำนวนฐานคู่นั้นมีการเสนอตัวดำเนินการทางเลขคณิตเพียงกระบวนการบวก และกระบวนการคูณเท่านั้น เนื่องด้วยการดำเนินการลบมีความยุ่งยากมากกว่าการดำเนินการบวก ความยุ่งยากของการดำเนินการลบคือการดำเนินการลบไม่สามารถใช้หลักการการซ้อนทับของตารางตัวตั้งและตารางตัวลบในการหักล้างของตำแหน่งที่มีค่าเพียงอย่างเดียวได้ เนื่องจากมีโอกาสที่มีตำแหน่งที่มีค่าคงเหลืออยู่ในรูปแบบแทนจำนวนของตัวลบ ทำให้ต้องมีการปรับเปลี่ยนรูปแบบแทนจำนวนของตัวตั้งหรือตัวลบโดยประยุกต์ใช้กฎการลดแถวหรือกฎการลดหลัก โดยการเปลี่ยนรูปแบบแทนจำนวนจำเป็นต้องทำการเปลี่ยนรูปเพื่อให้เกิดการหักล้างกันของตำแหน่งที่มีค่าจากการซ้อนทับกันของตารางตัวตั้งและตัวลบด้วย ซึ่งการดำเนินการลบด้วยการเปลี่ยนรูปแบบแทนจำนวนดังกล่าวนั้นไม่มีวิธีการปรับเปลี่ยนที่คงที่ทำให้ไม่สามารถคาดการณ์ความซับซ้อนเชิงเวลาของอัลกอริทึมได้และอาจมีหลายวิธีในการปรับเปลี่ยนรูปแบบแทนค่าในการดำเนินการลบ พิจารณาจากตัวอย่างการดำเนินการลบด้วยวิธีการเปลี่ยนรูปแบบแทนจำนวนดังนี้

ตัวอย่างที่ 1 ค่าเชิงตัวเลขที่มีค่าน้อยแต่อาจจะมีรูปแบบแทนจำนวนที่มีตำแหน่งที่มีค่าอยู่ทางฝั่งขวามือของตาราง แต่ค่าเชิงตัวเลขมากกว่าแต่อาจจะมีรูปแบบแทนจำนวนที่มีตำแหน่งที่มีค่าอยู่ในทางฝั่งซ้ายมือของตารางดังรูปที่ 3.1 และรูปที่ 3.2 ตามลำดับ



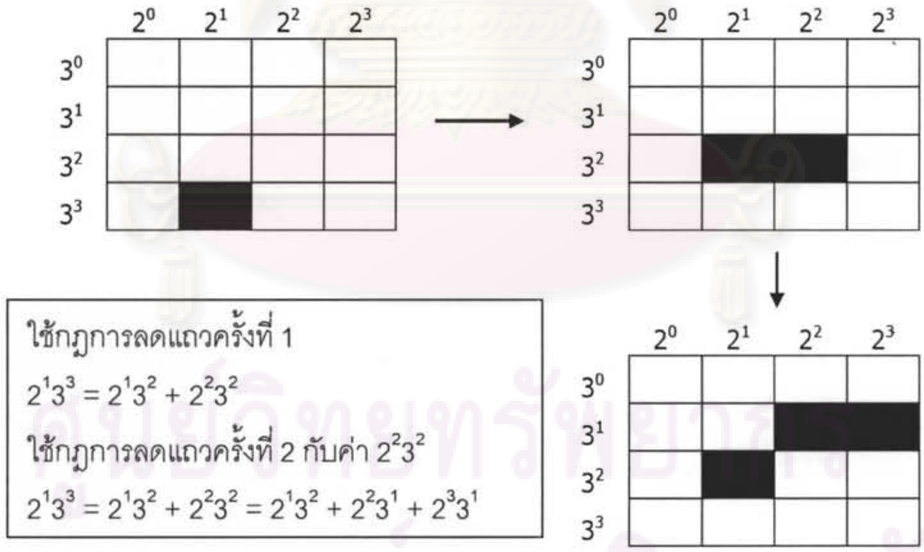
รูปที่ 3.1 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 24



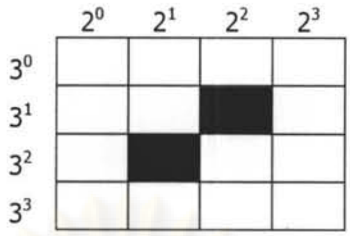
รูปที่ 3.2 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 54

เมื่อพิจารณาตัวอย่างที่ 1 ถ้าจะทำการลบกันระหว่างรูปแบบแทนจำนวนของค่าเชิงตัวเลข 54 กับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 24 จะพบว่าการดำเนินการลบโดยใช้วิธีการซ้อนทับกันของตารางสองตารางเพียงอย่างเดียวไม่สามารถหาผลลัพธ์จากการลบได้ เพราะว่ายังมีตำแหน่งที่มีค่าคงเหลืออยู่ในตารางของรูปแบบแทนจำนวนของตัวลบ ดังนั้นจึงจำเป็นต้องมีการปรับเปลี่ยนรูปแบบแทนจำนวนเพื่อให้สามารถหาผลลัพธ์จากการลบได้ โดยการดำเนินการลบมีได้หลายวิธีขึ้นอยู่กับวิธีการปรับเปลี่ยนรูปแบบแทนจำนวน เช่น

วิธีที่ 1 จะนำกฎการลดแถวมาใช้กับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 54 สองครั้งเพื่อปรับเปลี่ยนรูปแบบแทนจำนวนให้สามารถทำการลบด้วยการวิธีซ้อนทับของตารางได้ ดังที่แสดงตามรูปที่ 3.3 เมื่อปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 54 ซึ่งเป็นตัวตั้งแล้ว จากนั้นนำตารางของรูปแบบแทนจำนวนของค่าเชิงตัวเลข 24 มาซ้อนทับกัน โดยตำแหน่งที่ $2^3 3^1$ เป็นตำแหน่งที่เกิดการซ้อนทับกันของตำแหน่งที่มีค่าของทั้งสองตาราง (collision of active cell) ซึ่งจะได้ผลลัพธ์เป็นไม่มีค่าในตำแหน่งนั้นในตารางของตัวตั้งและตารางของตัวลบ ดังนั้นตารางของตัวลบในทุกตำแหน่งจะไม่มีค่าเหลืออยู่เลยหรือมีค่าเชิงตัวเลขเป็นศูนย์ ดังนั้นผลลัพธ์ของการลบก็คือตารางของตัวตั้งหลังจากที่ถูกหักตำแหน่งที่มีค่าที่เกิดการซ้อนทับออกไปแล้ว จะมีค่าเชิงตัวเลขเป็น 30 สามารถแสดงอยู่ในรูปแบบแทนจำนวนได้ดังรูปที่ 3.4

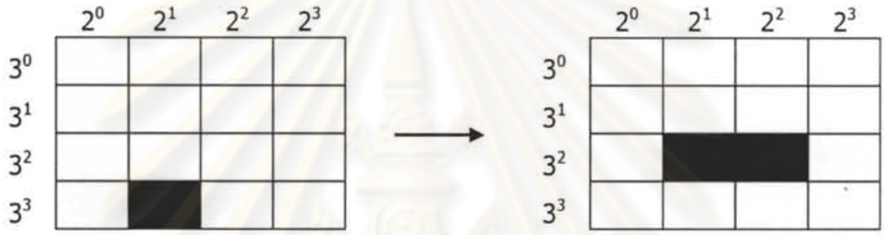


รูปที่ 3.3 การปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 54 เพื่อใช้สำหรับการลบ

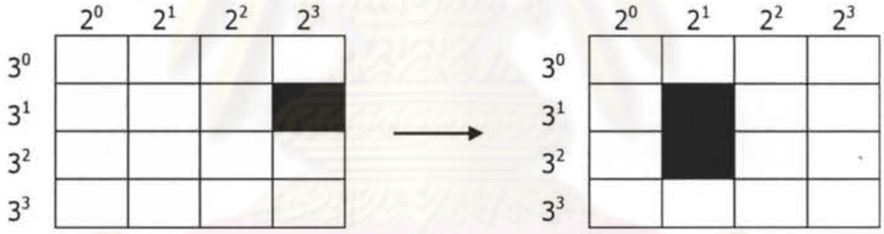


รูปที่ 3.4 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 30 ที่เป็นผลลบด้วยวิธีที่ 1

วิธีที่ 2 ใช้กฎการลดแถวกับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 54 และใช้กฎการลดหลักกับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 24 ดังรูปที่ 3.5 และ 3.6 ตามลำดับ



รูปที่ 3.5 การใช้กฎการลดแถวกับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 54 เพื่อใช้สำหรับการลบ



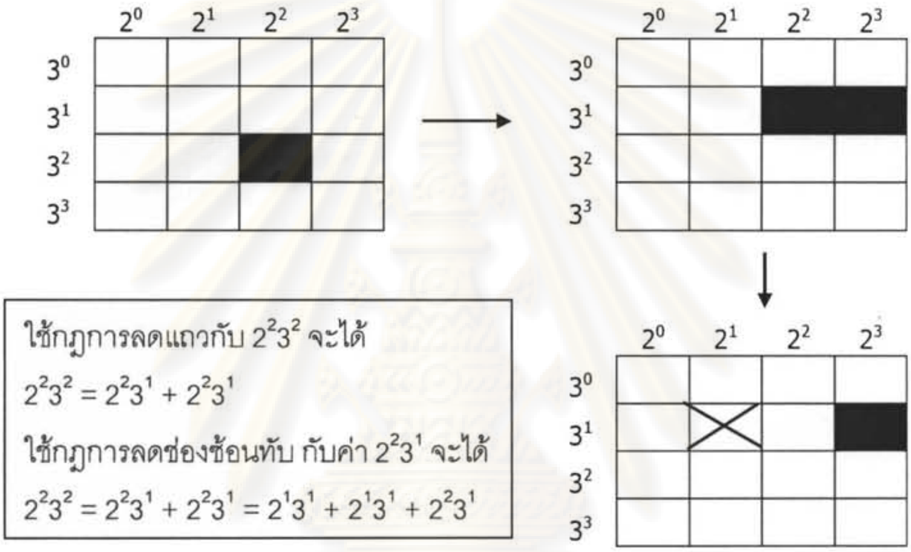
รูปที่ 3.6 การใช้กฎการลดหลักกับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 24 เพื่อใช้สำหรับการลบ

หลังจากนั้นทำการลบโดยการซ้อนทับกันของสองตาราง จะได้ผลลัพธ์เป็นรูปแบบแทนจำนวนของค่าเชิงตัวเลข 36 ลบด้วยรูปแบบแทนจำนวนของค่าเชิงตัวเลข 6 ดังรูปที่ 3.7

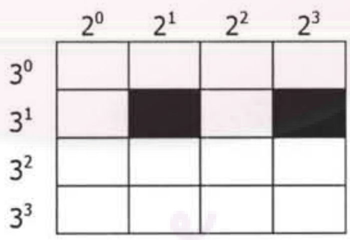
รูปแบบแทนจำนวนของค่าเชิงตัวเลข 36 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 6

รูปที่ 3.7 การผลลัพธ์จากการลบด้วยการซ้อนทับกันของสองตาราง

แต่เนื่องจากว่ายังมีตำแหน่งที่มีค่าเหลืออยู่ในตารางของรูปแบบแทนจำนวนของตัวลบ ดังนั้นจึงต้องทำการลบต่อไป โดยนำกฎการลดแถวมาใช้กับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 36 และต่อด้วยกฎการลดช่องที่ซ้อนทับกันในตำแหน่งของ $2^2 3^1$ จะได้ดังรูปที่ 3.8 หลังจากนั้นนำตารางรูปแบบแทนจำนวนของค่าเชิงตัวเลข 36 ที่ได้ทำการปรับเปลี่ยนรูปแบบแล้วมาซ้อนทับกับตารางรูปแบบแทนจำนวนของค่าเชิงตัวเลข 6 จะทำให้ไม่มีตำแหน่งที่มีค่าเหลืออยู่ในตารางรูปแบบแทนจำนวนของตัวลบ ดังนั้นผลลัพธ์สุดท้ายของการลบคือรูปแบบแทนจำนวนของค่าเชิงตัวเลข 30 ดังรูปที่ 3.9



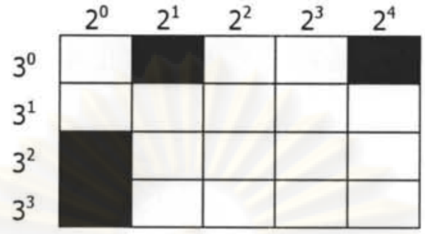
รูปที่ 3.8 การปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 36 เพื่อใช้สำหรับการลบ



รูปที่ 3.9 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 30 ที่เป็นผลจากการลบด้วยวิธีที่ 2 □

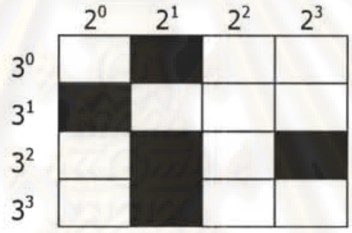
จากตัวอย่างที่ 1 ได้การแสดงวิธีในการดำเนินการลบโดยให้ผลลัพธ์ที่ถูกต้อง โดยที่ผลลัพธ์ของทั้งสองวิธีก็มีรูปแบบแทนจำนวนที่แตกต่างกัน เนื่องจากแต่ละวิธีมีความต่างกันของการนำกฎต่างๆ มาใช้ในการปรับเปลี่ยนรูปแบบแทนจำนวนเพื่อใช้ในการดำเนินการลบ สังเกตได้ว่าวิธีที่ 1 มีขั้นตอนในการดำเนินการลบที่น้อยกว่าวิธีที่ 2 แต่การที่จะเปลี่ยนรูปแบบแทนจำนวนของตัวตั้งเพียงตัวเดียวอย่างที่วิธีที่ 1ทำก็ใช้ไม่ได้กับทุกกรณีเสมอไป ถ้าตำแหน่งที่มีค่าของตัวตั้งมีจำนวนมากดังรูปที่ 3.10 ก็จะทำให้การเปลี่ยนรูปแบบแทนจำนวนของตัวตั้งมีความยุ่งยากมากขึ้น

