แนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน

นายกิตติภพ พละการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2561

Alternative Redundant Residue Number System Construction with Redundant Residue Representations

Mr. Kittiphop Phalakarn

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2018

หัวข้อวิทยานิพนธ์	แนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน
โดย	นายกิตติภพ พละการ
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.เศรษฐา ปานงาม)

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

..... กรรมการภายนอกมหาวิทยาลัย
(รองศาสตราจารย์ ดร.อานนท์ รุ่งสว่าง)

กิตติภพ พละการ : แนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน. (Alternative Redundant Residue Number System Construction with Redundant Residue Representations) อ.ที่ปรึกษาหลัก : ผศ. ดร.อรรถสิทธิ์ สุรฤกษ์

ระบบจำนวนเศษเหลือ เป็นระบบการแทนจำนวนเต็มที่สามารถแทนจำนวนเต็มขนาดใหญ่ด้วยจำนวนเต็มที่มีค่าน้อยกว่าหลาย ๆ จำนวน การคำนวณผลบวกและผลคูณในระบบดังกล่าวสามารถทำได้อย่างรวดเร็ว ทำให้ระบบจำนวนเศษเหลือถูกนำมาใช้อย่างแพร่หลายในงานต่าง ๆ เช่น การประมวลผลสัญญาณ งานด้านการสื่อสารและเครือข่าย และการเข้ารหัสลับ เป็นต้น ระบบจำนวนเศษเหลือได้ถูกพัฒนาเป็นระบบจำนวนเศษเหลือซ้ำซ้อน ซึ่งสามารถตรวจจับและแก้ไขความผิดพลาดได้ ทำให้เหมาะกับงานที่ต้องการความสามารถในการทนต่อความผิดพร่อง ปัจจุบันมีแนวทางหลัก 2 แนวทางในการแปลงจากระบบจำนวนเศษเหลือให้เป็นระบบจำนวนเศษเหลือซ้ำซ้อน วิทยานิพนธ์นี้จะเสนอแนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน วิธีการที่นำเสนอนี้ทำให้การประมวลผลบางอย่างสามารถทำได้รวดเร็วขึ้น เช่น การแปลงจำนวนในรูปเศษเหลือกลับเป็นจำนวนเต็ม และการตรวจจับความผิดพลาด เป็นต้น นอกจากนี้ยังทำให้ระบบสามารถเปรียบเทียบค่าจำนวนเต็มในรูปของเศษเหลือได้รวดเร็วมากขึ้นด้วย อย่างไรก็ตามวิธีการที่นำเสนอทำให้ใช้เวลาในการคำนวณผลบวกและผลคูณมากขึ้น วิทยานิพนธ์นี้ได้ทำการเปรียบเทียบข้อดีและข้อจำกัดของวิธีการแปลงจากระบบจำนวนเศษเหลือให้เป็นระบบจำนวนเศษเหลือซ้ำซ้อนแบบต่าง ๆ เพื่อให้ผู้ที่สนใจสามารถนำไปพัฒนาระบบจำนวนเศษเหลือซ้ำซ้อนให้มีประสิทธิภาพมากยิ่งขึ้น

สาขาวิชา วิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต .....

ปีการศึกษา 2561 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6070125021 : MAJOR COMPUTER ENGINEERING

KEYWORD: Residue Number System (RNS), Redundant Residue Number System (RRNS), Error Detection and Correction Code

Kittiphop Phalakarn : Alternative Redundant Residue Number System Construction with Redundant Residue Representations. Advisor: Asst. Prof. Athasit Surarerks, Ph.D.

Residue number system (RNS) is a number representation system that represents a large integer with several smaller integers. Due to its ability to perform addition and multiplication in parallel, RNS is widely used in signal processing, communication, and cryptography. To extend the ability of RNS, redundant residue number system (RRNS), which has abilities to detect and correct errors, is proposed to be used in fault tolerant applications. Currently, there are two major ways to construct RRNS from RNS. This thesis proposes an alternative way to do the construction by using redundant residue representations. Our proposed RRNS can perform certain operations more efficiently, for example, backward conversion and error detection, and can also perform a complex RNS operation, namely, comparing the values between two RRNS representations. However, it would have more costs to perform addition and multiplication on our RRNS. We also compare our work to the two previous works, and discuss their advantages and drawbacks. Further investigations are required to improve the performance of the proposed RRNS.

Field of Study:	Computer Engineering	Student's Signature .....
Academic Year:	2018	Advisor's Signature .....

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ที่แนะนำแนวทาง และให้ความช่วยเหลือในการทำวิจัย จนทำให้วิทยานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี

ขอกราบขอบพระคุณ รองศาสตราจารย์ ดร.เศรษฐา ปานงาม และ รองศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง ที่ได้สละเวลามาเป็นคณะกรรมการสอบวิทยานิพนธ์ และได้กรุณาให้คำแนะนำต่าง ๆ เพื่อทำให้วิทยานิพนธ์นี้มีคุณภาพมากยิ่งขึ้น

ขอกราบขอบพระคุณ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่สนับสนุนทุนอุดหนุนการศึกษาระดับบัณฑิตศึกษาสำหรับนิสิตเก่าวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย

และสุดท้ายขอกราบขอบพระคุณ คณาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ และพี่ ๆ น้อง ๆ ในห้องปฏิบัติการทางวิศวกรรมระบบนับได้เชิงทฤษฎี ที่ได้ให้ความช่วยเหลือ ให้คำแนะนำ และให้กำลังใจในการทำวิทยานิพนธ์นี้

กิตติภพ พละการ

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค

บทคัดย่อภาษาอังกฤษ ..... ง

กิตติกรรมประกาศ ..... จ

สารบัญ ..... ฉ

สารบัญตาราง ..... ซ

สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 2

1.3 ขั้นตอนการดำเนินงานวิจัย 2

1.4 ขอบเขตการดำเนินงาน 3

1.5 ประโยชน์ที่ได้รับจากงานวิจัย 3

1.6 ผลงานวิจัยที่ได้รับการเผยแพร่ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 พื้นฐานทางคณิตศาสตร์ที่เกี่ยวข้อง 4

2.2 ระบบจำนวนเศษเหลือ 5

3.3 รหัสที่สามารถตรวจจับและแก้ไขความผิดพลาดได้ 8

บทที่ 3 งานวิจัยที่เกี่ยวข้อง 11

3.1 งานวิจัยของบาร์ซิ – เมสตรินิ 12

3.2 งานวิจัยของแคตติ 13

3.3 เปรียบเทียบข้อดีและข้อจำกัด 16

บทที่ 4 วิธีการดำเนินงาน 17
4.1 แนวคิดในการดำเนินงาน 17
4.2 แนวคิดในการแบ่งรูปแบบการแทนจำนวนเต็มด้วยเศษเหลือออกเป็นกลุ่มย่อย 18
4.3 การแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ 20
4.4 การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็ม 23
4.5 การบวก 24
4.6 การคูณ 26
4.7 การตรวจจับและแก้ไขความผิดพลาด 28
4.8 การเปรียบเทียบค่าในรูปเศษเหลือ 32
บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ 34
5.1 การประเมินระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ 34
5.2 การเปรียบเทียบจำนวนบิตที่ใช้ 35
5.3 ข้อเปรียบเทียบอื่น ๆ 36
5.4 สรุปผลการวิจัย และข้อเสนอแนะ 37
บรรณานุกรม 39
ประวัติผู้เขียน 41

สารบัญตาราง

หน้า

ตารางที่ 1 ข้อเปรียบเทียบของระบบจำนวนเศษเหลือ 3 ระบบ 16

ตารางที่ 2 ระบบจำนวนเศษเหลือซ้ำซ้อนที่นำเสนอ เมื่อกำหนดลำดับตัวหารเป็น (2, 3, 5) และกำหนดระยะทางแฮมมิงเป็น 2 22

ตารางที่ 3 การเปรียบเทียบจำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือ ของระบบจำนวนเศษเหลือทั้ง 4 ระบบ 36

ตารางที่ 4 ข้อเปรียบเทียบของระบบจำนวนเศษเหลือทั้ง 4 ระบบ 37

สารบัญรูป

หน้า

รูปที่ 1 การคำนวณจำนวนรูปเศษเหลือในแต่ละกลุ่ม และจำนวนกลุ่มที่ได้ ตามนิยาม 35 19

รูปที่ 2 ขั้นตอนวิธีในการแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ 20

รูปที่ 3 ขั้นตอนวิธีในการแปลงรูปเศษเหลือกลับเป็นจำนวนเต็ม ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ 23

รูปที่ 4 ขั้นตอนวิธีในการคำนวณผลบวกของรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ 25

รูปที่ 5 ขั้นตอนวิธีในการคำนวณผลคูณของรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ 27

รูปที่ 6 ขั้นตอนวิธีในการตรวจจับความผิดพลาด ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ..... 29

รูปที่ 7 ขั้นตอนวิธีในการตรวจจับความผิดพลาดฉบับปรับปรุง ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ 30

รูปที่ 8 ขั้นตอนวิธีในการเปรียบเทียบค่าในรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ. 33

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของปัญหา

ระบบจำนวนเศษเหลือ (residue number system – RNS) เป็นระบบการแทนจำนวนเต็มโดยใช้เศษเหลือจากการหารจำนวนเต็มนั้นด้วยตัวหารชุดหนึ่ง ระบบจำนวนเศษเหลือนี้มีข้อดีหลายประการ เช่น เศษเหลือที่ใช้แทนจำนวนเต็มจะมีค่าน้อยกว่าจำนวนเต็มนั้นมาก เมื่อนำมาใช้ในการคำนวณค่าต่าง ๆ จะทำให้การคำนวณทำได้รวดเร็ว โดยเฉพาะการบวกและการคูณ และการคำนวณค่าเศษเหลือของตัวหารตัวหนึ่งไม่ส่งผลกระทบกับการคำนวณของตัวหารตัวอื่น ทำให้สามารถใช้การคำนวณแบบขนานในระบบจำนวนเศษเหลือได้ ดังได้กล่าวไว้ใน [1] [2] เป็นต้น

จากข้อดีของระบบจำนวนเศษเหลือที่กล่าวมา ทำให้ระบบจำนวนเศษเหลือถูกนำมาใช้อย่างแพร่หลายในงานต่าง ๆ ที่เน้นการคำนวณการบวกและการคูณเป็นหลัก เช่น การประมวลผลสัญญาณ งานด้านการสื่อสารและเครือข่าย และการเข้ารหัสลับ เป็นต้น มีหลายการใช้งานที่ต้องการให้ระบบจำนวนเศษเหลือมีความสามารถในการทนต่อความผิดพร่อง (fault tolerant) ตัวอย่างเช่น ในการคำนวณบนระบบฝังตัว วงจรของระบบฝังตัวนั้นมีความซับซ้อนและยากต่อการตรวจสอบความถูกต้องของการทำงาน การตรวจจับและแก้ไขความผิดพลาดที่เกิดขึ้นจึงมีความสำคัญ ด้วยเหตุผลนี้จึงทำให้ระบบจำนวนเศษเหลือซ้ำซ้อน (redundant residue number system – RRNS) ถูกคิดขึ้น โดยระบบดังกล่าวมีความสามารถในการตรวจจับและแก้ไขความผิดพลาดที่เกิดขึ้นได้

บาร์ซิ (Barsi) และ เมสตรินิ (Maestrini) [3] ได้พัฒนาระบบจำนวนเศษเหลือซ้ำซ้อนขึ้น โดยพวกเขาใช้วิธีการเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบจำนวนเศษเหลือ ทำให้ระบบดังกล่าวมีสมบัติเป็นระบบจำนวนเศษเหลือซ้ำซ้อน มีงานวิจัยมากมายที่นำแนวคิดของ บาร์ซิ และ เมสตรินิ ไปพัฒนาต่อ [4] [5] [6] [7] ข้อจำกัดของวิธีการนี้คือ ต้องมีการเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ

นอกจากวิธีของ บาร์ซิ และ เมสตรินิ แล้ว ยังมีอีกวิธีหนึ่งที่สามารถแปลงระบบจำนวนเศษเหลือให้เป็นระบบจำนวนเศษเหลือซ้ำซ้อนได้ แคตติ (Katti) [8] ได้นำเสนอระบบจำนวนเศษเหลือที่ตัวหารไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กัน การใช้ตัวหารตามแบบของแคตติจะสามารถแปลงระบบจำนวนเศษเหลือให้มีสมบัติเป็นระบบจำนวนเศษเหลือซ้ำซ้อนได้ โดยไม่จำเป็นต้องมีการเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ ข้อจำกัดของตัวหารตามงานวิจัยของแคตติคือจะต้องใช้ตัวหารที่มีค่ามาก

การเพิ่มตัวหารและการใช้ตัวหารที่มีค่ามาก ซึ่งเป็นข้อจำกัดของการสร้างระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ – เมสตรินิและของแคตตินั้น จะทำให้วงจรของระบบมีขนาดใหญ่ขึ้น [1]

งานวิจัยนี้จึงต้องการที่จะนำเสนอแนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน ซึ่งคาดว่าวิธีการใหม่ที่เสนอนี้จะทำให้ข้อจำกัดที่ได้กล่าวไว้ข้างต้นหมดไปได้

1.2 วัตถุประสงค์

เพื่อนำเสนอแนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน ซึ่งในการเพิ่มความซ้ำซ้อนเข้าไปในระบบที่เสนอนี้ จะต้องไม่มีการเพิ่มตัวหาร และต้องใช้ตัวหารที่มีค่าไม่มาก การวัดประสิทธิภาพของระบบที่เสนอจะใช้การเปรียบเทียบกับงานวิจัยของบาร์ซิ – เมสตรินิ [3] และของแคตติ [8]

1.3 ขั้นตอนการดำเนินงานวิจัย

ขั้นตอนการดำเนินงานวิจัยประกอบด้วย 4 ข้อ ดังนี้

1.3.1 <u>การออกแบบวิธีการนำค่าเศษเหลือซ้ำซ้อนมาใช้กับระบบจำนวนเศษเหลือ</u> ขั้นตอนนี้จะต้องออกแบบว่า จะใช้ค่าเศษเหลือที่มีความซ้ำซ้อนค่าใด ในกรณีใดบ้าง และจะมีวิธีการนำค่าเศษเหลือที่มีความซ้ำซ้อนไปรวมกับระบบจำนวนเศษเหลืออย่างไร เพื่อให้ระบบมีสมบัติเป็นระบบจำนวนเศษเหลือซ้ำซ้อน

1.3.2 <u>การออกแบบขั้นตอนวิธีในการคำนวณบนระบบจำนวนเศษเหลือซ้ำซ้อนที่คิดขึ้น</u> ขั้นตอนนี้จะเป็นการออกแบบขั้นตอนวิธีต่าง ๆ ที่จำเป็นสำหรับการคำนวณบนระบบจำนวนเศษเหลือซ้ำซ้อน ซึ่งประกอบด้วย การแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็ม การบวก การคูณ และการตรวจจับและแก้ไขความผิดพลาด

1.3.3 <u>การพิสูจน์ขั้นตอนวิธีต่าง ๆ ที่เสนอ ว่าสามารถทำงานได้ถูกต้อง</u> ขั้นตอนนี้จะทำการพิสูจน์ขั้นตอนวิธีที่ได้จากขั้นตอนที่ 1.3.2 ว่าสามารถทำงานได้ถูกต้องและมีความสอดคล้องกัน

1.3.4 <u>การวัดประสิทธิภาพ เพื่อเปรียบเทียบกับระบบจำนวนเศษเหลือจากการทบทวนวรรณกรรม</u> ขั้นตอนนี้จะกำหนดข้อเปรียบเทียบต่าง ๆ เพื่อนำระบบจำนวนเศษเหลือซ้ำซ้อนที่คิดขึ้น มาเปรียบเทียบกับระบบจำนวนเศษเหลือแบบเดิม ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ – เมสตรินิ และระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ

1.4 ขอบเขตการดำเนินงาน

ขอบเขตการดำเนินงานประกอบด้วย 2 ข้อ ดังนี้

1.4.1 ระบบจำนวนเศษเหลือที่พิจารณาในงานวิจัยนี้จะรองรับการแทนช่วงจำนวนเต็มที่ไม่ติดลบและเริ่มต้นด้วย 0 เท่านั้น

1.4.2 การดำเนินการคำนวณที่พิจารณา ประกอบด้วย การแปลงจากจำนวนเต็มให้อยู่ในรูปเศษเหลือ การแปลงจากรูปเศษเหลือให้กลับเป็นจำนวนเต็ม การบวก การคูณ และการตรวจจับและแก้ไขความผิดพลาด

1.5 ประโยชน์ที่ได้รับจากงานวิจัย

ประโยชน์ที่ได้รับจากงานวิจัยนี้มี 3 ข้อ ได้แก่

1.5.1 ได้ขั้นตอนวิธีในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน ซึ่งในการเพิ่มความซ้ำซ้อนเข้าไปในระบบที่เสนอนี้ จะใช้ตัวหารที่มีค่าไม่มากและไม่มีการเพิ่มตัวหาร

1.5.2 ผู้ที่ต้องการนำระบบจำนวนเศษเหลือซ้ำซ้อนไปใช้งาน มีทางเลือกมากขึ้นในการเลือกระบบจำนวนเศษเหลือซ้ำซ้อนให้เหมาะสมกับความต้องการของผู้ใช้

1.5.3 งานวิจัยนี้จะเป็นแนวทางในการพัฒนาระบบจำนวนเศษเหลือซ้ำซ้อนอื่น ๆ และระบบจำนวนอื่น ๆ ต่อไป

1.6 ผลงานวิจัยที่ได้รับการเผยแพร่

Phalakarn, K. and A. Surarerks, *Alternative redundant residue number system construction with redundant residue representations*. in *2018 3rd International Conference on Computer and Communication Systems (ICCCS)*. 2018. IEEE.

