อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง

นายพิสุทธิ์ รัตนนัฏ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2547

ISBN 974-53-1916-3

DIVISION ALGORITHM FOR CONTINUOUS VALUED NUMBER SYSTEM

Mr. Pisut Rattananat

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 2004
ISBN 974-53-1916-3

หัวข้อวิทยานิพนธ์	อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง
โดย	นายพิสุทธิ์ รัตนนัฏ
สาขาวิชา	วิทยาศาสตร์คอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.... คณบดีคณะวิศวกรรมศาสตร์

(ศาสตราจารย์ ดร. ดิเรก ลาวัณย์ศิริ)

คณะกรรมการสอบวิทยานิพนธ์

.... ประธานกรรมการ

(รองศาสตราจารย์ ดร. วันชัย ริ้วไพบูลย์)

.... อาจารย์ที่ปรึกษา

(อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์)

.... กรรมการ

(อาจารย์ ดร. เศรษฐา ปานงาม)

.... กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. อานนท์ รุ่งสว่าง)

นายพิสุทธิ์ รัตนนัฏ : อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง (DIVISION ALGORITHM FOR CONTINUOUS VALUED NUMBER SYSTEM) อ. ที่ปรึกษา : อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์, 59 หน้า. 974-53-1916-3.

ระบบสัญญาณแอนะล็อกมีการคำนึงถึงการลดพลังงานและสัญญาณรบกวน เนื่องจากสัญญาณรบกวนทำให้เกิดค่าความผิดพลาดในระบบการคำนวณแอนะล็อก ระบบจำนวนค่าต่อเนื่องจึงได้ถูกนำเสนอขึ้นมาเพื่อทำการกู้ค่าความผิดพลาดที่เกิดขึ้นจากสัญญาณรบกวน ระบบจำนวนนี้ถูกออกแบบมาสำหรับการคำนวณ ที่มีสัญญาณรบกวนต่ำโดยใช้ระบบจำนวนและตัวดำเนินการแอนะล็อกสัญญาณ แอนะล็อกหลายสัญญาณถูกนำมาใช้ในการเพิ่มค่าความถูกต้องภายใต้ข้อจำกัดของความคลาดเคลื่อน ที่ไม่เกินไปกว่าช่วงของ ค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม ตัวดำเนินการพื้นฐานเลขคณิตเช่น การบวก การลบ และการคูณได้ถูกเสนอเช่นเดียวกัน

ในงานวิจัยนี้ เสนอว่าตัวดำเนินการหารสามารถทำได้ในระบบจำนวนค่าต่อเนื่องโดยการนำเสนอ อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง โดยการหารจะเริ่มคำนวณจากดิจิตที่มีนัยสำคัญสูงสุด ทีละดิจิต และยังแสดงให้เห็นว่า ผลลัพธ์ที่ได้มีคุณสมบัติของการเป็นรูปแบบแทนจำนวนค่าต่อเนื่อง รวมถึงการวิเคราะห์ความสัมพันธ์ระหว่างจำนวนดิจิตที่ใช้เทียบกับค่าความผิดพลาด

ภาควิชาวิศวกรรมคอมพิวเตอร์ ลายมือชื่อนิสิต...
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่ออาจารย์ที่ปรึกษา...............................
ปีการศึกษา 2547

4671424821 : MAJOR COMPUTER SCIENCE

KEY WORD: CONTINUOUS VALUED NUMBER SYSTEM / ANALOG CIRCUIT / ERROR TOLERANCE

PISUT RATTANANAT: DIVISION ALGORITHM FOR CONTINUOUS VALUED NUMBER SYSTEM. THESIS ADVISOR: ATHASIT SURARERKS, Ph.D., 59 pp. ISBN 974-53-1916-3.

Analog circuit concerns about both of power reduction and system noise. Since system noise cause errors in the analog computing system, continuous valued number system (CVNS) has been introduced in order to recover the precision from such an error. This number system is built for low noise arithmetic using analog representations and operations. Several redundant signals are proposed in order to increase the precision. The correct result is obtained only in the case that the error does not exceed the error tolerance. Some fundamental arithmetic operations which are addition, subtraction and multiplication are also presented.

In this thesis, we demonstrate that division can be performed in the system by presenting a division algorithm for continuous valued number system. Division is performed sequentially starting from the most significant digit. We also show that the result satisfies the properties of the continuous valued number system. The relationship between the number of digits and the errors is also analyzed.

Department Computer Engineering
Field of study Computer Science
Academic year 2004

Student's signature..........................

Advisor's signature..........................

กิตติกรรมประกาศ

ผู้จัดทำวิทยานิพนธ์มีความภาคภูมิใจเป็นอย่างสูงที่วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงเป็นอย่างดี ด้วยความอนุเคราะห์ และความช่วยเหลืออย่างยิ่งจาก อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษา ซึ่งเป็นผู้ถ่ายทอดวิชาต่างๆ ให้คำปรึกษา ตลอดจนเป็นผู้ตรวจทานแก้ไข ขอขอบพระคุณ อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ เป็นอย่างสูงที่ให้ความเมตตา ช่วยเหลือ ในทุกๆ ด้านแก่ผู้จัดทำวิทยานิพนธ์เสมอมา

ขอขอบพระคุณ รศ. ดร. วันชัย ริ้วไพบูลย์ อาจารย์ ดร. เศรษฐา ปานงาม คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และผศ. ดร. อานนท์ รุ่งสว่าง คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์ ที่ได้กรุณาให้คำแนะนำในการแก้ไขวิทยานิพนธ์ให้มีคุณภาพยิ่งขึ้น วิทยานิพนธ์ฉบับนี้ไม่อาจจะสำเร็จได้หากไม่ได้รับความร่วมมือจากทุกท่าน ขอขอบคุณอาจารย์ทุกๆ ท่านในสาขาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ที่สอนวิชาต่างๆ ในหลักสูตรให้กับผู้จัดทำวิทยานิพนธ์ ขอขอบคุณเพื่อนๆ ทุกๆ คน ที่ร่วมทุกข์ร่วมสุข โดยเฉพาะอย่างยิ่ง นายสิร ศรีวนาสณฑ์ ซึ่งเป็นผู้ที่ให้คำแนะนำเพิ่มเติมเป็นอย่างดีเสมอมา

ท้ายนี้ขอขอบพระคุณ คุณพ่อปชา คุณแม่สุรีย์ ที่เป็นผู้ให้กำเนิด พี่เอก และน้องแอน ที่คอยให้กำลังใจ ช่วยเหลือในทุกๆ ด้านจนสามารถทำวิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงเป็นอย่างดี ผู้จัดทำวิทยานิพนธ์หวังว่าวิทยานิพนธ์ฉบับนี้จะเป็นประโยชน์แก่ผู้ศึกษาค้นคว้างานวิจัยที่เกี่ยวข้อง เพื่อก่อให้เกิดงานวิจัยต่างๆ ที่เป็นประโยชน์ต่อไป

สารบัญ

สารบัญตาราง

ตารางที่ หน้า

ตารางที่ หน้า

สารบัญภาพ

รูปที่ หน้า

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการใช้งานทางด้านคอมพิวเตอร์มีความต้องการให้อุปกรณ์ต่างๆ เช่น ตัวประมวลผล สามารถตอบสนองการทำงานได้อย่างรวดเร็ว ดังนั้นจึงมีการวิจัยเกี่ยวกับวงจรการคำนวณเลขคณิต ประกอบด้วยการคำนวณทางดิจิทัล (digital) และแอนะล็อก (analog)

การคำนวณเลขคณิตของตัวประมวลผลในระบบดิจิทัลคอมพิวเตอร์ในปัจจุบันอยู่บนลักษณะของวงจรที่เป็น ศูนย์ กับ หนึ่ง ข้อดีคือง่ายและมีความรวดเร็ว การส่งสัญญาณจะมีลักษณะของสายที่เป็นเส้นเดียว โดยจะมีค่า ศูนย์ กับ หนึ่ง ส่งไปตามสาย แต่การบวกกันของค่าต่างๆ ในวงจรดิจิทัล จำเป็นต้องมีตัวทดทำให้เกิดความล่าช้าในการทำงานในกรณีที่ข้อมูลมีปริมาณมาก (ค่ามาก) มีงานวิจัยบางงานที่นำเสนอรูปแบบของระบบจำนวนที่ลดขนาดของการทดลงได้ เช่น ในปี ค.ศ. 1961 ระบบจำนวนของ อวีเซียนิส (Avizienis) ใน [1] ได้ถูกเสนอขึ้น โดยใช้เครื่องหมายมาช่วยในการออกแบบ เป็นต้น แต่อย่างไรก็ตามเวลาในการคำนวณก็ยังคงใช้เวลามากเมื่อเทียบกับการทำงานด้วยระบบแอนะล็อก

วงจรแอนะล็อกนั้นในการส่งข้อมูลสามารถวัดสัญญาณได้ด้วยค่าตัวแปรทางไฟฟ้า เช่น ค่าความต่างศักย์ หรือกระแส ซึ่งค่าที่วัดได้เป็นจำนวนจริง ข้อดีคือมีความรวดเร็ว ในการทำงานโดยไม่ขึ้นกับค่าของข้อมูล เช่น การบวกเลข อาจทำได้โดยการรวมกันของสัญญาณแอนะล็อกสองสัญญาณ ค่าที่วัดได้จะมีค่ามากหรือน้อย ไม่ได้ขึ้นอยู่กับจำนวนหลักของตัวเลข และเป็นการคำนวณที่ปราศจากตัวทด [2] ข้อเสียของการคำนวณเลขคณิตของวงจรแอนะล็อกคือ สัญญาณรบกวน (noise) ซึ่งไม่สามารถกำจัดสัญญาณรบกวนได้ทั้งหมด แต่สามารถออกแบบการคำนวณให้มีความถูกต้องให้ได้มากที่สุด รวมถึงลดความผิดพลาดของการคำนวณ ซึ่งมีผลมาจากสัญญาณรบกวน ทั้งนี้ เนื่องจากการไม่ทราบค่าที่แท้จริงของสัญญาณรบกวน สิ่งที่สำคัญในการคำนวณในระบบแอนะล็อกคือ การออกแบบระบบแทนจำนวนที่มีประสิทธิภาพ ที่ทนต่อสัญญาณรบกวน และสิ่งสำคัญอีกประการคือ การออกแบบวงจร การบวก การลบ การคูณ และการหาร ที่ทนต่อสัญญาณรบกวนเช่นเดียวกัน ดังนั้นจึงมีการวิจัยที่นำเสนอระบบจำนวนค่าต่อเนื่อง [3] ซึ่งเป็นระบบจำนวนที่มีความสามารถในการลดค่าความผิดพลาดที่เกิดจากสัญญาณรบกวนลงได้ โดยค่าความผิดพลาดนั้นจะต้องไม่เกินไปกว่าช่วงของค่าความผิดพลาดคลาดเคลื่อน

ที่ยินยอม (error tolerance) พร้อมกันนั้นได้มีการเสนออัลกอริทึมการคำนวณเลขคณิตพื้นฐาน คือ การบวก ลบ คูณ แต่ยังไม่มีการออกแบบอัลกอริทึมสำหรับการหาร

งานวิจัยนี้จึงได้นำเสนออัลกอริทึมการหาร ของระบบจำนวนค่าต่อเนื่องนี้ โดยการหารนี้ต้องมีคุณสมบัติคงทนต่อค่าความผิดพลาดที่อยู่ในช่วงของค่าความผิดพลาดคลาดเคลื่อนที่ยอมรับได้ นอกจากนี้ คำตอบที่ได้จากการคำนวณจำเป็นต้องสอดคล้องกับคุณสมบัติของจำนวนในระบบจำนวนค่าต่อเนื่องด้วยเช่นกัน ในงานวิจัยนี้ยังศึกษาถึงรูปแบบของค่าความผิดพลาดที่เกิดขึ้นในระบบจำนวนด้วย พร้อมทั้งศึกษาถึงความสัมพันธ์ของขนาดของความผิดพลาดเทียบกับขนาดของรูปแบบแทนจำนวน (number representation) ด้วย

1.2 วัตถุประสงค์ของการวิจัย

นำเสนออัลกอริทึมสำหรับแสดงการคำนวณของตัวดำเนินการเลขคณิตพื้นฐาน คือ การหาร ของระบบจำนวนค่าต่อเนื่อง เนื่องจากงานวิจัยระบบจำนวนค่าต่อเนื่อง ยังไม่ได้มีการนำเสนอ โดยผลลัพธ์ที่ได้จากการคำนวณยังคงสามารถที่จะกู้ค่าความผิดพลาดให้ได้ค่าที่ยอมรับได้ และผลลัพธ์ที่ได้ต้องคงคุณสมบัติความเป็นดิจิตค่าต่อเนื่องด้วยเช่นกัน

1.3. ขอบเขตของการวิจัย

1. ในการวิจัยนี้เป็นการนำเสนออัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง
2. ศึกษาความสัมพันธ์ของผลการหารเทียบกับขนาดของค่าความผิดพลาดที่เกิดจากสัญญาณรบกวน

1.4 ขั้นตอนการวิจัย

1. ศึกษาแนวคิดรวมถึงทฤษฎีงานวิจัยระบบจำนวนค่าต่อเนื่อง
2. ออกแบบอัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง
3. พิสูจน์อัลกอริทึมการหารที่ได้ออกแบบว่ามีความถูกต้อง
4. เขียนวิทยานิพนธ์ และดำเนินการสอบวิทยานิพนธ์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้อัลกอริทึมการหารที่ระบบจำนวนค่าต่อเนื่อง
2. ทฤษฎีความสัมพันธ์ของผลการหารเทียบกับขนาดของค่าความผิดพลาดที่เกิดจากสัญญาณรบกวน

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 บทนำ

ในบทนี้จะนำเสนอ ความรู้พื้นฐานที่จำเป็นต่อการทำความเข้าใจในงานวิจัยนี้ โดยเริ่มต้นจากความเป็นมาของงานวิจัยระบบแอนะล็อกและงานวิจัยที่เกี่ยวข้อง โดยเฉพาะอย่างยิ่งระบบจำนวนค่าต่อเนื่อง [2, 3, 4] พร้อมทั้งอัลกอริทึมการกู้ค่าความผิดพลาด รวมไปถึง อัลกอริทึมการคำนวณเลขคณิตพื้นฐาน คือ การบวก การลบ และการคูณ

2.2 งานวิจัยที่เกี่ยวข้อง

เนื่องจากในปัจจุบัน งานวิจัยส่วนใหญ่มีเป้าหมายในการปรับปรุงระบบจำนวนแบบดิจิทัลเป็นอย่างมาก แต่สำหรับระบบจำนวนแบบแอนะล็อก การคำนวณทางคณิตศาสตร์เป็นการคำนวณที่ปราศจากตัวทด ซึ่งบนระบบจำนวนแบบดิจิทัลจะมีการแพร่กระจายตัวทด (carry propagation) ในปี 1997 เดือนสิงหาคม Saed Ahmadi Jullien และ Miller ได้นำเสนองานวิจัยเรื่อง “Overlap Resolution: Continuous Valued Digits for Hybrid Architectures” [2] ซึ่งเป็นงานวิจัยที่เกี่ยวกับระบบจำนวนแบบแอนะล็อก โดยมีชื่อว่าระบบจำนวนมีค่าซ้อนทับกัน (overlap resolution number system) ระบบจำนวนนี้อาศัยหลักการในการเพิ่มสัญญาณเข้าไปในระบบจำนวนซึ่งสัญญาณดังกล่าวจะถูกเรียกว่าเป็นดิจิต และดิจิตแต่ละดิจิตนี้มีค่าเป็นแบบต่อเนื่อง โดยดิจิตที่เพิ่มเข้าไปนั้นจะถูกเรียกว่าดิจิตซ้ำซ้อน (redundancy digit) โดยที่ดิจิตแรกเท่านั้นที่มีหน้าที่ในการบอกค่าแอนะล็อกที่เราต้องการที่จะแสดงทั้งหมด ส่วนในดิจิตอื่นๆ จะบอกค่าในส่วนของรายละเอียดที่ลดลงไปเรื่อยๆ ซึ่งก็หมายความว่าถ้าดิจิตแรกนั้นสามารถที่จะแสดงค่าที่ถูกต้องได้แบบไม่มีความผิดพลาด ก็ไม่มีความจำเป็นที่จะต้องมีดิจิตอื่นๆ ที่เหลือ โดยดิจิตซ้ำซ้อนที่เพิ่มเข้ามานั้นมีหน้าที่ในการปรับปรุงค่าของดิจิตที่มีความผิดพลาดเกิดขึ้น โดยวิธีลดค่าความผิดพลาดนั้นจะเริ่มจากดิจิตที่มีลำดับน้อยที่สุด ทำการปรับปรุงค่าของดิจิตที่อยู่ติดกันไปเรื่อยๆ จนถึงการปรับปรุงค่าของดิจิตที่มีลำดับสูงสุด ซึ่งดิจิตลำดับสูงสุดเป็นดิจิตเพียงดิจิตเดียวที่มีหน้าที่ในการแทนค่าจำนวนที่ต้องการแสดงเอาไว้ ทำให้ค่าความซับซ้อน (complexity) ของการลดค่าความผิดพลาดนั้นเป็นแบบเชิงเส้นตรงตามจำนวนของดิจิตซ้ำซ้อน

โดยสรุปคืองานวิจัย [2] ได้เสนอระบบจำนวนที่มีค่าซ้อนทับกันที่สามารถทำให้เทคนิคในกระบวนการของระบบจำนวนแบบแอนะล็อก ถูกประยุกต์ขึ้นมาในลักษณะของดิจิตที่มีค่า

ต่อเนื่อง เพื่อทำให้การส่งค่าสัญญาณมีลักษณะขนานกันไปหลายๆ ค่าในเวลาเดียวกัน ทำให้เกิดความรวดเร็ว และทำให้ได้ค่าที่ถูกต้องเช่นเดียวกับเทคนิคของระบบจำนวนแบบดิจิทัล