เนื่องจากว่าไม่สามารถรู้ได้ว่าควรจะใช้กฎใดกับตำแหน่งใดก่อนในการเปลี่ยนรูปเพื่อใช้ในการดำเนินการลบให้ประสบผลสำเร็จได้

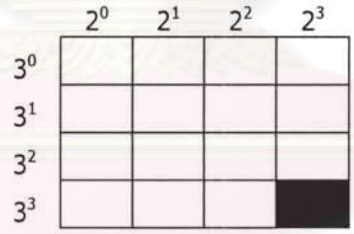


รูปที่ 3.10 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 54

ตัวอย่างที่ 2 รูปแบบการแทนจำนวนของค่าเชิงตัวเลขที่น้อยกว่าแต่มีจำนวนของตำแหน่งที่มีค่าอยู่มากกว่าจำนวนของตำแหน่งที่มีค่าในรูปแบบแทนจำนวนของค่าเชิงตัวเลขที่มากกว่าดังรูปที่ 3.11 และ รูปที่ 3.12 ตามลำดับ

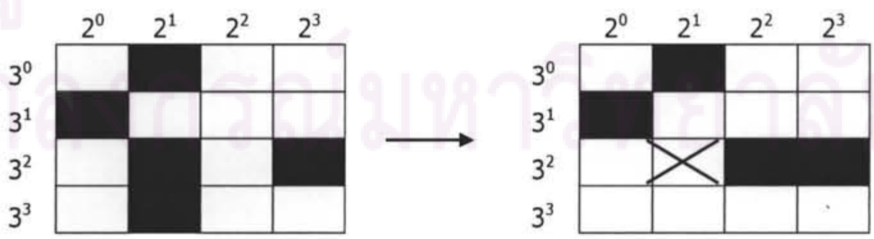


รูปที่ 3.11 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 149



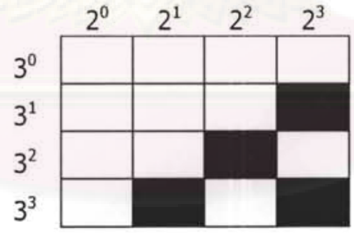
รูปที่ 3.12 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 216

จากตัวอย่างนี้ถ้าทำการปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 149 โดยการใช้กฎการลดแถวกับตำแหน่งที่ $2^1 3^3$ ก็จะทำให้เกิดปัญหาเรื่องสายการทดได้ ดังรูปที่ 3.13

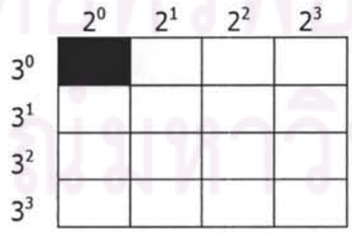


รูปที่ 3.13 การใช้กฎการลดแถวกับที่ตำแหน่ง $2^1 3^3$ ทำให้เกิดสายการทดได้

ดังนั้นถ้าจะทำการลบกันระหว่างรูปแบบแทนจำนวนของค่าเชิงตัวเลข 216 กับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 149 จะต้องทำการปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 216 แต่การที่จะปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 216 ให้สามารถทำการลบกับรูปแบบแทนจำนวนของค่าเชิงตัวเลข 149 ให้สำเร็จได้นั้นมีความยุ่งยากเนื่องจากต้องใช้กฎต่างๆ ในการเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 216 ให้มีตำแหน่งที่มีค่าตรงกับตำแหน่งที่มีค่าในรูปแบบแทนจำนวนของค่าเชิงตัวเลข 149 เพื่อทำการซ้อนทับกันได้ ซึ่งการที่จะเปลี่ยนรูปแบบแทนจำนวนดังกล่าวก็ไม่ใช่เรื่องง่าย ทั้งนี้เพราะไม่สามารถคาดเดาได้ว่าควรจะใช้กฎใดก่อนหรือใช้กฎที่ตำแหน่งใดก่อนถึงจะสามารถหารูปแบบแทนจำนวนที่สามารถทำให้เกิดการซ้อนทับกันของตำแหน่งที่มีค่าได้ และต้องทำการขจัดตำแหน่งที่มีค่าของรูปแบบแทนจำนวนของตัวลบให้หมดทุกตำแหน่งด้วย ในทางกลับกันถ้าจะทำการลบกันระหว่างรูปแบบการแทนจำนวนของค่าเชิงตัวเลขที่มากกว่าและก็มีจำนวนของตำแหน่งที่มีค่าอยู่มากกว่าจำนวนของตำแหน่งที่มีค่าในรูปแบบการแทนจำนวนของค่าเชิงตัวเลขที่น้อยกว่าดังรูปที่ 3.14 และรูปที่ 3.15 ตามลำดับ การปรับเปลี่ยนรูปแบบแทนจำนวนของค่าเชิงตัวเลข 1 ซึ่งเป็นตัวลบ ไม่สามารถทำได้เนื่องจากว่าตำแหน่งที่มีค่าของรูปแบบแทนจำนวนของค่าเชิงตัวเลข 1 ไม่สามารถปรับเปลี่ยนไปให้ตรงกับตำแหน่งที่มีค่าในรูปแบบแทนจำนวนของค่าเชิงตัวเลข 330 ที่เป็นตัวตั้งได้ ดังนั้นจำเป็นที่ต้องมีการปรับเปลี่ยนรูปแบบแทนจำนวนของตัวตั้งให้มีตำแหน่งที่มีค่าตรงกันกับตำแหน่งที่มีค่าของรูปแบบแทนจำนวนตัวลบเพื่อทำการหักออกจากกันได้



รูปที่ 3.14 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 330



รูปที่ 3.15 รูปแบบแทนจำนวนของค่าเชิงตัวเลข 1

จากตัวอย่างที่ผ่านมาจะพบว่าปัญหาที่เกิดขึ้นของการลบคือ

1. การเลือกกฎที่จะนำมาใช้ในการเปลี่ยนรูปแบบแทนจำนวนให้สามารถเกิดการซ้อนทับกันระหว่างตำแหน่งที่มีค่าของตัวตั้งและตัวลบเพื่อทำให้ตำแหน่งที่มีค่าของตัวลบหมดไป
2. การเลือกว่าควรจะทำการเปลี่ยนรูปแบบแทนจำนวนของตัวตั้งหรือตัวลบหรือต้องทำการปรับเปลี่ยนรูปแบบแทนจำนวนของทั้งสอง
3. การเลือกตำแหน่งในรูปแบบแทนจำนวนที่จะต้องถูกเปลี่ยนรูปไปตามกฎต่างๆ ที่เลือกมาใช้
4. การเปรียบเทียบค่าเชิงตัวเลขของรูปแบบแทนจำนวนในระบบฐานคู่

เพื่อขจัดปัญหาทั้งหมดดังที่กล่าวมาแล้วข้างต้น ผู้วิจัยจึงมีแนวคิดในการพัฒนาการดำเนินการลบที่มีลำดับที่แน่นอนด้วยการประยุกต์แนวคิดในการลบของระบบเลขฐานสองด้วยการลบในหลักที่มีค่าเชิงตัวเลขต่ำกว่าไปยังหลักที่มีค่าเชิงตัวเลขสูงกว่า ซึ่งการลบในระบบจำนวนฐานคู่จะทำการลบในระบบแนวตั้งหรือแนวหลักคือ $\sum_j 2^i 3^j$ เมื่อ i เป็นค่าคงที่ โดยทำการลบทีละหลักจากซ้ายไปขวา จนกว่าตำแหน่งที่มีค่าของรูปแบบแทนจำนวนของตัวลบทุกตำแหน่งจะถูกพิจารณา ซึ่งการลบในแนวหลักอาจจะเกิดสายการทดของหลักของตัวตั้งและหลักตัวลบไปยังหลักถัดไปได้ โดยที่การลบในแนวหลักนั้นจำเป็นต้องมีการเปรียบเทียบค่าเชิงตัวเลขของแนวหลักของตัวตั้งและแนวหลักของตัวลบว่าค่าเชิงตัวเลขของหลักใดมีค่ามากกว่ากันก่อนดำเนินการลบในแนวหลัก เพื่อความครอบคลุมกรณีทุกกรณีที่เป็นไปได้ในการลบในแนวหลัก การเปรียบเทียบค่าเชิงตัวเลขของแนวหลัก 2 หลักใดๆ มีรายละเอียดดังนี้

3.1 การเปรียบเทียบค่าในแต่ละหลัก

บทตั้งที่ 3.1: *กำหนดให้ a_k และ b_k เป็นรูปแบบแทนจำนวนในแนวหลักที่ k ในระบบจำนวนฐานคู่ เมื่อ $a_k = \sum_j d_j 2^k 3^j$ และ $b_k = \sum_j e_j 2^k 3^j$ โดยที่ $d_i \neq e_i$ จะได้ว่า*

$$a_k < b_k \text{ ถ้า } m < n$$

เมื่อ m คือ $\max\{ j \mid d_j = 1\}$ และ $n = \max\{ j \mid e_j = 1\}$ และในทางตรงกันข้าม

พิสูจน์ จาก $\sum_{j=0}^{n} 1 \times 2^k 3^j = 2^k (3^n-1)/(3-1) = 2^k (3^n-1)/2$ ทำให้สามารถสรุปได้ว่า $\sum_{j=0}^{n} 1 \times 2^k 3^j < 2^k 3^{n+1}$ ■

จากบทตั้งที่ 3.1 เป็นการแสดงการเปรียบเทียบค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักที่ k พิจารณาได้จากตำแหน่งที่มีค่าของทั้งสองหลักว่าตำแหน่งที่มีค่าที่มากที่สุดในแนวหลักมีค่าของ j มากกว่ากัน

3.2 การลบในแนวหลัก

จากแนวคิดในการลบจะดำเนินการโดยกระทำการลบในแนวหลัก ซึ่งจะอาศัยการเปรียบเทียบค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักที่ได้แสดงไว้ในบทตั้งที่ 3.1 โดยการลบในแนวหลักจะแยกพิจารณาการลบเป็น 4 กรณี ดังนี้

1. ค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวตั้งมากกว่า 0 และ ค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวลบเป็น 0
2. ค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวตั้งเป็น 0 และ ค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวลบมากกว่า 0
3. ค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวตั้งมากกว่าค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวลบ
4. ค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวตั้งน้อยกว่าค่าเชิงตัวเลขของรูปแบบแทนจำนวนในแนวหลักของตัวลบ

ทฤษฎีบทที่ 3.1: *กำหนดให้ a_k และ b_k เป็นรูปแบบแทนจำนวนในแนวหลักที่ k ในระบบจำนวนฐานคู่ เมื่อ $a_k = \sum_j d_j 2^k 3^j$ และ $b_k = \sum_j e_j 2^k 3^j$ โดยที่ $d_j \neq e_j$ จะได้ว่า*

$$a_k - b_k = r_k + ca_{k+1} + ca_{k+2} - cb_{k+1}$$

เมื่อ ca_{k+1} และ ca_{k+2} คือสายการทดของตัวตั้ง และ cb_{k+1} คือสายการทดของตัวลบ

อัลกอริทึม 3.1 อัลกอริทึมการลบในแนวหลัก (column subtraction algorithm)

Input: a_i is a i^{th}-column representation in DBNS such that

$$a_i = \sum_j d_j 2^i 3^j \;\; ; d \in \{0,1\}$$

b_i is a i^{th}-column representation in DBNS such that

$$b_i = \sum_j e_j 2^i 3^j \;\; ; e \in \{0,1\}$$

when $d_j \neq e_j$

Output: r_i is a i^{th}-column representation in DBNS such that $r_i = g 2^i 3^0$; $g \in \{0,1\}$

a'_i is a representation in DBNS such that $a'_i = \sum_{l,j} d'_{l,j} 2^l 3^j$; $d' \in \{0,1\}, l > i$

b'_i is a representation in DBNS such that $b'_i = \sum_{l,j} e'_{l,j} 2^l 3^j$; $e' \in \{0,1\}, l > i$

Consider two inputs a_i and b_i; they can be classified as 4 cases.

Case 1: $a_i > 0$, $b_i = 0$

In this case we will transform a_i to be $r_i + a'_i$.

Step 1: for ($j=n$ to $j>0$ step -1) do

If $d_j = 1$ and $d_{j-1} = 0$ then $d_j = 0$, $d_{j-1} = 1$, $d'_{j+1,j-1} = 1$

If $d_j = 1$ and $d_{j-1} = 1$ then $d_j = 0$, $d_{j-1} = 0$, $d'_{j+2,j-1} = 1$

Step 2: if $d_{i,0} = 1$ then $g = 1$ else $g = 0$

Case 2: $a_i = 0$, $b_i > 0$

In this case we will transform b_i to be $r_i + a'_i - b'_i$.

Step 1: For any j^{th} row,

If $e_j = 1$ then $d'_{i,j} = 1$ and $e'_{i+1,j} = 1$

Step 2: Apply *Case 1* for each a'_i

Step 3: $r_i \leftarrow r_i$ from result in step 2 and $a'_i \leftarrow a'_i$ from result in step 2

Case 3: $a_i > b_i$

In this case we will subtract a_i by b_i. One-complement technique will be applied in this case. Given $\overline{a}_n$ is the complement of $a_i + t_{i+1}$ where $t_{i+1} = 0$ then,

$$\overline{a}_n = \sum_{n,j} \overline{d}_{n,j} 2^n 3^j \quad ; \overline{d} \in \{0,1\},\ n \in \{i, i+1\}.$$

Define: p is max$\{ j \mid d_j = 1\}$

Step 1: Finding p value

for ($j = 0$ to $j < n$ step -1)

if $d_j = 1$ then $p = j$

Step 2: for ($j=0$ to $j \leq$ p step +1) do

If $d_j = 1$ then $\overline{d}_{i+1,j} = 1$

Else $\overline{d}_{i,j+1} = 1$

Step 3: for ($j=0$ to $j \leq$ p step +1) do

If $\overline{d}_{i,j} = 1$ and $e_j = 1$ then $\overline{d}_{i,j} = 0$, $\overline{d}_{i+1,j} = 1$

If $\overline{d}_{i,j} = 0$ and $e_j = 1$ then $\overline{d}_{i,j} = 1$

Step 4: for ($j=0$ to $j \leq$ p step +1) do

If $\overline{d}_{i,j} = 1$ then $d'_{i,j} = 0$ else $d'_{i,j} = 1$

If $\overline{d}_{i+1,j} = 1$ then $d'_{i+1,j} = 0$ else $d'_{i+1,j} = 1$

Step 5: $a'_i, b'_i \leftarrow \text{split}(a'_i)$

Step 6: Apply addition algorithm for (a'_i, b'_i) and keep result to a'_i

Step 7: Apply *Case 1* for each a'_i

Step 8: $a'_i \leftarrow a'_{i+1} + a'_i$ from result in step 7 and $r_i \leftarrow r_i$ from result in step 7 and $b'_i \leftarrow 0$

Case 4: $a_i < b_i$

This case is similar to *Case 3* but the addend and subtracter need to be switched.