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในบทนี้ จะกล่าวถึงนิยามและทฤษฎีต่าง ๆ ที่เกี่ยวข้องกับระบบจำนวนเศษเหลือ และรหัสที่สามารถตรวจจับและแก้ไขความผิดพลาดได้ ซึ่งจะใช้ในการอธิบายงานวิจัยของบาร์ซิ – เมสตรินิ และของแคตติในบทถัดไป

2.1 พื้นฐานทางคณิตศาสตร์ที่เกี่ยวข้อง

พื้นฐานทางคณิตศาสตร์ที่เกี่ยวข้องกับงานวิจัยนี้จะอยู่ในหัวข้อทฤษฎีจำนวนเรื่องการหารและเศษจากการหาร เพื่อให้เข้าใจได้ตรงกัน จึงขอกำหนดนิยามและสัญลักษณ์ต่าง ๆ ดังนี้

นิยาม 1 กำหนดให้ a เป็นจำนวนจริงใด ๆ สัญลักษณ์ $\lfloor a \rfloor$ แทน จำนวนเต็มที่มีค่ามากที่สุดที่น้อยกว่าหรือเท่ากับ a

นิยาม 2 (เศษจากการหาร) กำหนดให้ a เป็นจำนวนเต็ม และ m เป็นจำนวนเต็มบวก เศษจากการหาร a ด้วย m ซึ่งแทนด้วยสัญลักษณ์ $|a|_m$ มีค่าเท่ากับ $a - (\lfloor a / m \rfloor \times m)$

นิยาม 3 (ความสัมพันธ์สมภาค) กำหนดให้ a และ b เป็นจำนวนเต็ม และ m เป็นจำนวนเต็มบวก a และ b จะมีความสัมพันธ์สมภาค (congruence) บนตัวหาร m ซึ่งแทนด้วยสัญลักษณ์ $a \equiv b \pmod{m}$ ก็ต่อเมื่อ $|a|_m = |b|_m$

ตัวอย่าง 4 เศษจากการหารจำนวนเต็ม 33 และ 18 ด้วย 5 มีค่าเท่ากับ $|33|_5$ และ $|18|_5$ ซึ่งทั้งคู่มีค่าเท่ากับ 3 จำนวนเต็มทั้งสองมีความสัมพันธ์สมภาคบนตัวหาร 5 ซึ่งแทนด้วยสัญลักษณ์ $33 \equiv 18 \pmod{5}$ เนื่องจาก $|33|_5 = |18|_5 = 3$ □

นิยาม 5 (ตัวหารร่วมมาก) กำหนดให้ a และ b เป็นจำนวนเต็มที่ไม่ติดลบ จำนวนเต็มบวก m จะเป็นตัวหารร่วมมาก (greatest common divisor – GCD) ของ a และ b ซึ่งแทนด้วยสัญลักษณ์ $m = GCD(a, b)$ ก็ต่อเมื่อ m เป็นจำนวนเต็มบวกที่มีค่ามากที่สุดที่ $|a|_m = |b|_m = 0$

นิยาม 6 (ตัวคูณร่วมน้อย) กำหนดให้ a และ b เป็นจำนวนเต็มที่ไม่ติดลบ จำนวนเต็มบวก m จะเป็นตัวคูณร่วมน้อย (least common multiple – LCM) ของ a และ b ซึ่งแทนด้วยสัญลักษณ์ m = LCM (a, b) ก็ต่อเมื่อ m เป็นจำนวนเต็มบวกที่มีค่าน้อยที่สุดที่ $|m|_a = |m|_b = 0$

ตัวอย่าง 7 ตัวหารร่วมมากของ 12 และ 15 เขียนแทนด้วย GCD (12, 15) มีค่าเท่ากับ 3 เนื่องจาก 3 เป็นจำนวนเต็มบวกที่มีค่ามากที่สุดที่ทำให้สมการ $|12|_m = |15|_m = 0$ เป็นจริง

ตัวคูณร่วมน้อยของ 12 และ 15 เขียนแทนด้วย LCM (12, 15) มีค่าเท่ากับ 60 เนื่องจาก 60 เป็นจำนวนเต็มบวกที่มีค่าน้อยที่สุดที่ทำให้สมการ $|m|_{12} = |m|_{15} = 0$ เป็นจริง □

นิยาม 8 (จำนวนเฉพาะสัมพัทธ์) กำหนดให้ a และ b เป็นจำนวนเต็มบวก a และ b เป็นจำนวนเฉพาะสัมพัทธ์กัน ก็ต่อเมื่อ GCD (a, b) = 1

ตัวอย่าง 9 จำนวนเต็ม 9, 10 และ 11 เป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด เนื่องจาก GCD (9, 10) = GCD (9, 11) = GCD (10, 11) = 1 แต่จำนวนเต็ม 7 และ 14 ไม่เป็นจำนวนเฉพาะสัมพัทธ์กัน เนื่องจาก GCD (7, 14) = 7 □

นิยาม 10 (สัญลักษณ์ของผลบวกและสัญลักษณ์ของผลคูณ) สัญลักษณ์ของผลบวกสามารถเขียนได้ว่า $\sum_{(i=1,\ldots,k)} a_i = a_1 + a_2 + \ldots + a_k$ และสัญลักษณ์ของผลคูณสามารถเขียนได้ว่า $\prod_{(i=1,\ldots,k)} a_i = a_1 \times a_2 \times \ldots \times a_k$

2.2 ระบบจำนวนเศษเหลือ

ระบบจำนวนเศษเหลือ (residue number system – RNS) เป็นระบบการแทนจำนวนเต็มโดยใช้เศษเหลือจากการหารจำนวนเต็มนั้นด้วยตัวหารชุดหนึ่ง ต่อไปนี้จะเป็นนิยามและทฤษฎีบทต่าง ๆ ที่เกี่ยวข้อง รายละเอียดเพิ่มเติมสามารถดูได้ที่ [1] [2]

นิยาม 11 (ระบบจำนวนเศษเหลือ) ระบบจำนวนเศษเหลือจะถูกกำหนดด้วยลำดับของตัวหาร (moduli sequence) และช่วงของจำนวนเต็มที่ต้องการแทน

ลำดับของตัวหาร คือลำดับ $(m_1, m_2, \ldots, m_k)$ ซึ่งประกอบด้วยจำนวนเต็มบวก k จำนวน ทุกจำนวนเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด

ระบบจำนวนเศษเหลือนี้จะสามารถแทนจำนวนเต็มได้ $M_k = \prod_{(i=1,\ldots,k)} m_i$ จำนวน ความหมายว่าช่วงจำนวนเต็มที่ระบบสามารถแทนได้คือ N ถึง $N + M_k - 1$ โดยในงานนี้จะใช้ $N = 0$

ระบบจำนวนเศษเหลือนี้จะแทนจำนวนเต็ม X ด้วยลำดับของเศษจากการหาร $(x_1, x_2, \ldots, x_k) = (|X|_{m1}, |X|_{m2}, \ldots, |X|_{mk})$ ซึ่งเกิดจากการนำจำนวนเต็ม X หารด้วยตัวหารแต่ละตัวในลำดับของตัวหาร รูปแบบการแทนจำนวนเต็มทั้งหมดจะมี M_k รูปแบบที่แตกต่างกัน

ตัวอย่าง 12 กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหารเป็น (2, 3, 5, 7) ระบบจำนวนเศษเหลือนี้จะสามารถแทนจำนวนเต็มได้ $2 \times 3 \times 5 \times 7 = 210$ จำนวน ซึ่งจะใช้แทนจำนวนเต็ม 0 ถึง 209 จำนวนเต็ม 117 สามารถแทนด้วยลำดับของเศษจากการหาร $(|117|_2, |117|_3, |117|_5, |117|_7) = (1, 0, 2, 5)$ □

จะเห็นได้ว่าระบบจำนวนเศษเหลือมีความแตกต่างกับระบบจำนวนที่อ้างอิงตำแหน่ง (positional number system) เช่น ระบบเลขฐาน (radix/base) เนื่องจากระบบจำนวนเศษเหลือไม่มีการใช้ค่าประจำตำแหน่ง และค่าเศษเหลือที่มีค่ามากไม่ได้หมายความว่าจำนวนเต็มนั้นจะมีค่ามากตามไปด้วย ด้วยสาเหตุนี้ทำให้การเปรียบเทียบความมากกว่าน้อยกว่าบนระบบจำนวนเศษเหลือทำได้ไม่สะดวก

การคำนวณบนระบบจำนวนเศษเหลือจะเกิดขึ้นแยกกันระหว่างเศษเหลือของแต่ละตัวหาร ซึ่งเป็นผลมาจากสมบัติของความสัมพันธ์สมภาค ทฤษฎีบท 13 จะนำไปสู่วิธีการบวก ลบ และคูณบนระบบจำนวนเศษเหลือในทฤษฎีบท 14 สำหรับการดำเนินการหารด้วยทฤษฎีบท 13 และทฤษฎีบท 14 นั้น จะต้องมีการหาตัวผกผันการคูณก่อน และจะได้ผลหารถูกต้องเมื่อเป็นการหารลงตัวเท่านั้น จึงทำให้การหารบนระบบจำนวนเศษเหลือทำได้ไม่สะดวก

ทฤษฎีบท 13 กำหนด a, b, c และ d เป็นจำนวนเต็ม และ m เป็นจำนวนเต็มบวก ถ้า $a \equiv b \pmod{m}$ และ $c \equiv d \pmod{m}$ จะได้ว่า $a \bullet c \equiv b \bullet d \pmod{m}$ เมื่อ ● แทนการดำเนินการบวก ลบ หรือคูณ

ทฤษฎีบท 14 กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร $(m_1, m_2, \ldots, m_k)$ โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และจำนวนเต็ม X และ Y ซึ่งแสดงในระบบจำนวนเศษเหลือนี้ได้เป็น $(x_1, x_2, \ldots, x_k)$ และ $(y_1, y_2, \ldots, y_k)$ ตามลำดับ จะได้ว่า $|X \bullet Y|_{Mk}$ สามารถแทนในรูปเศษเหลือได้เป็น $(|x_1 \bullet y_1|_{m1}, |x_2 \bullet y_2|_{m2}, \ldots, |x_k \bullet y_k|_{mk})$ เมื่อ ● แทนการดำเนินการบวก ลบ หรือคูณ

ตัวอย่าง 15 กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับตัวหารเป็น (11, 13, 17, 19) และจำนวนเต็ม 179 กับ 203 ที่มีการแทนในรูปเศษเหลือเป็น (3, 10, 9, 8) และ (5, 8, 16, 13) ตามลำดับ การคำนวณผลบวกและผลคูณในรูปเศษเหลือสามารถคำนวณได้เป็น ($|3 + 5|_{11}$, $|10 + 8|_{13}$, $|9 + 16|_{17}$, $|8 + 13|_{19}$) = (8, 5, 8, 2) และ ($|3 \times 5|_{11}$, $|10 \times 8|_{13}$, $|9 \times 16|_{17}$, $|8 \times 13|_{19}$) = (4, 2, 8, 9) ผลลัพธ์จากการดำเนินการตรงกับรูปเศษเหลือของจำนวนเต็ม 382 และ 36337 ซึ่งเป็นผลบวกและผลคูณของ 179 และ 203 ตามลำดับ □

จากตัวอย่าง 15 จะเห็นได้ว่า ค่าเศษเหลือในแต่ละตัวหารมีค่าน้อยเมื่อเทียบกับจำนวนเต็มตั้งต้น การคำนวณสามารถทำได้รวดเร็ว เกิดขึ้นแยกกันระหว่างตัวหาร ไม่มีการทดค่าจากตัวหารหนึ่งไปอีกตัวหารหนึ่ง และทุกตัวหารสามารถคำนวณได้พร้อมกัน ข้อดีต่าง ๆ เหล่านี้ทำให้ระบบจำนวนเศษเหลือถูกนำไปใช้ในการประมวลผลสัญญาณ งานด้านการสื่อสารและเครือข่าย และการเข้ารหัสลับ ซึ่งการคำนวณส่วนใหญ่เป็นการบวก ลบ และคูณ

นอกจากการดำเนินการบวก ลบ และคูณแล้ว อีกหนึ่งการดำเนินการที่สำคัญในระบบจำนวนเศษเหลือคือการแปลงจำนวนในรูปเศษเหลือกลับเป็นจำนวนเต็ม ทฤษฎีบทที่เป็นที่รู้จักกันดีที่ใช้ในการแปลงคือ ทฤษฎีบทเศษเหลือของจีน (Chinese remainder theorem – CRT) นอกจากนี้ยังมีวิธีการอื่น ๆ เช่น การแปลงฐานผสม (mixed radix conversion – MRC) [9] เป็นต้น แต่จะไม่ลงรายละเอียดวิธีอื่น ๆ ในที่นี้

ทฤษฎีบท 16 (ทฤษฎีบทเศษเหลือของจีน) กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร (m_1, m_2, ..., m_k) โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และจำนวนเต็มซึ่งแสดงในระบบจำนวนเศษเหลือ (x_1, x_2, ..., x_k) เราสามารถแปลงจำนวนในรูปแบบจำนวนเศษเหลือกลับเป็นจำนวนเต็มได้โดยใช้สมการดังต่อไปนี้

$$X = \left| \sum_{(i=1,\ldots,k)} \left(a_i \frac{M_k}{m_i} x_i \right) \right|_{M_k}$$

โดย a_i เป็นตัวผกผันการคูณของ (M_k / m_i) บนตัวหาร m_i นั่นคือ a_i เป็นจำนวนเต็มที่ a_i (M_k / m_i) ≡ 1 (mod m_i)

ตัวอย่าง 17 กำหนดระบบจำนวนเศษเหลือที่มีลำดับของตัวหารเป็น (2, 3, 5, 7) และจำนวนที่แสดงในรูปเศษเหลือในระบบนี้ได้เป็น (1, 0, 2, 5) เราสามารถคำนวณค่าตัวผกผันการคูณทั้งสี่ค่าได้เป็น a_1 = 1, a_2 = 1, a_3 = 3 และ a_4 = 4 ดังนั้นจำนวนเต็มที่สอดคล้องกับรูปเศษเหลือ (1, 0, 2, 5) คือ 117 จากการคำนวณตามทฤษฎีบทเศษเหลือของจีน □

3.3 รหัสที่สามารถตรวจจับและแก้ไขความผิดพลาดได้

หัวข้อนี้จะนำเสนอความรู้จากทฤษฎีรหัส (coding theory) เพื่อใช้ในการอธิบายรหัสที่สามารถตรวจจับและแก้ไขความผิดพลาดได้ (error detection and correction code) นิยามคำสำคัญต่าง ๆ มีดังนี้ รายละเอียดเพิ่มเติมสามารถดูได้ที่ [10]

นิยาม 18 (อักษรรหัส และ รหัส) กำหนดเซตของจำนวนเต็ม A และจำนวนเต็มบวก n สมาชิกในเซต A^n เรียกว่าอักษรรหัส (codeword) ที่มีความยาว n

รหัส (code) ที่มีความยาว n เป็นเซตย่อย (subset) ของ A^n การเลือกว่าอักษรรหัสใดจะเป็นสมาชิกของรหัส จะขึ้นอยู่กับการนำรหัสนั้นไปใช้งาน

นิยาม 19 (ระยะทางแฮมมิง) เมื่อพิจารณาอักษรรหัสที่มีความยาวเท่ากัน $(x_1, x_2, ..., x_k)$ และ $(y_1, y_2, ..., y_k)$ ระยะทางแฮมมิง (Hamming distance) ระหว่างอักษรรหัสทั้งสอง มีค่าเท่ากับจำนวนตำแหน่งที่ $x_i \neq y_i$ สำหรับทุก $1 \leq i \leq k$

ในการคำนวณระยะทางแฮมมิงของรหัส ให้คำนวณระยะทางแฮมมิงระหว่างอักษรรหัสทุกคู่ที่อยู่ในรหัสนั้น ระยะทางแฮมมิงที่น้อยที่สุดจากการคำนวณนี้จะเป็นระยะทางแฮมมิงของรหัส

ตัวอย่าง 20 กำหนดเซตของจำนวนเต็ม A = {0, ..., 5} ตัวอย่างของอักษรรหัสที่มีความยาว 4 ซึ่งเป็นสมาชิกของ A^4 เช่น $c_1 = (1, 0, 2, 5)$, $c_2 = (3, 0, 0, 4)$ และ $c_3 = (2, 3, 0, 2)$ เป็นต้น และกำหนดรหัส $C = \{c_1, c_2, c_3\}$ ซึ่งเป็นเซตย่อยของ A^4 ระยะทางแฮมมิงระหว่าง c_1 กับ c_2 มีค่าเท่ากับ 3, ระหว่าง c_1 กับ c_3 มีค่าเท่ากับ 4 และระหว่าง c_2 กับ c_3 มีค่าเท่ากับ 3 ดังนั้นระยะทางแฮมมิงของรหัส C จึงมีค่าเท่ากับ 3 ซึ่งเป็นระยะทางแฮมมิงที่น้อยที่สุดจากการคำนวณระยะทางแฮมมิงระหว่างอักษรรหัสทุกคู่ในรหัส C □

นิยามและทฤษฎีบทถัดไป จะแสดงความสัมพันธ์ระหว่างระยะทางแฮมมิงของรหัสกับสมบัติการตรวจจับและแก้ไขความผิดพลาดของรหัสนั้น