ในเดือนพฤศจิกายน ปี 1997 Saed Ahmadi Jullien และ Miller ได้นำเสนองานวิจัยที่มีชื่อว่า “Overlap Resolution: Arithmetic with Continuous Valued Digits in Hybrid Architectures” [4] โดยแสดงถึงระบบจำนวนค่าซ้อนทับกัน (overlap resolution number system) เช่นเดียวกับงานวิจัย [2] แต่งานวิจัยนี้ได้แสดงถึงตัวดำเนินการทางเลขคณิตพื้นฐานสองตัว คือ การบวก และการคูณ เพื่อแสดงว่าระบบจำนวนค่าต่อเนื่องนั้น สามารถคำนวณได้บนตัวดำเนินการพื้นฐาน และเมื่อเกิดสัญญาณรบกวนขึ้นในระบบ โดยที่สัญญาณรบกวนนั้นต้องมีค่าไม่เกินค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม

ในปี 1999 Saed Ahmadi Jullien และ Miller ได้นำเสนองานวิจัยเรื่อง “Arithmetic with Signed Analog Digits” [5] เพื่อนำเสนอ การคำนวณเลขคณิตพื้นฐานบนระบบจำนวนแบบค่าซ้อนทับกัน โดยการใช้โครงสร้างการคำนวณที่ปราศจากตัวทดด้วยความสามารถของวงจรแอนะล็อกเอง ทำให้ความเร็วที่ได้จากการคำนวณของวงจรแอนะล็อกนั้นไม่ต่างจากการคำนวณในวงจรดิจิทัลด้วยระบบจำนวนดิจิตมีเครื่องหมาย (signed digit number system) ที่สามารถจะทำการจำกัดการแพร่กระจายของตัวทดได้ แนวคิดในการออกแบบระบบจำนวนคือการสร้างดิจิตหลายๆ ดิจิตที่กำหนดรูปแบบความสัมพันธ์ได้แน่นอน และจากความสัมพันธ์ดังกล่าวนำไปสู่ความสามารถในการลดค่าความผิดพลาดได้

ในปี 2002 ดังปรากฏในงานวิจัยเรื่อง “A Number System with Continuous Valued Digits and Modulo Arithmetic” [3] ได้มีการเปลี่ยนชื่อจากระบบจำนวนค่าซ้อนทับกันเป็นระบบจำนวนค่าต่อเนื่อง (continuous valued number system) ซึ่งมีการนำเสนอระบบจำนวนค่าต่อเนื่อง เพื่อลดความผิดพลาดที่เกิดขึ้นในระบบจำนวนแบบแอนะล็อก ที่เกิดจากสัญญาณรบกวน โดยทราบช่วงของความผิดพลาดที่เกิดจากสัญญาณรบกวนว่าไม่มากไปกว่าค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม ซึ่งเมื่อเกิดสัญญาณรบกวนขึ้นในแต่ละดิจิตค่าต่อเนื่อง งานวิจัยนี้ได้นำเสนออัลกอริทึมสำหรับกู้ค่าความผิดพลาดให้ได้ค่าใกล้เคียงค่าเดิม และยังได้นำเสนออัลกอริทึมการคำนวณพื้นฐาน คือ การบวก การลบ และการคูณ

2.3 **ระบบจำนวนค่าต่อเนื่อง** (continuous valued number system)

การใช้ระบบจำนวนค่าต่อเนื่อง ซึ่งมีการใช้สัญญาณหลายค่าสัญญาณในการแทนค่าจำนวนจริงแต่ละจำนวน โดยแต่ละสัญญาณจะเรียกว่าดิจิต เมื่อนำไปเทียบกับวงจรดิจิทัล โดยจะมีเพียงดิจิตเดียวที่จะทำการแทนค่าจำนวนจริงนั้นทั้งจำนวน โดยที่ดิจิตอื่นๆ ที่เพิ่มขึ้นมานั้นจะถูก

เรียกว่า *ดิจิตซ้ำซ้อน* (redundancy digit) ซึ่งมีหน้าที่ในการปรับปรุงค่าให้ถูกต้องมากขึ้น ในกรณีที่มีความผิดพลาดเกิดขึ้นในระบบ ซึ่งสมมุติค่า X เป็น *ค่าระยะสูงสุดแบบพลวัต* (maximum dynamic range) ซึ่ง $|x| < X$ และค่าจำนวนจริง x นั้นจะประกอบไปด้วย *ดิจิตค่าต่อเนื่อง* (continuous valued digit) x_n หลายดิจิตซึ่ง $K \le n \le L$ โดยที่ K และ L เป็นจำนวนเต็มที่ $K < 0$ และ $0 \le L$ สามารถคำนวณหาค่า L ได้จากอสมการที่ 2.1

$$X \le \beta^{L+1} \tag{2.1}$$

การแสดงค่าของ x มีรูปแบบดังนี้

$$(x_L, x_{L-1}, \ldots, x_0 \mid x_{-1}, \ldots, x_{K+1}, x_K)$$

การแยกส่วนที่เป็นจำนวนเต็มทางด้านซ้ายออกจากส่วนที่เป็นทศนิยมทางด้านขวาจะใช้เครื่องหมาย | การสร้างดิจิตค่าต่อเนื่องในแต่ละดิจิต จะแสดงค่าเริ่มต้นตั้งแต่ $x_L \ldots x_K$ จากมากไปน้อย ค่าดิจิตเริ่มต้นสามารถคำนวณได้จากสมการที่ 2.2

$$x_L = \beta \times \frac{x}{X} \tag{2.2}$$

ค่าดิจิตอื่นที่เหลือสามารถคำนวณได้จากสมการที่ 2.3

$$x_n = (x_{n+1} - \lfloor x_{n+1} \rfloor)\beta \tag{2.3}$$

จากสมการที่ 2.3 จะเห็นว่า ค่าของ x_n สามารถคำนวณได้จาก x_{n+1} และจากสมการที่ 2.2 และ 2.3 สามารถสรุปได้ว่า $|x_n| < \beta$ และ $|\lfloor x_n \rfloor| \le \beta - 1$

การคำนวณค่า L จะเลือก L ที่น้อยที่สุดที่สอดคล้องกับสมการที่ 2.1 ตัวอย่างเช่น กำหนดให้ค่าระยะสูงสุดแบบพลวัต X = 100 และ ฐาน β = 10 ค่า L ที่น้อยที่สุดที่สอดคล้องคือ 2 ในกรณีที่มีการเลือกค่า L ที่มากกว่า 2 ดิจิตที่เลขลำดับที่มากกว่า L ที่คำนวณได้จะถูกเรียกว่าเป็น *ดิจิตที่ถูกสร้างเกิน* (excessively evolved digits, EEDs) ซึ่งใช้สำหรับการคำนวณที่ซึ่งคำตอบของการดำเนินการนั้นต้องการดิจิตที่เพิ่มขึ้น หรืออีกนัยหนึ่งเป็นการเพิ่มค่าระยะสูงสุดแบบพลวัต

เนื่องจากดิจิตค่าต่อเนื่องสามารถแสดงค่าได้อย่างต่อเนื่อง บางครั้งดิจิตเหล่านี้อาจถูกเรียกว่า *แอนะล็อกดิจิต* ในทางปฏิบัติ ดิจิตแต่ละดิจิตจะหมายถึงสัญญาณที่เกิดขึ้นในระบบ ซึ่งอาจหมายถึงตัวแปรทางไฟฟ้า เช่น ค่ากระแส หรือแรงดันนับเป็นจำนวนโวลต์ เป็นต้น ถ้าตัวแปรทางไฟฟ้ามีลักษณะเป็นแบบเชิงเส้นมีค่าระหว่าง 0 ถึง $\pm Q$ หน่วย การแทนค่าของแต่ละดิจิตด้วยค่าทางไฟฟ้านั้นสามารถคำนวณได้จาก

$$q_n = x_n \, (Q \,/\, \beta)$$

โดยที่ q_n เป็นค่าทางไฟฟ้า เพื่อความสะดวกและความเข้าใจในการทำงาน ในงานวิจัยนี้จะใช้ ค่าดิจิตค่าต่อเนื่อง x_n แทนค่า q_n

ตัวอย่างที่ 2.1 *ถ้า x = 22.4525 ซึ่งมีค่าระยะสูงสุดแบบพลวัต X = 100 ต้องการที่จะทำการแสดงค่าให้อยู่ในรูปของดิจิตค่าต่อเนื่องในช่วง 0 μA ถึง 50 μA บนฐาน β = 2*

คำนวณค่า L ที่เหมาะสมตามสมการที่ 2.1 ได้ L = 6 สำหรับฐาน 2 ซึ่งผลการคำนวณแสดงได้ดังตารางที่ 2.1 ในตัวอย่างนี้ จำนวนดิจิตที่ใช้มีทั้งหมด 11 ดิจิต (K = -4)

ตารางที่ 2.1 แสดงค่าดิจิตที่มีค่าต่อเนื่อง และค่าตัวแปรทางไฟฟ้า

	ฐานสอง	
n	x_n	q_n (μA)
6	0.44905	11.22625
5	0.8981	22.4525
4	1.7962	44.905
3	1.5924	39.81
2	1.1848	29.62
1	0.3696	9.24
0	0.7392	18.48
-1	1.4784	36.96
-2	0.9568	23.92
-3	1.9136	47.84
-4	1.8272	45.68

นั่นคือ รูปแบบแทนจำนวนของ 22.4525 ในระบบจำนวนค่าต่อเนื่อง คือ (0.44905, 0.8981, 1.7962, 1.5924, 1.1848, 0.3696 | 1.4784, 0.9568, 1.9136, 1.8272) □

จากตัวอย่างที่ 2.1 ในกรณีที่ต้องการคำนวณค่าที่ถูกแสดงในรูปแบบแทนจำนวนในระบบจำนวนค่าต่อเนื่องที่กำหนดมาให้นั้น สามารถคำนวณได้จากสมการต่อไปนี้

$$x = x_L\ (X / \beta)$$

จากตัวอย่างที่ 2.1 L = 6 นำค่า x_6 ซึ่งมีค่าเท่ากับ 0.44905 นำมาแทนในสมการจะได้

$$x = 0.44905 \times (\ 100 / 2\ )$$

จะได้ค่าดั้งเดิมคือ x = 22.4525

แต่สำหรับวงจรแอนะล็อกแล้ว สัญญาณรบกวนสามารถหลีกเลี่ยงได้ยาก ซึ่งจากที่กล่าวมาแล้วว่า สัญญาณรบกวนเหล่านี้จะทำให้เกิดค่าความผิดพลาดในการคำนวณได้ ดังนั้น

ค่าที่ตรวจวัดได้จึงมีความคลาดเคลื่อนไปจากค่าที่ต้องการแสดงที่แท้จริง จะเห็นว่าระบบจำนวนค่าต่อเนื่องมีการใช้ดิจิตมากกว่าหนึ่งดิจิต ความสัมพันธ์ของดิจิตที่เพิ่มขึ้นมานี้ จะมีประโยชน์ในการนำไปคำนวณลดค่าความผิดพลาดได้ ในหัวข้อถัดไปจะกล่าวถึงวิธีการในการลดค่าความผิดพลาดในระบบการคำนวณ

2.4 การกู้ค่าความผิดพลาด

จากงานวิจัยระบบจำนวนค่าต่อเนื่อง ได้มีการพิจารณาถึงรูปแบบของค่าความผิดพลาดที่เกิดขึ้นในระบบ โดยสามารถแสดงค่าความผิดพลาดอยู่ในรูปของสมการ

$$x'_n = x_n + \varepsilon_n$$

ซึ่ง x'_n หมายถึงค่าของ x_n ที่เกิดสัญญาณรบกวนทำให้เกิดความผิดพลาด และ ε_n คือค่าความผิดพลาดในแต่ละดิจิต สำหรับวิธีการกู้ค่าความผิดพลาดของดิจิตลำดับที่ n ของระบบจำนวนค่าต่อเนื่องนั้น สามารถทำได้โดยสมมุติให้ดิจิตที่มีลำดับต่ำกว่าที่ติดกันเป็นค่าที่ถูกต้องหรือไม่มีค่าความผิดพลาด $x'_{n-1} = x_{n-1}$ และด้วยเงื่อนไขเพิ่มเติมอีกหนึ่งเงื่อนไขคือ

$$\lfloor x'_n \rfloor = \lfloor x_n \rfloor \tag{2.4}$$

ทำให้สามารถทำการกู้คืนค่าของ x'_n ออกมาเป็น x''_n ที่มีค่าเท่ากับ x_n ดังสมการ

$$x''_n = \begin{cases} \lfloor x'_n \rfloor + \left(x'_{n-1} / \beta\right) & n > K \\ x'_K & n = K \end{cases} \tag{2.5}$$

วิธีการกู้คืนข้อมูลเรียกว่า *ขบวนการวิวัฒนาการย้อนกลับ* (reverse evolution process) ซึ่งขบวนการนี้จะสามารถกู้คืนได้อย่างถูกต้องก็ต่อเมื่อการคำนวณฟังก์ชันพื้น (floor function) ตามสมการที่ 2.4 เป็นจริง ดังนั้นเทคนิคการกู้ค่าความผิดพลาดนี้จะผิดพลาดในกรณีที่

$$\lfloor x_n + \varepsilon_n \rfloor > \lfloor x_n \rfloor \text{ หรือ } \lfloor x_n + \varepsilon_n \rfloor < \lfloor x_n \rfloor$$

ดังนั้นจึงได้ปรับปรุงการกู้ค่าความผิดพลาด โดยเปลี่ยนฟังก์ชันพื้นให้เป็นฟังก์ชันการถูกปัดเศษ (rounded function) แทน $\lfloor x'_n \rfloor$ ได้ดังนี้

$$\lfloor x'_n \rfloor_R = \left[x'_n - \left(x'_{n-1} / \beta\right)\right]_{Rounded}$$

ด้วยเงื่อนไขที่ว่า

$$\left| \varepsilon_n - \left(\varepsilon_{n-1} / \beta\right) \right| < 1/2 \tag{2.6}$$

และเมื่อนำสมการที่ 2.6 ไปแทนในสมการที่ 2.5 จะได้สมการในการกู้ค่าความผิดพลาดดังนี้

$$[x'']_R = \begin{cases} \lfloor x'_n \rfloor_R + (x''_{n-1} / \beta) & n > K \\ x'_K & n = K \end{cases}$$

โดยที่ $\lfloor x'_n \rfloor_R = [\, x'_n - (x''_{n-1} / \beta)\,]_{Rounded}$ ซึ่งเทคนิคที่ใช้ในการกู้ค่าความผิดพลาดนั้นจะทำการปรับปรุงค่าความแม่นยำของดิจิตที่มีลำดับสูงกว่าที่อยู่ติดกันทีละดิจิต

ตัวอย่างที่ 2.2 *กำหนดให้เกิดค่าความผิดพลาดขึ้นในวงจรเท่ากับ 0.0025 ในฐาน* $\beta = 2$ *ด้วย* $X = 100$ *ทำการเลือกค่า* $L = 6$ *และ* $K = -4$ *การแทนค่าของ* $x = 22.4525$

ค่าของ x ที่ถูกใส่ค่าความผิดพลาดแทนด้วย x'_n ในระบบจำนวนค่าต่อเนื่อง พร้อมทั้ง ค่าของ x หลังจากกู้ค่าความผิดพลาดแล้วแทนด้วย x''_n ได้ถูกแสดงไว้ดังตารางที่ 2.2

ตารางที่ 2.2 แสดงค่าของ x = 22.4525 ในระบบที่มีค่าความผิดพลาดเท่ากับ 0.0025

n	x_n	x'_n	$[x'']_R$
6	0.44905	0.45155	0.44905244
5	0.8981	0.9006	0.89810488
4	1.7962	1.7987	1.79620975
3	1.5924	1.5949	1.5924195
2	1.1848	1.1873	1.184839
1	0.3696	0.3721	0.369678
0	0.7392	0.7417	0.739356
-1	1.4784	1.4809	1.4787125
-2	0.9568	0.9593	0.957425
-3	1.9136	1.9161	1.91485
-4	1.8272	1.8297	1.8297

ค่าที่คำนวณได้จากการกู้ค่าความผิดพลาด คือ 0.44905244 เมื่อนำไปแทนในสมการที่ 2.2 กรณีที่ L=6 จะได้

$$x = 0.44905244 \times (100 / 2) = 22.452622$$

□

สังเกตได้ว่าค่าที่ได้จากการกู้ค่าความผิดพลาดมีค่าเท่ากับ 22.452622 เมื่อเทียบกับค่าดั้งเดิมคือ 22.4525 แล้วไม่ได้ค่าที่ถูกต้อง และจากตัวอย่างที่ 2.2 ค่าที่ทำการกู้ค่ากลับคืนหลังจากเกิดสัญญาณรบกวนมีค่าเท่ากับ (0.44905244, 0.89810488, 1.79620975, 1.5924195, 1.184839, 0.369678, 0.739356 | 1.4787125, 0.957425, 1.91485, 1.8297) เมื่อเทียบกับค่าดั้งเดิม (0.44905, 0.8981, 1.7962, 1.5924, 1.1848, 0.3696, 0.7392 | 1.4784, 0.9568, 1.9136, 1.8272) ก็จะเห็นว่าค่าที่กู้มายังไม่เท่ากับค่าเดิม เนื่องจากยังมีความผิดพลาดซึ่งจะทำให้กระทบกับทุกๆ ดิจิตที่จะทำการกู้ค่าความผิดพลาด ทำให้คำตอบที่ได้ไม่ตรงกับค่าที่แท้จริง การเกิดค่าความผิดพลาดจากดิจิตที่มีลำดับน้อยที่สุดกระจายไปยังทุกๆ ดิจิตนั้นจะเรียกว่า *การแพร่กระจายของความผิดพลาดที่เกิดจากดิจิตที่มีค่าความสำคัญน้อยที่สุด* (least significant digit error propagation) ซึ่งค่าความผิดพลาดนี้เกิดจากการที่ไม่สามารถจะทำการปรับปรุงค่าของดิจิต x_K ได้ตามสมการการกู้ค่าความผิดพลาดในกรณีที่ $n = K$