Step 1: Apply *Case 3* by sending b_i and a_i as the input a_i and b_i for *Case 3* but it does not need to apply step 7-8 in *Case 3*.

Step 2: $b'_i \leftarrow a'_i$ from result of Step 1

Step 3: Apply *Case 2* for b'_i

Step 4: $b'_i \leftarrow b'_{i+1} + b'_i$ from result in step 3 and $r_i \leftarrow r_i$ from result in step 3 and $a'_i \leftarrow a'_i$ from result in step 3.

พิสูจน์ จะต้องพิสูจน์ว่า ค่าเชิงตัวเลขของ $a_i - b_i$ เท่ากับค่าเชิงตัวเลขของ $r_i + a'_i - b'_i$ และผลลัพธ์จากอัลกอริทึมนี้ไม่มีเลข 1 ติดกันในแถวเดียวกัน

1) ทำการพิจารณาแต่ละกรณีดังนี้

กรณีที่ 1 จะพบว่าค่าเชิงตัวเลขของ a_i ถูกเปลี่ยนรูปโดยประยุกต์การใช้กฎการลดแถวและกฎการลดหลัก โดยเปลี่ยนรูปเป็น $r_i + a'_i$ โดยค่าเชิงตัวเลขไม่มีการเปลี่ยนแปลง ดังนั้น ค่าเชิงตัวเลขของ $a_i - b_i$ เท่ากับค่าเชิงตัวเลขของ $r_i + a'_i - b'_i$ *และ* a'_i จะมีตำแหน่งที่มีค่าในหลักที่ k+1 หรือ k+2 เท่านั้น โดยที่ a'_i มีรูปแบบพร้อมบวก

กรณีที่ 2 เป็นการเปลี่ยนรูปของ b_i เป็น $a'_i - b'_i$ และทำการเปลี่ยนรูป a'_i เป็น $r_i + a'_i$ อีกครั้งทำให้ค่าเชิงตัวเลขไม่มีการเปลี่ยนแปลงในระหว่างขั้นตอน ดังนั้น ค่าเชิงตัวเลขของ $a_i - b_i$ เท่ากับค่าเชิงตัวเลขของ $r_i + a'_i - b'_i$ *และ* a'_i จะมีตำแหน่งที่มีค่าในหลักที่ k+1 หรือ k+2 ส่วน b'_i จะมีตำแหน่งที่มีค่าในหลักที่ k+1 เท่านั้น

กรณีที่ 3 ใช้ one-complement เทคนิคและการเปลี่ยนรูปเหมือนกับกรณีที่ 1 ซึ่ง one-complement เทคนิคนั้นไม่ทำให้ค่าเชิงตัวเลขของการลบเปลี่ยนแปลง สมมติให้ $U = A + \bar{A}$ ดังนั้น $A - B = U - \bar{A} - B = U - (\bar{A} + B) = \overline{(\bar{A} + B)}$ เมื่อ $0 \le B \le A \le$

U ดังนั้นค่าเชิงตัวเลขของ $a_i - b_i$ เท่ากับค่าเชิงตัวเลขของ $r_i + a'_i - b'_i$ และจากอัลกอริทึมจะได้ผลลัพธ์เป็น a'_i ที่เป็นสายการทดสำหรับตัวตั้งที่มีตำแหน่งที่มีค่าในหลักที่ k+1 หรือ k+2 เท่านั้น โดยจะเห็นว่าขั้นตอนที่ 1 จะทำการหาค่าตำแหน่งที่มีค่ามากที่สุดในหลักของตัวตั้ง ขั้นที่ 2 เป็นการทำสลับบิตมีค่ากับไม่มีค่าโดยที่มีการปรับรูปแบบให้พร้อมสำหรับการบวกกับหลักของตัวลบ โดยที่ในแต่ละแถวจะมีโอกาสที่เป็นไปได้ทั้งหมด 4 แบบคือ แบบที่1 $\overline{d}_{i,j} = 0$ และ $\overline{d}_{i+1,j} = 0$ เกิดจาก $d_0 = 0$ หรือ $d_{j-1} = 1$ และ $d_j = 0$ ซึ่งเมื่อนำ e_j เข้ามาบวกจะไม่เกิดสายการทด แบบที่2 $\overline{d}_{i,j} = 0$ และ $\overline{d}_{i+1,j} = 1$ เกิดจาก $d_j = 1$ ซึ่งเมื่อนำ e_j เข้ามาบวกจะไม่เกิดสายการทด แบบที่3 $\overline{d}_{i,j} = 1$ และ $\overline{d}_{i+1,j} = 0$ เกิดจาก $d_{j-1} = 0$ และ $d_j = 0$ ซึ่งเมื่อนำ $e_j = 1$ เข้ามาบวกจะทำให้เกิดการทดไปยังตำแหน่งถัดไปแต่ไม่เกิน ตำแหน่งที่ k+1 แบบที่4 $\overline{d}_{i,j} = 1$ และ $\overline{d}_{i+1,j} = 1$ เกิดจาก $d_{j-1} = 0$ และ $d_j = 1$ แต่เนื่องจากว่า $d_j \neq e_j$ ดังนั้นจึงไม่เกิดสายการทด ดังนั้นจากขั้นตอนที่ 1 ถึง 3 จะไม่ทำให้เกิดตำแหน่งที่มีค่าเกินขนาดของ $2 \times p$ คือมี 2 หลักและ p แถว เมื่อทำการสลับบิตอีกครั้งในขั้นตอนที่ 4 จะให้ผลที่มีตำแหน่งที่มีค่าอยู่ในหลักที่ k และ k+1 เท่านั้น และในขั้นตอนที่ 5 และ 6 เป็นการประยุกต์อัลกอริทึมการบวกเพื่อทำให้ผลลัพธ์จากการทำงานในขั้นที่ 4 อยู่ในรูปพร้อมบวกหลังจากนั้นหลักที่ k ของผลลัพธ์จะถูกนำไปหาผลลัพธ์โดยทำกรณีที่ 1 ซึ่งจะได้ผลลัพธ์การลบและสายการทดที่เกิดในหลักที่ k+1 และ k+2 เมื่อนำ a'_i ที่เกิดจาก ขั้นตอนที่ 6 และ ขั้นตอนที่ 7 ซึ่งเป็นรูปแบบพร้อมบวกทั้งคู่มาบวกกันจะไม่มีทางที่ขนาดจะเกิน k+2 ดังนั้น a'_i จะมีตำแหน่งที่มีค่าในหลักที่ k+1 หรือ k+2 เท่านั้น

กรณีที่ 4 นั้นใช้หลักการเดียวกับกรณีที่ 3 แต่ จะเกิดสายการทดของตัวลบโดยที่ b'_i จะมีตำแหน่งที่มีค่าในหลักที่ k+1 เท่านั้น เนื่องจากในขั้นตอนที่ 2 เป็นการทำงานแบบเดียวกันในขั้นตอนที่ 1 ถึง 6 ของกรณีที่ 3 ซึ่งจะได้ผลลัพธ์ที่มีตำแหน่งที่มีค่าไม่เกินหลักที่ k+1 และผลลัพธ์อยู่ในรูปพร้อมบวก หลังจากนั้นนำผลลัพธ์ที่อยู่ในรูปพร้อมบวกในหลักที่ k ไปทำงานด้วยกรณีที่ 2 ซึ่งจะได้ผลลัพธ์การลบและตัวทดของตัวลบที่ถูกเลื่อนบิตจากหลักที่ k ไปเป็นหลักที่ k+1 เมื่อนำ b'_i จากขั้นตอนแรก มารวมกับ b'_i ที่ได้จากขั้นตอนที่ 4 จะได้ผลลัพธ์ในหลักที่ k+1 เท่านั้น

2) ในแต่ละกรณีจะให้ผลลัพธ์ 3 ค่าคือ r_i, a'_i และ b'_i โดยที่ค่า r_i มีโอกาสที่จะมีเลข 1 ที่เดียวคือตำแหน่งบนสุดคือ $2^i 3^0$ ส่วนค่า a'_i และ b'_i จะพิจารณาแต่ละกรณีดังนี้

กรณีที่ 1 a'_i จะไม่มีเลข 1 ติดกัน เนื่องจาก ถ้า $d_j = 1$ และ $d_{j-1} = 0$ แล้วตำแหน่งที่ (i+1, j-1) เป็นเลข 1 แต่ตำแหน่งที่ (i+2, j-1) จะไม่สามารถเป็นเลข 1 เนื่องจากว่า d_j ถูกเปลี่ยน

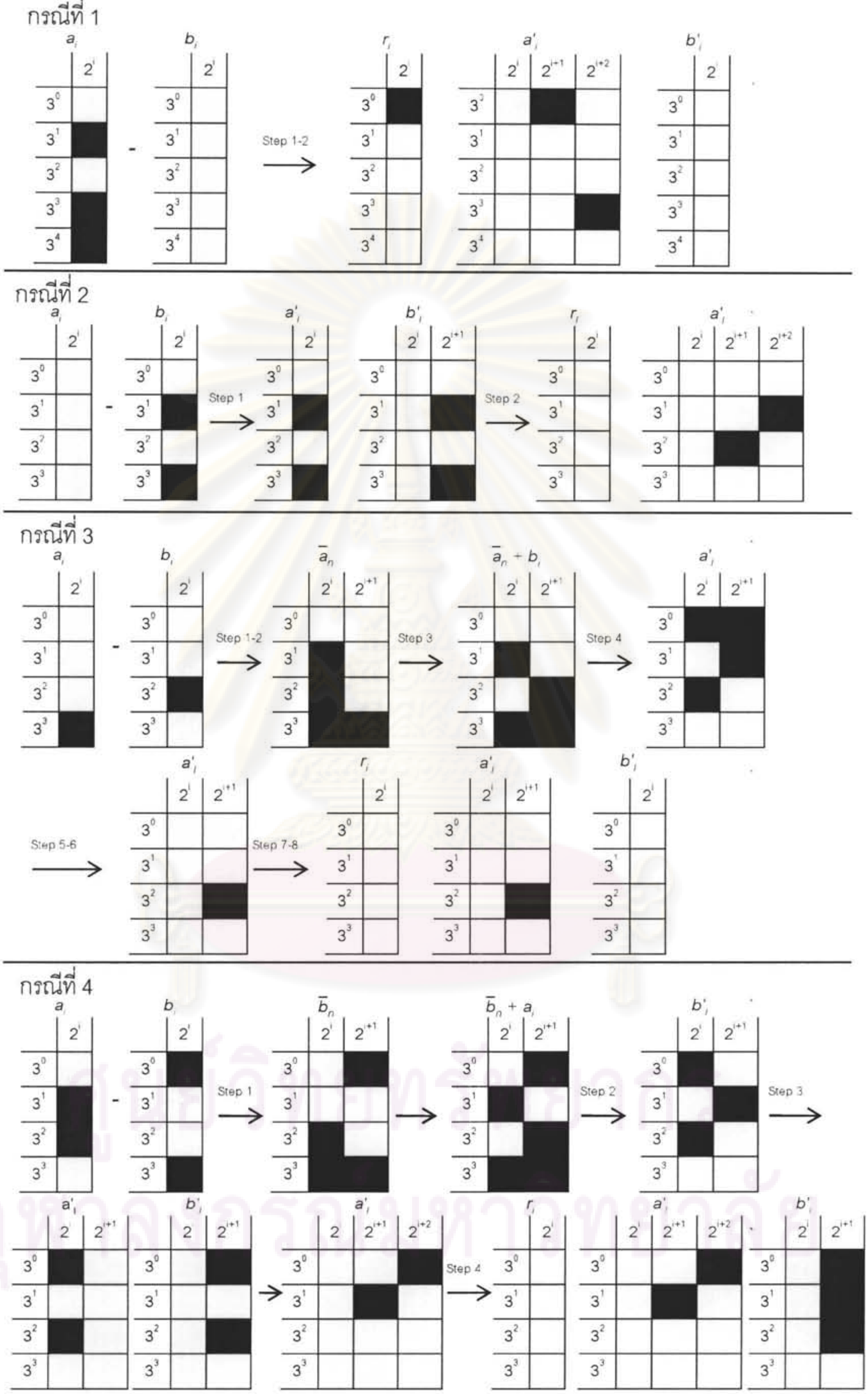
ค่าเป็น 0 และ d_{j-1} จะถูกเปลี่ยนค่าเป็น 1 ซึ่งไม่ทำให้กรณีที่เปลี่ยนค่าตำแหน่งที่ (i+2, j-1) เป็นเลข 1 เป็นจริง ส่วน b'_i มีค่าเป็น 0 ดังนั้นจึงไม่มีเลข 1 เลย

กรณีที่ 2 b'_i ไม่มีเลข 1 ติดกัน เนื่องจากว่า มีเลข 1 ที่หลักที่ i+1 หลักเดียวเท่านั้น ส่วน a'_i เกิดจากอัลกอริทึมกรณีที่ 1 ซึ่งได้พิสูจน์แล้วว่าไม่มีเลข 1 ติดกัน