นิยาม 21 กำหนดอักษรรหัส $c = (c_1, c_2, ..., c_k)$ ความผิดพลาด p ตำแหน่งเป็นการบวกอักษรรหัสดังกล่าวกับความผิดพลาด $e = (..., 0, ..., e_1, ..., 0, ..., e_p, ..., 0, ...)$ ซึ่งความผิดพลาด e นี้จะมี p ตำแหน่งที่เป็นจำนวนเต็มที่มีค่าไม่เท่ากับ 0 และตำแหน่งอื่น ๆ มีค่าเป็น 0 ผลลัพธ์จากการบวกจะเป็นอักษรรหัสที่มีความผิดพลาด $c' = c + e$ การบวกที่ใช้อาจแตกต่างกันตามการนำรหัสไปใช้งาน

ทฤษฎีบท 22 [10] กำหนดรหัส C ที่มีค่าระยะทางแฮมมิงเท่ากับ d และอักษรรหัส c ซึ่งเป็นสมาชิกของ C หากมีความผิดพลาดเกิดขึ้นกับอักษรรหัส c ไม่เกิน d – 1 ตำแหน่ง จากความผิดพลาด e อักษรรหัสที่มีความผิดพลาด c' = c + e จะไม่เป็นสมาชิกของ C นั่นคือระบบจะสามารถตรวจจับความผิดพลาดได้ หากมีความผิดพลาดเกิดขึ้นกับอักษรรหัสไม่เกิน d – 1 ตำแหน่ง

นอกจากนี้ ระบบยังสามารถแก้ไขความผิดพลาดได้ หากมีความผิดพลาดเกิดขึ้นกับอักษรรหัสไม่เกิน $\lfloor (d-1)/2 \rfloor$ ตำแหน่ง ในการแก้ไขความผิดพลาดของอักษรรหัส c' ให้แก้ไขเป็นอักษรรหัส c ซึ่งเป็นสมาชิกของรหัส C และมีระยะทางแฮมมิงระหว่าง c' กับ c น้อยที่สุด อักษรรหัส c ที่เป็นไปตามเงื่อนไขนี้จะมีเพียงแบบเดียวเท่านั้น หลักในการแก้ไขนี้เรียกว่าหลักการความควรจะเป็นสูงสุด (maximum likelihood principle)

ตัวอย่าง 23 กำหนดอักษรรหัส $c_1 = (1, 0, 2, 5)$, $c_2 = (3, 0, 0, 4)$, $c_3 = (2, 3, 0, 2)$ และรหัส $C = \{c_1, c_2, c_3\}$ ซึ่งมีค่าระยะทางแฮมมิงเท่ากับ 3 นั่นคือเราสามารถตรวจจับความผิดพลาดได้ หากมีความผิดพลาดเกิดขึ้นกับอักษรรหัสไม่เกิน 3 – 1 = 2 ตำแหน่ง เช่น หากได้รับอักษรรหัส $c' = c_2 + (1, 0, 0, 0) = (4, 0, 0, 4)$ และ $c'' = c_3 + (0, 0, 2, 3) = (2, 3, 2, 5)$ เราจะทราบว่ามีความผิดพลาดเกิดขึ้น เนื่องจาก c' และ c" ไม่เป็นสมาชิกของรหัส C

เราสามารถแก้ไขความผิดพลาดได้ หากมีความผิดพลาดเกิดขึ้นกับอักษรรหัสไม่เกิน $\lfloor (3-1)/2 \rfloor = 1$ ตำแหน่ง หากได้รับอักษรรหัส $c' = (4, 0, 0, 4)$ เราจะทราบว่ามีความผิดพลาดเกิดขึ้น เนื่องจาก c' ไม่เป็นสมาชิกของรหัส C ในการแก้ไขความผิดพลาด ให้แก้ไขเป็นอักษรรหัส (3, 0, 0, 4) เนื่องจากเป็นอักษรรหัสที่เป็นสมาชิกของรหัส C และมีระยะทางแฮมมิงจากอักษรรหัส c' น้อยที่สุด

หากได้รับอักษรรหัส c" = (2, 3, 2, 5) เราจะทราบว่ามีความผิดพลาดเกิดขึ้น เนื่องจาก c" ไม่เป็นสมาชิกของรหัส C แต่เนื่องจาก c" เกิดจากความผิดพลาด 2 ตำแหน่ง ในการแก้ไขความผิดพลาด จะพบว่า c" มีระยะห่างจาก c_1 และ c_3 น้อยที่สุดเท่ากัน ในกรณีนี้จึงสามารถตรวจจับความผิดพลาดได้ แต่ไม่สามารถแก้ไขความผิดพลาดได้ □

ระบบจำนวนเศษเหลือสามารถมองในมุมของทฤษฎีรหัสได้ โดยการแทนจำนวนเต็มหนึ่งจำนวนในรูปเศษเหลือเทียบเท่าได้กับอักษรรหัส และเซตของการแทนจำนวนเต็มที่เป็นไปได้ทั้งหมดเทียบเท่าได้กับรหัส

ตัวอย่าง 24 กำหนดให้เซต A เป็นเซตของตัวเลข {0, 1, ..., 6} ตัวอย่างของอักษรรหัสที่มีความยาวเท่ากับ 4 เช่น $c_1 = (1, 0, 2, 5)$ และ $c_2 = (3, 2, 6, 4)$ เมื่อพิจารณารหัส C สำหรับระบบจำนวน

เศษเหลือที่ใช้ลำดับตัวหารเป็น (2, 3, 5, 7) จะได้ว่าอักษรรหัส c_1 อยู่ในรหัส C แต่อักษรรหัส c_2 ไม่อยู่ในรหัส C จากนิยาม 11 จะได้ว่ารหัส C ประกอบด้วยอักษรรหัสทั้งสิ้น 210 อักษรรหัส □

บทที่ 3
งานวิจัยที่เกี่ยวข้อง

ในบทนี้ จะกล่าวถึงงานวิจัยเกี่ยวกับระบบจำนวนเศษเหลือซ้ำซ้อน (redundant residue number system – RRNS) ระบบจำนวนเศษเหลือซ้ำซ้อนเป็นระบบจำนวนเศษเหลือที่มีลักษณะพิเศษ คำอธิบายกว้าง ๆ ของระบบจำนวนเศษเหลือซ้ำซ้อนเป็นดังนี้

ระบบจำนวนเศษเหลือซ้ำซ้อนเป็นระบบจำนวนเศษเหลือประเภทหนึ่ง ซึ่งจากรูปแบบทั้งหมดในการแทนจำนวนเต็มด้วยเศษเหลือ จะมีการแทนบางแบบที่ไม่ถูกนำมาใช้ ผลของการไม่ใช้รูปเศษเหลือบางแบบ เมื่อมองในมุมของทฤษฎีรหัส จะเป็นการลดขนาดเซตรหัสลง ซึ่งเมื่อคำนวณระยะทางแฮมมิงของระบบ จะทำให้ค่าระยะทางแฮมมิงเพิ่มมากขึ้น และทำให้ระบบมีความสามารถในการตรวจจับและแก้ไขความผิดพลาด

คำว่า “ซ้ำซ้อน (redundant)” ในที่นี้ ไม่ได้หมายความว่า จำนวนเต็มหนึ่งจำนวนจะมีการแทนในรูปเศษเหลือได้มากกว่าหนึ่งแบบ แต่หมายถึง จากรูปแบบการแทนที่เป็นไปได้ทั้งหมดจะมีการแทนบางรูปแบบที่ไม่ถูกนำมาใช้งาน

ตัวอย่าง 25 กำหนดระบบจำนวนเศษเหลือที่ใช้ลำดับตัวหาร (2, 3, 5, 7) โดยปกติแล้วระบบนี้จะสามารถแทนจำนวนเต็มได้ 210 จำนวน และระยะทางแฮมมิงของระบบจะมีค่าเท่ากับ 1 หากนำระบบนี้มาแทนจำนวนเต็มเพียงสี่จำนวน คือ 0 ถึง 3 ซึ่งเขียนในรูปเศษเหลือได้ว่า (0, 0, 0, 0), (1, 1, 1, 1), (0, 2, 2, 2) และ (1, 0, 3, 3) ระบบจำนวนเศษเหลือซ้ำซ้อนนี้จะมีระยะทางแฮมมิงเป็น 3 ซึ่งจะทำให้สามารถตรวจจับความผิดพลาดได้ เมื่อมีความผิดพลาดเกิดขึ้นไม่เกิน 2 ตำแหน่งและสามารถแก้ไขความผิดพลาดได้เมื่อมีความผิดพลาดเกิดขึ้นไม่เกิน 1 ตำแหน่ง □

จากตัวอย่าง 25 จะพบปัญหา เช่น จะใช้ระบบจำนวนเศษเหลือซ้ำซ้อนแทนช่วงจำนวนเต็มใดจึงจะเหมาะสม และระยะทางแฮมมิงที่ได้มีค่าเป็นอย่างไร จากการทบทวนวรรณกรรม พบว่ามีงานวิจัยสองงาน ได้แก่งานวิจัยของบาร์ซิ – เมสตรินิ [3] และงานวิจัยของแคตติ [8] ที่เสนอแนวทางการแปลงระบบจำนวนเศษเหลือให้เป็นระบบจำนวนเศษเหลือซ้ำซ้อน โดยงานวิจัยทั้งสองงานนี้ได้พิสูจน์สมบัติของระบบจำนวนเศษเหลือซ้ำซ้อนไว้อย่างชัดเจน

3.1 งานวิจัยของบาร์ซิ – เมสตรินิ

ก่อนหน้างานวิจัยของบาร์ซิและเมสตรินิ มีงานวิจัยที่ได้เสนอแนวคิดเกี่ยวกับระบบจำนวนเศษเหลือซ้ำซ้อนไว้อยู่แล้ว แต่งานวิจัยของบาร์ซิและเมสตรินิเป็นงานวิจัยแรกที่เสนอระบบจำนวนเศษเหลือซ้ำซ้อนอย่างเป็นระบบ และมีการพิสูจน์สมบัติการตรวจจับและแก้ไขความผิดพลาดไว้อย่างชัดเจน งานวิจัยนี้ได้เสนอให้เพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ โดยไม่เพิ่มช่วงการแทนจำนวนเต็มทั้งหมดที่เป็นไปได้ ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิและเมสตรินิเป็นดังนี้

นิยาม 26 (ระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ลำดับตัวหารซ้ำซ้อน [3]) ระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ลำดับตัวหารซ้ำซ้อน จะมีลำดับตัวหารสองชุด คือ ลำดับตัวหารข้อมูล (information moduli sequence) (m_1, m_2, ..., m_k) ซึ่งมีการใช้งานตามระบบจำนวนเศษเหลือปกติ และลำดับตัวหารซ้ำซ้อน (redundant moduli sequence) (m_{k+1}, m_{k+2}, ..., m_{k+r}) ซึ่งจะเพิ่มความซ้ำซ้อนให้กับระบบ เงื่อนไขในการเลือกลำดับตัวหารคือ ตัวหารทุกตัวจากทั้งสองลำดับต้องเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และตัวหารทุกตัวในลำดับตัวหารซ้ำซ้อนต้องมีค่ามากกว่าตัวหารในลำดับตัวหารข้อมูล

ระบบจำนวนเศษเหลือโดยทั่วไปที่ใช้ลำดับตัวหาร (m_1, ..., m_k, m_{k+1}, ..., m_{k+r}) จะสามารถแทนจำนวนเต็มได้ทั้งหมด $M_n = M_k \times M_r = \prod_{(i=1,...,k)} m_i \times \prod_{(j=k+1,...,k+r)} m_j$ จำนวน สำหรับระบบจำนวนเศษเหลือซ้ำซ้อน จะใช้ลำดับตัวหารเดียวกันนี้ในการแทนจำนวนเต็มเพียง $M_k = \prod_{(i=1,...,k)} m_i$ จำนวนเท่านั้น โดยช่วงจำนวนเต็มที่ระบบสามารถแทนได้ $[0, M_k)$ จะเรียกว่าช่วงที่เป็นไปตามกฎ (legitimate range) และช่วงจำนวนเต็มที่ไม่นำมาใช้ $[M_k, M_n)$ จะเรียกว่าช่วงที่ไม่เป็นไปตามกฎ (illegitimate range)

ทฤษฎีบทต่อไปนี้ เป็นทฤษฎีบทสำคัญสำหรับระบบจำนวนเศษเหลือซ้ำซ้อนประเภทนี้ มีที่มาจากทฤษฎีบท 22 บทพิสูจน์ของทฤษฎีบทนี้สามารถดูได้ใน [3]

ทฤษฎีบท 27 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อน ที่ใช้ลำดับตัวหาร (m_1, ..., m_k, m_{k+1}, ..., m_{k+r}) ตามนิยาม 26 ตัวหารซ้ำซ้อนจะเพิ่มระยะทางแฮมมิงของระบบให้มีค่าเท่ากับ r + 1

กำหนดให้จำนวนเต็ม X อยู่ในช่วงที่เป็นไปตามกฎ $[0, M_k)$ ถ้ามีความผิดพลาดเกิดขึ้นกับรูปเศษเหลือของ X อย่างน้อย 1 ตำแหน่ง แต่ไม่เกิน r ตำแหน่ง เมื่อแปลงรูปเศษเหลือของ X ที่มีข้อผิดพลาดกลับเป็นจำนวนเต็ม จะได้ค่าอยู่ในช่วงที่ไม่เป็นไปตามกฎ $[M_k, M_n)$

การแทนจำนวนเต็มในรูปเศษเหลือ การดำเนินการบวก การคูณ และการแปลงจากรูปเศษเหลือกลับเป็นจำนวนเต็มบนระบบจำนวนเศษเหลือซ้ำซ้อนประเภทนี้ สามารถทำได้ในลักษณะเดียวกันกับระบบจำนวนเศษเหลือแบบปกติ การตรวจจับความผิดพลาดสามารถทำได้โดยตรวจสอบว่าจำนวนเต็มอยู่ในช่วงที่เป็นไปตามกฎหรือไม่ และการแก้ไขความผิดพลาดสามารถทำได้โดยใช้หลักการความควรจะเป็นสูงสุด มีงานวิจัยต่าง ๆ พยายามปรับปรุงวิธีการตรวจจับและแก้ไขความผิดพลาดให้มีประสิทธิภาพดีขึ้น [4] [5] [6] [7] แต่จะไม่ลงรายละเอียดในที่นี้

ตัวอย่าง 28 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อน โดยมีลำดับตัวหารข้อมูล (5, 7, 8, 9) และลำดับตัวหารซ้ำซ้อน (11, 13) ในกรณีระบบจำนวนเศษเหลือปกติ ระบบนี้ควรจะแทนจำนวนเต็มได้ทั้งหมด $5 \times 7 \times 8 \times 9 \times 11 \times 13 = 360360$ จำนวน แต่ในกรณีของระบบจำนวนเศษเหลือซ้ำซ้อนนี้จะใช้เพียง $5 \times 7 \times 8 \times 9 = 2520$ จำนวน ช่วงที่เป็นไปตามกฎคือ [0, 2520) ช่วงที่ไม่เป็นไปตามกฎคือ [2520, 360360) และได้ค่าระยะทางแฮมมิงเป็น 3

ตัวอย่างการแทนจำนวนเต็มในรูปเศษเหลือ เช่น จำนวนเต็ม 2345 สามารถแทนในรูปเศษเหลือได้เป็น (0, 0, 1, 5, 2, 5) หากมีความผิดพลาดเกิดขึ้นกับการแทนจำนวนเต็ม เช่น ระหว่างการส่งข้อมูล การแทนในรูปเศษเหลือเกิดความผิดพลาด 1 ตำแหน่ง เป็น (3, 0, 1, 5, 2, 5) เมื่อแปลงกลับเป็นจำนวนเต็มจะได้ค่า 290633 ซึ่งอยู่ในช่วงที่ไม่เป็นไปตามกฎ การแก้ไขความผิดพลาดสามารถใช้หลักการความควรจะเป็นสูงสุด ซึ่งจะได้ว่ารูปเศษเหลือ (0, 0, 1, 5, 2, 5) เป็นการแทนจำนวนเต็มที่ถูกต้อง เพราะรูปเศษเหลือนี้อยู่ในช่วงที่เป็นไปตามกฎ และมีระยะทางแฮมมิงห่างจาก (3, 0, 1, 5, 2, 5) น้อยที่สุด □

3.2 งานวิจัยของแคตติ

นอกจากงานวิจัยของบาร์ซิและเมสตรินิแล้ว ยังมีอีกงานวิจัยหนึ่งที่เกี่ยวกับระบบจำนวนเศษเหลือซ้ำซ้อน แคตติ [8] ได้เสนอระบบจำนวนเศษเหลือซ้ำซ้อนที่ตัวหารไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กัน โดยระบบที่เขาเสนอไม่ต้องมีการเพิ่มตัวหารซ้ำซ้อน การใช้ลำดับตัวหารที่ไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กันนั้น จะทำให้มีรูปเศษเหลือบางแบบที่ไม่สามารถใช้งานได้ แต่การที่รูปเศษเหลือบางแบบไม่สามารถใช้งานได้ อาจเพิ่มหรือไม่เพิ่มระยะทางแฮมมิงให้กับระบบก็ได้ พิจารณาตัวอย่างต่อไปนี้

ตัวอย่าง 29 กำหนดลำดับตัวหารที่ไม่เป็นจำนวนเฉพาะสัมพัทธ์กัน (6, 12, 15) ระบบจำนวนเศษเหลือที่ใช้ลำดับตัวหารนี้จะสามารถแทนจำนวนเต็มได้ LCM (6, 12, 15) = 60 จำนวน สาเหตุที่