สมการที่ 2.6 ได้กำหนดขอบเขตของค่าความผิดพลาดของดิจิตที่อยู่ติดกัน ที่จะทำให้ฟังก์ชันการปัดเศษ คำนวณค่าได้ถูกต้องตามสมมุติฐานที่กำหนดไว้ ในกรณีที่ค่าความผิดพลาดนี้อยู่นอกขอบเขตจะทำให้การกู้ค่าความผิดพลาดนั้นไม่สามารถทำให้คำตอบที่ถูกต้องได้ จากข้อจำกัดดังกล่าวทำให้เกิดช่วงของ*ค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม* (error tolerance) ดังสมการที่ 2.7

$$\vartheta = \left|\frac{1}{2\beta}\right| \tag{2.7}$$

จะเห็นได้ว่า ค่าฐานที่แตกต่างกันมีค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมไม่เท่ากัน ยกตัวอย่างเช่นฐานสอง จะมีค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมเท่ากับ 0.25 ซึ่งหมายความว่าในระบบจำนวนค่าต่อเนื่องของฐานสอง นั้นสามารถที่จะมีค่าผิดพลาดที่เกิดขึ้นในระบบได้น้อยกว่า 25% ของแต่ละดิจิต ถ้าเกิดมีค่าความผิดพลาดเกินจากค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมจะทำให้คำตอบที่ได้จากการกู้ค่าความผิดพลาดนั้นไม่ถูกต้อง

2.5 การคำนวณเลขคณิตของระบบจำนวนค่าต่อเนื่อง

นอกจากงานวิจัยระบบจำนวนค่าต่อเนื่องที่ได้นำเสนอในส่วนของการคำนวณดิจิตค่าต่อเนื่อง กรณีดิจิตแต่ละดิจิตมีค่าต่อเนื่องเกิดความผิดพลาดที่เกิดจากสัญญาณรบกวน รวมถึงการกู้ค่าความผิดพลาดไปแล้ว ยังได้นำเสนอทฤษฎีการคำนวณเลขคณิตพื้นฐาน (fundamental arithmetic) ของระบบจำนวนค่าต่อเนื่อง โดยนำเสนออัลกอริทึมในส่วนของ การบวก การลบ และการคูณ จะนำเสนอตามลำดับดังนี้

2.5.1 การบวกของระบบจำนวนค่าต่อเนื่อง

ทฤษฎีบทที่ 2.1 *กำหนดให้ x และ y เป็นจำนวนในระบบจำนวนค่าต่อเนื่องฐาน β ผลบวกของ x และ y คือ $z = x + y$ สามารถคำนวณได้จากสมการที่ 2.8*

$$z_n = (x_n + y_n) \bmod \beta \tag{2.8}$$

จากทฤษฎีบทที่ 2.1 จะได้ว่าการบวกสามารถทำได้โดยการบวกของแต่ละดิจิตที่ตรงกัน และนอกจากนี้ เมื่อเกิดค่าความผิดพลาดขึ้นในระบบการบวก การกู้ค่าความผิดพลาดก็สามารถกู้ค่ากลับมาได้ใกล้เคียง

ตัวอย่างที่ 2.3 *การบวก x = 12.24 และ y = 23.45 ด้วยค่า X = 100 สำหรับฐาน β = 2 ด้วยค่าความผิดพลาดหลังการทำการบวกเท่ากับ 0.0842*

ในกรณีที่เลือกค่า L = 6 และ K = -2 ผลลัพธ์ของการคำนวณถูกแสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 แสดงการบวก x = 12.24 และ y = 23.45 ซึ่งมีค่าความผิดพลาดเท่ากับ 0.0842

n	x_n	y_n	$z_n=(x_n + y_n)$	z'_n	z''_n
6	0.2448	0.469	0.7138	0.798	0.71412891
5	0.4896	0.938	1.4276	1.5118	1.42825782
4	0.9792	1.876	0.8552	0.9394	0.85651563
3	1.9584	1.752	1.714	1.7982	1.71303125
2	1.9168	1.504	1.4208	1.505	1.4260625
1	1.8336	1.008	0.8416	0.9258	0.852125
0	1.6672	0.016	1.6832	1.7674	1.70425
-1	1.3344	0.032	1.3664	1.4506	1.4085
-2	0.6688	0.064	0.7328	0.817	0.817

ผลบวกที่ได้จากการคำนวณในกรณีที่ไม่มีค่าความผิดพลาดคือ 35.69 ค่าที่คำนวณได้จากการกู้ค่าความผิดพลาด คือ 0.71412891 เมื่อนำไปแทนในสมการที่ 2.2 กรณี L=6 จะได้

$$x = 0.71412891 \times (100 / 2)$$

$$x = 35.7064455$$

□

2.5.2 การลบของระบบจำนวนค่าต่อเนื่อง

การลบสองจำนวนในระบบจำนวนแบบมีค่าต่อเนื่องนั้น สามารถทำได้ด้วยการบวกจำนวนที่ติดลบ โดยใช้ทฤษฎีบทที่ 2.1 เพื่อกำหนดการลบระหว่างดิจิตแบบมีค่าต่อเนื่อง ดังนี้

ทฤษฎีบทที่ 2.2 *กำหนดให้ x และ y เป็นจำนวนในระบบจำนวนค่าต่อเนื่องฐาน β ผลบวกของ x และ y คือ $z = x + (-y)$ สามารถคำนวณได้จากสมการที่ 2.9*

$$z_n = (x_n + (-y_n)) \bmod \beta = (x_n - y_n) \bmod \beta \qquad (2.9)$$

แต่เนื่องจากเครื่องหมายของจำนวนหลังจากทำการลบนั้นจะมีเพียงดิจิตที่มีค่าลำดับสูงสุดดิจิตเดียวเท่านั้นที่จะสามารถรับรองได้ว่ามีเครื่องหมายของคำตอบที่ถูกต้อง เนื่องจากดิจิตที่เหลือไม่จำเป็นที่จะต้องมีเครื่องหมายถูกต้องตามตัวแรกเสมอไป จึงจำเป็นที่จะต้องใช้เครื่องหมายของดิจิตที่มีลำดับสูงสุดเพื่อทำการแปลงเครื่องหมายของดิจิตที่เหลือของคำตอบให้ถูกต้อง เพื่อที่จะทำให้เครื่องหมายเหล่านั้นถูกต้อง จึงได้มีการทำปรับปรุงวิธีการคำนวณการบวกในสมการที่ 2.8 ซึ่งสามารถที่จะทำให้ค่าเครื่องหมายนั้นถูกต้องได้ด้วยตัวดำเนินการ mod

$$z_n = \begin{cases} (x_n + y_n) \operatorname{mod}^+ \beta & (x_L + y_L) \geq 0 \\ (x_n + y_n) \operatorname{mod}^- \beta & (x_L + y_L) < 0 \end{cases}$$

โดยที่ฟังก์ชัน mod^+ และ mod^- ได้ถูกนิยามไว้ดังต่อไปนี้

$$(a) \operatorname{mod}^+ \beta = a + I\beta \quad \text{ในกรณีที่} \quad 0 \leq (a)\operatorname{mod}^+\beta < \beta$$
$$(a) \operatorname{mod}^- \beta = a + I\beta \quad \text{ในกรณีที่} \quad -\beta < (a)\operatorname{mod}^-\beta \leq 0$$

และ I เป็นจำนวนเต็ม

ตัวอย่างที่ 2.4 *การลบ $x = 12.24$ และ $y = 23.45$ ด้วยค่า $X = 100$ สำหรับฐาน $\beta = 2$ ด้วยค่าความผิดพลาดหลังการทำการคำนวณทางคณิตศาสตร์เท่ากับ 0.0842*

ทำการเลือกค่า $L = 6$ และ $K = -2$ ผลลัพธ์ของการคำนวณถูกแสดงได้ดังตารางที่ 2.4

ตารางที่ 2.4 แสดงการลบ x = 12.24 และ y = 23.45 ซึ่งมีค่าความผิดพลาดเท่ากับ 0.0842

n	x_n	y_n	$z_n=(x_n - y_n)$	$z_n=(x_n - y_n)\text{mod}^{-}\beta$	z'_n	z''_n
6	0.2448	0.469	-0.2242	-0.2242	-0.3084	-0.22452894
5	0.4896	0.938	-0.4484	-0.4484	-0.5326	-0.44905788
4	0.9792	1.876	-0.8968	-0.8968	-0.981	-0.89811575
3	1.9584	1.752	0.2064	-1.7936	-1.8778	-1.7962315
2	1.9168	1.504	0.4128	-1.5872	-1.6714	-1.592463
1	1.8336	1.008	0.8256	-1.1744	-1.2586	-1.184925
0	1.6672	0.016	1.6512	-0.3488	-0.433	-0.36985
-1	1.3344	0.032	1.3024	-0.6976	-0.7818	-0.7397
-2	0.6688	0.064	0.6048	-1.3952	-1.4794	-1.4794

ดังนั้น ผลลบที่ได้จากการคำนวณในกรณีที่ไม่มีค่าความผิดพลาดคือ -11.21 ค่าที่คำนวณได้จากการกู้ค่าความผิดพลาด คือ -0.22452894 เมื่อนำไปแทนในสมการที่ 2.2 กรณี L=6 จะได้

$$x = -0.22452894 \times (100 / 2)$$

$$x = -11.226447 \qquad \square$$

2.5.3 การคูณของระบบจำนวนค่าต่อเนื่อง

การคูณในระบบจำนวนค่าต่อเนื่อง [4] ไม่สามารถที่จะคำนวณแต่ละดิจิตแยกออกจากกันได้เหมือนกับการบวกและการลบ ดังนั้นงานวิจัยนี้จึงได้ทำการเสนออัลกอริทึมการคูณโดยการรวมของผลคูณย่อย (partial product) ซึ่งสามารถที่จะเป็นไปได้เนื่องจากทฤษฎีดังต่อไปนี้

ทฤษฎีบทที่ 2.3 *กำหนดให้ x เป็นจำนวนในระบบจำนวนค่าต่อเนื่อง ฐานβ ดิจิตค่าต่อเนื่องของ λx ซึ่ง λ เป็นจำนวนเต็มสามารถคำนวณได้จากสมการที่ 2.10*

$$(\lambda x)_n = (\lambda . x_n) mod\ \beta \qquad (2.10)$$

กำหนดให้ $y = \sum_{k=n}^{m} \lambda_k \beta^k$ *แทนแต่ละดิจิตของรูปแบบการแทนค่า* $y_n\, y_{n-1} \ldots y_0 . y_{-1} \ldots y_m$ *โดยที่ค่า λ มีค่าเป็นจำนวนเต็ม ดังนั้นวิธีการคูณค่า x และ y ในระบบจำนวนค่าต่อเนื่องสามารถที่จะทำได้ดังทฤษฎีบทดังต่อไปนี้*

ทฤษฎีบทที่ 2.4 *กำหนดให้ x และ y เป็นจำนวนในระบบจำนวนค่าต่อเนื่องฐาน β ดิจิตค่าต่อเนื่องของผลลัพธ์การคูณ $z = x \,.\, y$ สามารถคำนวณได้จากสมการ ที่ 2.11*

$$(y \cdot x)_n = \left(\sum_{\forall k} \lambda_k \cdot x_{n-k} \right) \operatorname{mod} \beta = \left(\sum_{i=n}^{m} \vec{\lambda}_i x_n \right) \operatorname{mod} \beta \qquad (2.11)$$

โดยที่สัญลักษณ์ $\vec{\lambda}_i$ หมายถึงทำการเลื่อนค่า x_n ไปทางซ้าย i ตำแหน่งในกรณีที่ i เป็นบวก หรือทำการเลื่อนค่า x_n ไปทางขวา i ตำแหน่งในกรณีที่ i เป็นลบ

ตัวอย่างที่ 2.5 *การคูณ x = 12.4 และ y = 5.5 ด้วยค่า X = 100 สำหรับฐาน β = 2 ด้วยค่าความผิดพลาดหลังการทำการคำนวณทางคณิตศาสตร์เท่ากับ 0.0842*

ทำการเลือกค่า L = 6 และ K = -2 ผลลัพธ์ของการคำนวณถูกแสดงได้ดังตารางที่ 2.5

ตารางที่ 2.5 แสดงการคูณ x = 12.4 และ y = 5.5 ซึ่งมีค่าความผิดพลาดเท่ากับ 0.0842

n	x_n	$(\vec{\lambda}_2 . x_n)$	$(\vec{\lambda}_1 . x_n)$	$(\vec{\lambda}_0 . x_n)$	$(\vec{\lambda}_{-1} . x_n)$	z_n	z'_n	z''_n
6	0.248	0.992	0	0.248	0.124	1.364	1.4482	1.364328907
5	0.496	1.984	0	0.496	0.248	0.728	0.8122	0.728657813
4	0.992	1.968	0	0.992	0.496	1.456	1.5402	1.457315625
3	1.984	1.936	0	1.984	0.992	0.912	0.9962	0.91463125
2	1.968	1.872	0	1.968	1.984	1.824	1.9082	1.8292625
1	1.936	1.744	0	1.936	1,968	1.648	1.7322	1.658525
0	1.872	1.488	0	1.872	1.936	1.296	1.3802	1.31705
-1	1.744	0.976	0	1.744	1.872	0.592	0.6762	0.6341
-2	1.488	1.952	0	1.488	1.744	1.184	1.2682	1.2682

ดังนั้น ผลคูณที่ได้จากการคำนวณในกรณีที่ไม่มีค่าความผิดพลาด คือ 68.2 ค่าที่คำนวณได้จากการกู้ค่าความผิดพลาด คือ 1.364328907 เมื่อนำไปแทนในสมการที่ 2.2 กรณี L=6 จะได้

$$x = 1.364328907 \times (100 / 2)$$

$$x = 68.21644535$$

□

บทที่ 3

อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง

งานวิจัยระบบจำนวนค่าต่อเนื่อง Saed Ahmadi Jullien และ Miller ได้ออกแบบอัลกอริทึมการคำนวณเลขคณิตพื้นฐานคือ การบวก การลบ และการคูณ เมื่อเกิดค่าความคลาดเคลื่อนจากสัญญาณรบกวนในทุก ๆ ดิจิตค่าต่อเนื่อง สามารถกู้ค่าความผิดพลาดกลับมาได้ใกล้เคียงค่าเดิม ซึ่งได้อธิบายในบทที่ 2 ไปแล้วนั้น งานวิจัยดังกล่าวยังไม่ได้มีการนำเสนออัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง ซึ่งถือว่าเป็นตัวดำเนินการเลขคณิตพื้นฐานอีกหนึ่งตัวที่มีความสำคัญ

ในบทนี้ได้ทำการออกแบบอัลกอริทึมการหารเพื่อแสดงให้เห็นว่า การหารสามารถทำได้บนระบบจำนวนค่าต่อเนื่อง โดยผลลัพธ์ที่ได้ยังคงคุณสมบัติของดิจิตค่าต่อเนื่อง และสามารถคำนวณได้บนฐานใด ๆ ก็ตาม พร้อมทั้งบทพิสูจน์อัลกอริทึมดังกล่าว

3.1 อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง

แนวคิดการคำนวณของตัวดำเนินการทางเลขคณิตพื้นฐานของการหาร เกิดขึ้นจากการวิเคราะห์การหารตัวเลข ระหว่าง x กับ y ยกตัวอย่าง เช่น กำหนดให้ x มีค่าเท่ากับ 735 และ y มีค่าเท่ากับ 25 โดย β มีค่าเท่ากับ 10 การหารตัวเลขเกิดจากนำค่าของตัวเลขในแต่ละหลักของตัวตั้งใส่ฟังก์ชันพื้น กำหนดให้ x_n แทนค่าของหลัก โดย n แทนจำนวนหลัก คูณด้วย β^n หารด้วย y จะได้ค่าในแต่ละหลักดังนี้

$$\left(\frac{7\times10^2}{25}\right)+\left(\frac{3\times10^1}{25}\right)+\left(\frac{5\times10^0}{25}\right)$$

มีค่าเท่ากับ

$$\frac{700}{25}+\frac{30}{25}+\frac{5}{25}$$

ผลลัพธ์ที่ได้ของการหารคือ การนำผลรวมของการหารในแต่ละหลักมารวมเข้าด้วยกัน

ทฤษฎีบทที่ 3.1 *กำหนดให้ x และ y เป็นจำนวนจริงบนระบบจำนวนค่าต่อเนื่องบนฐาน β โดยที่รูปแบบการแทนค่าของ $x = (x_L\ x_{L-1}\ x_{L-2} \ldots | \ldots x_K)$ และรูปแบบการแทนค่าของ $y = (y_L\ y_{L-1}\ y_{L-2} \ldots | \ldots y_K)$ กำหนดให้รูปแบบการแทนค่าของ $z = (z_L\ z_{L-1}\ z_{L-2} \ldots | \ldots z_K)$ โดย z เป็นผลลัพธ์ที่ได้จากการหาร x ด้วย y ซึ่งจำนวนในระบบจำนวนค่าต่อเนื่องฐาน β ผลหารของ x ด้วย y คือ $z = x / y$ สามารถคำนวณได้จากสมการที่ 3.1*

$$z_n = \begin{cases} Q_n & : n = L \\ \left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right)\bmod \beta + Q_n\right)\bmod \beta & : n < L \end{cases} \tag{3.1}$$