กรณีที่ 3 a'_i จะไม่มีเลข 1 ติดกัน เนื่องจากว่า a'_i เกิดจากการบวกกันระหว่าง ค่า a'_{i+1} จากขั้นตอนที่ 4 และ ค่า a'_i จากขั้นตอนที่ 5 โดยที่ทั้งสองค่าไม่มีเลข 1 ติดกัน เพราะว่าค่าแรกใช้เพียงหลักเดียว ส่วนค่าที่สองเกิดจากอัลกอริทึมกรณีที่ 1 ซึ่งได้พิสูจน์แล้วว่าไม่มีเลข 1 ติดกัน เมื่อมาบวกกันด้วยอัลกอริทึมการบวกซึ่งพิสูจน์แล้วว่าผลลัพธ์จากการบวกจะอยู่ในรูปแบบพร้อมบวก ส่วน b'_i มีค่าเป็น 0 ดังนั้นจึงไม่มีเลข 1 เลย

กรณีที่ 4 a'_i จะไม่มีเลข 1 ติดกัน เนื่องจากว่า a'_i เกิดจากอัลกอริทึมกรณีที่ 2 ซึ่งได้พิสูจน์แล้วว่าไม่มีเลข 1 ติดกัน ส่วน b'_i จะไม่มีเลข 1 ติดกัน เนื่องจากว่า มีเลข 1 ที่หลักที่ i+1 หลักเดียวเท่านั้น ■

ตัวอย่างการลบในแนวหลักแสดงได้ดังรูปที่ 3.16 ซึ่งได้แจกแจงการลบทั้ง 4 กรณี โดยได้แสดงการทำงานแต่ละขั้นตอนและผลลัพธ์ที่ได้ในแต่ละกรณี



รูปที่ 3.16 ขั้นตอนการทำงานของอัลกอริทึมการลบในแนวหลักแต่ละกรณี

3.3 อัลกอริทึมการลบสำหรับระบบจำนวนฐานคู่

จากอัลกอริทึมการลบในแนวหลักจะเห็นว่าการลบนั้นจะได้ผลลัพธ์สองค่าคือผลลัพธ์จากการลบในแนวตั้งและสายการทดในหลักของตัวตั้งและตัวลบ โดยที่ผลลัพธ์ของการลบในแนวตั้งสามารถเป็นไปได้เพียงบิตเดียวคือ 2^k3^0 ซึ่งผลลัพธ์นั้นจะไม่มีผลต่อการลบในหลักถัดไปเนื่องจากกว่าค่าเชิงตัวเลขของผลลัพธ์จะมีน้อยกว่าค่าเชิงตัวเลขที่เป็นไปได้ทั้งหมดในหลักถัดไป ซึ่งทำให้ค่าผลลัพธ์ของการลบในแนวตั้งไม่มีนัยสำคัญในการการลบในหลักถัดๆ ไป โดยที่สายการทดของการลบในแนวหลักอาจเกิดได้ทั้งสายการทดในหลักถัดไปของตัวตั้งและหลักถัดไปของตัวลบ โดยที่สายการทดในหลักถัดไปของตัวตั้งสามารถเกิดได้ในทุกกรณี โดยที่อาจเกิดสายการทดที่มีตำแหน่งที่มีค่าในหลักที่ k+1 และ k+2 เท่านั้น ส่วนสายการทดในหลักของตัวลบจะมีตำแหน่งที่มีค่าในหลักที่ k+1 และเกิดในกรณีที่ 2 และ 4 เท่านั้น ซึ่งเป็นกรณีที่ค่าเชิงตัวเลขของหลักของตัวลบมีค่ามากว่าค่าเชิงตัวเลขของหลักของตัวตั้ง ซึ่งการที่สายการทดของตัวลบเกิดขึ้นในหลักที่ k+1 เพียงหลักเดียวทำให้สายการทดถูกนำไปใช้ในการลบในหลักถัดไปคือหลักที่ k+1 ได้ทันที ดังนั้นถ้าค่าเชิงตัวเลขของตัวตั้งมีค่ามากกว่าตัวลบแล้วเมื่อดำเนินการลบในแนวหลักไปยังหลักที่ n+1 ค่าเชิงตัวเลขของตัวลบจะเท่ากับ 0

ดังนั้นการดำเนินการลบจำเป็นต้องพิจารณาตำแหน่งที่มีค่าของตัวลบทุกตำแหน่ง ดังนั้นอัลกอริทึมต้องทำการลบในแนวหลักเป็นจำนวน n+1 รอบ โดยที่ n คือจำนวนหลักในรูปแบบแทนจำนวนของตัวลบ โดยที่ถ้าเกิดดำเนินการลบในแนวหลักจนครบจำนวน n+1 รอบแล้วยังมีค่าเหลือในสายการทดของตัวลบ แสดงว่าค่าเชิงตัวเลขของ

ทฤษฎีบทที่ 3.2: การลบในระบบจำนวนฐานคู่ ทำได้และให้ผลลัพธ์ขนาดไม่เกิน (N+3) × M เมื่อ N คือจำนวนหลักของข้อมูลตัวตั้งและตัวลบ และ M คือจำนวนแถวของข้อมูลตัวตั้งและตัวลบ โดยอัลกอริทึมการลบ

อัลกอริทึม 3.2 อัลกอริทึมการลบ (Subtraction algorithm)

Input: A and B, two WAR forms with size $n \times n$

$A = \sum_{i,j} d_{i,j} 2^i 3^j \mid d \in \{0,1\}$

$B = \sum_{i,j} e_{i,j} 2^i 3^j \mid e \in \{0,1\}$

Output: R represented in DBNR with size (n+3) x n and

$$\|R\| = \|A\| - \|B\| \; ; \; \|A\| \geq \|B\|$$

Define: a_i is a column i^{th} of A such that $a_i = \sum_j d_j 2^i 3^j \mid d \in \{0,1\}$

b_i is a column i^{th} of B such that $b_i = \sum_j e_j 2^i 3^j \mid e \in \{0,1\}$

r_i is a i^{th}-column representation that is result from column subtraction algorithm such that $r_i = \sum_j f_j 2^i 3^j \mid f \in \{0,1\}$

ca_{i+1} is a $i+1^{th}$-column representation that are carries of addend result from column subtraction algorithm such that $ca_{i+1} = \sum_j g1_j 2^{i+1} 3^j \mid g1 \in \{0,1\}$

ca_{i+2} is a $i+2^{th}$-column representation that are carries of addend result from column subtraction algorithm such that $ca_{i+2} = \sum_j g2_j 2^{i+2} 3^j \mid g2 \in \{0,1\}$

cb_i+1 is a $i+1^{th}$-column representation that are carries of subtracter result from column subtraction algorithm such that $cb_{i+1} = \sum_j h_j 2^{i+1} 3^j \mid h \in \{0,1\}$

$T1$ and $T2$, two WAR forms with size $(n+3) \times n$

$t1_i$ is a column i^{th} of $T1$ such that $t1_i = \sum_j k_j 2^i 3^j \mid k \in \{0,1\}$

n = the amount of rows or columns in representation

Step 1: $i = 0$

Step 2: while ($i<n+1$) do

Step 3: for any j^{th} row,

If $d_{i,j} = 1$ and $e_{i,j} = 1$ then $d_{i,j} = 0$, $e_{i,j} = 0$

Step 4: for ($j = n$ to $j > 0$ step -1) do

If $d_{i,j} = 1$ then

apply case 4 in column subtraction for (a_i,b_i) and exit for loop

else if $e_{i,j} - 1$ then

apply case 3 in column subtraction for (a_i,b_i) and exit for loop

Step 5: set $a_i = 0$ and $b_i = 0$

Step 6: $B = B + cb_{i+1}$

Step 7: $R = R + r_i$

Step 8: for any j^{th} row,

If $g1_j = 1$ and $e_{i+1,j} = 1$ then $g1_j = 0$, $e_{i+1,j} = 0$

If $g2_j = 1$ and $e_{i+1,j} = 1$ then $g2_j = 0$, $e_{i+1,j} = 0$

Step 9: $T1 = T1 + ca_{i+1} + ca_{i+2}$

Step 10: $A = A + t1_{i+1}$

Step 11: Set $t1_{i+1} = 0$

Step 12: $i = i+1$

End while loop

Step 13: if $cb_{i+1} = 0$ then

Return the output $R + A + T1$ that using addition algorithm

else

Return invalid subtraction because $A < B$

พิสูจน์ จะต้องพิสูจน์ว่า การลบให้ค่าที่ถูกต้อง และผลลัพธ์อยู่ในรูปแบบพร้อมบวก

1) อัลกอริทึมนี้เกิดจากอัลกอริทึมย่อยการลบในแนวหลัก โดยนำผลลัพธ์จากการการลบในแนวหลักที่ได้พิสูจน์ความถูกต้องของค่าการลบในทฤษฎีบทที่ 3.1 ซึ่งการทำงานในแต่ละรอบจะนำผลลัพธ์ที่เป็นสายการทดของตัวตั้งและตัวลบไปใช้ในการคำนวณในหลักถัดไปเสมอ ดังนั้นการทำงานแต่ละรอบจึงให้ผลลัพธ์ถูกต้อง โดยที่อัลกอริทึมมีการทำงานทั้งหมด n+1 รอบซึ่งเป็นการทำงานในรอบสุดท้ายของตัวลบ เนื่องจากว่าตัวลบสามารถมีสายการทดได้มากที่สุด 1 หลัก เมื่อพิจารณาการลบในหลักที่ 1 ถ้าตัวลบมีค่ามากกว่าตัวบวกจะทำให้เกิดสายการทดของตัวลบไปยังหลักที่ 2 แต่ถ้าตัวตั้งมีค่ามากกว่าตัวลบทำให้เกิดสายการทดของตัวตั้งไปยังหลักที่ 2 และ 3 พิจารณาการลบในหลักที่ n-1 ถ้าตัวลบมีค่ามากกว่าตัวตั้ง และจากสมติฐานว่าค่าเชิงตัวเลขของตัวตั้งมีค่ามากกว่าค่าเชิงตัวเลขของตัวลบ ดังนั้นตัวตั้งจะต้องมีค่าที่มากกว่าตัวลบที่เหลือในหลักที่ n และ n+1 โดยที่ค่าในหลักที่ n+1 เป็นสายการทดในการลบของหลักที่ n-2 ซึ่งการลบในหลักที่ n-1 จะทำให้เกิดสายการทดของตัวลบไปยังหลักที่ n และเมื่อรวมกับค่าของตัวลบในหลักที่ n แล้วทำให้ค่าของตัวลบในหลักที่ n มากกว่าค่าของตัวตั้งในหลักที่ n แล้วจะเกิดสายการทดของตัวลบไปยังหลักที่ n+1 ซึ่งค่าของตัวตั้งในหลักที่ n+1 จะต้องมากกว่าค่าของตัวลบในหลักที่ n+1 เสมอ เมื่อพิจารณาการลบในหลักที่ n ถ้าค่าของตัวลบในหลักที่ n มีค่ามากกว่าค่าของตัวตั้งในหลักที่ n แล้วค่าของตัวตั้งจะต้องมีสายการทดที่มีค่ามากว่าค่าของตัวลบซึ่งสายกการทดของตัวตั้งจะต้องเกิดการการลบหลักก่อนหน้าคือหลักที่ n-1 โดยจะให้สายการทดของตัวตั้งในหลักที่ n และ n+1 ดังนั้นการลบเป็นจำนวน n+1 รอบจะทำให้ค่าของตัวลบเป็น 0 ซึ่งทำให้ผลการลบเป็นผลที่ถูกต้อง
2) อัลกอริทึมนี้เกิดจากอัลกอริทึมย่อยการลบในแนวหลัก โดยนำผลลัพธ์จากการการลบในแนวหลักที่ได้พิสูจน์แล้วว่าผลลัพธ์ที่เกิดขึ้นจะไม่มีเลข 1 ติดกัน โดยอัลกอริทึมการลบจะนำผลจากอัลกอริทึมย่อยการลบในแนวหลักมาบวกกันด้วยอัลกอริทึมการบวก ซึ่งได้พิสูจน์แล้วว่าผลจากอัลกอริทึมการบวกเป็นจำนวนพร้อมบวก ดังนั้นผลลัพธ์จากอัลกอริทึมการลบจะให้ผลลัพธ์อยู่ในรูปแบบจำนวนพร้อมบวกด้วยเสมอ

ดังนั้นจึงสรุปได้ว่าอัลกอริทึมการลบให้ค่าที่ถูกต้อง และผลลัพธ์อยู่ในรูปแบบพร้อมบวก ■

เมื่อพิจารณาจากอัลกอริทึมจะเห็นว่าอัลกอริทึมจะทำงานเป็นจำนวนรอบขนาดเท่ากับ m+1 รอบ โดยที่ m เป็นขนาดของหลัก และในแต่ละรอบจะใช้การลบในแนวหลัก ซึ่งเมื่อพิจารณาอัลกอริทึมการลบในแนวหลักจะเห็นว่าจะทำงานเป็นจำนวนรอบขนาดเท่ากับ n รอบโดยที่ n เป็นขนาดของแถว แต่การลบในแนวหลักจะมีการทำงานอย่างมากที่สุด 2 หลัก ดังนั้นการทำงานของอัลกอริทึมการลบในแนวหลักจะวนทั้งหมด 2m รอบ ดังนั้นความซับซ้อนเชิงเวลาของอัลกอริทึมการลบจะเป็น O(mn) หรือ $O(n^2)$ ซึ่งมีขนาดไม่เกินเวลาพหุนาม

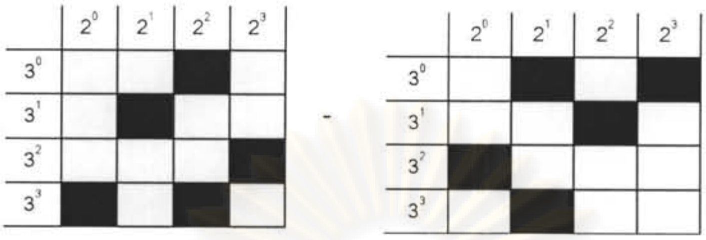
ตัวอย่างของการลบเป็นไปดังรูปที่ 3.17 ซึ่งแสดงผลของ 217 และ 85 โดยแจกแจงผลของแต่ละขั้นตอน และได้ผลลัพธ์การลบมีค่าเป็น 132