ไม่สามารถแทนจำนวนเต็มได้ $6 \times 12 \times 15 = 1080$ จำนวน เนื่องจากมีรูปเศษเหลือบางแบบที่จะผิดเงื่อนไขความสัมพันธ์สมภาค เช่น รูปเศษเหลือ (0, 1, 2) เพราะไม่สามารถแก้ระบบความสัมพันธ์ $X \equiv 0 \pmod 6$, $X \equiv 1 \pmod{12}$ และ $X \equiv 2 \pmod{15}$ ได้ อย่างไรก็ตาม ระบบจำนวนเศษเหลือนี้มีระยะทางแฮมมิงเท่ากับ 1 □

ในการใช้งานระบบจำนวนเศษเหลือซ้ำซ้อนที่ตัวหารไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กัน เมื่อผู้ใช้กำหนดลำดับตัวหารแล้ว ผู้ใช้สามารถเลือกช่วงจำนวนเต็มที่ต้องการแทนหรือระยะทางแฮมมิงของระบบได้ หากเลือกให้ช่วงจำนวนเต็มที่ต้องการแทนมีขนาดใหญ่ ค่าระยะทางแฮมมิงของระบบจะมีค่าน้อย ในทางกลับกัน ถ้าเลือกให้ระยะทางแฮมมิงของระบบมีค่ามาก ช่วงจำนวนเต็มที่ระบบสามารถแทนได้จะมีขนาดเล็กลง ความสัมพันธ์นี้เป็นไปตามทฤษฎีบท 30 สังเกตว่าทฤษฎีบทนี้จะครอบคลุมทฤษฎีบท 27 จากงานของบาร์ซิและเมสตรินิ

ทฤษฎีบท 30 (ระบบจำนวนเศษเหลือซ้ำซ้อนที่ตัวหารไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กัน [8]) กำหนดลำดับตัวหารที่ไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กัน $(m_1, m_2, \ldots, m_k)$ ให้สัญลักษณ์ S_c แทนเซตของวิธีการเลือกตัวหาร c ตัวจากทั้งหมด k ตัว ให้คำนวณค่าตัวคูณร่วมน้อยของแต่ละสมาชิกใน S_c ได้เป็นลำดับของตัวคูณร่วมน้อย $(l_1, l_2, \ldots, l_n)$ โดย $n = k! / (c! (k-c)!)$ และให้ $L_c = \min (l_1, l_2, \ldots, l_n)$ แทนค่าตัวคูณร่วมน้อยที่น้อยที่สุดที่คำนวณได้ ถ้าใช้ลำดับตัวหารนี้แทนจำนวนเต็ม L_c จำนวน ระบบจำนวนเศษเหลือซ้ำซ้อนที่ได้จะมีระยะทางแฮมมิงเท่ากับ $k-c+1$ ในทางกลับกัน หากต้องการสร้างระบบจำนวนเศษเหลือซ้ำซ้อนที่มีระยะทางแฮมมิงเท่ากับ d ให้สร้างเซต S_{k-d+1}

จากทฤษฎีบท 30 การเลือกให้ระบบสามารถแทนจำนวนเต็มเพียง L_c จำนวน จากทั้งหมด LCM $(m_1, m_2, \ldots, m_k)$ จำนวน จะเป็นการเพิ่มความซ้ำซ้อนของระบบ และเพิ่มระยะทางแฮมมิงให้กับระบบ ตัวอย่างการนำทฤษฎีบท 30 มาใช้ในการเลือกช่วงจำนวนเต็มที่ต้องการแทนหรือระยะทางแฮมมิงของระบบเป็นดังนี้

ตัวอย่าง 31 กำหนดลำดับตัวหารที่ไม่เป็นจำนวนเฉพาะสัมพัทธ์กัน (6, 10, 12, 15) ระบบจำนวนเศษเหลือที่ใช้ลำดับตัวหารนี้สามารถแทนจำนวนเต็มได้ LCM (6, 10, 12, 15) = 60 จำนวน โดยจะมีค่าระยะทางแฮมมิงเท่ากับ 1

ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อนที่ระยะทางแฮมมิงมีค่ามากกว่า 1 ถ้าทดลองสร้าง S_2 ได้เป็น {(6, 10), (6, 12), (6, 15), (10, 12), (10, 15), (12, 15)} และคำนวณค่าตัวคูณร่วมน้อย

ของแต่ละสมาชิกได้เป็นลำดับ (30, 12, 30, 60, 30, 60) ดังนั้น L_2 แทนค่าตัวคูณร่วมน้อยที่มีค่าน้อยที่สุดคือ min (30, 12, 30, 60, 30, 60) = 12 ถ้าใช้ระบบจำนวนเศษเหลือซ้ำซ้อนนี้แทนจำนวนเต็ม 12 จำนวน จะได้ระบบที่มีค่าระยะทางแฮมมิงเท่ากับ 4 – 2 + 1 = 3

อีกกรณีหนึ่ง ถ้าต้องการระบบจำนวนเศษเหลือซ้ำซ้อนที่มีค่าระยะทางแฮมมิงเท่ากับ 2 ให้สร้าง $S_{4-2+1} = S_3 = \{(6, 10, 12), (6, 10, 15), (6, 12, 15), (10, 12, 15)\}$ และคำนวณค่าตัวคูณร่วมน้อยของแต่ละสมาชิกได้เป็นลำดับ (60, 30, 60, 60) ดังนั้น L_3 แทนค่าตัวคูณร่วมน้อยที่มีค่าน้อยที่สุดคือ min (60, 30, 60, 60) = 30 ถ้าใช้ระบบจำนวนเศษเหลือซ้ำซ้อนนี้แทนจำนวนเต็ม 30 จำนวน จะได้ระบบที่มีค่าระยะทางแฮมมิงเท่ากับ 2 ตามที่ต้องการ □

ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อนที่ตัวหารไม่จำเป็นต้องเป็นจำนวนเฉพาะสัมพัทธ์กัน งานวิจัยของแคตติได้อธิบายวิธีการแปลงระบบจำนวนเศษเหลือแบบเดิมที่ใช้ลำดับตัวหาร $(m_1, m_2, \ldots, m_k)$ ให้เป็นระบบจำนวนเศษเหลือซ้ำซ้อนที่มีค่าระยะทางแฮมมิงเป็น k – 1 อย่างไรก็ตาม จากการศึกษางานวิจัยของแคตติ และทฤษฎีบท 30 ผู้วิจัยสามารถอนุมานวิธีการแปลงระบบจำนวนเศษเหลือแบบเดิมให้เป็นระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติที่มีระยะทางแฮมมิงค่าอื่น ๆ ได้ดังนี้

กำหนดระบบจำนวนเศษเหลือที่ตัวหารแต่ละตัวเป็นจำนวนเฉพาะสัมพัทธ์กัน ถ้าต้องการระบบจำนวนเศษเหลือซ้ำซ้อนแบบแคตติที่มีระยะทางแฮมมิงเท่ากับ d ให้สร้างลำดับตัวหารใหม่จากลำดับตัวหารเดิม ลำดับตัวหารใหม่นี้จะมีจำนวนตัวหารเท่ากับลำดับตัวหารเดิม โดยตัวหารใหม่แต่ละตัวเกิดจากการนำตัวหารเดิมที่แตกต่างกัน d ตัวมาคูณกัน และตัวหารเดิมจะต้องถูกใช้เป็นตัวคูณทั้งหมด d ครั้งพอดี พิจารณาตัวอย่างต่อไปนี้

ตัวอย่าง 32 กำหนดระบบจำนวนเศษเหลือที่ใช้ลำดับตัวหาร (2, 3, 5, 7) ซึ่งเป็นจำนวนเฉพาะสัมพัทธ์กัน ระบบนี้จะสามารถใช้แทนจำนวนเต็มได้ $2 \times 3 \times 5 \times 7 = 210$ จำนวน และมีระยะทางแฮมมิงเท่ากับ 1

ถ้าต้องการระบบจำนวนเศษเหลือซ้ำซ้อนแบบแคตติที่มีระยะทางแฮมมิงเท่ากับ 2 ให้สร้างลำดับตัวหารใหม่ที่เกิดจากการนำตัวหารเดิมสองตัวมาคูณกัน และตัวหารเดิมแต่ละตัวจะถูกนำมาเป็นตัวคูณสองครั้งพอดี ได้เป็น $(2 \times 5, 2 \times 7, 3 \times 5, 3 \times 7) = (10, 14, 15, 21)$ ลำดับตัวหารใหม่อาจสร้างได้มากกว่าหนึ่งแบบ อีก 2 ลำดับที่เป็นไปได้คือ $(2 \times 3, 2 \times 5, 3 \times 7, 5 \times 7) = (6, 10, 21, 35)$ และ $(2 \times 3, 2 \times 7, 3 \times 5, 5 \times 7) = (6, 14, 15, 35)$

ถ้าต้องการระบบจำนวนเศษเหลือซ้ำซ้อนแบบแคตติที่มีระยะทางแฮมมิงเท่ากับ 3 ให้สร้างลำดับตัวหารใหม่ที่เกิดจากการนำตัวหารเดิมสามตัวมาคูณกัน และตัวหารเดิมแต่ละตัวจะถูกนำมาเป็นตัวคูณสามครั้งพอดี ซึ่งก็คือ $(2 \times 3 \times 5,\ 2 \times 3 \times 7,\ 2 \times 5 \times 7,\ 3 \times 5 \times 7) = (30, 42, 70, 105)$

สังเกตว่าเมื่อคำนวณตามทฤษฎีบท 30 จะพบว่าระบบจำนวนเศษเหลือซ้ำซ้อนทั้งสองระบบสามารถแทนจำนวนเต็มได้ 210 จำนวนเหมือนกับระบบจำนวนเศษเหลือเดิม □

3.3 เปรียบเทียบข้อดีและข้อจำกัด

ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ – เมสตรินิ และระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ มีข้อดีและข้อจำกัดที่แตกต่างกัน การนำงานวิจัยของบาร์ซิ – เมสตรินิมาใช้ จะต้องมีการเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ ส่วนงานวิจัยของแคตตินั้น ไม่มีการเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ แต่สังเกตได้ว่าตัวหารที่ใช้นั้นจะมีค่ามากเมื่อเปรียบเทียบกับระบบจำนวนเศษเหลือเดิมและระบบของบาร์ซิ – เมสตรินิ ตารางที่ 1 สรุปข้อดีและข้อจำกัดของระบบทั้งสอง รวมถึงระบบจำนวนเศษเหลือแบบเดิม เครื่องหมาย ✓ ในตารางแสดงข้อดีของระบบนั้น ๆ

ตารางที่ 1 ข้อเปรียบเทียบของระบบจำนวนเศษเหลือ 3 ระบบ

ข้อเปรียบเทียบ	ระบบจำนวนเศษเหลือ	ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ–เมสตรินิ	ระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ
สามารถตรวจจับและแก้ไขความผิดพลาดได้		✓	✓
ไม่ต้องเพิ่มตัวหารซ้ำซ้อน	✓		✓
ตัวหารมีค่าน้อย	✓	✓	

บทที่ 4
วิธีการดำเนินงาน

4.1 แนวคิดในการดำเนินงาน

จากตารางที่ 1 จะเห็นได้ว่าระบบจำนวนเศษเหลือซ้ำซ้อนทั้งสองแบบมีข้อดีและข้อจำกัดที่แตกต่างกัน การเพิ่มตัวหารและการใช้ตัวหารที่มีค่ามาก จะทำให้วงจรของระบบมีขนาดใหญ่ขึ้น [1] งานวิจัยนี้จึงต้องการจะเสนอแนวทางในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อนแบบใหม่ ซึ่งไม่มีข้อจำกัดเหมือนกับงานวิจัยที่เคยมีการเสนอมาแล้ว

ผู้วิจัยได้สังเกตว่าการเพิ่มความซ้ำซ้อนเข้าไปในระบบจำนวนเศษเหลือสามารถทำได้สามวิธี คือ การเพิ่มตัวหารซ้ำซ้อน การเพิ่มค่าของตัวหารให้มีค่ามาก และอีกวิธีหนึ่งคือการเพิ่มรูปแบบของเศษเหลือที่ใช้ในการแทนจำนวนเต็ม ซึ่งยังไม่มีงานวิจัยใดที่นำวิธีการสุดท้ายนี้มาใช้ ข้อสังเกตนี้จึงนำมาสู่แนวคิดของงานวิจัย คือการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน เพื่อให้เห็นภาพมากขึ้น ให้พิจารณาตัวอย่างของค่าเศษเหลือที่มีความซ้ำซ้อนดังนี้

ตัวอย่าง 33 โดยทั่วไป การหารด้วย 2 จะมีเศษเหลือสองแบบคือ 0 และ 1 ถ้ากำหนดให้สามารถใช้ค่าเศษเหลือที่มีความซ้ำซ้อนได้ เราอาจจะใช้ค่าเศษเหลือมากกว่าสองแบบ ตัวอย่างเช่น ถ้าใช้ค่า 0, 1, 2 และ 3 เป็นเศษเหลือจากการหารด้วย 2 จะได้ว่าเศษจากการหาร 7 ด้วย 2 อาจเป็น 1 หรือ 3 ก็ได้ สังเกตว่าเศษที่นำมาใช้นั้นจะมีความสัมพันธ์สมภาคบนตัวหารที่ใช้ เช่นในตัวอย่างนี้ 1 และ 3 มีความสัมพันธ์สมภาคบนการหารด้วย 2 □

ในมุมของระบบจำนวนเศษเหลือ ค่าเศษเหลือที่มีความซ้ำซ้อนจะทำให้จำนวนเต็มสามารถเขียนในรูปเศษเหลือได้มากกว่าหนึ่งแบบ การเลือกใช้งานรูปเศษเหลือบางแบบ และเลือกไม่ใช้งานรูปเศษเหลืออื่น ๆ จะเป็นการเพิ่มความซ้ำซ้อนให้กับระบบ ซึ่งจะเพิ่มระยะทางแฮมมิงของระบบ และทำให้ระบบมีความสามารถในการตรวจจับและแก้ไขข้อผิดพลาดได้ พิจารณาตัวอย่างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อนดังนี้

ตัวอย่าง 34 กำหนดระบบจำนวนเศษเหลือที่ใช้ลำดับตัวหาร (2, 3, 5) ซึ่งเป็นจำนวนเฉพาะสัมพัทธ์กัน โดยปกติแล้วระบบจำนวนเศษเหลือนี้จะต้องแทนจำนวนเต็ม 7 ด้วยรูปเศษเหลือ (1, 1, 2) ถ้ามีการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน เช่น ให้เศษเหลือที่เป็นไปได้จากการหารด้วย 2 ได้แก่ 0, 1, 2 และ 3 และเศษเหลือที่เป็นไปได้จากการหารด้วย 3 ได้แก่ 0, 1, 2, 3, และ 4 ส่วนเศษเหลือ

ที่เป็นไปได้จากการหารด้วย 5 ให้คงไว้เป็น 0, 1, 2, 3 และ 4 ตามเดิม จำนวนเต็ม 7 จะแทนได้สี่รูปแบบ ได้แก่ (1, 1, 2), (3, 1, 2), (1, 4, 2) และ (3, 4, 2) เป็นต้น หากเลือกใช้รูปเศษเหลือเพียงหนึ่งรูปแบบ และไม่ใช้สามรูปแบบที่เหลือ จะเป็นการเพิ่มความซ้ำซ้อนและระยะทางแฮมมิงให้กับระบบได้ สำหรับการแทนจำนวนเต็มอื่น ๆ จะเป็นในลักษณะเดียวกัน □

จากตัวอย่าง 34 จะเห็นว่า ระบบที่เสนอมีสมบัติเป็นระบบจำนวนเศษเหลือซ้ำซ้อนโดยไม่ต้องเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ และไม่ต้องเพิ่มค่าของตัวหารให้มีค่ามากขึ้น ทำให้ระบบที่เสนอเป็นไปตามจุดประสงค์ที่ตั้งไว้ แต่การใช้ค่าเศษเหลือที่มีความซ้ำซ้อนอาจทำให้การคำนวณของระบบมีความซับซ้อนมากขึ้นได้

เพื่อให้การเสนอระบบจำนวนเศษเหลือซ้ำซ้อนที่คิดขึ้นเป็นไปได้อย่างถูกต้อง จะขออธิบายแนวคิดในการแบ่งรูปแบบการแทนจำนวนเต็มด้วยเศษเหลือออกเป็นกลุ่มย่อย ๆ และหลังจากนั้นจะนำการแบ่งกลุ่มนี้ไปใช้สร้างเป็นระบบจำนวนเศษเหลือซ้ำซ้อนต่อไป ซึ่งจะประกอบด้วยขั้นตอนวิธีในการแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็ม การบวก การคูณ การตรวจจับและแก้ไขความผิดพลาด และการเปรียบเทียบค่าในรูปเศษเหลือ

4.2 แนวคิดในการแบ่งรูปแบบการแทนจำนวนเต็มด้วยเศษเหลือออกเป็นกลุ่มย่อย

จากหัวข้อที่ 4.1 จะพบปัญหาว่า ถ้ากำหนดลำดับตัวหารและค่าอื่น ๆ ที่จำเป็นมาให้ จะเลือกใช้ค่าเศษเหลือที่มีความซ้ำซ้อนค่าใดบ้าง ด้วยเหตุนี้ แนวคิดในการแบ่งรูปแบบการแทนจำนวนเต็มด้วยเศษเหลือออกเป็นกลุ่มย่อย ๆ จึงเกิดขึ้น