โดยกำหนดให้ T_n เป็นค่าผลหารในหลักที่ n โดย $K \le n \le L$ ดังสมการที่ 3.2

$$T_n = \frac{\lfloor x_n \rfloor \times \beta^n}{y} \tag{3.2}$$

และกำหนดให้ Q_n เป็นตัวแปรสำหรับเก็บความสัมพันธ์ของการหาร x_n ด้วยตัวหาร y_n ดังสมการที่ 3.3

$$Q_n = \left(\frac{x_n}{y}\right)\bmod \beta \tag{3.3}$$

พิสูจน์

จากสมการที่ 2.2 สำหรับการหาค่าดิจิตแรก คือ x_L ซึ่งแสดงด้วยสมการที่ 3.4 และจากสมการที่ 2.3 สำหรับการหาค่าดิจิตอื่นที่เหลือ ซึ่งแสดงด้วยสมการที่ 3.5 สำหรับการพิสูจน์ทฤษฎีบทที่ 3.1

$$x_L = \beta \times (x / X) \tag{3.4}$$

$$x_n = (x_{n+1} - \lfloor x_{n+1} \rfloor) \times \beta \tag{3.5}$$

จากสมการที่ 3.4 และ 3.5 สามารถเขียนเป็นสมการสำหรับการคำนวณดิจิตค่าต่อเนื่องใหม่ ในรูปแบบทั่วไป โดยสามารถคำนวณหาค่าดิจิตที่ n ใดๆ ดังสมการที่ 3.6

$$x_n = \left(\frac{x}{X}\beta^{L-n+1}\right)\bmod \beta \tag{3.6}$$

จากสมการที่ 3.6 นำมาหาค่า x_n โดย x_m ดังสมการที่ 3.7

$$x_m = \left(x_n \times \beta^{L-n}\right) \bmod \beta \tag{3.7}$$

จากการที่กำหนด $z = x/y$ และสมการที่ 3.1 ในกรณีที่ $n = L$

$$z_L = \left(\frac{x_L}{y}\right) \bmod \beta$$

ทำการแทนค่า x_L จากสมการที่ 3.4 จะได้สมการที่ 3.8

$$z_L = \left(\left(\frac{x}{y}\right) \times \frac{\beta}{X}\right) \bmod \beta \tag{3.8}$$

สำหรับกรณี $n < L$ ทำการแทนค่า x_n ค่าในสมการที่ 3.7 จะได้สมการที่ 3.9

$$z_n = \left(\left(\frac{x}{y}\right) \times \frac{\beta}{X} \times \beta^{L-n}\right) \bmod \beta \tag{3.9}$$

จากหลักการของระบบจำนวนค่าต่อเนื่อง ค่าดิจิตในแต่ละหลักใส่ฟังก์ชันพื้น คูณด้วย β^n จะได้ค่าของแต่ละหลัก ดังสมการที่ 3.10

$$x_L \beta^L = \lfloor x_L \rfloor \beta^L + \lfloor x_{L-1} \rfloor \beta^{L-1} + \ldots + x_n \beta^n \tag{3.10}$$

นำสมการที่ 3.4 คูณด้วย β^n หลังจากนั้นนำค่าจากสมการที่ 3.10 แทนในสมการที่ 3.4 ดังสมการที่ 3.11

$$x = \frac{\left(\lfloor x_L \rfloor \beta^L + \lfloor x_{L-1} \rfloor \beta^{L-1} + \ldots + x_n \beta^n\right) \times X}{\beta \times \beta^L} \tag{3.11}$$

นำค่า x จากสมการที่ 3.11 แทนที่ในสมการที่ 3.9 ดังสมการที่ 3.12

$$z_n = \left(\left(\left(\sum_{i=n+1}^{L} \frac{\frac{\lfloor x_i \rfloor \beta^i}{y}}{\beta^n}\right) \bmod \beta\right) + \left(\frac{x_n}{y}\right) \bmod \beta\right) \bmod \beta \tag{3.12}$$

จากที่ได้กำหนดตัวแปรสองตัวคือ T_n และ Q_n ในสมการที่ 3.2 และสมการที่ 3.3 ตามลำดับ สามารถนำตัวแปรทั้งสองไปแทนในสมการที่ 3.12 ทำให้สามารถสรุปสมการสุดท้ายสำหรับทุกกรณีของ $n < L$ ดังสมการข้างล่างนี้

$$z_n = \left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right)\bmod \beta + Q_n\right)\bmod \beta$$

เป็นการสิ้นสุดการพิสูจน์ทฤษฎีบทที่ 3.1 ■

ทฤษฎีบทถัดไป จะแสดงให้เห็นว่า คำตอบที่ได้จากการหารในทฤษฎีบทที่ 3.1 นั้นจะยังคงคุณสมบัติของจำนวนในระบบจำนวนค่าต่อเนื่องด้วย

ทฤษฎีบทที่ 3.2 *ผลลัพธ์การหารจากทฤษฎีบทที่ 3.1 มีคุณสมบัติของรูปแบบแทนจำนวนในระบบจำนวนค่าต่อเนื่องฐาน* β

พิสูจน์

กำหนดให้ $z = (z_L\ z_{L-1}\ z_{L-2} \ldots \mid \ldots z_K)$ เป็นผลลัพธ์ที่ได้จากการหาร x ด้วย y ในระบบจำนวนค่าต่อเนื่อง ฐาน β จะต้องพิสูจน์ว่า

$$z_n = (z_{n+1} - \lfloor z_{n+1} \rfloor)\beta$$

สำหรับทุกกรณีของ $K \leq n < L$ การพิสูจน์แบ่งออกเป็น 2 กรณีคือ

กรณีที่ $n = L$-1

จากอัลกอริทึมของการหาร z_n สามารถคำนวณได้จาก

$$z_{L-1} = \left(\left(\left(\sum_{i=L}^{L}\frac{T_i}{\beta^{L-1}}\right)\bmod \beta\right) + Q_{L-1}\right)\bmod \beta$$

$$z_{L-1} = \left(\left(\left(\sum_{i=L}^{L}\frac{T_i}{\beta^{L-1}}\right)\bmod \beta\right) + \left(\left(\frac{x_L}{Y}\right)\bmod \beta\right)\right)\bmod \beta$$

$$z_{L-1} = \left(\frac{\lfloor x_L \rfloor \times \beta^L}{Y \times \beta^{L-1}} + \frac{(x_L - \lfloor x_L \rfloor) \times \beta}{Y}\right)\bmod \beta$$

$$z_{L-1} = \left(\frac{\left(\lfloor x_L \rfloor \times \beta\right) + \left(x_L \times \beta\right) - \left(\lfloor x_L \rfloor \times \beta\right)}{Y}\right) \bmod \beta$$

$$z_{L-1} = \left(\left(\frac{x_L}{Y}\right) \times \beta\right) \bmod \beta$$

$$z_{L-1} = \left(\left(\left(\frac{x_L}{Y}\right) \bmod \beta\right) \times \beta\right) \bmod \beta$$

$$z_{L-1} = \left(z_L \times \beta\right) \bmod \beta$$

$$z_{L-1} = \left(z_L \times \beta\right) - \left\lfloor \frac{z_L \times \beta}{\beta} \right\rfloor \times \beta$$

$$z_{L-1} = \left(z_L \times \beta\right) - \left(\lfloor z_L \rfloor \times \beta\right)$$

$$z_{L-1} = \left(z_L - \lfloor z_L \rfloor\right)\beta$$

กรณีที่ $n < L$-1

$$z_n = \left(\left(\left(\sum_{i=n+1}^{L} \frac{T_i}{\beta^{n1}}\right) \bmod \beta\right) + \left(\left(\frac{x_n}{y}\right) \bmod \beta\right)\right) \bmod \beta$$

$$z_n = \left(\frac{\lfloor x_L \rfloor \times \beta^L + \ldots + \lfloor x_{n+1} \rfloor \times \beta^{n+1}}{Y \times \beta^n} + \frac{\left(x_{n+1} - \lfloor x_{n+1} \rfloor\right) \times \beta}{Y}\right) \bmod \beta$$

$$z_n = \left(\frac{\lfloor x_L \rfloor \times \beta^L + \ldots + \lfloor x_{n+2} \rfloor \times \beta^{n+2}}{Y \times \beta^n} + \left(\frac{x_{n+1}}{Y}\right) \times \beta\right) \bmod \beta$$

$$z_n = \left(\left(\frac{\lfloor x_L \rfloor \times \beta^L + \ldots + \lfloor x_{n+2} \rfloor \times \beta^{n+2}}{Y \times \beta^{n+1}} + \frac{x_{n+1}}{Y}\right) \times \beta\right) \bmod \beta$$

$$z_n = \left(\left(\left(\frac{\lfloor x_L \rfloor \times \beta^L + \ldots + \lfloor x_{n+2} \rfloor \times \beta^{n+2}}{Y \times \beta^{n+1}} + \frac{x_{n+1}}{Y}\right) \bmod \beta\right) \times \beta\right) \bmod \beta$$

$$z_n = \left(\left(\left(\frac{\lfloor x_L \rfloor \times \beta^L + \ldots + \lfloor x_{n+2} \rfloor \times \beta^{n+2}}{Y \times \beta^{n+1}}\right) \bmod \beta + \left(\frac{x_{n+1}}{Y}\right) \bmod \beta\right) \times \beta\right) \bmod \beta$$

แทนค่า $\left(\frac{\lfloor x_L \rfloor \times \beta^L + \ldots + \lfloor x_{n+2} \rfloor \times \beta^{n+2}}{Y \times \beta^{n+1}}\right) \bmod \beta + \left(\frac{x_{n+1}}{Y}\right) \bmod \beta$ ด้วย z_{n+1}

$$z_n = (z_{n+1} \times \beta) \bmod \beta$$

$$z_{L-1} = (z_L \times \beta) - \left\lfloor \frac{z_L \times \beta}{\beta} \right\rfloor \times \beta$$

$$z_n = (z_{n+1} - \lfloor z_{n+1} \rfloor)\beta$$

เป็นการสิ้นสุดการพิสูจน์ทฤษฎีบทที่ 3.2 ■

3.2 ตัวอย่างการหารบนระบบจำนวนค่าต่อเนื่อง

เพื่อการทำความเข้าใจอัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่องให้เข้าใจมากขึ้น ซึ่งจะมีตัวอย่างโดยจะแสดงบนฐานสิบ และฐานสอง เพื่อแสดงให้เห็นว่าอัลกอริทึมที่ได้นำเสนอสามารถคำนวณได้ทั้งหมดบนฐานใดๆ โดยตัวอย่างที่ยกมานี้ ยังมิได้มีการคำนึงถึงค่าความคลาดเคลื่อนจากสัญญาณรบกวน รวมถึงการกู้ค่าความผิดพลาดที่เกิดขึ้น ซึ่งจะแสดงในบทที่ 4 ต่อไป เพียงแต่แสดงให้เห็นว่า การหารสามารถคำนวณได้จริงตามอัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่องที่ได้ออกแบบไว้

ตัวอย่างที่ 3.1 *การหาร x = 735.2 และ y = 25 ด้วยค่า X = 1000 สำหรับฐาน β = 10 ด้วยเลือกค่า L = 2 และ K = -1*

ผลลัพธ์ของการคำนวณถูกแสดงได้ดังตารางที่ 3.1

ตารางที่ 3.1 แสดงการหาร x = 735.2 และ y = 25 สำหรับฐาน β = 10

n	2	1	0	-1
x_n	7.352	3.52	5.2	2
y_n	0.25	2.5	5	0
T_n	28	1.2	0.2	0.008
Q_n	0.29408	0.1408	0.208	0.08
z_n	0.29408	2.9408	9.408	4.08

คำตอบที่ได้จากการคำนวณนี้ คือ (0.29408, 2.9408, 9.408 | 4.08) ถ้าพิจารณาคำตอบที่ถูกต้องของ 735.2 หารด้วย 25 จะได้ค่าเท่ากับ 29.408 โดยที่ 29.408 ในระบบจำนวนค่าต่อเนื่อง X = 1000 และฐาน β = 10 นั้น ค่าของ 29.408 เป็นค่าของผลลัพธ์ของการหารที่แท้จริง

นำค่าดิจิตที่ต่อเนื่องหลักที่ n = 2 มาคำนวณในสมการที่ 3.4 โดยนำค่า z_n มาแทนค่าในสมการดังกล่าวจะได้ผลลัพธ์ดังนี้

$$z_n = z_L(X/\beta)$$
$$z_n = 0.29498(1000/10)$$
$$z_n = 29.408$$

สำหรับดิจิตถัดไปคำนวณในสมการที่ 3.14 จะได้ดิจิตค่าแบบต่อเนื่องดังนี้

$$z_1 = (0.29408 - \lfloor 0.29408 \rfloor)2 = 2.9408$$
$$z_0 = (2.9408 - \lfloor 2.9408 \rfloor)2 = 9.408$$
$$z_{-1} = (9.408 - \lfloor 9.408 \rfloor)2 = 4.08$$

สังเกตเห็นว่าคำตอบที่ได้ตรงกับคำตอบที่แท้จริง คือ 29.408 และเป็นการยืนยันว่าอัลกอริทึมที่ได้ออกแบบไว้ มีคุณสมบัติของดิจิตที่มีค่าต่อเนื่อง ต่อไปจะนำเสนอตัวอย่างบนฐาน 2 ดังตัวอย่างที่ 3.2 □

ตัวอย่างที่ 3.2 *การหาร x = 73.52 และ y = 25 ด้วยค่า X = 100 สำหรับฐาน β = 2 เลือกค่า L = 6 และ K = -2*

วิธีทำ

คำนวณหาค่าดิจิตค่าต่อเนื่องของ x_n และ y_n บนฐาน 2 หาค่า T_n, Q_n และ z_n เพื่อได้ผลหารที่เป็นดิจิตค่าต่อเนื่อง บนฐาน 2

วิธีคำนวณ

จากสมการ $x_L = \beta \times \frac{x}{X}$ สามารถคำนวณหาค่าดิจิตเริ่มต้น x_L สำหรับดิจิตต่อไปหาได้จากสมการ $x_n = (x_{n+1} - \lfloor x_{n+1} \rfloor)\beta$ ดังนี้

$$x_6 = 2 \times \frac{73.52}{100} = 1.4704$$

สำหรับดิจิตต่อไปคำนวณได้ดังแสดงในตารางที่ 3.2

จากวิธีการเดียวกันกับการหาดิจิตต่อเนื่อง x_n สามารถคำนวณหาดิจิต y_n ได้ดังนี้

$$y_6 = 2 \times \frac{25}{100} = 0.5$$

สำหรับดิจิตต่อไปคำนวณได้ดังแสดงในตารางที่ 3.3

ตารางที่ 3.2 แสดงการคำนวณค่า x_n

n	$(x_{n+1} - \lfloor x_{n+1} \rfloor)\beta$	x_n
5	$(1.4704 - \lfloor 1.4704 \rfloor) \times 2$	0.9408
4	$(0.9408 - \lfloor 0.9408 \rfloor) \times 2$	1.8816
3	$(1.8816 - \lfloor 1.8816 \rfloor) \times 2$	1.7632
2	$(1.7632 - \lfloor 1.7632 \rfloor) \times 2$	1.5264
1	$(1.5264 - \lfloor 1.5264 \rfloor) \times 2$	1.0528
0	$(1.0528 - \lfloor 1.0528 \rfloor) \times 2$	0.1056
-1	$(0.1056 - \lfloor 0.1056 \rfloor) \times 2$	0.2112
-2	$(0.2112 - \lfloor 0.2112 \rfloor) \times 2$	0.4224

ตารางที่ 3.3 แสดงการคำนวณค่า y_n

n	$(y_{n+1} - \lfloor y_{n+1} \rfloor)\beta$	y_n
5	$(0.5 - \lfloor 0.5 \rfloor) \times 2$	1
4	$(1 - \lfloor 1 \rfloor) \times 2$	0
3	$(0 - \lfloor 0 \rfloor) \times 2$	0
2	$(0 - \lfloor 0 \rfloor) \times 2$	0
1	$(0 - \lfloor 0 \rfloor) \times 2$	0
0	$(0 - \lfloor 0 \rfloor) \times 2$	0
-1	$(0 - \lfloor 0 \rfloor) \times 2$	0
-2	$(0 - \lfloor 0 \rfloor) \times 2$	0

หลังจากได้ดิจิตค่าแบบต่อเนื่อง x_n และ y_n แล้วทำการคำนวณหาค่า T_n จากสมการ $T_n = \frac{\lfloor x_n \rfloor \times \beta^n}{Y}$ แสดงวิธีการคำนวณดังตารางที่ 3.4

ตารางที่ 3.4 แสดงการคำนวณ T_n

n	$\frac{\lfloor x_n \rfloor \times \beta^n}{Y}$	T_n
6	$\frac{\lfloor 1.4704 \rfloor \times 2^6}{25}$	2.56
5	$\frac{\lfloor 0.9408 \rfloor \times 2^5}{25}$	0
4	$\frac{\lfloor 1.8816 \rfloor \times 2^4}{25}$	0.64
3	$\frac{\lfloor 1.7632 \rfloor \times 2^3}{25}$	0.32
2	$\frac{\lfloor 1.5264 \rfloor \times 2^2}{25}$	0.16
1	$\frac{\lfloor 1.0528 \rfloor \times 2^1}{25}$	0.08
0	$\frac{\lfloor 0.1056 \rfloor \times 2^0}{25}$	0
-1	$\frac{\lfloor 0.2112 \rfloor \times 2^{-1}}{25}$	0
-2	$\frac{\lfloor 0.4224 \rfloor \times 2^{-2}}{25}$	0

หาค่า Q_n จากสมการ $Q_n = \left(\frac{x_n}{y}\right) \bmod \beta$ แสดงการคำนวณดังตารางที่ 3.5

ตารางที่ 3.5 แสดงการคำนวณค่า Q_n

n	$\left(\frac{x_n}{y}\right) \bmod \beta$	Q_n
6	$\left(\frac{1.4704}{25}\right) \bmod 2$	0.058816
5	$\left(\frac{0.9408}{25}\right) \bmod 2$	0.037632
4	$\left(\frac{1.8816}{25}\right) \bmod 2$	0.075264
3	$\left(\frac{1.7632}{25}\right) \bmod 2$	0.070528
2	$\left(\frac{1.5264}{25}\right) \bmod 2$	0.061056
1	$\left(\frac{1.0528}{25}\right) \bmod 2$	0.042112
0	$\left(\frac{0.1056}{25}\right) \bmod 2$	0.004224
-1	$\left(\frac{0.2112}{25}\right) \bmod 2$	0.008448
-2	$\left(\frac{0.4224}{25}\right) \bmod 2$	0.016896

เมื่อได้ตัวแปรทั้งหมดแล้วสามารถหาค่า z_n จากสมการ ที่ 3.1

$$z_n = \begin{cases} Q_n & : n = L \\ \left(\left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right)\right) \bmod \beta + Q_n\right) \bmod \beta & : n < L \end{cases}$$