จากทฤษฎีบทที่ 3.2 จะพบว่ามีอัลกอริทึมที่สามารถลบจำนวนในรูปแบบพร้อมบวกสองจำนวน แล้วให้ผลลัพธ์อยู่ในรูปแบบพร้อมบวก และมีการทำงานเชิงกำหนด การลบแบบนี้ค่าเชิงตัวเลขของตัวตั้งจะต้องมีค่ามากกว่าค่าเชิงตัวเลขของตัวลบ โดยผ่านไปจำนวนหนึ่ง ซึ่งเท่ากับเวลาในการลบแต่ละหลักจนครบทุกหลัก และการรวมค่าผลลัพธ์ของการลบในแต่ละหลัก และได้คำตอบที่ถูกต้อง แต่ถ้าค่าเชิงตัวเลขของตัวลบมีค่ามากกว่าอัลกอริทึมสามารถหยุดและแสดงผลไม่สามารถลบได้

ข้อเสียของการลบแบบนี้ คือ ผลลัพธ์ของการลบจะอยู่ในแถวแรกและหลักทางขวามือเสมอ และการลบนี้จะทำงานได้ถูกต้องกับระบบจำนวนฐานคู่ที่มีฐานเป็น 2 และ 3 เท่านั้น

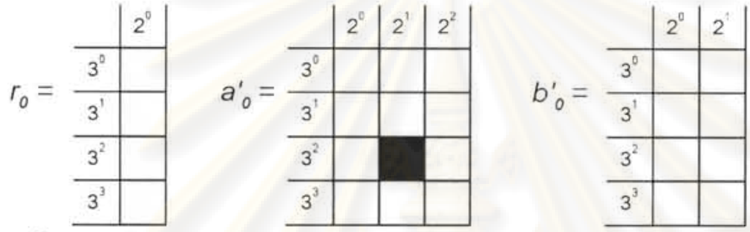
รูปที่ 3.17 ขั้นตอนการทำงานของอัลกอริทึมการลบ

A = 217 B = 85

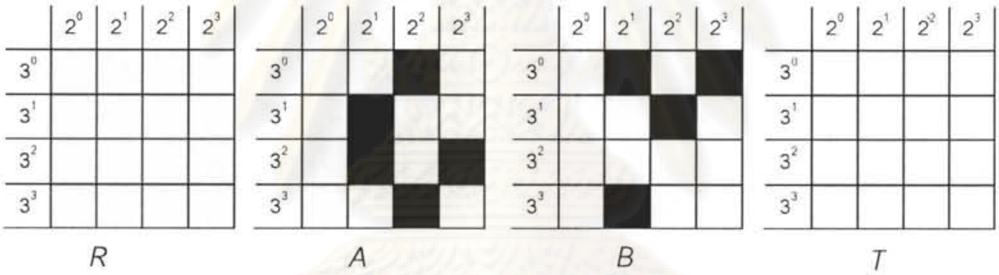


รอบที่ 1 พิจารณาหลักที่ 2^0

- ขั้นตอนที่ 4 ใช้ case 3 ของอัลกอริทึมการลบในแนวหลัก

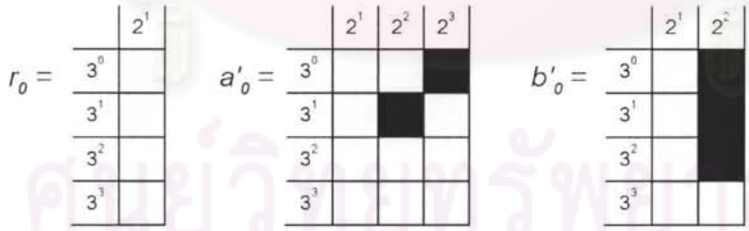


- ขั้นตอนที่ 5-10 การรวมค่าของผลลัพธ์จากการลบในแนวหลัก

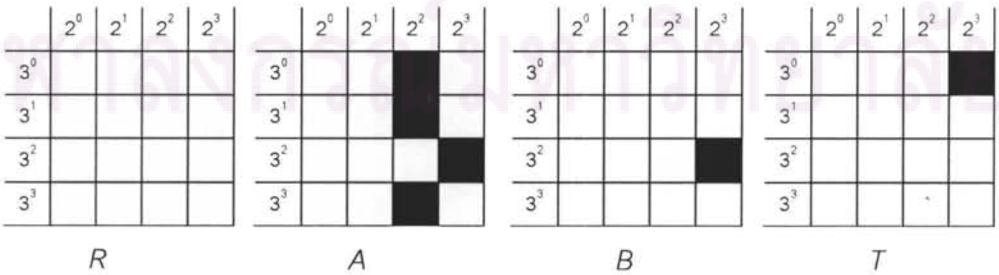


รอบที่ 2 พิจารณาหลักที่ 2^1

- ขั้นตอนที่ 4 ใช้ case 4 ของอัลกอริทึมการลบในแนวหลัก

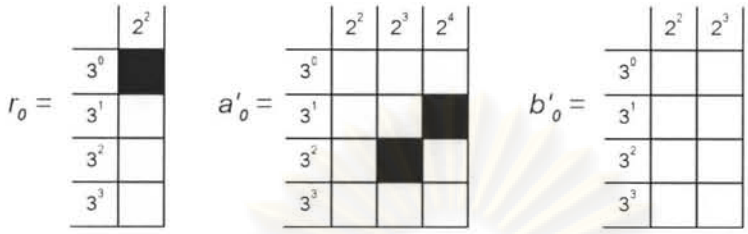


- ขั้นตอนที่ 5-10 การรวมค่าของผลลัพธ์จากการลบในแนวหลัก

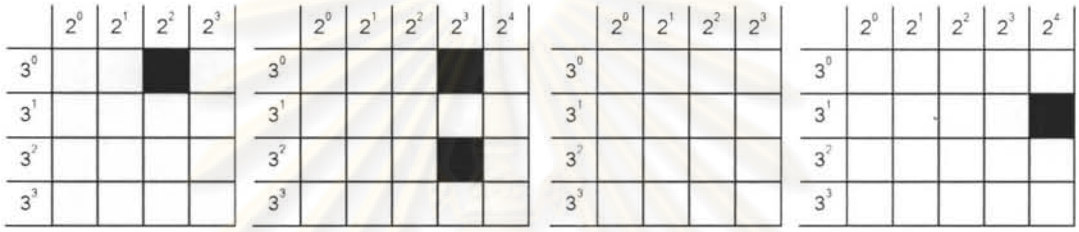


รอบที่ 3 พิจารณาหลักที่ 2^2

- ขั้นตอนที่ 4 ใช้ case 1 ของอัลกอริทึมการลบในแนวหลัก



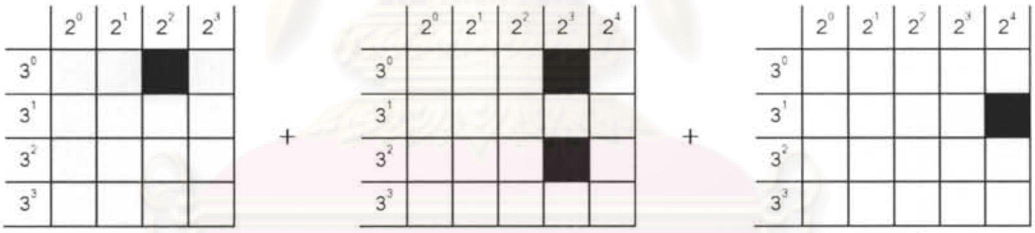
- ขั้นตอนที่ 5-10 การรวมค่าของผลลัพธ์จากการลบในแนวหลัก



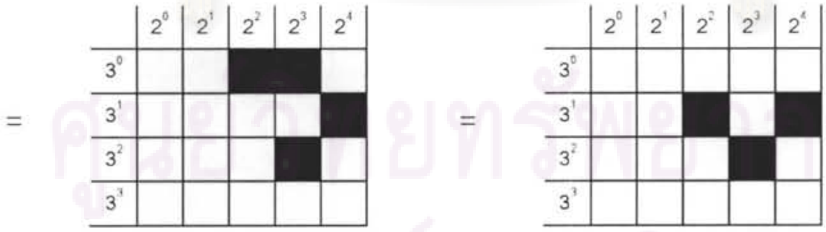
R A B T

รอบที่ 4 เนื่องด้วย $B = 0$ ดังนั้นจบการวนรอบ

- ขั้นตอนที่ 13 รวมผลลัพธ์



R A T



R R

บทที่ 4

อัลกอริทึมการหารสำหรับจำนวนฐานคู่

ในบทนี้ เราจะนำเสนออัลกอริทึมการหารสำหรับระบบจำนวนฐานคู่ โดยได้เสนอกระบวนการการประมาณค่าของรูปแบบจำนวนฐานคู่ที่เป็นค่าของผลบวกของแต่ละตำแหน่งในตารางแสดงค่าให้เหลือเพียงตำแหน่งเดียว โดยที่มีค่าเชิงตัวเลขมากกว่าค่าจริง และได้เสนออัลกอริทึมการหารโดยใช้ค่าประมาณที่ได้จากข้างต้น เพื่อให้ได้ผลลัพธ์และเศษของการหารที่อยู่ในรูปแบบแทนจำนวนในระบบจำนวนฐานคู่

4.1 การเปรียบเทียบค่าเชิงตัวเลขของ 2 ตำแหน่งในตารางแสดงค่าในระบบจำนวนฐานคู่

บทตั้งที่ 4.1: *ให้* $A=\sum_{i,j} d_{i,j}2^i3^j$ *โดยที่* $d \in \{0,1\}$ *และให้* $x = 1\cdot(2^m3^n)$ *โดยที่* $m \leq I$ *และ* $n \leq j$ *แล้วในทุกแถว* j *จะมี* $y_1 = 1X(2^{m+a}3^{n+b})$ *และ* $y_2 = 1X(2^{m+a}3^{n+b})$ *โดยที่* $y_1 \leq x \leq y_2$ *และ* $a,b \in Z$

บทพิสูจน์ จะต้องพิสูจน์ว่าในแต่ละแถวสามารถหาค่า y_1และ y_2 ได้เสมอ

พิจารณาสมการ $\frac{2^a}{3^b} \leq 1 \leq \frac{2^{a+1}}{3^b}$ สมมติให้ $p=\frac{2^a}{3^b}$ และ $q=\frac{2^{a+1}}{3^b}$ ถ้า $p=1$ซึ่งเป็นค่าที่มากที่สุดที่เป็นไปได้ของ p ดังนั้นค่า q ที่มากกว่า p ที่อยู่ในรูป 2^i3^j โดยที่มีค่าของ j เท่ากันคือ 2 ซึ่งสมการข้างต้นยังเป็นจริง ถ้าเพิ่ม b อีก 1 แล้วค่าของ p และ q จะเป็น 1/3 และ 2/3 ตามลำดับ ซึ่งถ้านำ 2^1 คูณค่า p และ q จะทำให้สมการข้างต้นเป็นจริง ในทางกลับกันถ้าให้ $q=1$ ซึ่งเป็นค่าที่น้อยที่สุดที่เป็นไปได้ของ q ค่า p จะเท่ากับ 0.5 ซึ่งสมการข้างต้นยังเป็นจริง ถ้าเพิ่ม b อีก 1 แล้วค่าของ p และ q จะเป็น 0.5/3 และ 1/3 ตามลำดับ ซึ่งถ้านำ 2^2 คูณค่า p และ q จะทำให้สมการข้างต้นเป็นจริง จะเห็นว่าจากขอบเขตของค่าที่มากที่สุด และค่าที่น้อยที่สุด ถ้าเพิ่ม b ขึ้น 1 จะต้องเพิ่มค่าของ a ขึ้นอีก 1 หรือ 2 ก็จะทำให้สมการดังกล่าวเป็นจริงเสมอ โดยตารางที่ 1 เป็นการแสดงการค่าของ a เมื่อเพิ่มค่าของ b ทีละ 1 ที่ทำให้สมการ $\frac{2^a}{3^b} \leq 1 \leq \frac{2^{a+1}}{3^b}$ เป็นจริง ดังนั้นการที่ b เพิ่มขึ้นทีละ 1 จะสามารถหาค่า a ที่ทำให้สมการเป็นจริงได้เสมอ ดังนั้นถ้านำ $\frac{2^a}{3^b}$ และ $\frac{2^{a+1}}{3^b}$ ไปคูณกับค่าของ x ซึ่งจะทำให้ได้ค่า $\frac{2^a}{3^b}\cdot x \leq x \leq \frac{2^{a+1}}{3^b}\cdot x$ ดังนั้นค่า y_1 คือ $\frac{2^a}{3^b}\cdot x$ และ y_2 คือ $\frac{2^{a+1}}{3^b}\cdot x$ ดังนั้นสรุปได้ว่าในแต่ละแถว j สามารถหาค่า y_1และ y_2 ได้เสมอ ■

จากบทตั้งบทที่ 4.1 ทำให้สามารถทำการเปรียบค่าเชิงตัวเลขของจำนวน 2 บิต ได้ดังนี้

สมมติ $x = 2^i3^j$ และ $y = 2^k3^l$ เป็นบิตหนึ่งในตารางแสดงค่าของระบบจำนวนฐานคู่ โดยที่ $j > l$ ซึ่ง b มีค่าเท่ากับ $j - l$ แล้วสามารถหาค่า a โดยดูจากตารางที่ 1 สมมติว่าเท่ากับ ω ดังนั้นจะได้ว่า $2^{k+\omega}3^{l+(j-l)} \leq 2^i3^j \leq 2^{k+\omega+1}3^{l+(j-l)}$ ซึ่งถ้า $k<=i+\omega$ แสดงว่า $x>y$ แต่ถ้า $k>i+\omega$ แสดงว่า $x<y$

b	a	$2^a/3^b$	$2^{a+1}/3^b$
1	1	0.666666667	1.333333333
2	3	0.888888889	1.777777778
3	4	0.592592593	1.185185185
4	6	0.790123457	1.580246914
5	7	0.526748971	1.053497942
6	9	0.702331962	1.404663923
7	11	0.936442615	1.872885231
8	12	0.624295077	1.248590154
9	14	0.832393436	1.664786872
10	15	0.554928957	1.109857915
11	17	0.739905276	1.479810553
12	19	0.986540369	1.973080737
13	20	0.657693579	1.315387158
14	22	0.876924772	1.753849544
15	23	0.584616515	1.169233029
16	25	0.779488686	1.558977373
17	26	0.519659124	1.039318248
18	28	0.692878832	1.385757664
19	30	0.923838443	1.847676886
20	31	0.615892295	1.231784591
21	33	0.821189727	1.642379454
22	34	0.547459818	1.094919636
23	36	0.729946424	1.459892848
24	38	0.973261899	1.946523798
25	39	0.648841266	1.297682532
26	41	0.865121688	1.730243376