จากทฤษฎีบท 27 เมื่อกำหนดระบบจำนวนเศษเหลือที่ใช้ลำดับตัวหาร $(m_1, m_2, \ldots, m_k)$ หากต้องการระบบจำนวนเศษเหลือซ้ำซ้อนที่มีระยะทางแฮมมิงเท่ากับ d โดยไม่ต้องเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ เราสามารถแบ่งตัวหารที่ได้มาออกเป็นลำดับตัวหารข้อมูล $(m_1, \ldots, m_{k-d+1})$ และลำดับตัวหารซ้ำซ้อน $(m_{k-d+2}, \ldots, m_k)$ จะเห็นได้ว่าจำนวนเต็มที่ระบบสามารถแทนได้ ลดลงจาก $\prod_{(i=1,\ldots,k)} m_i$ เป็น $\prod_{(i=1,\ldots,k-d+1)} m_i$

เพื่อให้ระบบสามารถแทนจำนวนเต็มได้ $\prod_{(i=1,\ldots,k)} m_i$ จำนวนเท่าเดิม ขอให้สังเกตว่าการแทนกลุ่มของจำนวนเต็ม $\prod_{(i=1,\ldots,k-d+1)} m_i$ จำนวน จะได้ระยะทางแฮมมิงเท่ากับ d ดังนั้น หากเราแบ่งจำนวนเต็มที่มีอยู่ทั้งหมด $\prod_{(i=1,\ldots,k)} m_i$ จำนวน ออกเป็นกลุ่ม กลุ่มละ $\prod_{(i=1,\ldots,k-d+1)} m_i$ จำนวน และให้หมายเลขกับแต่ละกลุ่ม เมื่อระบุหมายเลขกลุ่มที่ต้องการ เราก็จะสามารถแทนจำนวนเต็ม $\prod_{(i=1,\ldots,k-d+1)} m_i$ จำนวนในกลุ่มนั้นได้ โดยมีระยะทางแฮมมิงเท่ากับ d เมื่อรวมรูปเศษเหลือที่มีอยู่และ

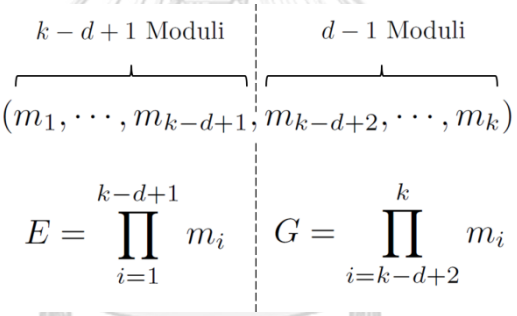
หมายเลขกลุ่มเข้าด้วยกัน ระบบจำนวนเศษเหลือซ้ำซ้อนที่ได้ก็จะสามารถแทนจำนวนเต็มได้ $\prod_{(i=1,\ldots,k)} m_i$ จำนวน และมีระยะทางแฮมมิงเท่ากับ d

เพื่อให้เข้าใจได้ตรงกันและสะดวกในการอธิบายต่อไป ขอกำหนดนิยามต่าง ๆ ดังนี้

นิยาม 35 (หมายเลขกลุ่มของระบบจำนวนเศษเหลือ) กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร (m_1, m_2, ..., m_k) โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และ $m_i < m_{i+1}$ สำหรับทุกค่า i ในช่วง $1 \le i \le k-1$

จากรูปแบบการแทนจำนวนเต็มทั้งหมด $M_k = \prod_{(i=1,\ldots,k)} m_i$ จำนวน จะแบ่งรูปเศษเหลือออกเป็นกลุ่ม กลุ่มละ $E = \prod_{(i=1,\ldots,k-d+1)} m_i$ จำนวน จะได้ว่ามีจำนวนกลุ่มทั้งสิ้น $G = M_k / E$ กลุ่ม กำหนดให้หมายเลขกลุ่มของจำนวนเต็ม X มีค่าเท่ากับ $\lfloor X / E \rfloor$

ภาพรวมของนิยาม 35 สามารถแสดงได้ดังรูปที่ 1



รูปที่ 1 การคำนวณจำนวนรูปเศษเหลือในแต่ละกลุ่ม และจำนวนกลุ่มที่ได้ ตามนิยาม 35

ตัวอย่าง 36 กำหนดระบบจำนวนเศษเหลือที่มีลำดับตัวหารเป็น (2, 3, 5) ซึ่งระบบนี้สามารถแทนจำนวนเต็มได้ 30 จำนวน หากต้องการให้ระบบมีระยะทางแฮมมิงเป็น 2 สามารถแบ่งลำดับตัวหารออกเป็นลำดับตัวหารข้อมูล (2, 3) และลำดับตัวหารซ้ำซ้อน (5) วิธีการนี้จะทำให้ระบบสามารถแทนจำนวนเต็มได้เพียง 6 จำนวนเท่านั้น

เพื่อให้ระบบสามารถแทนจำนวนเต็มได้ 30 จำนวน จะทำการแบ่งรูปเศษเหลือออกเป็นกลุ่ม เมื่อคำนวณค่าตามนิยาม 35 จะได้ว่ารูปแบบการแทนจำนวนเต็มด้วยเศษเหลือของระบบที่เสนอ จะถูกแบ่งออกเป็นกลุ่มละ E = 6 จำนวน และมีทั้งหมด G = 5 กลุ่ม

เมื่อคำนวณหมายเลขกลุ่มตามนิยาม 35 จะได้ว่ารูปเศษเหลือที่แทนจำนวนเต็ม 5 จะอยู่ในกลุ่มหมายเลข $\lfloor 5 / 6 \rfloor = 0$ และรูปเศษเหลือที่แทนจำนวนเต็ม 20 จะอยู่ในกลุ่มหมายเลข $\lfloor 20 / 6 \rfloor = 3$ สังเกตว่าหมายเลขกลุ่มจะมีค่าตั้งแต่ 0 จนถึง G–1 □

4.3 การแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ

เมื่อได้หมายเลขกลุ่มจากหัวข้อที่ 4.2 แล้ว ขั้นตอนต่อไปจะเป็นการนำหมายเลขกลุ่มที่ได้ มารวมกับระบบจำนวนเศษเหลือที่มีอยู่เดิม เพื่อให้กลายเป็นระบบจำนวนเศษเหลือซ้ำซ้อน ขั้นตอนวิธีในการแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ เป็นไปตามรูปที่ 2

Algorithm 1 Converting integer into RRNS representation with redundant residue representations

Input: pairwise co-prime moduli sequence (m_1, ..., m_k), Hamming distance d, and integer X
Output: RRNS representation of X as (x_1, ..., x_k)

1: **for** i = 1 to k **do**
2: $x_i \leftarrow | X |_{mi}$
3: **end for**
4: $E \leftarrow \Pi_{(i=1,...,k-d+1)}\, m_i$
5: $g \leftarrow \lfloor X / E \rfloor$
6: **for** i = 1 to d **do**
7: $x_i \leftarrow x_i + g \times m_i$
8: **end for**
9: **return** (x_1, ..., x_k)

รูปที่ 2 ขั้นตอนวิธีในการแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

ขั้นตอนวิธีตามรูปที่ 2 สามารถอธิบายได้ดังนี้ กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร (m_1, m_2, ..., m_k) โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และ $m_i < m_{i+1}$ สำหรับทุกค่า i ในช่วง $1 \le i \le k-1$ กำหนดให้ระยะทางแฮมมิงของระบบจำนวนเศษเหลือซ้ำซ้อนที่ต้องการมีค่าเท่ากับ d และจำนวนเต็มที่ต้องการแปลงคือ X

ในบรรทัดที่ 1 – 3 จะทำการคำนวณรูปเศษเหลือของ X ในระบบจำนวนเศษเหลือปกติก่อน จากนั้นในบรรทัดที่ 4 – 5 จะทำการคำนวณหมายเลขกลุ่มของ X ซึ่งมีค่าเท่ากับ $\lfloor X / E \rfloor$ และในบรรทัดที่ 6 – 8 จะทำการรวมหมายเลขกลุ่มเข้าไปในค่าของเศษเหลือ d ตัวแรก โดยคูณหมายเลขกลุ่มกับตัวหารแต่ละตัว แล้วบวกเข้ากับเศษเหลือที่ได้จากบรรทัดที่ 1 – 3 เมื่อรวมหมายเลขกลุ่มเข้ากับค่าเศษเหลือ d ตัวแรกครบแล้ว ก็เป็นอันเสร็จสิ้นขั้นตอนวิธี

จะเห็นว่ารูปเศษเหลือจากขั้นตอนวิธีในรูปที่ 2 มีความสัมพันธ์สมภาคกับรูปเศษเหลือที่ได้จากระบบจำนวนเศษเหลือปกติ ทฤษฎีบทต่อไปนี้จะแสดงว่าขั้นตอนวิธีที่นำเสนอ จะให้ระบบจำนวนเศษเหลือซ้ำซ้อนที่มีระยะทางแฮมมิงมากกว่าหรือเท่ากับ d จริง

ทฤษฎีบท 37 ระบบจำนวนเศษเหลือซ้ำซ้อนที่ได้จากขั้นตอนวิธีในรูปที่ 2 มีระยะทางแฮมมิงมากกว่าหรือเท่ากับ d

บทพิสูจน์ กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร $(m_1, m_2, ..., m_k)$ โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และ $m_i < m_{i+1}$ สำหรับทุกค่า i ในช่วง $1 \le i \le k-1$ และกำหนดค่า d เป็นข้อมูลนำเข้าของขั้นตอนวิธี จะแสดงให้เห็นว่า เมื่อเลือกจำนวนเต็ม X และ Y ที่มีค่าไม่เท่ากัน และอยู่ในช่วง $[0, M_k)$ รูปเศษเหลือของ X และ Y ที่ได้จากขั้นตอนวิธีจะมีระยะทางแฮมมิงมากกว่าหรือเท่ากับ d เสมอ แบ่งการพิสูจน์เป็น 2 กรณี ได้แก่

กรณีที่ 1 ถ้า X และ Y มีหมายเลขกลุ่มเท่ากัน จากทฤษฎีบท 27 จะได้ว่าการใช้ระบบจำนวนเศษเหลือซ้ำซ้อนที่มีลำดับตัวหารข้อมูล $(m_1, ..., m_{k-d+1})$ และลำดับตัวหารซ้ำซ้อน $(m_{k-d+2}, ..., m_k)$ แทนจำนวนเต็มในช่วง $[N, N+E)$ เมื่อ N เป็นจำนวนเต็มใด ๆ และ $E = \prod_{(i=1,...,k-d+1)} m_i$ จะมีระยะทางแฮมมิงเท่ากับ d และเนื่องจาก X และ Y มีหมายเลขกลุ่มเท่ากัน ดังนั้น X และ Y มีค่าต่างกันไม่เกิน E และระยะทางแฮมมิงระหว่างรูปเศษเหลือของ X และ Y ก็จะต้องมีค่ามากกว่าหรือเท่ากับ d ด้วย เมื่อบวกหมายเลขกลุ่มเข้าไปในรูปเศษเหลือของ X และ Y ระยะทางแฮมมิงก็ยังคงมากกว่าหรือเท่ากับ d

กรณีที่ 2 ถ้า X และ Y มีหมายเลขกลุ่มที่ต่างกัน จากบรรทัดที่ 6 – 8 ของขั้นตอนวิธี มีการบวกหมายเลขกลุ่มเข้ากับเศษเหลือ d ตัวแรก พิจารณาตัวหาร m_i ถ้าหมายเลขกลุ่มของ X คือ g_X และหมายเลขกลุ่มของ Y คือ g_Y จะได้รูปเศษเหลือของ X และ Y เป็น $|X|_{mi} + g_X \times m_i$ และ $|Y|_{mi} + g_Y \times m_i$ ตามลำดับ และเนื่องจาก g_X ไม่เท่ากับ g_Y และ $|X|_{mi}$ กับ $|Y|_{mi}$ มีค่าน้อยกว่า m_i ดังนั้น $|X|_{mi} + g_X \times m_i$ ต้องมีค่าไม่เท่ากับ $|Y|_{mi} + g_Y \times m_i$ สรุปได้ว่ารูปเศษเหลือ d ตัวแรกของ X และ Y แตกต่างกัน นั่นคือระยะทางแฮมมิงระหว่างรูปเศษเหลือของ X และ Y มีค่ามากกว่าหรือเท่ากับ d

จากทั้ง 2 กรณีสามารถสรุปได้ว่ารูปเศษเหลือของ X และ Y จะมีระยะทางแฮมมิงมากกว่าหรือเท่ากับ d เสมอ ดังนั้นระยะทางแฮมมิงของระบบจำนวนเศษเหลือซ้ำซ้อนจึงมากกว่าหรือเท่ากับ d ด้วย ■

จากขั้นตอนวิธีในรูปที่ 2 จะเห็นได้ว่าระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ ไม่มีการเพิ่มตัวหารซ้ำซ้อนเข้าไปในระบบ และไม่ต้องเพิ่มค่าของตัวหาร และจะเห็นได้ว่าเศษเหลือของตัวหาร d ตัวแรกสามารถมีค่ามากกว่าตัวหารได้ นั่นคือเราได้เพิ่มรูปเศษเหลือที่มีความซ้ำซ้อนเข้าไปนั่นเอง เพื่อให้เห็นภาพมากยิ่งขึ้น ขอยกตัวอย่างต่อไปนี้

ตัวอย่าง 38 กำหนดระบบจำนวนเศษเหลือที่มีลำดับตัวหาร (2, 3, 5) ถ้าต้องการให้ระยะทางแฮมมิงของระบบมีค่าเป็น 2 จากตัวอย่าง 36 จะได้ค่า E = 6 และ G = 5 เมื่อพิจารณาจำนวนเต็มแต่ละจำนวน จะสามารถคำนวณหมายเลขกลุ่ม และรูปเศษเหลือในระบบจำนวนเศษเหลือซ้ำซ้อนตามขั้นตอนวิธีในรูปที่ 2 ได้ดังตารางที่ 2 □

ตารางที่ 2 ระบบจำนวนเศษเหลือซ้ำซ้อนที่นำเสนอ เมื่อกำหนดลำดับตัวหารเป็น (2, 3, 5) และกำหนดระยะทางแฮมมิงเป็น 2

จำนวนเต็ม	รูปเศษเหลือปกติ	หมายเลขกลุ่ม	รูปเศษเหลือจากขั้นตอนวิธีในรูปที่ 2
0	(0, 0, 0)	0	(0, 0, 0)
1	(1, 1, 1)	0	(1, 1, 1)
2	(0, 2, 2)	0	(0, 2, 2)
3	(1, 0, 3)	0	(1, 0, 3)
4	(0, 1, 4)	0	(0, 1, 4)
5	(1, 2, 0)	0	(1, 2, 0)
6	(0, 0, 1)	1	(2, 3, 1)
7	(1, 1, 2)	1	(3, 4, 2)
...	...	...	...
11	(1, 2, 1)	1	(3, 5, 1)
12	(0, 0, 2)	2	(4, 6, 2)
...	...	...	...
17	(1, 2, 2)	2	(5, 8, 2)
18	(0, 0, 3)	3	(6, 9, 3)
...	...	...	...
23	(1, 2, 3)	3	(7, 11, 3)
24	(0, 0, 4)	4	(8, 12, 4)
...	...	...	...
28	(0, 1, 3)	4	(8, 13, 3)
29	(1, 2, 4)	4	(9, 14, 4)

4.4 การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็ม

เนื่องจากขั้นตอนวิธีในการแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือของระบบที่เสนอ ได้คงความสัมพันธ์สมภาคของจำนวนเต็มตั้งต้นและเศษเหลือแต่ละตัว ดังนั้นการแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มสามารถใช้ทฤษฎีบทเศษเหลือของจีนกับค่าเศษเหลือทุกตัวได้ เหมือนกับระบบจำนวนเศษเหลือปกติ

อย่างไรก็ตาม จากการสังเกตตารางที่ 2 จะพบว่า ในระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ เมื่อกำหนดตัวหาร k ตัว และกำหนดระยะทางแฮมมิงให้มีค่าเท่ากับ d ค่าเศษเหลือ k–d+1 ตัวแรกของรูปเศษเหลือในแต่ละกลุ่มจะมีรูปแบบเหมือนกัน และในแต่ละกลุ่มค่าเศษเหลือ k–d+1 ตัวแรกของรูปเศษเหลือจะมีรูปแบบที่ไม่ซ้ำกัน ทั้งนี้เนื่องจากเราได้แบ่งกลุ่มของรูปเศษเหลือออกเป็นกลุ่มละ $E = \prod_{(i=1,\ldots,k-d+1)} m_i$ จำนวน

จากสมบัติของระบบจำนวนเศษเหลือซ้ำซ้อนที่ได้กล่าวไป ทำให้การใช้หมายเลขกลุ่มและค่าเศษเหลือ k–d+1 ตัวแรก เพียงพอต่อการระบุจำนวนเต็มแต่ละตัว การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มจึงใช้การคำนวณตามทฤษฎีบทเศษเหลือของจีนกับค่าเศษเหลือ k–d+1 ตัวแรกเท่านั้น ไม่จำเป็นต้องคำนวณกับค่าเศษเหลือทั้ง k ตัวเหมือนกับระบบเดิม ขั้นตอนวิธีในการแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มสามารถแสดงได้ดังรูปที่ 3

Algorithm 2 Converting RRNS representation with redundant residue representations to integer

Input: pairwise co-prime moduli sequence $(m_1, \ldots, m_k)$, Hamming distance d, and RRNS representation of X as $(x_1, \ldots, x_k)$