สามารถคำนวณได้ดังตารางที่ 3.6

ตารางที่ 3.6 แสดงการคำนวณค่า z_n

n	$\left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right)\bmod \beta + Q_n\right)\bmod \beta$	z_n
6	Q_2	0.058816
5	$\left(\left(\sum_{i=5+1}^{6}\left(\frac{T_i}{2^5}\right)\right)\bmod 2 + 0.037632\right)\bmod 2$	0.117632
4	$\left(\left(\sum_{i=4+1}^{6}\left(\frac{T_i}{2^4}\right)\right)\bmod 2 + 0.075264\right)\bmod 2$	0.235264
3	$\left(\left(\sum_{i=3+1}^{6}\left(\frac{T_i}{2^3}\right)\right)\bmod 2 + 0.070528\right)\bmod 2$	0.470528
2	$\left(\left(\sum_{i=2+1}^{6}\left(\frac{T_i}{2^2}\right)\right)\bmod 2 + 0.061056\right)\bmod 2$	0.941056
1	$\left(\left(\sum_{i=1+1}^{6}\left(\frac{T_i}{2^1}\right)\right)\bmod 2 + 0.042112\right)\bmod 2$	1.882112
0	$\left(\left(\sum_{i=0+1}^{6}\left(\frac{T_i}{2^0}\right)\right)\bmod 2 + 0.004224\right)\bmod 2$	1.764224
-1	$\left(\left(\sum_{i=-1+1}^{6}\left(\frac{T_i}{2^{-1}}\right)\right)\bmod 2 + 0.008448\right)\bmod 2$	1.528448
-2	$\left(\left(\sum_{i=-2+1}^{6}\left(\frac{T_i}{2^{-2}}\right)\right)\bmod 2 + 0.016896\right)\bmod 2$	1.056896

นำค่าที่คำนวณได้ทั้งหมดแสดงในตารางที่ 3.7

ตารางที่ 3.7 แสดงการหาร x = 73.52 และ y = 25 สำหรับฐาน β = 2

n	x_n	y_n	T_n	Q_n	z_n
6	1.4704	0.5	2.56	0.058816	0.058816
5	0.9408	1	0	0.037632	0.117632
4	1.8816	0	0.64	0.075264	0.235264
3	1.7632	0	0.32	0.070528	0.470528
2	1.5264	0	0.16	0.061056	0.941056
1	1.0528	0	0.08	0.042112	1.882112
0	0.1056	0	0	0.004224	1.764224
-1	0.2112	0	0	0.008448	1.528448
-2	0.4224	0	0	0.016896	1.056896

คำตอบที่ได้จากการคำนวณนี้ คือ (0.058816, 0.117632, 0.235264, 0.470528, 0.941056, 1.882112, 1.764224 | 1.528448, 1.056896) ถ้าพิจารณาคำตอบที่ถูกต้องของ 73.52 หารด้วย 25 จะได้ค่าเท่ากับ 2.9408 โดยที่ 2.9408 ในระบบ X = 100 และฐาน β = 2 นั้น ค่าของ 2.9408 เป็นค่าของผลลัพธ์ของการหารที่แท้จริง

เช่นเดียวกันกับตัวอย่างที่แล้ว นำค่าดิจิตค่าต่อเนื่องหลักที่ n=6 มาคำนวณด้วยสมการที่ 3.4 โดยนำค่า z_n มาแทนค่าในสมการดังกล่าวจะได้ผลลัพธ์ดังนี้

$$z_n = z_L(X/\beta)$$
$$z_n = 0.058816(100/2)$$
$$z_n = 2.9408$$

สำหรับดิจิตถัดไปคำนวณในสมการ 3.14 จะได้ดิจิตที่มีค่าแบบต่อเนื่องดังนี้

$$z_5 = (1.4704 - \lfloor 1.4704 \rfloor)2 = 0.9408$$
$$z_4 = (0.9408 - \lfloor 0.9408 \rfloor)2 = 1.8816$$
$$z_3 = (1.8816 - \lfloor 1.8816 \rfloor)2 = 1.7632$$
$$z_2 = (1.7632 - \lfloor 1.7632 \rfloor)2 = 1.5264$$
$$z_1 = (1.5264 - \lfloor 1.5264 \rfloor)2 = 1.0528$$
$$z_0 = (1.0528 - \lfloor 1.0528 \rfloor)2 = 0.1056$$
$$z_{-1} = (0.1056 - \lfloor 0.1056 \rfloor)2 = 0.2112$$
$$z_{-2} = (0.2112 - \lfloor 0.2112 \rfloor)2 = 0.4224$$

คำตอบที่ได้สังเกตเห็นว่าตรงกับผลหารที่แท้จริง ซึ่งการคำนวณบนฐานใดๆ สามารถแสดงผลลัพธ์ออกมาได้ถูกต้อง และยังคงคุณสมบัติของดิจิตค่าต่อเนื่องอีกด้วย □

ในบทต่อไปจะกล่าวถึงการคำนวณของการหารกรณีเกิดสัญญาณรบกวนขึ้นในระบบ รวมถึงการกู้ค่าความผิดพลาด โดยจะแสดงวิธีการคำนวณอย่างละเอียด

บทที่ 4

การคำนวณกรณีเกิดค่าความผิดพลาดและการกู้ค่าความผิดพลาด

จากบทที่ 3 ได้แสดงอัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง ซึ่งแสดงให้เห็นว่าการหารสามารถคำนวณได้บนฐานใดๆ และมีคุณสมบัติเป็นดิจิตค่าต่อเนื่อง ในบทนี้จะแสดงการคำนวณจากอัลกอริทึมการหารที่ได้นำเสนอไปแล้วในบทที่ 3 ซึ่งมีสัญญาณรบกวนเกิดขึ้นในทุกๆ ดิจิตค่าต่อเนื่อง โดยค่าความผิดพลาดที่เกิดขึ้นจากสัญญาณรบกวน จะต้องไม่เกินค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม เมื่อเกิดความผิดพลาดเกิดขึ้นในแต่ละดิจิตแล้ว สามารถทำการกู้ค่าความผิดพลาดโดยใช้อัลกอริทึมในการกู้ค่าความผิดพลาด ที่ได้แสดงในบทที่ 2 เพื่อแสดงให้เห็นว่าอัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง สามารถกู้ค่าความผิดพลาดได้ผลลัพธ์ใกล้เคียงค่าเดิม

4.1 การคำนวณกรณีเกิดความผิดพลาดและการกู้ค่าความผิดพลาด

จากบทที่ 2 ได้กล่าวถึงการแสดงการคำนวณดิจิตค่าต่อเนื่อง เมื่อเกิดค่าความผิดพลาดขึ้นในแต่ละดิจิต ค่าความผิดพลาดที่เกิดขึ้นจะถูกนำมารวมเข้ากับดิจิตแต่ละดิจิตของ z_n มีรูปแบบ $z'_n = z_n \times \varepsilon$ โดย z'_n คือค่าของ z_n ที่เกิดสัญญาณรบกวนทำให้เกิดความผิดพลาดขึ้น และ ε คือค่าความผิดพลาดในแต่ละดิจิต

ในบทนี้ จะพิจารณาถึงรูปแบบของความผิดพลาดอีกแบบหนึ่ง นั่นคือ ความผิดพลาดที่อยู่ในรูปของ $z'_n = z_n + \varepsilon$ ดังแสดงในทฤษฎีบทต่อไป

ทฤษฎีบทที่ 4.1 *กำหนดให้ $x = (x_L\ x_{L-1}\ x_{L-2} \ldots \mid \ldots x_K)$ เป็นรูปแบบแทนจำนวนในระบบจำนวนค่าต่อเนื่องฐาน β ในกรณีที่ระบบมีค่าความผิดพลาดในแต่ละดิจิตค่าต่อเนื่องเป็น*

$$x'_n = x_n + \varepsilon$$

โดย ε เป็นค่าความผิดพลาดในช่วงค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม ค่าของ x''_n จากการกู้ค่าความผิดพลาด จะได้ผลลัพธ์ที่ต่างจาก x เป็น

$$\left| x'' - x \right| \le \frac{1}{2\beta^{L-K}} \tag{4.1}$$

พิสูจน์

สมมุติให้ $x' = (x'_L\ x'_{L-1}\ x'_{L-2} \dots \mid \dots x'_K)$ เป็นรูปแบบแทนจำนวนในระบบจำนวนค่าต่อเนื่องฐาน β ที่มีค่าความผิดพลาด ε โดยที่

$$x'_n = x_n + \varepsilon$$

และ

$$\varepsilon < \left|\frac{1}{2\beta}\right|\beta$$

จากอัลกอริทึมกู้ค่าความผิดพลาดในตำแหน่งที่ K

$$x''_K = x'_K = x_K + \varepsilon$$

ค่าความผิดพลาด (ε_K) ในตำแหน่งที่ K คือ

$$\varepsilon_n = x''_K - x_K = \varepsilon$$

ในการกู้ค่าความผิดพลาด

$$x''_n = \lfloor x'_n \rfloor + (x''_{n-1} / \beta)$$

พิจารณาตำแหน่งที่ n ในกรณีที่มีค่า ε_n เกิดขึ้นจากการกู้ค่าความผิดพลาดตำแหน่งที่ n-1

$$x''_n = \lfloor x'_n \rfloor_R + \frac{(x_{n-1} + \varepsilon_{n-1})}{\beta}$$

$$x''_n = \lfloor x'_n \rfloor_R + \left(\frac{x_{n-1}}{\beta}\right) + \left(\frac{\varepsilon_{n-1}}{\beta}\right)$$

จากเงื่อนไขของการกู้ค่าความผิดพลาดตามสมการที่ 2.4 ที่ว่า $\lfloor x'_n \rfloor_R = \lfloor x_n \rfloor$ จะได้ว่า

$$x''_n = \lfloor x_n \rfloor + \left(\frac{x_{n-1}}{\beta}\right) + \left(\frac{\varepsilon_{n-1}}{\beta}\right)$$

$$x''_n = \lfloor x_n \rfloor + \frac{(x_n - \lfloor x_n \rfloor)\beta}{\beta} + \frac{\varepsilon_{n-1}}{\beta}$$

$$x''_n = x_n + \frac{\varepsilon_{n-1}}{\beta}$$

เพราะฉะนั้น

$$x''_n = x_n + \frac{\varepsilon_n}{\beta}\ ; k < n \le L$$

โดย $$x''_K = x_K + \varepsilon_K \text{ และ } |\varepsilon_K| \le \frac{1}{2}$$

ดังนั้น $$x''_L = \frac{\varepsilon_K}{\beta^{L-K}} \le \frac{1}{2\beta^{L-K}}$$ ■

4.2 ตัวอย่างการคำนวณ

ตัวอย่าง 4.1 *แสดงการกู้ค่าความผิดพลาด เมื่อกำหนดค่าความผิดพลาด ε_n = 0.0106875 เทียบกับค่าความผิดพลาด ε_n = 6% สำหรับ x = 34.2 y = 12 บนฐาน β = 2 และ X = 32 ผลลัพธ์ของการหาร z = 2.85*

แสดงการกู้ค่าความผิดพลาดทั้งสองกรณีในตารางที่ 4.1

ตารางที่ 4.1 แสดงการกู้ค่าความผิดพลาดจากการบวกและคูณค่าความผิดพลาด

n	4	3	2	1	0	-1
z_n	0.178125	0.35625	0.7125	1.425	0.85	1.7
$z'_n = z_n + \varepsilon_n$	0.1888125	0.3669375	0.7231875	1.4356875	0.8606875	1.7106875
z''_n	0.17845899	0.35691798	0.71383595	1.4276719	0.8553438	1.7106875
$z'_n = z_n \times \varepsilon_n$	0.1888125	0.377625	0.75525	1.5105	0.901	1.802
Z''_n	0.1813125	0.362625	0.72525	1.4505	0.901	1.802

จากตารางที่ 4.1 แสดงให้เห็นว่า อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง เมื่อเกิดค่าความผิดพลาด การกู้ค่าความผิดพลาดสามารถทำได้ทั้งสองกรณี โดยการเกิดค่าความผิดพลาดกรณีค่าความผิดพลาดรวมเข้าไปในแต่ละดิจิต ทำให้การกู้ค่าความผิดพลาดกลับมาได้ค่าใกล้เคียงกว่ากรณีที่ค่าความผิดพลาดคูณเข้าไปในแต่ละดิจิต □

ตัวอย่าง 4.2 *การหารบนระบบจำนวนค่าต่อเนื่อง โดยกำหนดให้ x = 645.2 และ y = 42 ด้วยค่า X=1000 สำหรับฐาน β = 10 ค่าความผิดพลาดที่เกิดขึ้นในระบบเท่ากับ 3% จากสมการ $X = \beta^{L+1}$ คำนวณได้ค่า L = 2 และกำหนดให้ K = -1*

วิธีทำ

คำนวณหาค่าดิจิตค่าต่อเนื่องของ x_n และ y_n บนฐาน 10 หาค่า T_n, Q_n และ z_n เพื่อได้ผลหารที่เป็นดิจิตค่าต่อเนื่อง บนฐาน 10 หลังจากนั้นใส่ค่าความผิดพลาดที่ z_n ในแต่ละดิจิต แล้วทำการกู้ค่าความผิดพลาด

วิธีคำนวณ

จากสมการ $x_L = \beta \times \frac{x}{X}$ สามารถคำนวณหาค่าดิจิตเริ่มต้น x_L สำหรับดิจิตต่อไปหาได้จากสมการ $x_n = (x_{n+1} - \lfloor x_{n+1} \rfloor)\beta$ ดังนี้

$$x_2 = 10 \times \frac{645.2}{1000} = 6.452$$

สำหรับดิจิตต่อไปคำนวณได้ดังแสดงในตารางที่ 4.2

จากวิธีการเดียวกันกับการหาดิจิตต่อเนื่อง x_n สามารถคำนวณหาดิจิต y_n ได้ดังนี้

$$y_2 = 10 \times \frac{42}{1000} = 0.42$$

สำหรับดิจิตต่อไปคำนวณได้ดังแสดงในตารางที่ 4.3

ตารางที่ 4.2 แสดงการคำนวณค่า x_n

n	$(x_{n+1} - \lfloor x_{n+1} \rfloor)\beta$	x_n
1	$(6.452 - \lfloor 6.452 \rfloor) \times 10$	4.52
0	$(4.52 - \lfloor 4.52 \rfloor) \times 10$	5.2
-1	$(5.2 - \lfloor 5.2 \rfloor) \times 10$	2

ตารางที่ 4.3 แสดงการคำนวณค่า y_n

n	$(y_{n+1} - \lfloor y_{n+1} \rfloor)\beta$	y_n
1	$(0.42 - \lfloor 0.42 \rfloor) \times 10$	4.2
0	$(4.2 - \lfloor 4.2 \rfloor) \times 10$	2
-1	$(2 - \lfloor 2 \rfloor) \times 10$	0

หลังจากได้ดิจิตค่าต่อเนื่อง x_n และ y_n แล้วทำการคำนวณหาค่า T_n จากสมการที่ 4.2 แสดงวิธีการคำนวณดังตารางที่ 4.4

ตารางที่ 4.4 แสดงการคำนวณ T_n

n	$\dfrac{\lfloor x_n \rfloor \times \beta^n}{Y}$	T_n
2	$\dfrac{\lfloor 6.452 \rfloor \times 10^2}{42}$	14.2857143
1	$\dfrac{\lfloor 4.52 \rfloor \times 10^1}{42}$	0.952380952
0	$\dfrac{\lfloor 5.2 \rfloor \times 10^0}{42}$	0.119047619
-1	$\dfrac{\lfloor 2 \rfloor \times 10^{-1}}{42}$	0.0047619047

หาค่า Q_n จากสมการ $Q_n = \left(\dfrac{x_n}{y}\right) \bmod \beta$ แสดงการคำนวณดังตารางที่ 4.5

ตารางที่ 4.5 แสดงการคำนวณค่า Q_n

n	$\left(\dfrac{x_n}{y}\right) \bmod \beta$	Q_n
2	$\left(\dfrac{6.452}{42}\right) \mathrm{mod} 10$	0.153619048
1	$\left(\dfrac{4.52}{42}\right) \mathrm{mod} 10$	0.107619048
0	$\left(\dfrac{5.2}{42}\right) \mathrm{mod} 10$	0.123809524
-1	$\left(\dfrac{2}{42}\right) \mathrm{mod} 10$	0.0476190476

เมื่อได้ตัวแปรทั้งหมดแล้วสามารถหาค่า z_n จากสมการ ที่ 3.1

$$z_n = \begin{cases} Q_n & : n = L \\ \left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right) \bmod \beta + Q_n\right) \bmod \beta & : n < L \end{cases}$$

สามารถคำนวณได้ดังตารางที่ 4.6

ตารางที่ 4.6 แสดงการคำนวณค่า z_n

N	$\left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right) \bmod \beta + Q_n\right) \bmod \beta$	z_n
2	Q_2	0.153619048
1	$\left(\left(\sum_{i=1+1}^{2}\left(\frac{T_i}{10^1}\right)\right) \bmod 10 + 0.107619048\right) \bmod 10$	1.53619048
0	$\left(\left(\sum_{i=0+1}^{2}\left(\frac{T_i}{10^0}\right)\right) \bmod 10 + 0.123809524\right) \bmod 10$	5.3619048
-1	$\left(\left(\sum_{i=0}^{2}\left(\frac{T_i}{10^0}\right)\right) \bmod 10 + 0.047619047\right) \bmod 10$	3.619048