ตารางที่ 4.1 แสดงค่า a เมื่อ b เพิ่มขึ้นทีละ 1 ซึ่งทำให้ $\frac{2^a}{3^b} \le 1 \le \frac{2^{a+1}}{3^b}$ เป็นจริง

4.2 การหาประมาณค่าของ 2 ตำแหน่งในตารางแสดงค่าในระบบจำนวนฐานคู่ เป็นหนึ่งตำแหน่ง

สมมติให้ $x = 2^a3^b$ และ $y = 2^c3^d$ เป็นบิตหนึ่งในระบบจำนวนฐานคู่ โดยที่ $x > y$ เราสามารถหาค่า $s = 2^m3^n$ เป็นค่าประมาณของ $x + y$ ได้โดยที่ $s > x + y$ ซึ่งเมื่อทำพิจารณาสมการ $2^a3^b + 2^c3^d = 2^a3^b\,(1 + 2^{c-a}3^{d-b})$ จะพบว่า $(1 + 2^{c-a}3^{d-b})$ จะมีค่า อยู่ในช่วง (1, 2] เสมอเนื่องจาก $x > y$ ดังนั้น เราจะประมาณค่า $(1 + 2^{c-a}3^{d-b})$ ด้วยค่าคงที่ค่าหนึ่งใน 3 ค่าที่อยู่ในช่วง (1, 2] ที่ได้กำหนดขึ้น ได้แก่ $2^{-3}3^2$, $2^{-1}3^1$ และ 2^13^0 โดยจะทำการประมาณค่าดังนี้

กรณีที่ 1 ประมาณค่า $(1 + 2^{c-a}3^{d-b})$ ด้วย$2^{-3}3^2$ ถ้าค่า $2^{c-a}3^{d-b} <= 2^{-3}3^0$

ดังนั้นค่าประมาณ $s = 2^{a-3}3^{b+2}$

กรณีที่ 2 ประมาณค่า $(1 + 2^{c-a}3^{d-b})$ ด้วย$2^{-1}3^1$ ถ้า $2^{-3}3^0 < 2^{c-a}3^{d-b} <= 2^{-1}3^0$

ดังนั้นค่าประมาณ $s = 2^{a-1}3^{b+1}$

กรณีที่ 3 ประมาณค่า $(1 + 2^{c-a}3^{d-b})$ ด้วย2^13^0 ถ้า $2^{-1}3^0 < 2^{c-a}3^{d-b} <= 2^03^0$

ดังนั้นค่าประมาณ $s = 2^{a+1}3^b$

4.3 อัลกอริทึมการหารสำหรับระบบจำนวนฐานคู่

หลักการของการหารของอัลกอริทึมนี้จะนำรูปแบบแทนจำนวนของตัวหารมาหาค่าประมาณแล้วทำการหารด้วยการเลื่อนบิตจนกว่าค่าผลลัพธ์จะมีค่าน้อยกว่าค่าที่กำหนดไว้

ทฤษฎีบทที่ 4.1 การดำเนินการหารของรูปแบบแทนค่าในระบบจำนวนฐานคู่ 2 จำนวน สามารถหาผลลัพธ์และเศษจากการหารที่เป็นรูปแบบแทนค่าในระบบจำนวนฐานคู่

อัลกอริทึม 4.1 อัลกอริทึมการหาร (division algorithm)

ให้ *A* และ *B* เป็นรูปแบบแทนค่าในระบบจำนวนฐานคู่ที่มีขนาด $n \times m$

C และ *R* เป็นรูปแบบแทนค่าในระบบจำนวนฐานคู่ โดยที่ *C* เป็นผลลัพธ์ของการหาร *A* ด้วย *B* และ *R* เป็นเศษเหลือของการหาร *A* ด้วย *B*

ขั้นตอนการหาร

1. หาค่าประมาณของ *B*

ให้ $A = \sum_{i,j} d_{i,j}2^i3^j$ โดยที่ $d \in \{0,1\}$, $0 \le i \le m$, $0 \le j \le n$

ให้ $a,b,c,d,x,y \in I$

ขั้นตอน

$a = b = -1$

for ($i = 0$ to $i \le m$ step $i+1$) do for (j = 0 to $j \le n$ step j+1) do

if $d_{i,j} = 1$ then

if $a = -1$ then

$x = a = i$

$y = b = j$

else

$c = i$

$d = j$

apply การหาค่าประมาณของ 2 ตำแหน่ง ผลลัพธ์ให้กับ x และ y

end if

end if

end for

end for

2. นำ S ที่ค่าประมาณของ B มาหาร A โดยการเลื่อนบิตของ A

ให้ $S = f_{k,l}2^k3^l$ โดยที่ $f_{k,l}$=1, $0 \le k \le m+1$, $0 \le l \le n+1$

$A = \sum_{i,j} d_{i,j}2^i3^j$ โดยที่ $d \in \{0,1\}$, $0 \le i \le m$, $0 \le j \le n$

และให้ผลลัพธ์เป็น

$C = \sum_{p,q} g_{p,q}2^p3^q$ โดยที่ $g \in \{0,1\}$, $-k \le p \le m$, $-l \le q \le n$

ขั้นตอน

for (i = 0 to $i \le m$ step i+1) do

for (j = 0 to $j \le n$ step j+1) do

if $d_{i,j} = 1$ then

$g_{i-k,j-l} = 1$

end if

end for

end for

3. หาเศษของการหาร A ด้วย S

ให้ $A = \sum_{i,j} d_{i,j}2^i3^j$ โดยที่ $d \in \{0,1\}$, $0 \le i \le m$, $0 \le j \le n$

$B = \sum_{i,j} e_{i,j}2^i3^j$ โดยที่ $e \in \{0,1\}$, $0 \le i \le m$, $0 \le j \le n$

$C = \sum_{p,q} g_{p,q} 2^p 3^q$ โดยที่ $g \in \{0,1\}$, $-k \leq p \leq m$, $-l \leq q \leq n$

และให้เศษเป็น

$R = \sum_{t,u} h_{p,q} 2^t 3^u$ โดยที่ $h \in \{0,1\}$, $-k \leq t \leq mn + 3$, $-l \leq u \leq mn$

ขั้นตอน

$R = A - (B * C)$

4. นำเศษของการหารในขั้นตอนที่ 3 มาทำการหารต่อด้วยขั้นตอนที่ 2 และ 3 จนกว่าผลลัพธ์ในขั้นตอนที่ 2 จะน้อยกว่าค่าที่กำหนดไว้
5. รวมผลลัพธ์ของแต่ละรอบในขั้นตอนที่ 2 เป็นผลลัพธ์จากการหาร

พิสูจน์ จะต้องพิสูจน์ว่า การหารให้ค่าที่ถูกต้อง

อัลกอริทึมนี้เกิดจากการประมาณค่า อัลกอริทึมการลบและการคูณ โดยอัลกอริทึมการลบได้พิสูจน์ความถูกต้องในทฤษฎีบทที่ 3.2 และการคูณได้พิสูจน์แล้วว่าให้ผลลัพธ์ที่ถูกต้อง ซึ่งในการทำงานนั้นจะใช้ค่าประมาณมาทำการเลื่อนบิต โดยที่การใช้ค่าประมาณมาใช้ในการหารนั้นสามารถแสดงการหารค่าเชิงตัวเลขของ A หารด้วย B ได้ผลลัพธ์เป็นค่าเชิงตัวเลขของ C และเศษ R ได้ดังนี้

สมมติ $A/B = C + (R/B)$

จาก $C = A/S$ และ $R = A-(B*C)$

จะได้ $A/B \quad = (A/S) + ((A-(B*(A/S)))/B)$

$\qquad\qquad = (A/S) + (A/B) - (A/S)$

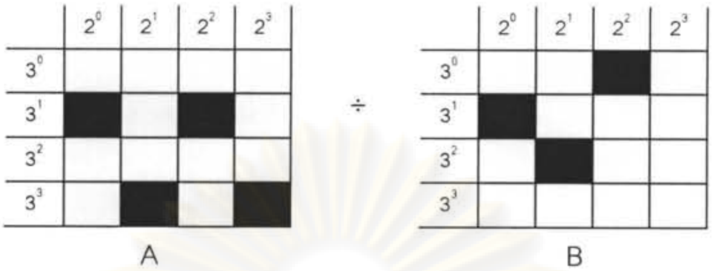
$\qquad\qquad = A/B$

ดังนั้นจึงสรุปได้ว่าอัลกอริทึมการหารให้ค่าที่ถูกต้อง ■

ตัวอย่างของการหารเป็นไปดังรูปที่ 4.1 ซึ่งแสดงผลของ 285 หารด้วย 25 โดยแจกแจงผลของแต่ละขั้นตอน โดยกำหนดค่า 1 เป็นค่าที่ใช้หยุดอัลกอริทึม และได้ผลลัพธ์การหารมีค่าเป็น 11.3374 และได้เศษการหารเป็น 1.5638

จากทฤษฎีบทที่ 4.1 จะพบว่ามีอัลกอริทึมที่สามารถหารจำนวนในรูปแบบพร้อมบวกสองจำนวน แล้วให้ผลลัพธ์อยู่ในรูปแบบพร้อมบวกสองจำนวนคือผลการหารและเศษจากการหาร อัลกอริทึมนี้มีข้อจำกัดคือความละเอียดของผลของการหารขึ้นอยู่กับค่าคงที่ที่กำหนดไว้สำหรับหยุดอัลกอริทึม

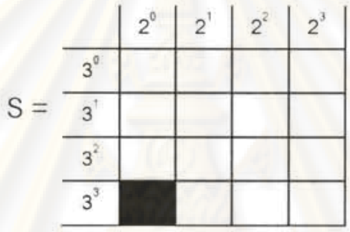
รูปที่ 4.1 แสดงขั้นตอนการทำงานของอัลกอริทึมการหาร



ขั้นที่ 1 หาค่าประมาณของ B

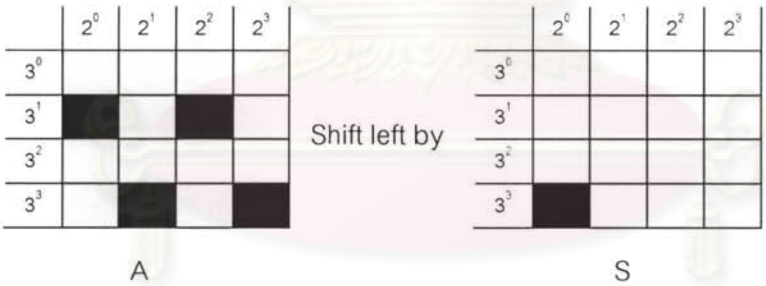
รอบที่ 1: $2^2 3^0 + 2^0 3^1 = 2^2 3^0(1+2^{-2}3^1)$ เนื่องจาก $2^{-2}3^1 > 2^{-1}3^0$ ดังนั้นค่าประมาณคือ $2^3 3^0$

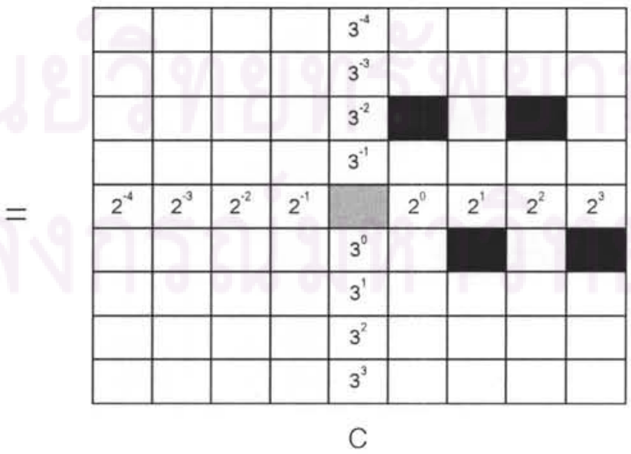
รอบที่ 2: $2^1 3^2 + 2^2 3^0 = 2^1 3^2(1+2^2 3^{-2})$ เนื่องจาก $2^{-3}3^0 < 2^2 3^{-2} <= 2^{-1}3^0$ ดังนั้นค่าประมาณคือ $2^0 3^3$