Output: integer value of X

1: $h \leftarrow \mathrm{CRT}(m_1, \ldots, m_{k-d+1}, x_1, \ldots, x_{k-d+1})$
2: $g \leftarrow \lfloor x_1 / m_1 \rfloor$
3: $E \leftarrow \prod_{(i=1,\ldots,k-d+1)} m_i$
4: $X \leftarrow h + g \times E$
5: **return** X

รูปที่ 3 ขั้นตอนวิธีในการแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

ขั้นตอนวิธีตามรูปที่ 3 สามารถอธิบายได้ดังนี้ กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร $(m_1, m_2, \ldots, m_k)$ โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และ $m_i < m_{i+1}$ สำหรับทุกค่า i ในช่วง $1 \le i \le k-1$ กำหนดให้ระยะทางแฮมมิงของระบบ

จำนวนเศษเหลือซ้ำซ้อนมีค่าเท่ากับ d และรูปเศษเหลือที่ต้องการแปลงกลับเป็นจำนวนเต็มคือ $(x_1, x_2, ..., x_k)$

ในบรรทัดที่ 1 จะทำการคำนวณตามทฤษฎีบทเศษเหลือของจีน ได้เป็นอันดับของรูปเศษเหลือในกลุ่มที่รูปเศษเหลือนั้นอยู่ บรรทัดที่ 2 ทำการคำนวณหมายเลขกลุ่ม บรรทัดที่ 3 – 4 คำนวณค่าจำนวนเต็ม X โดยหาผลคูณระหว่างจำนวนรูปเศษเหลือในแต่ละกลุ่ม (E) และจำนวนกลุ่มที่อยู่ก่อนหน้า เมื่อบวกด้วยอันดับของรูปเศษเหลือในกลุ่มที่รูปเศษเหลือนั้นอยู่ ก็จะได้อันดับของรูปเศษเหลือนั้นเทียบกับรูปเศษเหลือทั้งหมดที่เป็นไปได้ ซึ่งมีค่าเท่ากับค่าจำนวนเต็มของรูปเศษเหลือนั้น เพื่อให้เห็นภาพได้ชัดเจนมากขึ้น พิจารณาตัวอย่างการทำงานตามขั้นตอนวิธีดังนี้

ตัวอย่าง 39 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อนเช่นเดียวกันกับตัวอย่าง 38 เมื่อต้องการแปลงรูปเศษเหลือ (3, 5, 1) กลับเป็นจำนวนเต็ม ตามขั้นตอนวิธีในรูปที่ 3 จะคำนวณตามทฤษฎีบทเศษเหลือของจีน เพื่อแก้ความสัมพันธ์สมภาค $x \equiv 3 \pmod 2$ และ $x \equiv 5 \pmod 3$ ซึ่งจะได้คำตอบเป็น $x = 5$ และคำนวณหมายเลขกลุ่มจาก $g = \lfloor 3 / 2 \rfloor = 1$ ให้สังเกตว่ารูปเศษเหลือ (3, 5, 1) อยู่เป็นอันดับที่ 5 ในกลุ่มหมายเลข 1 ดังนั้นรูปเศษเหลือ (3, 5, 1) จึงอยู่เป็นอันดับที่ $x + g \times E = 5 + 1 \times 6 = 11$ ในระบบ ซึ่งหมายความว่ารูปเศษเหลือ (3, 5, 1) แทนค่าจำนวนเต็ม 11 ด้วย □

4.5 การบวก

เนื่องจากขั้นตอนวิธีในการแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือของระบบที่เสนอ ได้คงความสัมพันธ์สมภาคของจำนวนเต็มตั้งต้นและเศษเหลือแต่ละตัว ดังนั้นการบวกก็อาจจะสามารถใช้หลักการของทฤษฎีบท 14 ได้ แต่เนื่องจากระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ มีการรวมหมายเลขกลุ่มเข้าไปในรูปแบบการแทนจำนวนเต็มด้วยเศษเหลือ ดังนั้นการดำเนินการบวกจะต้องมีความซับซ้อนมากกว่าเดิม ขอให้พิจารณาทฤษฎีบทต่อไปนี้

ทฤษฎีบท 40 มีขั้นตอนวิธีการคำนวณผลบวกของรูปเศษเหลือ สำหรับระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

บทพิสูจน์ ขั้นตอนวิธีการคำนวณผลบวกของรูปเศษเหลือเป็นไปตามขั้นตอนวิธีในรูปที่ 4

Algorithm 3 Addition operation for proposed RRNS

Input: pairwise co-prime moduli sequence $(m_1, ..., m_k)$, Hamming distance d, RRNS representations of X and Y as $(x_1, ..., x_k)$ and $(y_1, ..., y_k)$
Output: RRNS representation of $Z = X + Y$ as $(z_1, ..., z_k)$

1: **for** $i = 1$ to k **do**
2: $z_i \leftarrow | x_i + y_i |_{mi}$
3: **end for**
4: $E \leftarrow \prod_{(i=1,...,k-d+1)} m_i$
5: $G \leftarrow \prod_{(i=k-d+2,...,k)} m_i$
6: $g_x \leftarrow \lfloor x_1/m_1 \rfloor$
7: $g_y \leftarrow \lfloor y_1/m_1 \rfloor$
8: $h_x \leftarrow \mathrm{CRT}(m_1, ..., m_{k-d+1}, x_1, ..., x_{k-d+1})$
9: $h_y \leftarrow \mathrm{CRT}(m_1, ..., m_{k-d+1}, y_1, ..., y_{k-d+1})$
10: $g_z \leftarrow g_x + g_y$
11: **if** $h_x + h_y \geq E$ **then**
12: $g_z \leftarrow | g_z + 1 |_G$
13: **end if**
14: **for** $i = 1$ to d **do**
15: $z_i \leftarrow z_i + g_z \times m_i$
16: **end for**
17: **return** $(z_1, ..., z_k)$

รูปที่ 4 ขั้นตอนวิธีในการคำนวณผลบวกของรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

ขั้นตอนวิธีตามรูปที่ 4 สามารถอธิบายได้ดังนี้ กำหนดระบบจำนวนเศษเหลือ ที่มีลำดับของตัวหาร (m_1, m_2, ..., m_k) โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และ $m_i < m_{i+1}$ สำหรับทุกค่า i ในช่วง $1 \leq i \leq k-1$ กำหนดให้ระยะทางแฮมมิงของระบบจำนวนเศษเหลือซ้ำซ้อนมีค่าเท่ากับ d และรูปเศษเหลือของจำนวนเต็ม X และ Y ที่ต้องการหาผลบวกคือ (x_1, x_2, ..., x_k) และ (y_1, y_2, ..., y_k) ตามลำดับ

ในบรรทัดที่ 1 – 3 จะทำการหาผลบวกตามทฤษฎีบท 14 เช่นเดียวกันกับระบบจำนวนเศษเหลือปกติ การคำนวณในลักษณะนี้จะทำให้คงความสัมพันธ์สมภาคของจำนวนเต็มตั้งต้นและเศษเหลือแต่ละตัว จากนั้นจะทำการคำนวณหมายเลขกลุ่มของผลบวก บรรทัดที่ 4 – 9 เป็นการคำนวณค่า $E = \prod_{(i=1,...,k-d+1)} m_i$, $G = \prod_{(i=k-d+2,...,k)} m_i$ และหมายเลขกลุ่มและอันดับในกลุ่มของ X และ Y ตามลำดับ บรรทัดที่ 10 – 13 จะทำการคำนวณหมายเลขกลุ่มของผลบวก โดยหมายเลขกลุ่มของผลบวก จะมีค่าเท่ากับผลรวมของหมายเลขกลุ่มของ X และ Y แต่มีโอกาสร้อยละ 50 ที่ผลรวมของอันดับในกลุ่มของ X และ Y จะมีค่าตั้งแต่ E ขึ้นไป ในกรณีนี้จะต้องบวก 1 เข้าไปที่หมายเลขกลุ่มของผลบวกด้วย ทั้งนี้ผลรวมของอันดับในกลุ่มของ X และ Y จะมีค่าไม่ถึง 2E เนื่องจากอันดับในกลุ่ม

ของ X และ Y มีค่าได้ตั้งแต่ 0 ถึง E–1 เงื่อนไขที่กล่าวมาสามารถเขียนได้ตามที่แสดงในบรรทัดที่ 11 และสามารถอธิบายได้ด้วยสมการดังนี้

เนื่องจาก $X = g_x \times E + h_x$

$Y = g_y \times E + h_y$

$Z = X + Y$

ดังนั้น $g_z = \lfloor Z / E \rfloor$

$= g_x + g_y + \lfloor (h_x + h_y) / E \rfloor$

และเนื่องจาก $0 \le h_x + h_y < 2E$

ดังนั้น $0 \le \lfloor (h_x + h_y) / E \rfloor \le 1$

เมื่อได้หมายเลขกลุ่มของผลบวกแล้ว บรรทัดที่ 14 – 16 จะนำหมายเลขกลุ่มไปรวมกับรูปเศษเหลือที่ได้จากบรรทัดที่ 1 – 3 และได้ผลลัพธ์สุดท้ายเป็นรูปเศษเหลือของผลบวกในระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอตามที่ต้องการ ■

เพื่อให้เห็นภาพได้ชัดเจนมากขึ้น พิจารณาตัวอย่างการทำงานตามขั้นตอนวิธีดังนี้

ตัวอย่าง 41 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อนเช่นเดียวกันกับตัวอย่าง 38 พิจารณาการบวกระหว่าง 11 และ 15 ซึ่งมีรูปเศษเหลือเป็น (3, 5, 1) และ (5, 6, 0) ตามลำดับ ในบรรทัดที่ 1 – 3 จะทำการบวกตามทฤษฎีบท 14 ได้ผลบวกในเบื้องต้นเป็น (0, 2, 1) ในบรรทัดที่ 4 – 9 คำนวณค่าต่าง ๆ ได้ E = 6, G = 5, $g_x = 1$, $g_y = 2$, $h_x = 5$, $h_y = 3$ และในบรรทัดที่ 10 คำนวณหมายเลขกลุ่มของผลบวกได้ $g_z = g_x + g_y = 3$ เมื่อพิจารณาเงื่อนไขในบรรทัดที่ 11 อันดับในกลุ่มของ 11 และ 15 มีค่าเป็น 5 และ 3 ตามลำดับ ผลรวมของอันดับมีค่าเท่ากับ 8 ซึ่งมากกว่าหรือเท่ากับ E จึงต้องเพิ่มค่า g_z อีก 1 ทำให้ได้ $g_z = 4$ และสุดท้าย บรรทัดที่ 14 – 16 นำหมายเลขกลุ่ม $g_z = 4$ มารวมกับผลบวกในเบื้องต้น (0, 2, 1) ได้เป็นผลลัพธ์สุดท้าย (8, 14, 1) ซึ่งตรงกับรูปเศษเหลือของ 26 ตามระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ □

4.6 การคูณ

การคูณในระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ มีแนวคิดคล้ายกับการบวก แต่การคำนวณหมายเลขกลุ่มของผลคูณจะมีความซับซ้อนมากกว่า

ทฤษฎีบท 42 มีขั้นตอนวิธีการคำนวณผลคูณของรูปเศษเหลือ สำหรับระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

บทพิสูจน์ ขั้นตอนวิธีการคำนวณผลคูณของรูปเศษเหลือเป็นไปตามขั้นตอนวิธีในรูปที่ 5

Algorithm 4 Multiplication operation for proposed RRNS

Input: pairwise co-prime moduli sequence $(m_1, ..., m_k)$, Hamming distance d, RRNS representations of X and Y as $(x_1, ..., x_k)$ and $(y_1, ..., y_k)$

Output: RRNS representation of $Z = X \times Y$ as $(z_1, ..., z_k)$

1: **for** $i = 1$ to k **do**
2: $\quad z_i \leftarrow | x_i \times y_i |_{mi}$
3: **end for**
4: $E \leftarrow \prod_{(i=1,...,k-d+1)} m_i$
5: $G \leftarrow \prod_{(i=k-d+2,...,k)} m_i$
6: $g_x \leftarrow \lfloor x_1/m_1 \rfloor$
7: $g_y \leftarrow \lfloor y_1/m_1 \rfloor$
8: $h_x \leftarrow \mathrm{CRT}(m_1, ..., m_{k-d+1}, x_1, ..., x_{k-d+1})$
9: $h_y \leftarrow \mathrm{CRT}(m_1, ..., m_{k-d+1}, y_1, ..., y_{k-d+1})$
10: $g_z \leftarrow | g_x g_y E + g_x h_y + g_y h_x + \lfloor h_x h_y / E \rfloor |_G$
11: **for** $i = 1$ to d **do**
12: $\quad z_i \leftarrow z_i + g_z \times m_i$
13: **end for**
14: **return** $(z_1, ..., z_k)$

รูปที่ 5 ขั้นตอนวิธีในการคำนวณผลคูณของรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

การทำงานของขั้นตอนวิธีในรูปที่ 5 จะคล้ายกับขั้นตอนวิธีในรูปที่ 4 ขอให้สังเกตว่าบรรทัดที่ 1, 3 – 9 และ 11 – 14 ของขั้นตอนวิธีในรูปที่ 5 เหมือนกับขั้นตอนวิธีในรูปที่ 4 ส่วนที่แตกต่างกันคือ บรรทัดที่ 2 เป็นการคำนวณผลคูณในเบื้องต้นตามทฤษฎีบท 14 และบรรทัดที่ 10 ซึ่งเป็นการคำนวณหมายเลขกลุ่มของผลคูณ สำหรับแนวคิดในการคำนวณหมายเลขกลุ่มของผลคูณสามารถอธิบายได้ด้วยสมการในลักษณะเดียวกันกับขั้นตอนวิธีในรูปที่ 4 ดังนี้

เนื่องจาก $\quad X = g_x \times E + h_x$

$\quad Y = g_y \times E + h_y$

$\quad Z = X \times Y$

ดังนั้น $g_z = \lfloor Z / E \rfloor$

$= \lfloor (g_x \times E + h_x) (g_y \times E + h_y) / E \rfloor$

$= g_x g_y E + g_x h_y + g_y h_x + \lfloor h_x h_y / E \rfloor$ ■

เพื่อให้เห็นภาพได้ชัดเจนมากขึ้น พิจารณาตัวอย่างการทำงานตามขั้นตอนวิธีดังนี้

ตัวอย่าง 43 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อนเช่นเดียวกันกับตัวอย่าง 38 พิจารณาการคูณระหว่าง 3 และ 7 ซึ่งมีรูปเศษเหลือเป็น (1, 0, 3) และ (3, 4, 2) ตามลำดับ ในบรรทัดที่ 1 – 3 จะทำการคูณตามทฤษฎีบท 14 ได้ผลคูณในเบื้องต้นเป็น (1, 0, 1) ในบรรทัดที่ 4 – 9 คำนวณค่าต่าง ๆ ได้ $E = 6$, $G = 5$, $g_x = 0$, $g_y = 1$, $h_x = 3$, $h_y = 1$ ในบรรทัดที่ 10 คำนวณหมายเลขกลุ่มของผลคูณตามสมการได้ $g_z = 3$ และสุดท้าย บรรทัดที่ 11 – 13 นำหมายเลขกลุ่ม $g_z = 3$ มารวมกับผลบวกในเบื้องต้น (1, 0, 1) ได้เป็นผลลัพธ์สุดท้าย (7, 9, 1) ซึ่งตรงกับรูปเศษเหลือของ 21 ตามระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ □

4.7 การตรวจจับและแก้ไขความผิดพลาด

จากทฤษฎีบท 22 ถ้าระบบจำนวนเศษเหลือซ้ำซ้อนที่มีลำดับตัวหาร (m_1, m_2, ..., m_k) มีระยะทางแฮมมิงเท่ากับ d ระบบจะสามารถตรวจจับความผิดพลาดได้ หากมีความผิดพลาดเกิดขึ้นไม่เกิน d – 1 ตำแหน่ง และระบบยังสามารถแก้ไขความผิดพลาดได้ หากมีความผิดพลาดเกิดขึ้นไม่เกิน $\lfloor (d - 1) / 2 \rfloor$ ตำแหน่ง ในการแก้ไขความผิดพลาดจะใช้หลักการที่เรียกว่าหลักการความควรจะเป็นสูงสุด (maximum likelihood principle)

เนื่องจากมีการเพิ่มข้อมูลหมายเลขกลุ่มเข้าไปในรูปเศษเหลือ จึงสามารถนำหมายเลขกลุ่มนี้มาช่วยในการตรวจจับและแก้ไขความผิดพลาดได้ ขั้นตอนการตรวจจับความผิดพลาดจะเริ่มจากการตรวจสอบหมายเลขกลุ่มของค่าเศษเหลือ d ตัวแรกก่อน ในกรณีที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกไม่ตรงกัน จะสามารถสรุปได้ทันทีว่ามีความผิดพลาดเกิดขึ้นกับรูปเศษเหลือที่กำลังตรวจสอบ ขอให้สังเกตว่าในกรณีนี้ระบบสามารถทำงานได้รวดเร็ว การที่ระบบมีค่าระยะทางแฮมมิง d ที่มาก จะทำให้เมื่อมีความผิดพลาดเกิดขึ้น จะมีโอกาสสูงที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกจะไม่ตรงกัน และระบบจะสามารถตรวจพบได้อย่างรวดเร็ว

ในกรณีที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกตรงกันทั้งหมด ยังไม่สามารถสรุปได้ว่าไม่มีความผิดพลาดเกิดขึ้นกับรูปเศษเหลือที่กำลังตรวจสอบ ให้ทำการแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มตามทฤษฎีบทเศษเหลือของจีน โดยใช้ค่าเศษเหลือทั้ง k ตัว เมื่อได้ค่าจำนวนเต็มแล้ว