นำค่า z_n ที่ได้จากการคำนวณใส่ค่าความผิดพลาดเท่ากับ 3% ทุกๆ ดิจิต ดังนี้

$$z'_2 = 0.153619048 \times 1.03 = 0.158227619$$

$$z'_1 = 1.53619048 \times 1.03 = 1.582276194$$

$$z'_0 = 5.3619048 \times 1.03 = 5.522761944$$

$$z'_{-1} = 3.619048 \times 1.03 = 3.72761944$$

เมื่อเกิดค่าความผิดพลาดขึ้นในระบบสามารถกู้ค่าความผิดพลาดจากสมการที่ 4.5 แสดงการคำนวณดังนี้

$$z''_{-1} = 3.72761944$$

$$z''_0 = \lfloor 5.15 \rfloor_R + (3.72761944/10) = 5.372761944$$

$$z_1'' = \lfloor 1.045 \rfloor_R + (5.372761944/10) = 1.53276194$$

$$z_2'' = \lfloor 0.004951425 \rfloor_R + (1.53276194/10) = 0.153276194$$

นำค่าที่คำนวณได้ทั้งหมดแสดงในตารางที่ 4.7

ตารางที่ 4.7 แสดงการหาร x = 645.2 และ y = 42 ซึ่งมีค่าความผิดพลาดเท่ากับ 3%

n	2	1	0	-1
x_n	6.452	4.52	5.2	2
y_n	0.42	4.2	2	0
T_n	14.2857143	0.952380952	0.119047619	0.00476190476
Q_n	0.153619048	0.107619048	0.123809524	0.0476190476
z_n	0.153619048	1.53619048	5.3619048	3.619048
z_n'	0.158227619	1.582276194	5.522761944	3.72761944
z_n''	0.153276194	1.53276194	5.372761944	3.72761944

คำตอบที่ได้จากการคำนวณนี้ก่อนที่จะเกิดค่าความผิดพลาด คือ (0.153619048, 1.53619048, 5.3619048 | 3.619048) เมื่อเกิดค่าความผิดพลาดในแต่ละดิจิตมีขนาด 0.562 ทำการกู้ค่าความผิดพลาดได้คำตอบคือ (0.153276194,1.53276194,5.372761944 | 3.72761944) เห็นว่าค่าที่กู้มายังไม่เท่ากับค่าเดิม เนื่องจากยังมีความผิดพลาด ซึ่งจะทำให้กระทบกับทุกๆ ดิจิตที่จะทำการกู้ค่าความผิดพลาด ทำให้ค่าคำตอบที่ได้ไม่ตรงกับค่าที่แท้จริง การเกิดค่าความผิดพลาดจากดิจิตที่มีลำดับน้อยที่สุดกระจายไปยังทุกๆ ดิจิตนั้นจะเรียกว่า การแพร่กระจายของความผิดพลาดที่เกิดจากดิจิตที่มีค่าความสำคัญน้อยที่สุด (Least significant digit error propagation) ซึ่งค่าความผิดพลาดนี้เกิดจากการที่ไม่สามารถจะทำการปรับปรุงค่าของดิจิต z_k ได้ตามสมการการกู้ค่าความผิดพลาดในกรณีที่ $n = K$ แต่สังเกตเห็นว่าดิจิตแต่ละดิจิตที่ทำการที่กู้ค่าความผิดพลาดได้คำตอบที่มีค่ายอมรับได้ □

นอกจากการแสดงการคำนวณบนฐาน 10 แล้ว ตัวอย่างต่อไปจะแสดงวิธีการคำนวณดิจิตค่าต่อเนื่อง เกิดค่าความผิดพลาดขึ้นแต่ละดิจิต พร้อมทั้งการกู้ค่าความผิดพลาด บนฐาน 2 อย่างละเอียด

ตัวอย่างที่ 4.3 *การหารบนระบบจำนวนค่าต่อเนื่อง โดยกำหนดให้ x= 92.2 และ y=32 ด้วยค่า* X *= 100 สำหรับฐาน* β *= 2 ค่าความผิดพลาดที่เกิดขึ้นในระบบ 0.0842 จากสมการ* $X = \beta^{L+1}$ *คำนวณได้ค่า* $L = 6$ *และกำหนดให้* K *= -2*

วิธีทำ

คำนวณหาค่าดิจิตค่าต่อเนื่องของ x_n และ y_n บนฐาน 2 หาค่า T_n , Q_n และ z_n เพื่อได้ผลหารที่เป็นดิจิตค่าต่อเนื่อง บนฐาน 2 หลังจากนั้นใส่ค่าความผิดพลาดที่ z_n ในแต่ละดิจิต แล้วทำการกู้ค่าความผิดพลาด

วิธีคำนวณ

จากสมการ $x_L = \beta \times \frac{x}{X}$ สามารถคำนวณหาค่าดิจิตเริ่มต้น x_L สำหรับดิจิตต่อไปหาได้จากสมการ $x_n = (x_{n+1} - \lfloor x_{n+1} \rfloor)\beta$ ดังนี้

$$x_6 = 2 \times \frac{92.2}{100} = 1.844$$

สำหรับดิจิตต่อไปคำนวณได้ดังแสดงในตารางที่ 4.8

จากวิธีการเดียวกันกับการหาดิจิตต่อเนื่อง x_n สามารถคำนวณหาดิจิต y_n ได้ดังนี้

$$y_6 = 2 \times \frac{32}{100} = 0.64$$

สำหรับดิจิตต่อไปคำนวณได้ดังแสดงในตารางที่ 4.9

ตารางที่ 4.8 แสดงการคำนวณค่า x_n

n	$(x_{n+1}-\lfloor x_{n+1}\rfloor)\beta$	x_n
5	$(1.844 - \lfloor 1.844 \rfloor)\times 2$	1.688
4	$(1.688 - \lfloor 1.688 \rfloor)\times 2$	1.376
3	$(1.376 - \lfloor 1.376 \rfloor)\times 2$	0.752
2	$(0.752 - \lfloor 0.752 \rfloor)\times 2$	1.504
1	$(1.504 - \lfloor 1.504 \rfloor)\times 2$	1.008
0	$(1.008 - \lfloor 1.008 \rfloor)\times 2$	0.016
-1	$(0.016 - \lfloor 0.016 \rfloor)\times 2$	0.032
-2	$(0.032 - \lfloor 0.032 \rfloor)\times 2$	0.064

ตารางที่ 4.9 แสดงการคำนวณค่า y_n

n	$(y_{n+1}-\lfloor y_{n+1}\rfloor)\beta$	y_n
5	$(0.64 - \lfloor 0.64 \rfloor)\times 2$	1.28
4	$(1.28 - \lfloor 1.28 \rfloor)\times 2$	0.56
3	$(0.56 - \lfloor 0.56 \rfloor)\times 2$	1.12
2	$(1.12 - \lfloor 1.12 \rfloor)\times 2$	0.24
1	$(0.24 - \lfloor 0.24 \rfloor)\times 2$	0.48
0	$(0.48 - \lfloor 0.48 \rfloor)\times 2$	0.96
-1	$(0.96 - \lfloor 0.96 \rfloor)\times 2$	1.92
-2	$(1.92 - \lfloor 1.92 \rfloor)\times 2$	1.84

หลังจากได้ดิจิตค่าแบบต่อเนื่อง x_n และ y_n แล้วทำการคำนวณหาค่า T_n จากสมการ $T_n = \frac{\lfloor x_n \rfloor \times \beta^n}{Y}$ แสดงวิธีการคำนวณดังตารางที่ 4.10

ตารางที่ 4.10 แสดงการคำนวณ T_n

n	$\frac{\lfloor x_n \rfloor \times \beta^n}{Y}$	T_n
6	$\frac{\lfloor 1.844 \rfloor \times 2^6}{32}$	2
5	$\frac{\lfloor 1.688 \rfloor \times 2^5}{32}$	1
4	$\frac{\lfloor 1.376 \rfloor \times 2^4}{32}$	0.5
3	$\frac{\lfloor 0.752 \rfloor \times 2^3}{32}$	0
2	$\frac{\lfloor 1.504 \rfloor \times 2^2}{32}$	0.125
1	$\frac{\lfloor 1.008 \rfloor \times 2^1}{32}$	0.0625
0	$\frac{\lfloor 0.016 \rfloor \times 2^0}{32}$	0
-1	$\frac{\lfloor 0.032 \rfloor \times 2^{-1}}{32}$	0
-2	$\frac{\lfloor 0.064 \rfloor \times 2^{-2}}{32}$	0

หาค่า Q_n จากสมการ $Q_n = \left(\frac{x_n}{y}\right) \bmod \beta$ แสดงการคำนวณดังตารางที่ 4.11

ตารางที่ 4.11 แสดงการคำนวณค่า Q_n

n	$\left(\frac{x_n}{y}\right) \bmod \beta$	Q_n
6	$\left(\frac{1.844}{32}\right) \bmod 2$	0.057625
5	$\left(\frac{1.688}{32}\right) \bmod 2$	0.05275
4	$\left(\frac{1.376}{32}\right) \bmod 2$	0.043
3	$\left(\frac{0.752}{32}\right) \bmod 2$	0.0235
2	$\left(\frac{1.504}{32}\right) \bmod 2$	0.047
1	$\left(\frac{1.008}{32}\right) \bmod 2$	0.0315
0	$\left(\frac{0.016}{32}\right) \bmod 2$	0.0005
-1	$\left(\frac{0.032}{32}\right) \bmod 2$	0.001
-2	$\left(\frac{0.064}{32}\right) \bmod 2$	0

เมื่อได้ตัวแปรทั้งหมดแล้วสามารถหาค่า z_n จากสมการ ที่ 3.1

$$z_n = \begin{cases} Q_n & : n = L \\ \left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right) \bmod \beta + Q_n\right) \bmod \beta & : n < L \end{cases}$$

สามารถคำนวณได้ดังตารางที่ 4.12

ตารางที่ 4.12 แสดงการคำนวณค่า z_n

n	$\left(\left(\sum_{i=n+1}^{L}\left(\frac{T_i}{\beta^n}\right)\right)\bmod \beta + Q_n\right)\bmod \beta$	z_n
6	Q_2	0.057625
5	$\left(\left(\sum_{i=5+1}^{6}\left(\frac{T_i}{2^5}\right)\right)\bmod 2 + 0.05275\right)\bmod 2$	0.11525
4	$\left(\left(\sum_{i=4+1}^{6}\left(\frac{T_i}{2^4}\right)\right)\bmod 2 + 0.043\right)\bmod 2$	0.2305
3	$\left(\left(\sum_{i=3+1}^{6}\left(\frac{T_i}{2^3}\right)\right)\bmod 2 + 0.0235\right)\bmod 2$	0.461
2	$\left(\left(\sum_{i=2+1}^{6}\left(\frac{T_i}{2^2}\right)\right)\bmod 2 + 0.047\right)\bmod 2$	0.922
1	$\left(\left(\sum_{i=1+1}^{6}\left(\frac{T_i}{2^1}\right)\right)\bmod 2 + 0.0315\right)\bmod 2$	1.844
0	$\left(\left(\sum_{i=0+1}^{6}\left(\frac{T_i}{2^0}\right)\right)\bmod 2 + 0.0005\right)\bmod 2$	1.688
-1	$\left(\left(\sum_{i=-1+1}^{6}\left(\frac{T_i}{2^{-1}}\right)\right)\bmod 2 + 0.001\right)\bmod 2$	1.376
-2	$\left(\left(\sum_{i=-2+1}^{6}\left(\frac{T_i}{2^{-2}}\right)\right)\bmod 2 + 0.002\right)\bmod 2$	0.752

นำค่า z_n ที่ได้จากการคำนวณใส่ค่าความผิดพลาดเท่ากับ 0.0842 ทุกๆดิจิต ดังนี้

$$z'_6 = 0.057682 + 0.0842 = 0.14825$$

$$z'_5 = 0.11525 + 0.0842 = 0.19945$$

$$z'_4 = 0.2305 + 0.0842 = 0.3147$$

$$z'_3 = 0.461 + 0.0842 = 0.5452$$

$$z'_2 = 0.922 + 0.0842 = 1.0062$$

$$z'_1 = 1.844 + 0.0842 = 1.9282$$

$$z'_0 = 1.688 + 0.0842 = 1.7722$$

$$z'_{-1} = 1.376 + 0.0842 = 1.4602$$

$$z'_{-2} = 0.752 + 0.0842 = 0.8362$$

เมื่อเกิดค่าความผิดพลาดขึ้นในระบบสามารถกู้ค่าความผิดพลาดจากสมการที่ 4.6 แสดงการคำนวณดังนี้

$$z''_{-2} = 0.8362$$

$$z''_{-1} = \lfloor 1.0421 \rfloor_R + (0.8362/2) = 1.4181$$

$$z''_0 = \lfloor 1.06315 \rfloor_R + (1.4181/2) = 1.70905$$

$$z''_1 = \lfloor 1.073675 \rfloor_R + (1.70905/2) = 1.854525$$

$$z''_2 = \lfloor 0.0789375 \rfloor_R + (1.854525/2) = 0.9272625$$

$$z''_3 = \lfloor 0.08156875 \rfloor_R + (0.9272625/2) = 0.46363125$$

$$z''_4 = \lfloor 0.08288438 \rfloor_R + (0.46363125/2) = 0.23181563$$

$$z''_5 = \lfloor 0.08354219 \rfloor_R + (0.23181563/2) = 0.11590782$$

$$z''_6 = \lfloor 0.08387109 \rfloor_R + (0.11590782/2) = 0.05795391$$

นำค่าที่คำนวณได้ทั้งหมดแสดงในตารางที่ 4.13

ตารางที่ 4.13 แสดงการหาร x = 92.2 และ y = 32 ซึ่งมีค่าความผิดพลาดเท่ากับ 0.0842

n	x_n	y_n	T_n	Q_n	z_n	z'_n	z''_n
6	1.844	0.64	2	0.057625	0.057625	0.14825	0.05795391
5	1.688	1.28	1	0.05275	0.11525	0.19945	0.11590782
4	1.376	0.56	0.5	0.043	0.2305	0.3147	0.23181563
3	0.752	1.12	0	0.0235	0.461	0.5452	0.46363125
2	1.504	0.24	0.125	0.047	0.922	1.0062	0.9272625
1	1.008	0.48	0.0625	0.0315	1.844	1.9282	1.854525
0	0.016	0.96	0	0.0005	1.688	1.7722	1.70905
-1	0.032	1.92	0	0.001	1.376	1.4602	1.4181
-2	0.064	1.84	0	0.002	0.752	0.8362	0.8362

คำตอบที่ได้จากการคำนวณนี้ก่อนที่จะเกิดค่าความผิดพลาด คือ (0.057625, 0.11525, 0.2305, 0.461, 0.922, 1.844, 1.688 | 1.376, 0.752) เมื่อเกิดค่าความผิดพลาดในแต่ละดิจิตมีขนาด 0.0842 ทำการกู้ค่าความผิดพลาดได้คำตอบคือ(0.05795391, 0.11590782, 0.23181563, 0.46363125, 0.9272625, 1.854525, 1.70905 | 1.4181, 0.8362) เห็นว่าค่าที่กู้มายังไม่เท่ากับค่าเดิม แต่ดิจิตแต่ละดิจิตที่ทำการที่กู้ค่าความผิดพลาดได้คำตอบมีค่าที่ใกล้เคียงค่าเดิม เช่นเดียวกับการคำนวณบนฐานสิบ ซึ่งเป็นการแสดงให้เห็นว่าอัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง สามารถคำนวณได้บนฐานใดๆ □

สำหรับการคำนวณบนฐานสอง สิ่งที่เป็นข้อสังเกตคือ การกำหนดค่า K โดยปกติจะกำหนดให้ค่า K ตามตำแหน่งของทศนิยม จากที่ได้กล่าวแล้วว่าการกำหนดค่า K ยิ่งมีจำนวนของ K มากๆ จะทำให้การกู้ค่าความผิดพลาดกลับมาใกล้เคียงค่าเดิมมากยิ่งขึ้น ตัวอย่างต่อไปจะใช้ตัวเลขเดียวกันกับตัวอย่างที่ 4.2 แต่จะกำหนดค่า K ให้มีจำนวนมากขึ้น ซึ่งหมายความว่ากำหนดค่า K ยิ่งต่ำลงมากๆ เวลาที่ใช้คำนวณก็จะมากขึ้นด้วยเช่นกัน

ตัวอย่าง 4.4 *จากตัวอย่างที่ 4.3 กำหนดค่า K = -2 สำหรับตัวอย่างนี้ จะกำหนดค่า K = -4*

ผลลัพธ์การกู้ค่าความผิดพลาดแสดงในตารางที่ 4.14

ตารางที่ 4.14 แสดงผลลัพธ์หลังจากกู้ค่าความผิดพลาดกรณีกำหนดค่า K มากขึ้น

n	z_n	z'_n	z''_n
6	0.057625	0.14825	0.05770724
5	0.11525	0.19945	0.11541448
4	0.2305	0.3147	0.23082895
3	0.461	0.5452	0.4616579
2	0.922	1.0062	0.9233158
1	1.844	1.9282	1.8466315
0	1.688	1.7722	1.693263
-1	1.376	1.4602	1.386525
-2	0.752	0.8362	0.77305
-3	1.504	1.5882	1.5461
-4	1.008	1.0922	1.0922

จากผลลัพธ์ของตัวอย่างนี้ สังเกตเห็นว่าค่าที่กู้ค่าความผิดพลาดกลับมา เมื่อเทียบดิจิตต่อดิจิต พบว่าการกำหนดค่า K ยิ่งต่ำลง จากตัวอย่างนี้ K = -4 ทำให้การกู้ค่าความผิดพลาดได้ใกล้เคียงกว่าการที่กำหนดค่า K = -2 แต่จะใช้เวลาการคำนวณมากขึ้น เนื่องจากต้องคำนวณดิจิตเพิ่มขึ้นอีก 2 หลัก □

4.3 บทสรุป

ในบทนี้ได้แสดงให้เห็นถึงการคำนวณอย่างละเอียด เมื่อดิจิตค่าต่อเนื่องเกิดความผิดพลาดขึ้นในทุกๆ ดิจิต สามารถกู้ค่าความผิดพลาดกลับมาได้ค่าที่ยอมรับได้ โดยค่าความผิดพลาดจะต้องน้อยกว่าค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม และยังได้แสดงให้เห็นถึงการคำนวณโดยกำหนดจำนวนดิจิต ถ้าคำนวณด้วยการกำหนดจำนวนดิจิตมากๆ จะทำให้การกู้ค่าความผิดพลาดใกล้เคียงกว่าการกำหนดดิจิตที่น้อยกว่า