รอบที่ 1

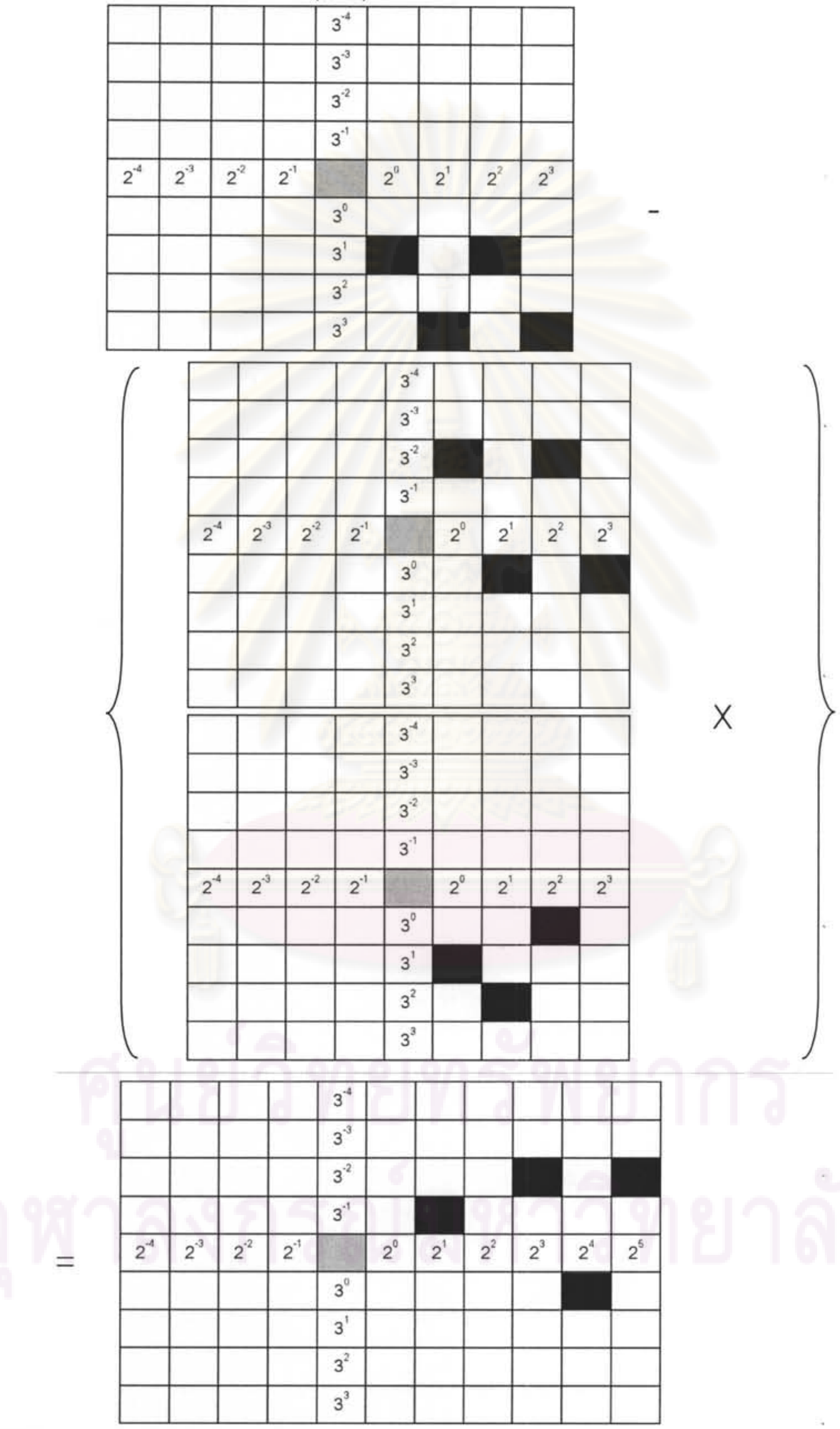
ขั้นที่ 2 หาค่าผลลัพธ์การหารด้วยการเลื่อนบิต





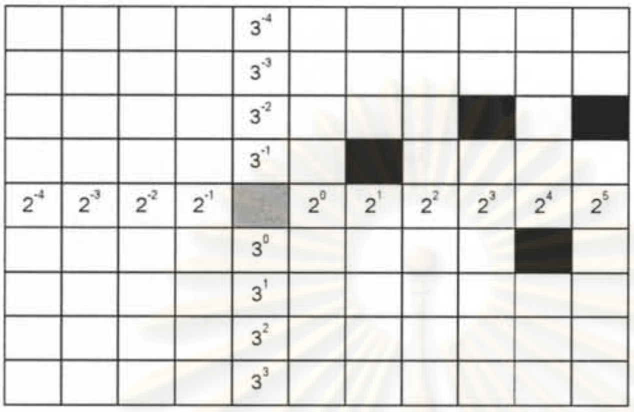
รอบที่ 1

ขั้นที่ 3 หาเศษของการหาร A-(C*B)



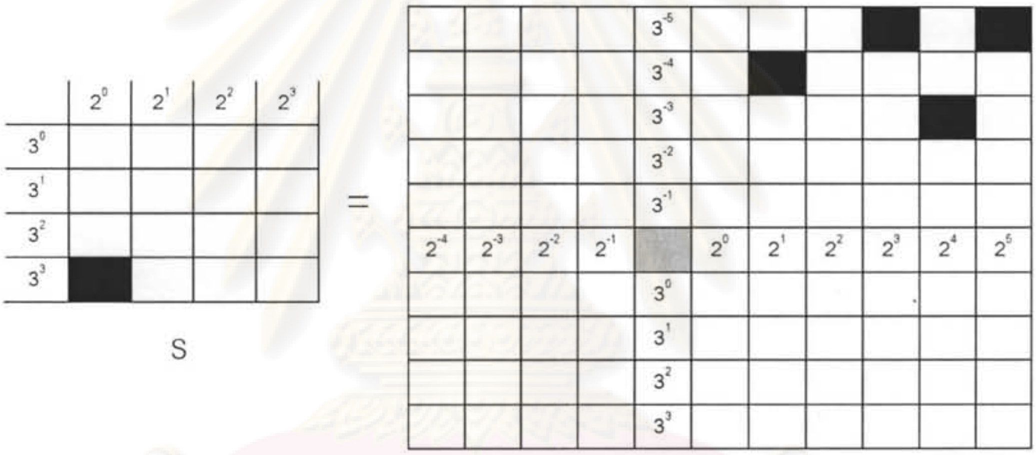
เนื่องด้วยผลลัพธ์ในขั้นตอนที่ 2 มากกว่าค่าที่กำหนดคือ 1 ดังนั้นทำการวนรอบที่ 2

ขั้นที่ 2 หาค่าผลลัพธ์การหารด้วยการเลื่อนบิต



Shift left by

A

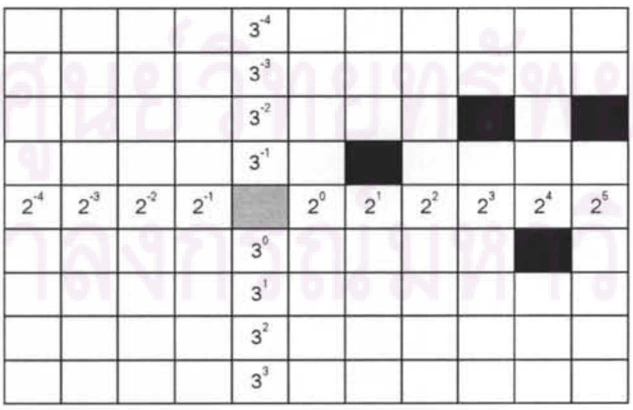


S

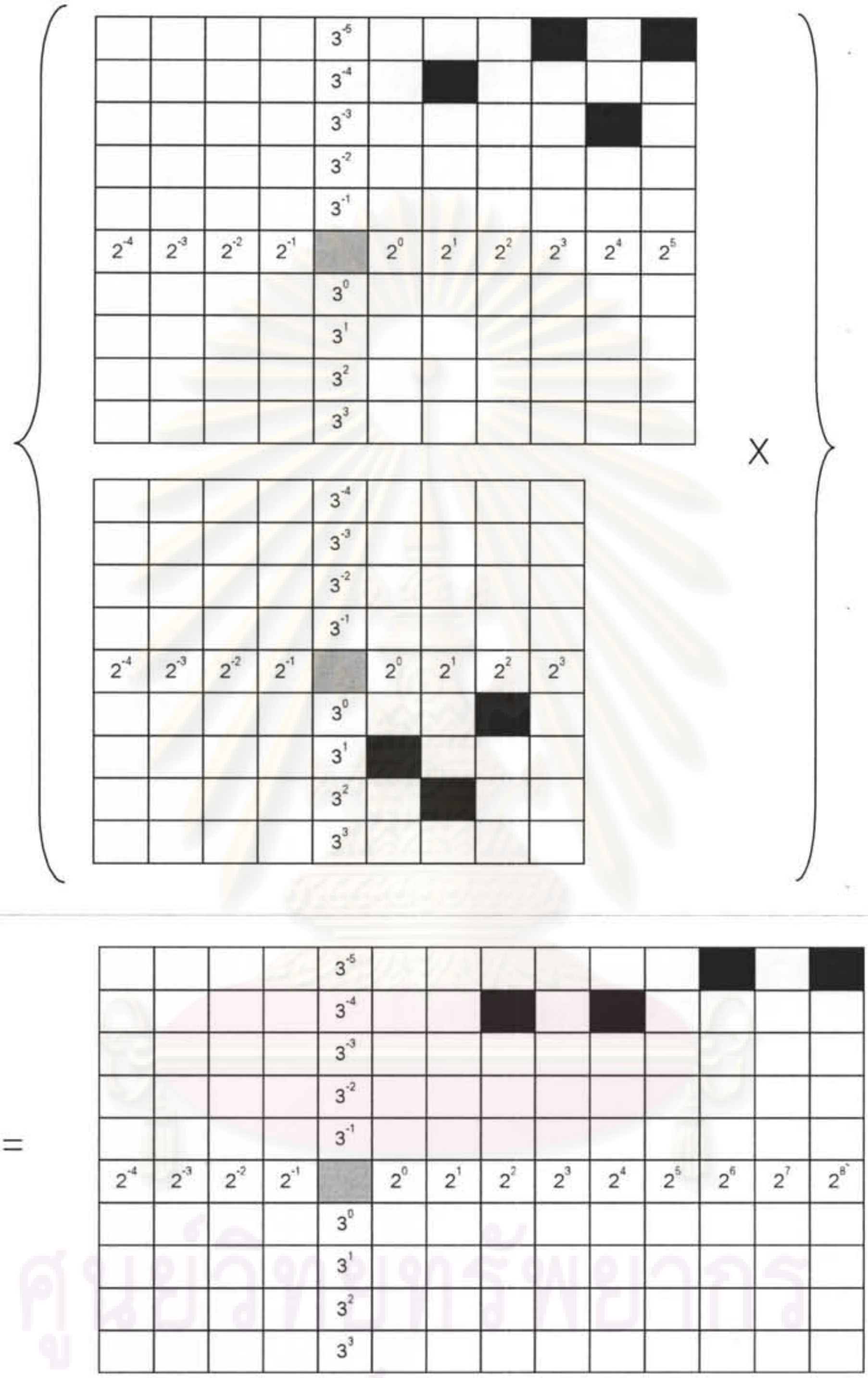
C

รอบที่ 2

ขั้นที่ 3 หาเศษของการหาร A-(C*B)

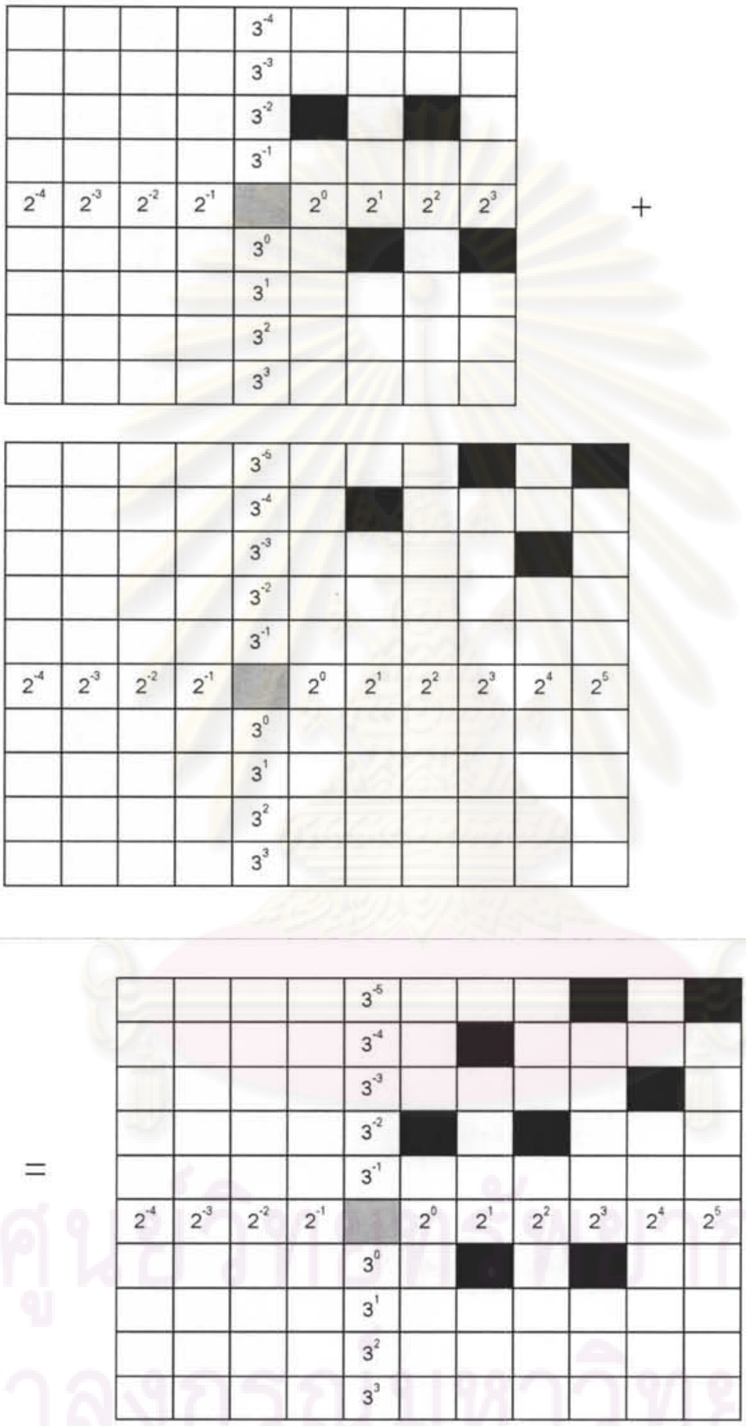


-



ขั้นที่ 4 เนื่องจากผลลัพธ์ในขั้นตอนที่ 2 มีค่าน้อยกว่าค่าที่กำหนดคือ 1 ดังนั้นจึงหยุดการวนรอบและทำขั้นที่ 5 ต่อ