ให้ตรวจสอบว่าหมายเลขกลุ่มของจำนวนเต็มนั้นตรงกับหมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกหรือไม่ ถ้าตรงกัน แสดงว่าไม่มีความผิดพลาดเกิดขึ้น แต่ถ้าหากไม่ตรงกัน แสดงว่ามีความผิดพลาดเกิดขึ้นกับรูปเศษเหลือ ขั้นตอนวิธีที่กล่าวมา สามารถเขียนได้ตามรูปที่ 6

Algorithm 5 Error detection process for proposed RRNS

Input: pairwise co-prime moduli sequence (m_1, ..., m_k), Hamming distance d, and RRNS representations (x_1, ..., x_k)
Output: Error detection result

1: **for** i = 1 to d **do**
2: $g_i \leftarrow \lfloor x_i / m_i \rfloor$
3: **end for**
4: **if** all g_i are not the same **then**
5: **return** "Error detected"
6: **end if**
7: $X \leftarrow \mathrm{CRT}(m_1, ..., m_k, x_1, ..., x_k)$
8: $E \leftarrow \Pi_{(i=1,...,k-d+1)}\, m_i$
9: $g_x \leftarrow \lfloor X/E \rfloor$
10: **if** g_x is not equal to other g_i **then**
11: **return** "Error detected"
12: **end if**
13: **return** "No error detected"

รูปที่ 6 ขั้นตอนวิธีในการตรวจจับความผิดพลาด ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

ทฤษฎีบท 44 ขั้นตอนวิธีในรูปที่ 6 สามารถทำงานได้ถูกต้องกับระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

บทพิสูจน์ แบ่งการพิสูจน์เป็น 2 กรณีดังนี้

ในกรณีที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกไม่ตรงกัน เราสามารถสรุปได้ทันทีว่ามีความผิดพลาดเกิดขึ้นกับรูปเศษเหลือ เนื่องจากผลลัพธ์ที่ถูกต้องของการประมวลผลทั้งหมด ได้แก่ การแปลงจำนวนเต็มให้อยู่ในรูปเศษเหลือ การบวก และการคูณ จะต้องมีหมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกตรงกัน

ในกรณีที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกตรงกัน สมมติให้จำนวนเต็มของรูปเศษเหลือก่อนที่จะเกิดความผิดพลาดมีค่าเท่ากับ X หมายเลขกลุ่มที่คำนวณได้ตรงกันมีค่าเท่ากับ g และเมื่อแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มตามขั้นตอนวิธีในบรรทัดที่ 7 ได้ผลลัพธ์เป็นจำนวนเต็ม X'

หากมีความผิดพลาดเกิดขึ้นอย่างน้อย 1 ตำแหน่ง แต่ไม่เกิน d−1 ตำแหน่ง ตามทฤษฎีบท 27 จะได้ว่า $|X - X'| \geq E$ นั่นคือ X และ X' ต้องมีหมายเลขกลุ่มที่ต่างกัน และเนื่องจากหมายเลข

กลุ่มที่คำนวณได้จากค่าเศษเหลือทั้ง d ตัวตรงกัน จึงสามารถสรุปได้ทันทีว่า หมายเลขกลุ่ม g ที่คำนวณได้ตรงกันนั้น เป็นหมายเลขกลุ่มที่ถูกต้องของ X (เพราะความผิดพลาดเกิดขึ้นไม่เกิน d–1 ตำแหน่ง จึงไม่สามารถทำให้หมายเลขกลุ่มของเศษเหลือ d ตัวแรกเปลี่ยนไปทั้งหมดได้) หากพบว่าหมายเลขกลุ่มของ X' ไม่ตรงกับ g จึงสามารถสรุปได้ทันทีว่ามีความผิดพลาดเกิดขึ้น

จากทั้ง 2 กรณี สามารถสรุปได้ว่าขั้นตอนวิธีในรูปที่ 6 สามารถทำงานได้ถูกต้องกับระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ ■

ขั้นตอนวิธีในรูปที่ 6 สามารถปรับปรุงให้มีประสิทธิภาพมากขึ้นได้ โดยในกรณีที่หมายเลขกลุ่มของเศษเหลือ d ตัวแรกตรงกัน จะแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มตามขั้นตอนวิธีในรูปที่ 3 และจะนำจำนวนเต็มที่ได้มาตรวจสอบความสัมพันธ์สมภาคกับเศษเหลือ (x_{k-d+2}, ..., x_k) ว่าตรงกันหรือไม่ ถ้าตรงกัน แสดงว่าไม่มีความผิดพลาดเกิดขึ้น แต่ถ้าหากไม่ตรงกัน แสดงว่ามีความผิดพลาดเกิดขึ้นกับรูปเศษเหลือ วิธีนี้จะทำให้ลดการคำนวณตามทฤษฎีบทเศษเหลือของจีนจากเดิมต้องใช้ค่าเศษเหลือ k ตัว ลดลงเป็น k–d+1 ตัว ขั้นตอนวิธีที่กล่าวมา สามารถเขียนได้ตามรูปที่ 7

Algorithm 6 Improved error detection process for proposed RRNS

Input: pairwise co-prime moduli sequence (m_1, ..., m_k), Hamming distance d, and RRNS representations (x_1, ..., x_k)
Output: Error detection result

1: **for** i = 1 to d **do**
2: $g_i \leftarrow \lfloor x_i / m_i \rfloor$
3: **end for**
4: **if** all g_i are not the same **then**
5: **return** "Error detected"
6: **end if**
7: $h \leftarrow \mathrm{CRT}(m_1, ..., m_{k-d+1}, x_1, ..., x_{k-d+1})$
8: $E \leftarrow \Pi_{(i=1,...,k-d+1)}\, m_i$
9: $X \leftarrow h + g_1 \times E$
10: **for** $i = k-d+2$ to k **do**
11: **if** $(X \bmod m_i) \neq x_i$ **then**
12: **return** "Error detected"
13: **end if**
14: **end for**
15: **return** "No error detected"

รูปที่ 7 ขั้นตอนวิธีในการตรวจจับความผิดพลาดฉบับปรับปรุงของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

ทฤษฎีบท 45 ขั้นตอนวิธีในรูปที่ 7 สามารถทำงานได้ถูกต้องกับระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

บทพิสูจน์ พิจารณาเฉพาะกรณีที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกตรงกัน สมมติให้จำนวนเต็มของรูปเศษเหลือก่อนที่จะเกิดความผิดพลาดมีค่าเท่ากับ X หมายเลขกลุ่มที่คำนวณได้ตรงกันมีค่าเท่ากับ g และเมื่อแปลงรูปเศษเหลือที่มีความผิดพลาด $(x'_1, x'_2, \ldots, x'_k)$ กลับเป็นจำนวนเต็มตามขั้นตอนวิธีในบรรทัดที่ 7 – 9 ได้ผลลัพธ์เป็นจำนวนเต็ม X'

การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มโดยใช้การคำนวณจากหมายเลขกลุ่มและค่าเศษเหลือ k–d+1 ตัวแรก $(x'_1, x'_2, \ldots, x'_{k-d+1})$ โดยไม่สนใจค่าเศษเหลือ d–1 ตัวที่เหลือ ตามขั้นตอนวิธีในบรรทัดที่ 7 – 9 เป็นการตีความว่า X' มีหมายเลขกลุ่มเดียวกันกับ X คือหมายเลขกลุ่มเท่ากับ g เมื่อคำนวณรูปเศษเหลือของ X และ X' โดยไม่รวมหมายเลขกลุ่มเข้าไป ตามทฤษฎีบท 27 จะได้ว่ารูปเศษเหลือที่ยังไม่รวมหมายเลขกลุ่มเข้าไปนี้ จะมีระยะทางแฮมมิงระหว่างกันตั้งแต่ d ขึ้นไป แต่เนื่องจาก X' เกิดจากความผิดพลาดไม่เกิน d–1 ตำแหน่งจาก X ดังนั้นเป็นไปไม่ได้ที่ X' จะมีหมายเลขกลุ่มเท่ากับ g นั่นคือ ในกลุ่ม g รูปเศษเหลือ k–d+1 ตัวแรก $(x'_1, x'_2, \ldots, x'_{k-d+1})$ จะต้องไม่ตามด้วยเศษเหลือ $(x'_{k-d+2}, \ldots, x'_k)$ เมื่อนำ X' มาตรวจสอบความสัมพันธ์สมภาคกับเศษเหลือ $(x'_{k-d+2}, \ldots, x'_k)$ จะต้องเกิดความผิดพลาดขึ้นอย่างแน่นอน ■

อีกวิธีการหนึ่งในการตรวจจับความผิดพลาดคือ ในกรณีที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกตรงกัน และหมายเลขกลุ่มมีค่าเท่ากับ g ให้นำรูปเศษเหลือของ $g \times E$ ไปลบออกจากรูปเศษเหลือที่ต้องการตรวจสอบ การทำเช่นนี้จะเป็นการแปลงระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ ให้กลายเป็นระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิและเมสตรินิ หลังจากนั้น สามารถใช้วิธีการในงานวิจัยต่าง ๆ [3] [4] [5] [6] [7] ในการตรวจจับความผิดพลาดได้

ตัวอย่าง 46 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อนที่มีลำดับตัวหาร (5, 7, 8, 9, 11, 13) และระยะทางแฮมมิงเท่ากับ 3 จะได้ว่าระบบสามารถตรวจจับความผิดพลาดได้ d – 1 = 2 ตำแหน่ง ในตัวอย่างนี้จะอธิบายการตรวจจับความผิดพลาดโดยใช้รูปเศษเหลือของ 5678 คือ (13, 15, 22, 8, 2, 10)

ในกรณีที่รูปเศษเหลือเกิดความผิดพลาดเป็น (15, 15, 22, 8, 2, 10) เมื่อคำนวณหมายเลขกลุ่มที่เศษเหลือ d = 3 ตัวแรก จะได้หมายเลขกลุ่มเป็น 3, 2 และ 2 ตามลำดับ เมื่อหมายเลขกลุ่มไม่ตรงกัน จึงสามารถสรุปได้ทันทีว่ามีข้อผิดพลาดเกิดขึ้น

ในกรณีที่รูปเศษเหลือเกิดความผิดพลาดเป็น (14, 15, 22, 8, 2, 10) เมื่อคำนวณหมายเลขกลุ่มที่เศษเหลือ d = 3 ตัวแรก จะได้หมายเลขกลุ่มเป็น 2 ทั้งหมด การที่หมายเลขกลุ่มตรงกันยังไม่เพียงพอที่จะสรุปว่าไม่มีข้อผิดพลาดเกิดขึ้น จึงต้องตรวจสอบต่อไป

ถ้าใช้ขั้นตอนวิธีการตรวจจับข้อผิดพลาดตามรูปที่ 6 เมื่อนำค่าเศษเหลือทั้ง 6 ตัว (14, 15, 22, 8, 2, 10) มาคำนวณตามทฤษฎีบทเศษเหลือของจีน จะได้จำนวนเต็มคือ 221894 ซึ่งจำนวนเต็มนี้ไม่ได้มีหมายเลขกลุ่มเท่ากับ 2 ดังนั้นจึงสรุปได้ว่ามีข้อผิดพลาดเกิดขึ้น

ถ้าใช้ขั้นตอนวิธีการตรวจจับข้อผิดพลาดตามรูปที่ 7 เมื่อนำค่าเศษเหลือ 4 ตัวแรก (14, 15, 22, 8) มาคำนวณตามทฤษฎีบทเศษเหลือของจีน จะได้จำนวนเต็มคือ 134 ซึ่งเป็นอันดับของรูปเศษเหลือในกลุ่มที่ 2 ดังนั้นจะได้จำนวนเต็มของรูปเศษเหลือนี้คือ $h + g \times E = 134 + 2 \times 2520 = 5174$ เมื่อนำ 5174 ไปหารด้วยตัวหาร 11 และ 13 จะได้เศษเหลือเท่ากับ 4 และ 0 ตามลำดับ ซึ่งไม่ตรงกับค่าเศษเหลือ 2 ตัวหลัง (2, 10) ดังนั้นจึงสรุปได้ว่ามีข้อผิดพลาดเกิดขึ้น

อีกวิธีหนึ่งคือนำรูปเศษเหลือ (14, 15, 22, 8, 2, 10) มาลบออกด้วยรูปเศษเหลือของค่า $g \times E = 2 \times 2520 = 5040$ ซึ่งก็คือ (10, 14, 16, 0, 2, 9) ได้เป็น (4, 1, 6, 8, 0, 1) จากนั้นจึงใช้วิธีในงานวิจัยอื่น ๆ เพื่อตรวจสอบข้อผิดพลาดที่เกิดขึ้นกับรูปเศษเหลือ (4, 1, 6, 8, 0, 1) ในระบบที่มีลำดับตัวหารข้อมูลเป็น (5, 7, 8, 9) และลำดับตัวหารซ้ำซ้อนเป็น (11, 13) □

สำหรับการแก้ไขความผิดพลาด จะใช้วิธีการที่คล้ายกับการตรวจจับความผิดพลาด เนื่องจากระบบจะสามารถแก้ไขข้อผิดพลาดได้เมื่อมีความผิดพลาดเกิดขึ้นไม่เกิน $\lfloor (d - 1) / 2 \rfloor$ ตำแหน่ง ดังนั้นเราสามารถหาหมายเลขกลุ่มที่ถูกต้องได้ โดยพิจารณาหมายเลขกลุ่มของเศษเหลือ d ตัวแรก และเลือกหมายเลขกลุ่มที่ปรากฏมากที่สุด หมายเลขกลุ่มนั้นจะต้องเป็นหมายเลขกลุ่มที่ถูกต้อง หลังจากนั้น ให้นำรูปเศษเหลือของ $g \times E$ ไปลบออกจากรูปเศษเหลือที่ต้องการแก้ไข และใช้วิธีการในงานวิจัยต่าง ๆ [3] [4] [5] [6] [7] ในการแก้ไขความผิดพลาด สุดท้ายจึงนำรูปเศษเหลือของ $g \times E$ ไปบวกเข้ากับรูปเศษเหลือที่แก้ไขแล้ว ได้เป็นผลลัพธ์การแก้ไขรูปเศษเหลือตามที่ต้องการ

4.8 การเปรียบเทียบค่าในรูปเศษเหลือ

โดยปกติแล้ว การเปรียบเทียบค่าในระบบจำนวนเศษเหลือเป็นสิ่งที่ทำได้ยาก แต่การเพิ่มหมายเลขกลุ่มเข้าไปในรูปเศษเหลือทำให้การเปรียบเทียบค่าในรูปเศษเหลือทำได้สะดวกมากขึ้น ในการเปรียบเทียบรูปเศษเหลือของจำนวนเต็ม X และ Y ซึ่งก็คือ $(x_1, x_2, \ldots, x_k)$ และ $(y_1, y_2, \ldots, y_k)$ เราสามารถเปรียบเทียบหมายเลขกลุ่มของ X และ Y ก่อน หากหมายเลขกลุ่มของ X และ Y แตกต่างกัน ค่าที่มีหมายเลขกลุ่มมากกว่าก็จะมีค่ามากกว่า จะเห็นได้ว่าถ้ามีจำนวนกลุ่มมาก โอกาสที่ X และ Y จะอยู่ต่างกลุ่มกันก็มีมากขึ้น ทำให้การเปรียบเทียบเป็นไปได้รวดเร็ว สำหรับในกรณีที่หมายเลขกลุ่มเท่ากัน ให้เปรียบเทียบอันดับของรูปเศษเหลือในกลุ่มตามทฤษฎีบทเศษเหลือของจีน โดยใช้ค่าเศษเหลือ k–d+1 ตัวแรก พิจารณาขั้นตอนวิธีตามรูปที่ 8 และตัวอย่างการเปรียบเทียบค่าดังนี้

Algorithm 7 Value comparison process for proposed RRNS

Input: pairwise co-prime moduli sequence $(m_1, ..., m_k)$, Hamming distance d, RRNS representations of X and Y as $(x_1, ..., x_k)$ and $(y_1, ..., y_k)$
Output: Comparison result

$g_x \leftarrow \lfloor x_1 / m_1 \rfloor$
$g_y \leftarrow \lfloor y_1 / m_1 \rfloor$
if $g_x > g_y$ **then**
    return "$X > Y$"
end if
if $g_x < g_y$ **then**
    return "$X < Y$"
end if
$h_x \leftarrow \text{CRT}(m_1, ..., m_{k-d+1}, x_1, ..., x_{k-d+1})$
$h_y \leftarrow \text{CRT}(m_1, ..., m_{k-d+1}, y_1, ..., y_{k-d+1})$
if $h_x > h_y$ **then**
    return "$X > Y$"
end if
if $h_x < h_y$ **then**
    return "$X < Y$"
end if
return "$X = Y$"

รูปที่ 8 ขั้นตอนวิธีในการเปรียบเทียบค่าในรูปเศษเหลือ ของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

ตัวอย่าง 47 กำหนดระบบจำนวนเศษเหลือซ้ำซ้อนที่มีลำดับตัวหาร (2, 3, 5, 7) และระยะทางแฮมมิงมีค่าเท่ากับ 2 ถ้าต้องการเปรียบเทียบค่าของรูปเศษเหลือ (3, 5, 2, 5) และ (4, 6, 3, 1) ให้เริ่มจากการคำนวณหมายเลขกลุ่มก่อน จะเห็นว่าหมายเลขกลุ่มของรูปเศษเหลือทั้งสองคือ 1 และ 2 ตามลำดับ ดังนั้นสามารถสรุปได้ว่าค่าของรูปเศษเหลือ (3, 5, 2, 5) น้อยกว่า (4, 6, 3, 1) เมื่อพิจารณาค่าที่เป็นจำนวนเต็มจะพบว่าค่าของรูปเศษเหลือทั้งสองคือ 47 และ 78 ตามลำดับ