บทที่ 5

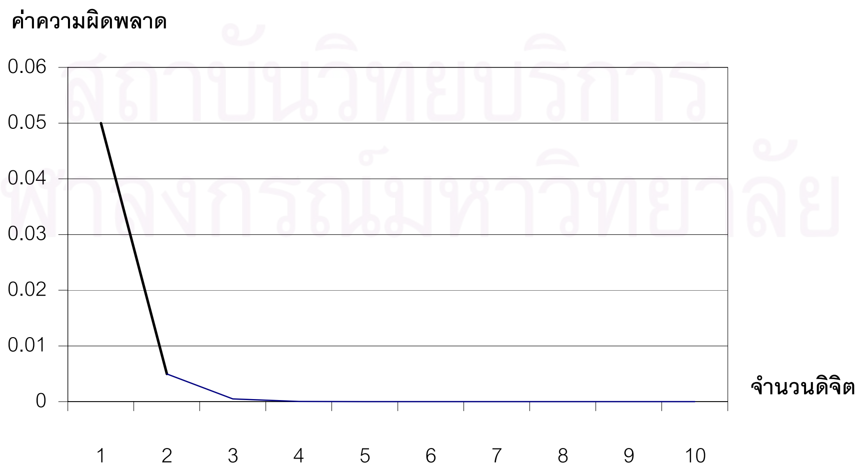
การวิเคราะห์การหารบนระบบจำนวนค่าต่อเนื่อง

จากบทที่ 4 ได้กล่าวถึงการคำนวณกรณีเกิดค่าความผิดพลาด และการกู้ค่าความผิดพลาด โดยการกำหนดจำนวนดิจิตยิ่งมาก จะทำให้การกู้ค่าความผิดพลาดกลับมาได้ใกล้เคียงมากกว่าการกำหนดดิจิตที่น้อยกว่า ซึ่งในบทนี้จะแสดงกราฟแสดงความสัมพันธ์ระหว่างจำนวนดิจิตกับค่าความผิดพลาดที่เกิดขึ้น นอกจากนี้จะแสดงการวิเคราะห์อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง โดยคำนึงถึงผลลัพธ์ที่ได้จากขั้นตอนของการกู้ค่าความผิดพลาดที่แตกต่างกัน คือกรณีที่หนึ่ง ตัวถูกดำเนินการจะถูกกู้ค่าความผิดพลาดก่อนที่จะดำเนินการหาร และกรณีที่สองจะดำเนินการหารของตัวถูกดำเนินการก่อนที่จะทำการกู้ค่าความผิดพลาด

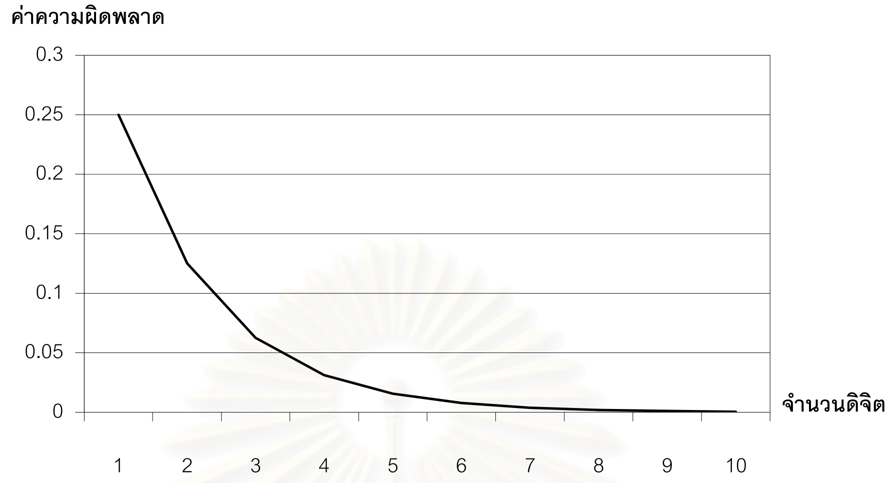
5.1 ความสัมพันธ์ระหว่างจำนวนดิจิตเทียบกับค่าความผิดพลาด

การแสดงความสัมพันธ์ระหว่างจำนวนดิจิตเทียบกับค่าความผิดพลาด เพื่อแสดงให้เห็นถึงการกำหนดจำนวนดิจิตในการคำนวณบนระบบจำนวนค่าต่อเนื่อง กรณีเกิดค่าความผิดพลาดขึ้นในแต่ละดิจิตค่าต่อเนื่อง

จากสมการที่ 4.1 แสดงความสัมพันธ์ระหว่างจำนวนดิจิตเทียบกับค่าความผิดพลาด ดังรูปที่ 5.1 สำหรับ $\beta = 10$ และรูปที่ 5.2 สำหรับ $\beta = 2$ โดยแกน x แทนค่าความผิดพลาดสูงสุด และแกน y แทนจำนวนดิจิต ซึ่งสรุปได้ว่าจำนวนดิจิตยิ่งมากขึ้นค่าความผิดพลาดสูงสุดจะลดลงเข้าใกล้ค่า 0



รูปที่ 5.1 กราฟแสดงความสัมพันธ์ระหว่างจำนวนดิจิตกับค่าความผิดพลาดสำหรับ $\beta = 10$



รูปที่ 5.2 กราฟแสดงความสัมพันธ์ระหว่างจำนวนดิจิตกับค่าความผิดพลาดสำหรับ $\beta = 2$

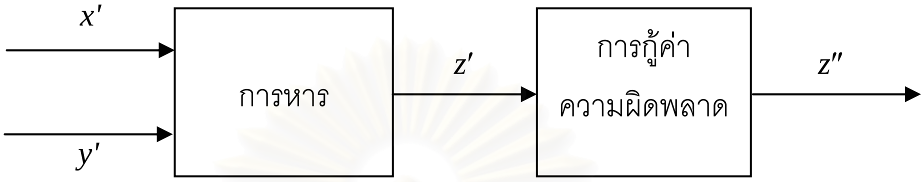
ในส่วนนี้สรุปได้ว่าในกรณีที่ใช้จำนวนดิจิตเท่ากัน ระบบจำนวนที่ใช้ค่าฐานที่สูงกว่าจะมีผลกระทบของค่าความผิดพลาดหลังจากการกู้ค่าความผิดพลาดน้อยกว่าระบบจำนวนบนฐานที่มีค่าน้อยกว่า เนื่องจากค่าความผิดพลาดที่เกิดขึ้นกับจำนวนในระบบนั้นจะแปรผกผันกับค่าของฐานที่ใช้และจำนวนของดิจิต ดังนั้นบนฐานที่มีค่าเท่ากันแต่ใช้จำนวนของดิจิตที่มากกว่าก็จะมีค่าความผิดพลาดที่เกิดขึ้นน้อยกว่าเช่นกัน นอกจากนี้เรื่องของฐานที่ใช้ในระบบจำนวนยังส่งผลกระทบต่อจำนวนดิจิตที่จะต้องใช้ในการแสดงรูปแบบการแทนค่าของจำนวนใดๆ ที่มีค่าระยะสูงสุดแบบพลวัตที่เท่ากัน ซึ่งระบบจำนวนบนฐานที่มีค่ามากกว่าจะใช้จำนวนดิจิตที่น้อยกว่าระบบจำนวนบนค่าฐานที่ต่ำกว่า และยังคงส่งผลกระทบต่อค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมโดยฐานที่มีค่าสูงกว่าจะมีค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมต่ำกว่าระบบที่ใช้ค่าฐานที่สูงกว่า

5.2 การวิเคราะห์อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง

ในส่วนนี้จะทำการวิเคราะห์การหารบนระบบจำนวนค่าต่อเนื่อง ซึ่งแบ่งออกเป็นสองกรณีคือ

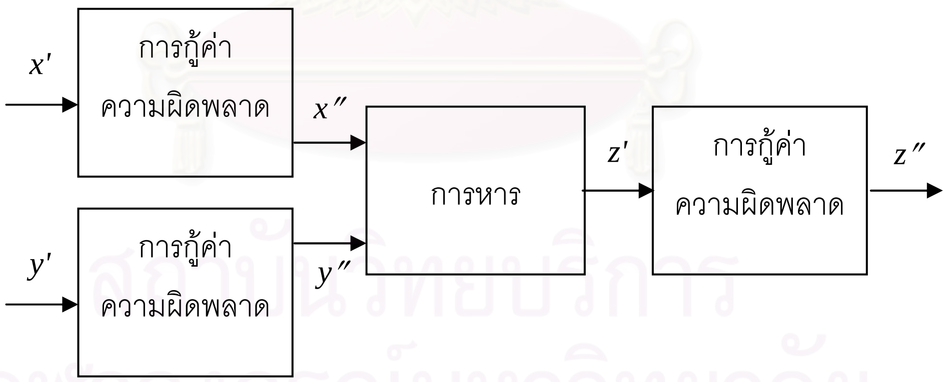
กรณีที่ 1 คือกรณีที่ดำเนินการหารกับตัวถูกดำเนินการ x' และ y' ซึ่งเป็นจำนวนที่เกิดค่าความผิดพลาดขึ้น จะได้ผลลัพธ์เป็นค่า z' ซึ่งเป็นผลลัพธ์ที่เกิดความผิดพลาดขึ้นเช่นกัน หลังจากนั้นจึงทำการกู้ค่าความผิดพลาดที่เกิดขึ้น แสดงได้ดังรูปที่ 5.3 ในสถาปัตยกรรมแบบนี้ ตัว

ถูกดำเนินการทั้งสองอาจมีค่าคลาดเคลื่อน และทำให้ไม่มีคุณสมบัติของการเป็นรูปแบบแทนจำนวนในระบบจำนวนค่าต่อเนื่อง ดังนั้นผลของการหาร และค่าความคลาดเคลื่อนในขั้นตอนการหารอาจมีผลต่อคำตอบที่ได้มาก นั่นคือ ช่วงของค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมของการหารจะลดลงขึ้นกับค่าความผิดพลาดที่เกิดขึ้นกับของตัวดำเนินการก่อนการหาร นอกจากนี้ ในกรณีที่ตัวถูกดำเนินการทั้งสองมีค่าคลาดเคลื่อนที่ต่างกันจะส่งผลถึงการคำนวณด้วย



รูปที่ 5.3 แผนภาพแสดงการหารกรณีการหารที่ดิจิตมีค่าความผิดพลาด

กรณีที่ 2 การหารกรณีที่แต่ละดิจิตผ่านการกู้ค่าความผิดพลาดมาก่อนแล้ว เมื่อได้ผลลัพธ์การหารแล้วเกิดค่าความผิดพลาด หลังจากนั้นทำการกู้ค่าความผิดพลาด แสดงดังแผนภาพที่ 5.4 จะเห็นได้ว่า ในสถาปัตยกรรมแบบนี้ ตัวถูกดำเนินการทั้งสองจำนวน อาจมีความคลาดเคลื่อนที่ไม่เท่ากันได้ ซึ่งจะไม่มีผลต่อค่าความผิดพลาดคลาดเคลื่อนที่ยินยอมของการหาร ทั้งนี้เพราะตัวดำเนินการทั้งสองผ่านกระบวนการกู้ค่าความผิดพลาดมาก่อน ทำให้จำนวนทั้งสองมีคุณสมบัติของการเป็นรูปแบบแทนจำนวนในระบบจำนวนค่าต่อเนื่อง



รูปที่ 5.4 แผนภาพแสดงการหารกรณีการหารที่ดิจิตยังไม่มีค่าความผิดพลาด

ตัวอย่างที่ 5.2 *การหาร* $x = 14.4$ *ด้วย* $y = 4.2$ *ค่าระยะสูงสุดแบบพลวัตเท่ากับ 16 สำหรับฐาน* $\beta = 2$ *กำหนดให้* $L = 3$ *และ* $K = -1$ *โดยค่าความผิดพลาดที่* x *เท่ากับ 5% และ* y *เท่ากับ 2% ซึ่งจะแสดงการหารทั้งสองกรณี*

ตารางที่ 5.1 แสดงการหารตามสถาปัตยกรรมรูปที่ 5.3 และสำหรับรูปที่ 5.4 การหารแสดงได้ดัง ตารางที่ 5.2

ตารางที่ 5.1 แสดงการหารกรณีการหารที่ดิจิตมีค่าความผิดพลาด

n	3	2	1	0	-1
x'_n	1.89	1.68	1.26	0.42	0.84
y'_n	0.5355	1.0815	0.103	0.206	0.412
z_n	0.462378638	0.924757275	1.84951455	1.6990291	1.3980582
z'_n	0.485497569	0.970995139	1.941990278	1.783980555	1.46796111
z''_n	0.46674757	0.933495139	1.866990278	1.555882353	1.46796111

ตารางที่ 5.2 แสดงการหารกรณีการหารที่ดิจิตยังไม่มีค่าความผิดพลาด

n	3	2	1	0	-1
x''_n	1.8025	1.605	1.21	0.42	0.84
y''_n	0.52575	1.0515	0.103	0.206	0.412
z_n	0.428554446	0.857108892	1.714217784	1.428435568	0.856871136
z'_n	0.449982168	0.899964337	1.799928673	1.499857346	0.899714693
z''_n	0.431232169	0.862464337	1.724928674	1.449857347	0.899714693

ผลลัพธ์ของการหาร x = 14.4 ด้วย y = 4.2 มีค่าเท่ากับ 3.42857143 ซึ่งถ้านำไปคำนวณหาค่า z_L จะมีค่าเท่ากับ 0.428571429 จะสังเกตได้ว่า ค่า z''_n ของกรณีที่ 2 มีค่าใกล้เคียงมากกว่ากรณีที่ 1 ซึ่งกรณีที่ 1 จะทำการหารดิจิตทั้งๆ ที่มีค่าความผิดพลาดเกิดขึ้น แตกต่างจากกรณีที่ 2 ที่ทำการกู้ค่าความผิดพลาดก่อน หลังจากนั้นถึงทำการหาร กระบวนการทำงานจะเพิ่มขั้นตอนมากกว่ากรณีที่ 1 แต่ผลลัพธ์ที่ได้ค่อนข้างที่แน่นอนกว่า สามารถคำนวณได้เสมอและได้คำตอบที่สามารถยอมรับได้

ตัวอย่างที่ 5.3 *การหาร x = 4.21 ด้วย y = 4.18 ค่าระยะสูงสุดแบบพลวัตเท่ากับ 16 สำหรับฐาน β = 2 กำหนดให้ L = 3 และ K = -1 โดยค่าความผิดพลาดที่ x เท่ากับ 1% และ y เท่ากับ 5% ซึ่งจะแสดงการหารทั้งสองกรณี*

การหารตามสถาปัตยกรรมรูปที่ 5.3 สามารถแสดงได้ดังตารางที่ 5.3 และการหารตามสถาปัตยกรรมรูปที่ 5.4 แสดงดังตารางที่ 5.4

ตารางที่ 5.3 แสดงการหารกรณีการหารที่ดิจิตมีค่าความผิดพลาด

n	3	2	1	0	-1
x'_n	0.5315125	1.063025	0.10605	0.2121	0.4242
y'_n	0.548625	1.09725	0.0945	0.189	0.378
z_n	0.121101048	0.242202096	0.484404192	0.968808384	1.937616768
z'_n	0.124734079	0.249468159	0.498936318	0.997872636	1.995745271
z''_n	0.12473408	0.249468159	0.498936318	0.997872636	1.995745271

ตารางที่ 5.4 แสดงการหารกรณีการหารที่ดิจิตยังไม่มีค่าความผิดพลาด

n	3	2	1	0	-1
x'_n	0.5265125	1.053025	0.10605	0.2121	0.4242
y'_n	0.523625	1.04725	0.0945	0.189	0.378
z_n	0.125689305	0.25137861	0.50275722	1.00551444	0.01102888
z'_n	0.129459984	0.258919968	0.517839937	1.035679873	0.011359746
z''_n	0.125709985	0.25419969	0.502839937	1.005679873	0.011359746

ผลลัพธ์การหาร x = 4.21 ด้วย y = 4.18 มีค่าเท่ากับ 1.00717703 สำหรับกรณีที่ 1 การเกิดค่าความผิดพลาดที่ดิจิตของตัวหารสามารถทำให้มีค่ามากกว่าตัวตั้งได้ ซึ่งจะทำให้ผลลัพธ์การหารที่ได้ มีค่าเท่ากับ 0.99787264 ทำให้ค่าผิดไปจากความเป็นจริง แต่สำหรับกรณีที่ 2 จะทำ

การกู้ค่าความผิดพลาดกลับมาก่อน จะทำให้ค่าของตัวตั้งและตัวหารใกล้เคียงค่าเดิม ทำให้ผลลัพธ์ของการหารที่ได้ มีค่าเท่ากับ 1.00567984 เป็นค่าที่ยอมรับได้ ทำให้สรุปได้ว่าการหารกรณีที่ 2 สามารถทำการหารได้ครอบคลุมกว่ากรณีที่ 1 และผลลัพธ์ที่ได้มีคุณสมบัติของดิจิตค่าต่อเนื่องอีกด้วย

5.4 บทสรุป

ในบทนี้ได้ทำการวิเคราะห์อัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง โดยแสดงให้เห็นถึงความสัมพันธ์ระหว่างจำนวนดิจิตกับค่าความผิดพลาดบนฐานสิบ และฐานสอง นอกจากนี้ได้แสดงความสัมพันธ์ระหว่างดิจิตกับค่าความผิดพลาดที่เกิดขึ้น ซึ่งแสดงให้เห็นว่าอัลกอริทึมการหารที่ได้นำเสนอ สามารถคำนวณการหารได้ ไม่ว่าจะเป็นกรณีที่ทำการหารก่อนแล้วเกิดค่าความผิดพลาด หรือกรณีเกิดค่าความผิดพลาดก่อนแล้วทำการหาร และคุณสมบัติของแต่ละดิจิตเป็นดิจิตค่าต่อเนื่องอีกด้วย

บทที่ 6

สรุปผลการวิจัย

6.1 สรุปผลการวิจัย

จากงานวิจัยระบบจำนวนค่าต่อเนื่องได้เสนอให้มีการนำเสนอดิจิตค่าต่อเนื่องมาใช้ทำงานบนระบบสัญญาณแอนะล็อก และยังได้เสนอการคำนวณทางเลขคณิตพื้นฐาน โดยเสนออัลกอริทึมการบวก การลบ และการคูณ ซึ่งได้แสดงให้เห็นว่าการคำนวณเลขคณิตพื้นฐานนั้นสามารถคำนวณได้บนดิจิตค่าต่อเนื่อง แต่ระบบสัญญาณแอนะล็อกนั้นมีข้อเสียคือ สัญญาณรบกวน ซึ่งจะเกิดขึ้นระหว่างการส่งสัญญาณ งานวิจัยระบบจำนวนค่าต่อเนื่องยังได้เสนออัลกอริทึมการกู้ค่าความผิดพลาด เมื่อเกิดสัญญาณรบกวนขึ้นในแต่ละดิจิตที่มีค่าต่อเนื่อง โดยค่าความผิดพลาดนั้นจะต้องไม่เกินค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม จึงจะสามารถกู้ค่าความผิดพลาดในแต่ละดิจิตกลับมาได้ใกล้เคียงค่าเดิม

สำหรับงานวิจัยนี้ ได้นำเสนออัลกอริทึมการหารบนระบบจำนวนค่าต่อเนื่อง ซึ่งเป็นตัวดำเนินการทางเลขคณิตพื้นฐานที่ยังไม่ได้นำเสนอในงานวิจัยระบบจำนวนค่าต่อเนื่อง โดยงานวิจัยนี้ได้ออกแบบอัลกอริทึมการหาร และทำการพิสูจน์อัลกอริทึมที่ได้นำเสนอว่าสามารถคำนวณบนดิจิตค่าต่อเนื่อง เมื่อแต่ละดิจิตเกิดความผิดพลาดขึ้น โดยค่าความผิดพลาดจะต้องไม่เกินค่าความผิดพลาดคลาดเคลื่อนที่ยินยอม การคำนวณสามารถทำการกู้ค่าความผิดพลาดได้ค่ากลับมาเป็นค่าที่ยอมรับได้ และยังสามารถคำนวณได้บนฐานใดๆ โดยยังคงคุณสมบัติความเป็นดิจิตค่าต่อเนื่องอีกด้วย

รายการอ้างอิง

[1] A. Avizienis, “Signed-digit number representation for fast parallel arithmetic,” *IRE Transaction on electronic computer,* **1961**, pp. 389-400.