ขั้นที่ 5 รวมผลลัพธ์จากรอบที่ 1 และ 2

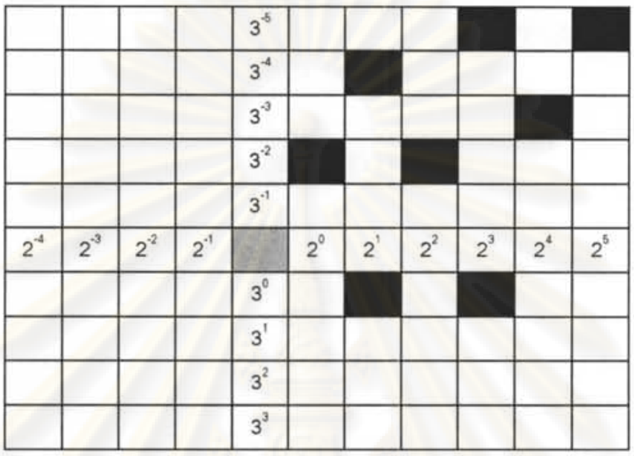


ซึ่งผลลัพธ์จากการหารจะได้ ค่าผลลัพธ์ และ เศษจากการหารโดยที่

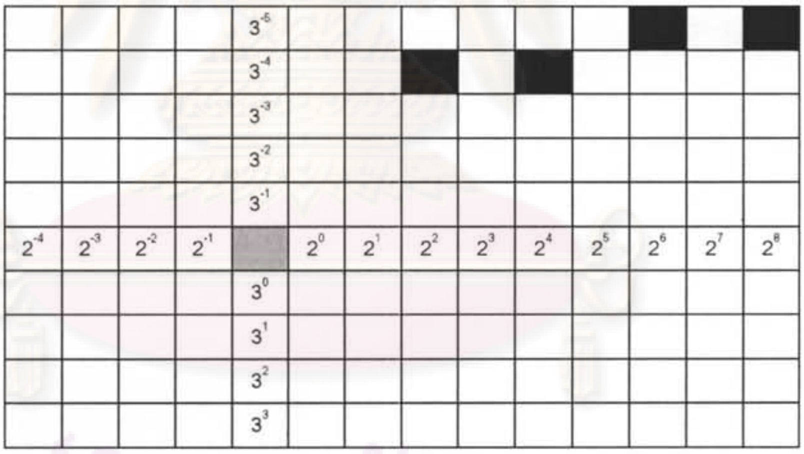
ค่าผลลัพธ์มีค่าเท่ากับ 11.3374

ค่าเศษการหารมีค่าเท่ากับ 1.5638

ผลลัพธ์ =



เศษ =



บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

5.1 สรุปผลการวิจัย

จากการนำเสนอระบบจำนวนฐานคู่โดยเป็นระบบจำนวนที่มีความซับซ้อนและยืดหยุ่นมาก อย่างไรก็ตามงานวิจัยที่เกี่ยวข้องกับระบบจำนวนนี้มีจำนวนน้อย โดยเฉพาะยังไม่มีการนำเสนอกระบวนการ และลำดับวิธีการที่แน่นอนสำหรับตัวปฏิบัติการพื้นฐานการลบ และการหาร งานวิจัยจึงมุ่งเน้นที่จะนำเสนอกระบวนการเชิงกำหนดสำหรับตัวปฏิบัติการทั้งสอง เพื่อให้ระบบจำนวนชนิดนี้มีความเหมาะสมในการใช้งานได้มากขึ้น

สิ่งที่นำเสนอเริ่มจากการลบในระบบจำนวนฐานคู่ โดยได้นำเสนออัลกอริทึมการลบที่มีลำดับที่แน่นอน แต่ยังมีข้อจำกัดคือ อัลกอริทึมสามารถให้ผลลัพธ์ของการลบที่ถูกต้องก็ต่อเมื่อค่าเชิงตัวเลขของตัวตั้งต้องมีค่ามากกว่าหรือเท่ากับค่าเชิงตัวเลขของตัวลบ อย่างไรก็ตามอัลกอริทึมสามารถตรวจจับในกรณีที่ค่าเชิงตัวเลขของตัวตั้งมีค่าน้อยกว่าค่าเชิงตัวเลขของตัวลบ ข้อจำกัดอีกอย่างของอัลกอริทึมการลบคือตัวตั้งและตัวลบต้องอยู่ในรูปแบบพร้อมบวก แต่ผลลัพธ์ก็อยู่ในรูปแบบพร้อมบวกเช่นกัน โดยขนาดของผลลัพธ์ของการลบจะมีขนาดที่เพิ่มขึ้น 3 หลัก ซึ่งเกิดมาจากการทดในระหว่างการดำเนินการลบ และรูปแบบแทนจำนวนของผลลัพธ์จะมีตำแหน่งมีค่าอยู่ในแนวแถวบน และหลักทางขวามือ

ความซับซ้อนเชิงเวลาของอัลกอริทึมการลบจะเป็นเวลาพหุนาม เนื่องจากอัลกอริทึมจะทำงานเป็นรอบตามจำนวนหลักของรูปแบบแทนจำนวน โดยในแต่ละรอบจะมีการทำงานเป็นรอบตามจำนวนแถวของรูปแบบแทนจำนวน ดังนั้นความซับซ้อนเชิงเวลาของอัลกอริทึมจะขึ้นกับขนาดของรูปแบบแทนจำนวนของตัวนำเข้าทั้งสองคือตัวตั้งและตัวลบ

เมื่อได้อัลกอริทึมการลบแล้ว จึงมีแนวคิดในการนำเสนอกระบวนการหาร เนื่องจากการหารนั้นจำเป็นต้องใช้กระบวนการลบเข้ามาช่วยด้วย โดยอัลกอริทึมการหารนั้นจะมีการใช้หลักการการประมาณค่าจากตำแหน่งที่มีค่าหลายตำแหน่งในรูปแบบแทนจำนวนให้เหลือเพียงตำแหน่งเดียว โดยที่ค่าเชิงตัวเลขของค่าประมาณจะมีค่ามากกว่าค่าเชิงตัวเลขของตัวถูกประมาณ โดยค่าประมาณจะถูกนำมาใช้ในการเลื่อนบิตของตัวตั้ง และนำผลลัพธ์ของการเลื่อนบิตมาหาค่าของเศษของการหาร ซึ่งอัลกอริทึมการหารจะสิ้นสุดก็ต่อเมื่อค่าเชิงตัวเลขของเศษของการหารมีค่าน้อยกว่าค่าที่กำหนดไว้ ข้อจำกัดของอัลกอริทึมนี้คือผลลัพธ์ของการหารจะมีสองจำนวนคือ ผลการหาร

และเศษของการหาร อย่างไรก็ตามความละเอียดของคำตอบของการหารสามารถถูกกำหนดได้ด้วยการเปลี่ยนค่าคงที่ที่ใช้ในการหยุดการวนรอบในอัลกอริทึม

ประสิทธิภาพของอัลกอริทึมการหารนั้นขึ้นอยู่กับ 2 ค่าคือ ค่าคงที่ใช้ในขั้นตอนการประมาณค่าซึ่งมีผลทำให้ค่าประมาณที่ได้มีค่าใกล้เคียงค่าเชิงตัวเลขของรูปแบบแทนจำนวนมากน้อยเพียงไร ถ้ามีการกำหนดจำนวนของค่าคงที่เพิ่มมากขึ้นจะทำให้ค่าประมาณมีค่าใกล้เคียงค่าจริงมากขึ้น ทำให้สามารถหาผลลัพธ์ของการหารได้รวดเร็วยิ่งขึ้น เนื่องจากถ้าค่าประมาณมีค่าใกล้เคียงค่าจริงมากเท่าไหร่ ผลของการหารโดยการเลื่อนบิตจะได้ค่าผลลัพธ์ที่ใกล้เคียงความเป็นจริงมากขึ้นและทำให้ค่าเศษเหลือน้อยลง ซึ่งทำให้จำนวนรอบในการทำงานลดลง อีกค่าคือค่าคงที่ใช้สำหรับหยุดวนเวียนของอัลกอริทึม ถ้าค่าคงที่มีค่าน้อยจะทำให้ผลลัพธ์ของการหารมีความถูกต้องมากยิ่งขึ้น

5.2 ข้อเสนอแนะ

ข้อเสนอแนะสำหรับงานวิจัยนี้ คือ อัลกอริทึมการลบสามารถเพิ่มประสิทธิภาพการทำงานให้ดียิ่งขึ้นได้ โดยการนำวิธีการอื่นมาประยุกต์ใช้ เช่น การนำสถาปัตยกรรมออนเดอะฟลาย (on-the-fly) มาประยุกต์ใช้ โดยที่จะมีการคำนวณผลลัพธ์บางส่วนไว้ล่วงหน้าก่อนที่จะทำการลบจริงในหลักนั้นๆ นอกจากนี้การลบที่นำเสนอไปนั้นสามารถให้ผลลัพธ์ของการลบได้ในกรณีที่ตัวตั้งมีค่ามากกว่าตัวลบ ซึ่งอาจจะต้องมีการคำนึงถึงหรือสร้างรูปแบบแทนจำนวนของจำนวนลบสำหรับระบบฐานคู่

สำหรับอัลกอริทึมการหารนั้น ได้มีการใช้การประมาณค่า โดยที่การประมาณค่าได้กำหนดค่าคงที่ 3 ค่าที่อยู่ในช่วง (1,2] ซึ่งได้แก่ $2^{-3}3^{2}$, $2^{-1}3^{1}$ และ $2^{1}3^{0}$ ถ้ามีการเพิ่มค่าคงที่ในช่วงดังกล่าวจะทำให้การประมาณค่ามีความละเอียดมากขึ้นทำให้ค่าใกล้เคียงค่าจริงมากขึ้น ซึ่งส่งผลให้รอบการทำงานลดลง และถ้ามีการปรับเปลี่ยนค่าคงที่ในหารหยุดการวนรอบโดยให้ค่าที่น้อยลงจะทำให้ความละเอียดของคำตอบเพิ่มขึ้นแต่ต้องแลกกับเวลาในการคำนวณที่อาจจะเพิ่มมากขึ้นด้วย ซึ่งการเลือกค่าในจุดดังกล่าวขึ้นอยู่กับความเหมาะสมกับงานที่นำไปใช้ด้วย

การหารนั้นสามารถปรับปรุงให้มีประสิทธิภาพเพิ่มขึ้นได้ด้วยการนำวิธีการอื่นมาประยุกต์ใช้ เช่น การนำสถาปัตยกรรมออนเดอะฟลาย (on-the-fly) มาประยุกต์ใช้ ด้วยทำการคำนวณการหารโดยรับบิตตัวตั้งและบิตตัวหารเข้ามาคำนวณทีละบิต ซึ่งทำให้การหารสามารถดำเนินการได้โดยที่ไม่จำเป็นต้องรอให้ตัวตั้งและตัวหารมีค่าครบสมบูรณ์ ทำให้มีความเร็วเพิ่มขึ้นเมื่อนำการหารไปคำนวณร่วมกันกับตัวปฏิบัติการตัวอื่นรวมถึงตัวหารด้วย

อภิธานศัพท์

สัญลักษณ์ทางคณิตศาสตร์

i	เลขดัชนีของแถว (row index)
j	เลขดัชนีของคอลัมน์ (column index)
$d_{i,j}$	ดิจิตในตำแหน่งแถวที่ i คอลัมน์ที่ j (digit in the i^{th} row and j^{th} column)
$\sum_{all\,i,j} d_{i,j} 2^i 3^j$	รูปแบบแทนจำนวนฐานคู่ (double base number representation)
β	ฐาน (base)
$m \times n$	ขนาดของตารางแทนจำนวน (size of table representation)
$\square$	เซลล์ไม่แอ็กทิฟ ดิจิตศูนย์ (non-active cell, zero digit)
$\blacksquare$	แอ็กทิฟเซลล์ ดิจิตที่ไม่เป็นศูนย์ (active cell, non-zero digit)
$\boxtimes$	เซลล์ที่เกิดการซ้อนทับกันของตำแหน่งมีค่า (collision cell)

ตัวย่อ

ARDBNR	Addition Ready Double Base Number System
DBNR	Double Base Number Representation
DBNS	Double Base Number System

รายการอ้างอิง

[1] V. S. Dimitrov, G. A. Julliena, J. Eskritt, and W.C. Miller. The use of The multi-Dimensional Logarithmic Number System in DSP Applications. *In Proceeding of the 15th IEEE Symposium on Computer Arithmetic* (1998) 1998: 247-254.

[2] P. Prathummal and A. Surarerks. An Extended Double Dimensional Logarithmic Number System. *In Proceeding of the 11th National Computer Science and Engineering Conference (NCSEC2007)* 2007.

[3] V. S. Dimitrov, G. A. Jullien, and W. C. Miller. Theory and applications of the double-base number system. *Computers, IEEE Transactions* 1999: 1098-1106.

[4] V. Berthé, L. Imbert, and G. A. Jullien. More on Converting Numbers to the Double-Base Number System. Research report LIRMM-04031 Montpellier France October 2004.

[5] G. Gilbert and J. M. P. Langlois. Multipath greedy algorithm for canonical representation of numbers in double base number system. 2005.

[6] M. Pankaala, A. Paasio, and M. Laiho. Implementation alternatives of a DBNS adder. 2005.

[7] Y. Ibrahim, W. C. Miller, G. A. Jullien, and V. S. Dimitrov. DBNS addition using cellular neural networks. 2005.

[8] K. Wangjitman, A. Surarerks. Addition transducer for double base number system. *In Proceeding of the IEEE International Symposium on Communications and Information Technologies 2006(IEEE-ISCIT2006)* October 18-20,2006.

ประวัติผู้เขียนวิทยานิพนธ์

นายเอกพล มลฑลจุลเกศ เกิดเมื่อวันที่ 30 เมษายน พ.ศ. 2525 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียนธัญรัตน์ อำเภอธัญบุรี จังหวัดปทุมธานี เข้าศึกษาต่อในระดับปริญญาบัณฑิต สาขาวิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยธรรมศาสตร์ จนสำเร็จการศึกษาในปี พ.ศ. 2547 และในปี พ.ศ. 2550 ได้เข้าศึกษาต่อในระดับปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ปัจจุบันทำงานที่ DST Worldwide Service (Thailand) Co., Ltd.