ถ้าต้องการเปรียบเทียบค่าของรูปเศษเหลือ (7, 9, 1, 6) และ (6, 10, 0, 2) ให้เริ่มจากการคำนวณหมายเลขกลุ่มก่อน จะเห็นว่าหมายเลขกลุ่มของรูปเศษเหลือทั้งสองคือ 3 เท่ากัน จึงคำนวณอันดับในกลุ่มเป็นขั้นตอนต่อไป พบว่าอันดับในกลุ่มของรูปเศษเหลือทั้งสองคือ 21 และ 10 ตามลำดับ ดังนั้นสามารถสรุปได้ว่าค่าของรูปเศษเหลือ (7, 9, 1, 6) มากกว่า (6, 10, 0, 2) เมื่อพิจารณาค่าที่เป็นจำนวนเต็มจะพบว่าค่าของรูปเศษเหลือทั้งสองคือ 111 และ 100 ตามลำดับ □

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

ในบทนี้ จะทำการประเมินระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ และนำระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ มาเปรียบเทียบกับระบบจำนวนเศษเหลือแบบเดิม ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ – เมสตรินิ และระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ ด้วยข้อเปรียบเทียบต่าง ๆ

5.1 การประเมินระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ

จากบทที่ 4 จะเห็นได้ว่าระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอรองรับการคำนวณต่าง ๆ ที่จำเป็น มีบางการประมวลผลที่สามารถลดจำนวนเศษเหลือที่ใช้ในการคำนวณ เมื่อเปรียบเทียบกับระบบจำนวนเศษเหลือซ้ำซ้อนอื่น ๆ เช่น การแปลงจำนวนในรูปเศษเหลือกลับเป็นจำนวนเต็ม และการตรวจจับความผิดพลาด เป็นต้น นอกจากนี้การเปรียบเทียบค่าในรูปเศษเหลือยังสามารถทำได้สะดวกมากขึ้นด้วย อย่างไรก็ตามวิธีในการคำนวณผลบวกและผลคูณมีความซับซ้อนมากขึ้น

จากขั้นตอนวิธีการบวกและการคูณในรูปที่ 4 และรูปที่ 5 จะเห็นว่าการคำนวณต้องใช้เวลามากขึ้น เนื่องจากมีการแปลงรูปเศษเหลือกลับเป็นจำนวนเต็มด้วยทฤษฎีบทเศษเหลือของจีน อย่างไรก็ตาม บางส่วนของการคำนวณนี้สามารถคำนวณค่าไว้ก่อนล่วงหน้าได้ และสามารถใช้การประมวลผลแบบขนานเพื่อช่วยลดเวลาในการคำนวณได้

ระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ มีข้อสังเกตในการเลือกค่าระยะทางแฮมมิงอยู่บางประการ การเลือกระยะทางแฮมมิงให้มีค่าเท่ากับ d จะทำให้การคำนวณตามทฤษฎีบทเศษเหลือของจีน ใช้ค่าเศษเหลือ k–d+1 ตัว ดังนั้นการเลือกระยะทางแฮมมิง d ให้มีค่ามาก จะทำให้จำนวนค่าเศษเหลือที่ต้องใช้ในการคำนวณตามทฤษฎีบทเศษเหลือของจีนลดลง การคำนวณผลบวกและผลคูณก็จะทำได้รวดเร็วขึ้น นอกจากนี้ ในการแบ่งกลุ่มของรูปเศษเหลือ จะได้จำนวนกลุ่มทั้งหมด $\Pi_{(i=k-d+2,\ldots,k)}$ m_i กลุ่ม การเลือกระยะทางแฮมมิง d ให้มีค่ามาก จะทำให้มีจำนวนกลุ่มของรูปเศษเหลือมาก ซึ่งจะส่งผลให้เมื่อเกิดข้อผิดพลาดกับรูปเศษเหลือ มีโอกาสมากขึ้นที่หมายเลขกลุ่มที่คำนวณได้จากค่าเศษเหลือ d ตัวแรกจะมีค่าไม่ตรงกัน และทำให้ตรวจจับความผิดพลาดได้รวดเร็ว การที่มีจำนวนกลุ่มมาก ยังส่งผลให้การเปรียบเทียบค่าในรูปเศษเหลือสามารถทำได้รวดเร็วขึ้นด้วย ในขณะเดียวกัน การที่ระยะทางแฮมมิง d มีค่ามาก ก็จะส่งผลให้จำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือมีค่ามาก (จะแสดงการคำนวณในหัวข้อถัดไป) ผู้ที่สนใจนำระบบจำนวนเศษเหลือที่นำเสนอไปปรับใช้ควรคำนึงถึงผลกระทบต่าง ๆ ที่จะเกิดขึ้นจากการเลือกค่าระยะทางแฮมมิงด้วย

5.2 การเปรียบเทียบจำนวนบิตที่ใช้

หัวข้อนี้จะทำการเปรียบเทียบจำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือของระบบจำนวนเศษเหลือแบบต่าง ๆ กำหนดให้ลำดับตัวหารที่จะพิจารณาคือ $(m_1, m_2, \ldots m_k)$ โดย m_i แต่ละตัวเป็นจำนวนเต็มบวกและเป็นจำนวนเฉพาะสัมพัทธ์กันทั้งหมด และ $m_i < m_{i+1}$ สำหรับทุกค่า i ในช่วง $1 \le i \le k-1$ สำหรับระบบจำนวนเศษเหลือซ้ำซ้อน กำหนดให้มีค่าระยะทางแฮมมิงเท่ากับ d และเพื่อให้สะดวกในการเปรียบเทียบ กำหนดให้ $E = \prod_{(i=1,\ldots,k-d+1)} m_i$ และ $G = \prod_{(i=k-d+2,\ldots,k)} m_i$

สำหรับระบบจำนวนเศษเหลือปกติ ระบบสามารถแทนค่าจำนวนเต็มได้ $\prod_{(i=1,\ldots,k)} m_i = EG$ รูปแบบ และมีค่าระยะทางแฮมมิงเป็น 1 จำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือสามารถประมาณได้เท่ากับ $\log_2(m_1) + \log_2(m_2) + \ldots + \log_2(m_k) = \log_2(\prod_{(i=1,\ldots,k)} m_i) = \log_2(EG)$

สำหรับระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ – เมสตรินิ จำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือสามารถประมาณได้เท่ากับ $\log_2(EG)$ เช่นเดียวกับระบบจำนวนเศษเหลือปกติ แต่ระบบสามารถแทนค่าจำนวนเต็มได้เพียง $\prod_{(i=1,\ldots,k-d+1)} m_i = E$ รูปแบบ และมีค่าระยะทางแฮมมิงเป็น d

สำหรับระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ จากวิธีการสร้างในทฤษฎีบท 30 ที่ว่า ให้สร้างลำดับตัวหารใหม่จากลำดับตัวหารเดิม ลำดับตัวหารใหม่นี้จะมีจำนวนตัวหารเท่ากับลำดับตัวหารเดิม โดยตัวหารใหม่แต่ละตัวเกิดจากการนำตัวหารเดิมที่แตกต่างกัน d ตัวมาคูณกัน และตัวหารเดิมจะต้องถูกใช้เป็นตัวคูณทั้งหมด d ครั้งพอดี ดังนั้นเมื่อคิดจำนวนบิตแล้ว จำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือสามารถประมาณได้เท่ากับ $\log_2(\prod_{(i=1,\ldots,k)} m_i^d) = \log_2(E^d G^d)$ โดยระบบสามารถแทนค่าจำนวนเต็มได้ $\prod_{(i=1,\ldots,k)} m_i = EG$ รูปแบบ และมีค่าระยะทางแฮมมิงเป็น d

สำหรับระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ เมื่อแบ่งรูปเศษเหลือออกเป็นกลุ่ม กลุ่มละ $E = \prod_{(i=1,\ldots,k-d+1)} m_i$ จำนวน จะได้ทั้งหมด $G = \prod_{(i=k-d+2,\ldots,k)} m_i$ กลุ่ม ในการรวมหมายเลขกลุ่มเข้าไปในรูปเศษเหลือ จะทำการบวกผลคูณของหมายเลขกลุ่มกับตัวหารเข้าไปในค่าเศษเหลือ d ตัวแรก ดังนั้นเมื่อพิจารณาค่าเศษเหลือตัวที่ 1 ถึง d ค่าเศษเหลือที่เป็นไปได้สำหรับเศษเหลือแต่ละตัวจะมี $G \times m_i$ รูปแบบ ส่วนค่าเศษเหลือตัวที่ d+1 ถึง k แต่ละตัวยังคงเป็นได้ m_i รูปแบบดังเดิม ดังนั้นจำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือสามารถประมาณได้เท่ากับ $\log_2(G \times m_1) + \ldots + \log_2(G \times m_d) + \log_2(m_{d+1}) + \ldots + \log_2(m_k) = \log_2(\prod_{(i=1,\ldots,d)} Gm_i \times \prod_{(i=d+1,\ldots,k)} m_i) = \log_2(EG^{d+1})$ โดยระบบสามารถแทนค่าจำนวนเต็มได้ $\prod_{(i=1,\ldots,k)} m_i = EG$ รูปแบบ และมีค่าระยะทางแฮมมิงเป็น d

การเปรียบเทียบจำนวนบิตที่ใช้ สามารถสรุปได้ดังตารางที่ 3

ตารางที่ 3 การเปรียบเทียบจำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือ
ของระบบจำนวนเศษเหลือทั้ง 4 ระบบ

ข้อเปรียบเทียบ	ระบบจำนวนเศษเหลือ	ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ–เมสตรินิ	ระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ	ระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ
รูปแบบจำนวนเต็มที่สามารถแทนได้	EG	E	EG	EG
ระยะทางแฮมมิง	1	d	d	d
จำนวนบิตที่ใช้ในการเก็บรูปเศษเหลือ	$\log_2(EG)$	$\log_2(EG)$	$\log_2(E^dG^d)$	$\log_2(EG^{d+1})$

จากค่าต่าง ๆ จะเห็นว่าระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอใช้จำนวนบิตมากกว่าระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ–เมสตรินิ ในทุกกรณี แต่จำนวนเต็มที่ระบบที่เสนอสามารถแทนได้ก็มากกว่าเช่นกัน เมื่อเปรียบเทียบกับระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ ระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอจะใช้จำนวนบิตน้อยกว่าเมื่อ $E^dG^d > EG^{d+1}$ หรือก็คือ $E^{d-1} > G$ อสมการนี้มีโอกาสสูงที่จะเป็นจริง เนื่องจากการเลือกลำดับตัวหาร มักเลือกให้ตัวหารแต่ละตัวมีค่าใกล้เคียงกัน [11] ดังนั้นระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอจะใช้จำนวนบิตน้อยกว่าระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติในเกือบทุกกรณี

5.3 ข้อเปรียบเทียบอื่น ๆ

ระบบจำนวนเศษเหลือทั้งหมดที่ได้กล่าวมา มีข้อดีและข้อจำกัดที่แตกต่างกัน ตารางที่ 4 สรุปข้อดีและข้อจำกัดของระบบต่าง ๆ โดยเครื่องหมาย ✓ ในตารางแสดงข้อดีของระบบนั้น ๆ หัวข้อที่พิจารณาได้แก่ความสามารถในการตรวจจับและแก้ไขความผิดพลาด ค่าของตัวหารและเศษเหลือในระบบ และประสิทธิภาพในการคำนวณต่าง ๆ

ตารางที่ 4 ข้อเปรียบเทียบของระบบจำนวนเศษเหลือทั้ง 4 ระบบ

ข้อเปรียบเทียบ	ระบบจำนวนเศษเหลือ	ระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ–เมสตรินิ	ระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ	ระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอ
สามารถตรวจจับและแก้ไขความผิดพลาดได้		✓	✓	✓
ไม่ต้องเพิ่มตัวหารซ้ำซ้อน	✓		✓	✓
ตัวหารมีค่าน้อย	✓	✓		✓
เศษเหลือที่เก็บมีค่าน้อย	✓	✓		
สามารถทำการคำนวณพื้นฐานได้รวดเร็ว	✓	✓	✓	
สามารถเปรียบเทียบค่าในรูปเศษเหลือได้รวดเร็ว				✓

5.4 สรุปผลการวิจัย และข้อเสนอแนะ

ผู้วิจัยได้เสนอแนวทางใหม่ในการสร้างระบบจำนวนเศษเหลือซ้ำซ้อน โดยการใช้ค่าเศษเหลือที่มีความซ้ำซ้อน วิธีการนี้ได้ใช้แนวคิดการแบ่งรูปเศษเหลือออกเป็นกลุ่มย่อย ๆ และใช้ค่าเศษเหลือเพิ่มเติมจากที่ใช้กันโดยทั่วไป วิธีการที่เสนอนี้ ไม่ต้องมีการเพิ่มตัวหารเหมือนกับระบบจำนวนเศษเหลือซ้ำซ้อนของบาร์ซิ–เมสตรินิ และใช้ตัวหารที่มีค่าไม่มาก เมื่อเปรียบเทียบกับระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ ค่าที่มีผลต่อประสิทธิภาพของระบบที่เห็นได้ชัดเจนคือค่าระยะทางแฮมมิง ซึ่งผู้ที่ต้องการนำระบบที่เสนอไปปรับใช้ ควรเลือกใช้ค่าต่าง ๆ ให้เหมาะสมกับงานที่ต้องการ

เมื่อเปรียบเทียบกับระบบจำนวนเศษเหลืออื่น ๆ บางขั้นตอนวิธีของระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอสามารถลดจำนวนเศษเหลือที่ต้องใช้ในการคำนวณตามทฤษฎีบทเศษเหลือของจีนได้ เช่น การแปลงรูปเศษเหลือกลับเป็นจำนวนเต็ม เป็นต้น การเพิ่มหมายเลขกลุ่มเข้าไปในรูปเศษเหลือทำให้สามารถตรวจจับข้อผิดพลาด และเปรียบเทียบค่าในรูปเศษเหลือได้รวดเร็วมากขึ้น และเมื่อเปรียบเทียบกับระบบจำนวนเศษเหลือซ้ำซ้อนของแคตติ ที่มีจำนวนรูปแบบการแทนจำนวนเต็มเท่ากัน พบว่าระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอใช้จำนวนบิตในการแทนรูปเศษเหลือน้อยกว่าระบบของแคตติในเกือบทุกกรณี อย่างไรก็ตาม ขั้นตอนวิธีในการบวกและการคูณของระบบที่เสนอมีความซับซ้อนมากกว่าระบบจำนวนเศษเหลืออื่น ๆ

สำหรับประเด็นที่สามารถนำระบบจำนวนเศษเหลือซ้ำซ้อนที่เสนอไปพัฒนาต่อ ได้แก่ การปรับปรุงขั้นตอนวิธีการบวกและการคูณ ให้สามารถทำงานได้ถูกต้อง โดยไม่ต้องแปลงรูป

เศษเหลือกลับเป็นจำนวนเต็ม หรืออาจจะปรับแนวคิดของหมายเลขกลุ่ม และวิธีการรวมหมายเลขกลุ่มเข้าไปในรูปเศษเหลือ เพื่อให้สามารถทำการบวกและการคูณได้ง่ายขึ้นก็ได้

บรรณานุกรม

1. Omondi, A.R. and B. Premkumar, *Residue number systems: theory and implementation*. Advances in Computer Science and Engineering: Texts. Vol. 2. 2007: Imperial College Press.
2. Mohan, P.V.A., *Residue number systems: algorithms and architectures*. Vol. 677. 2002: Springer Science & Business Media.
3. Barsi, F. and P. Maestrini, *Error correcting properties of redundant residue number systems.* IEEE Transactions on Computers, 1973. **C-22**(3): p. 307-315.
4. Mandelbaum, D.M., *On a class of arithmetic codes and a decoding algorithm (Corresp.).* IEEE Transactions on Information Theory, 1976. **22**(1): p. 85-88.
5. Goldreich, O., D. Ron, and M. Sudan, *Chinese remaindering with errors.* IEEE Transactions on Information Theory, 2000. **46**(4): p. 1330-1338.
6. Goh, V.T. and M.U. Siddiqi, *Multiple error detection and correction based on redundant residue number systems.* IEEE Transactions on Communications, 2008. **56**(3): p. 325-330.
7. Tay, T.F. and C.H. Chang, *A non-iterative multiple residue digit error detection and correction algorithm in RRNS.* IEEE Transactions on Computers, 2016. **65**(2): p. 396-408.
8. Katti, R.S., *A new residue arithmetic error correction scheme.* IEEE Transactions on Computers, 1996. **45**(1): p. 13-19.
9. Szabo, N.S. and R.I. Tanaka, *Residue arithmetic and its applications to computer technology*. 1967: McGraw-Hill.
10. Aydin, N., *An introduction to coding theory via Hamming codes: a computational science model*. 2007: Kenyon College.
11. Wang, J., et al. *A systemic performance evaluation method for residue number system*. in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. 2016. IEEE.

ประวัติผู้เขียน

ชื่อ-สกุล	กิตติภพ พละการ
วัน เดือน ปี เกิด	16 ธันวาคม 2537
สถานที่เกิด	กรุงเทพมหานคร
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	27/14 ซอยนวลจันทร์ 21 ถนนนวลจันทร์ แขวงนวลจันทร์ เขตบึงกุ่ม กรุงเทพมหานคร 10230