[2] A. Saed, M. Ahmadi, G.A. Jullien and W.C. Miller, “Overlap Resolution: Continuous Valued Digits for Hybrid Architectures,” *Proc. 40th Midwest Symp. Circuits and Systems,* vol. 1, **1997**, pp. 377-380.

[3] A. Saed, M. Ahmadi, G.A. Jullien, “A Number System with Continuous Valued Digits and Modulo Arithmetic,” *IEEE Transactions on Computers*, vol. 51, **2002**, pp. 1294-1305.

[4] A. Saed, M. Ahmadi, G.A. Jullien and W.C. Miller, “Overlap Resolution: Arithmetic with Continuous Valued Digits for Hybrid Architectures,” *Proc. 31st Asilomar Conf. Signals, Systems, and Computers,* vol. 2, **1997**, pp. 1188-1191.

[5] A. Saed, M. Ahmadi and G.A. Jullien, “Arithmetic with Signed Analog Digits,” *Proc. 14th IEEE Symp. Computer Arithmetic (ARITH14),* **1999**, pp. 134-141.

ภาคผนวก

ภาคผนวก ก

ผลงานตีพิมพ์ในงาน ANSCSE 2004

Division algorithm for continuous valued number system

Pisut Rattananat[1], Athasit Surarerks[2]
ELITE (Engineering Laboratory in Theoretical Enumerable System)
Department of Computer Engineering, Faculty of Engineering,
Chulalongkorn University, Phayathai Road
Bangkok 10330 Thailand
pisut@ombudsman.go.th[1], athasit@cp.eng.chula.ac.th[2]

Abstract

Analog circuit concerns about both of power reduction and system noise. Since system noise can make an error, continuous valued number system (CVNS) has been introduced in order to recover the precision from the error. CVNS is built for low noise arithmetic using analog representations and operations. It provides redundancy digits in order to increase the precision, by reducing errors. The correct result is obtained only in the case that the error does not exceed the error tolerance. CVNS also provide some fundamental arithmetic operations which are addition, subtraction and multiplication. In this paper, we prove that division can be performed in the system by presenting a division algorithm for continuous valued number system. Division is performed sequentially started by the least significant digit, digit by digit. We also show that the result is satisfied the properties of the continuous valued number system.

1. Introduction

Computer today needs hi-speed processors which response to speed of work. Most researches on computer arithmetic are aim to improve a digital system. In 1977, Saed et al. [2] have first introduced an *overlap resolution* number system (ORNS) which is an analog number system. The system concerns about both of power reduction and system noise. ORNS is built for low noise arithmetic using analog representations and operations. The principle characteristic of ORNS is to recover the precision from the error which is created from system noise. ORNS provides redundancy digits in order to increase the precision, by reducing errors. The redundancy digits are used to increase the precision from the error by refining the high order digits. The time used for refining increases up to the number of redundancy digits. The concept to parallel analog computations for addition and multiplication were placed in [3]. ORNS is replaced by a *continuous valued* number system (CVNS) in [3, 4]. The arithmetic operations for this number system are processed by using carry free arithmetic structure with circuit level redundancy [4] as same as the signed digit number system [1]. A Modified CVNS is also studied in [6]. Another analog number system called *redundant analog* number system (RANS) is also first introduced in [7].

Fundamental arithmetic operators such as addition, subtraction and multiplication for CVNS are already proposed in [5]. The division is still an important problem. Although division can be performed using a sequence of multiplications and subtractions, but this is not convenient. In this paper, we investigate an efficient division algorithm for CVNS. Two constraints must be focused. First, the time complexity should not be exploded. Second, the result obtained from the algorithm must be preserved the CVNS property. We also propose a linear-time division algorithm with respect to such constraints. Time complexity depends on the number of digits of the operands.

This paper is organized as follows. Section 2 recalls CVNS, analog implementation and error recovery method. A division algorithm is introduced in Section 3. Finally, Section 4 is the conclusion.

2. Preliminaries

In this section, we recall a definition of continuous valued number system and its fundamental arithmetic operators such as addition, subtraction and multiplication. An error recovery method is also presented in this section.

2.1 Continuous valued number system

A continuous valued number system [5] is characterized by a triple $<\beta, L, K>$ where β is the radix of the system, L and K are integers which $K \leq L$. A real number x represented in the system is composed of several *continuous valued digits* (CVDs), x_n, with $K \leq n \leq L$. It is usually used K and L are integers such that $K < 0$ and $0 \leq L$. That is the representation of x is as follows:

$$(x_L, x_{L-1}, \ldots, x_0 \mid x_{-1}, \ldots, x_{K+1}, x_K),$$

where bar (|) is used to distinguish the left part, called *integer part* from the right part, called *fractional part*. The maximum range of x is bounded with X such that $|x| < X$. That is

$$X = \beta^{L+1},$$

where β can be any real number. We select the minimum value of L such that $\beta^{L+1} \geq |X|$. K is used to increase the recovering precision.

The cascaded technique is applied to generate each digit of CVDs in [5]. The digit generation starts from the most significant digit (MSD) to the least significant digit (LSD). The first digit, x_L, can be generated as

$$x_L = \beta \times (x / X). \tag{1}$$

The other CVDs are generated by using the cascade rule as follow:

$$x_n = (x_{n+1} - \lfloor x_{n+1} \rfloor) \times \beta. \tag{2}$$

Because of the analog circuit, all digits are represented as electronic variable, for instance a current or voltage, in range 0 to $\pm Q$ units. Then each CVD of x can be matched to the analog variable q_n by

$$q_n = x_n \times (Q / \beta). \tag{3}$$

Example 1

Given <2, 4, -2> be the CVNS. That is $\beta = 2$, $L = 4$ and $K = -2$. Given a real number $x = 12.456$. Since x is less than $X = 2^5 = 32$, x can be represented in this system. The CVDs of x are illustrated by Table 1. The conversion from x into analog values in the range of 0 μA to 50 μA is computed using Equation (3) and shown in Table 1.

Table 1. CVDs and analog values for x = 12.456

	$\beta = 2$		
n	x_n	$\lfloor x_n \rfloor$	q_n (μA)
4	0.7785	0	19.4625
3	1.557	1	38.925
2	1.114	1	27.85
1	0.228	0	5.7
0	0.456	0	11.4
-1	0.912	0	22.8
-2	1.842	1	46.05

It means that 12.456 is represented as (0.7785 1.557 1.114 0.228 0.456 | 0.912 1.842) and the analog values are (19.4625μA 38.925μA 27.85μA 5.7μA 11.4μA | 22.8μA, 46.05μA).

2.2 Error recovery process

Errors are appeared in the system because of the inaccuracies in the hardware represented the analog digits. Let ε_n be an error applied to digit x_n. For any digit x_n, let x'_n denote the digit included the error, $x'_n = x_n + \varepsilon_n$. The objective of the error recovery process is to recover x'_n as close as the original value x_n with unknown ε_n. The value after recovery is denoted by x''_n. The value of x''_n is computed using the following equations:

$$\lfloor x'_n \rfloor_R = [x'_n - (x''_{n-1} / \beta)]_{\text{Rounded}} \tag{4}$$

and

$$[x'']_R = \begin{cases} \lfloor x'_n \rfloor_R + (x''_{n-1} / \beta) & n > K \\ x'_K & n = K \end{cases} . \tag{5}$$

The error recovery process starts from the LSD or x'_K by assuming this value is correct. This is that the process uses the lower order digit (x_{n-1}) to increase the precision of the higher order digit (x_n). Equation (4) is correct in the case that the following condition is satisfied:

$$|\varepsilon_n - (\varepsilon_{n-1}/\beta)| < 1/2. \tag{6}$$

The inequality (6) provides an error bound on the adjacent digits. This bound is the error that CVNS can tolerate. We assume that all errors of each CVD have substantially equal error probability density function (EPDF) [5]. This is reasonable when the analog operations are implemented with similar circuit over the same analog range.

Example 2

Suppose that the error in the circuit is 4%. Given <2, 4, -2> be the CVNS. The representation of x = 12.456 and x'_n, including errors, are illustrated by Table 2. The error recovering process is also demonstrated by Table 2.

Table 2. The error recovery for x = 12.456 with 4% of the error

	<2, 4, -2>		
n	x_n	x'_n	$[x'']_R$
4	0.7785	0.80964	0.7799325
3	1.557	1.61928	1.559865
2	1.114	1.15856	1.11973
1	0.228	0.23712	0.23946
0	0.456	0.47424	0.47892
-1	0.912	0.94848	0.95784
-2	1.842	1.91568	1.91568

The result of the error recovery process is (0.7799325 1.559865 1.11973 0.23946 0.47892 | 0.95784 1.91568) comparing with the original value (0.7785 1.557 1.114 0.228 0.456 | 0.912 1.842). The error in the result is called a *least significant digit error propagation* [5] along the CVDs.

Remark that the error propagation depends on the value of K. (*i.e.*, the error can be decreased if K is small). In this example it is clearly to see that the complexity for error recovery process is O(n) where n is the number of CVDs. Since the error recovery process refines only one digit, and digit-by-digit, the time needed increases up to the number of redundancy digits.

2.2 CVNS Arithmetic

Some fundamental arithmetic operators have been introduced in [5]. We recall now algorithms for addition, subtraction and multiplication. To simplify the algorithm, number in the system will be represented using analog digits (x_n) instead of using analog values (q_n).

2.2.1 Addition

Let x and y be two numbers represented in $<\beta, L, K>$ number system, where

$x = (x_L\, x_{L-1}\, x_{L-2} \ldots | \ldots x_K)$ and

$y = (y_L\, y_{L-1}\, y_{L-2} \ldots | \ldots y_K)$.

Addition of x and y can be computed as follow:

$$z_n = (x_n + y_n) \bmod \beta, \text{ for all } K \le n \le L, \tag{7}$$

where z is the result, $z = x + y$. It is clear that all digits can be computed in the same time. Then the time used function for addition is constant.

2.2.2 Subtraction

Subtraction can be performed using an addition of a number and one negative number. Let x and y be two numbers represented in $<\beta, L, K>$ number system, where

$x = (x_L\, x_{L-1}\, x_{L-2} \ldots | \ldots x_K)$ and

$y = (y_L\, y_{L-1}\, y_{L-2} \ldots | \ldots y_K)$.

Subtraction of y from x is performed by addition as follows:

$$z_n = \begin{cases} (x_n + y_n) \bmod \ \beta & (x_L + y_L) \ge 0 \\ \\ (x_n + y_n) \bmod^- \beta & (x_L + y_L) < 0 \end{cases}, \tag{8}$$

where a mod^{-} $b = b - (a \bmod b)$.

2.2.3 Multiplication

Multiplication cannot be directly performed independently digit-wise. Multiplication is performed by a summation of the partial product. The theoretical result in [5] shows that the CVD of a product λ.x (λ is an integer) is $(\lambda x)_n = (\lambda . x_n) \bmod \beta$, then the product of x and y can be computed by

$$(y.x)_n = (\sum_{\forall k} \lambda_k . x_{n-k}) \bmod \beta = (\sum_{\forall k} \vec{\lambda}_i . x_n) \bmod \beta ,$$

where $\lambda_i = \lfloor y_i \rfloor$ for all $K \le i \le L$.

3. Division algorithm

In this section, we focus on a division algorithm for CVNS. Division can be performed by summation of division result in each digit. The division of x by y can be computed by the following theorem.

Theorem 1:
Let x and y be two numbers represented in $<\beta, L, K>$ number system, where $x = (x_L\ x_{L-1}\ x_{L-2} \ldots | \ldots x_K)$ and $y = (y_L\ y_{L-1}\ y_{L-2} \ldots | \ldots y_K)$. A number $z = (z_L\ z_{L-1}\ z_{L-2} \ldots | \ldots z_K)$, the quotient of a dividend x with the divisor y, can be computed by the following equation:

$$Z_n = \begin{cases} Q_n & : n = L \\ \left(\left(\left(\sum_{i=L}^{n+1} \left(\frac{T_i}{\beta^n} \right) \right) \right) \bmod \beta + Q_n \right) \bmod \beta & : n < L \end{cases} , \tag{9}$$

where T_n (for all $K \le n \le L$) is defined as

$$T_n = \frac{\lfloor x_n \rfloor * \beta^n}{Y} , \tag{10}$$

and Q_n is the ratio of x_n and y_n:

$$Q_n = \left(\frac{x_n}{y} \right) \bmod \beta . \tag{11}$$

Proof: From Equation (1) and (2), the CVNS digit generation rule can be obtained as follow:

$$x_n = \left(\frac{x}{X} \beta^{L-n+1} \right) \bmod \beta . \tag{12}$$

We can also use Equation (12) to generalize digit generation to find x_n given x_m as follows:

$$x_m = \left(x_n * \beta^{L-n} \right) \bmod \beta . \tag{13}$$

Since $z = x/y$, and Equation (13), it is obtained that $Z_n = \frac{x_n}{y}$ in the case where $n = L$.

Now, substitute x_n in Equation (1), we obtain

$$Z_n = \left(\frac{x_n}{y}\right) * \frac{\beta}{X}. \tag{14}$$

In the case where $n < L$, the following equation can be obtained from Equation (13).

$$Z_n = \left(\left(\frac{x}{y}\right) * \frac{\beta}{X} * \beta^{L-n}\right) \bmod \beta. \tag{15}$$

From the generation rule of CVD, it is true that

$$x_L \beta^L = \lfloor x_L \rfloor \beta^L + \lfloor x_{L-1} \rfloor \beta^{L-1} + \ldots + x_n \beta^n. \tag{16}$$

Multiply (1) by β^n combining with Equation (16), then Equation (17) is obtained.

$$x = \frac{\left(\lfloor x_L \rfloor \beta^L + \lfloor x_{L-1} \rfloor \beta^{L-1} + \ldots + x_n \beta^n\right) * X}{\beta * \beta^L} \tag{17}$$

The following equation is derived from (15) and (17),

$$Z_n = \left(\left(\left(\sum_{i=n+1}^{L} \frac{\frac{\lfloor x_i \rfloor \beta^i}{y}}{\beta^n}\right) \bmod \beta\right) + \left(\frac{x_n}{y}\right) \bmod \beta\right) \bmod \beta. \tag{18}$$

Finally, from the definitions of T_n and Q_n in Equation (10) and (11), and (18), we can conclude in the case $n < L$ that

$$Z_n = \left(\left(\sum_{i=n+1}^{L} \frac{T_i}{\beta^n}\right) \bmod \beta + Q_n\right) \bmod \beta,$$

and the proof is completed. □

In order to understand the algorithm, the following example can be used to demonstrate the algorithm.

Example 3: Consider the division of x = 92.2 by y = 32, using a decimal radix of β = 2 with a range of X= 100. Suppose that an error in the system is fixed by ε = 0.0842 applied to each of the CVNS representations of x and y. Given L = 6 and K = -2, the representation of $z = x/y$ can be computed as shown in Table 3.

Table 3. The division of x = 92.2 by y = 32

n	x_n	y_n	T_n	Q_n	Z_n	Z'_n	Z''_n
6	1.844	0.64	2	0.057625	0.057625	0.14825	0.05795391
5	1.688	1.28	1	0.05275	0.11525	0.19945	0.11590782
4	1.376	0.56	0.5	0.043	0.2305	0.3147	0.23181563
3	0.752	1.12	0	0.0235	0.461	0.5452	0.46363125
2	1.504	0.24	0.125	0.047	0.922	1.0062	0.9272625
1	1.008	0.48	0.0625	0.0315	1.844	1.9282	1.854525
0	0.016	0.96	0	0.0005	1.688	1.7722	1.70905
-1	0.032	1.92	0	0.001	1.376	1.4602	1.4181
-2	0.064	1.84	0	0.002	0.752	0.8362	0.8362

The result of calculation is z = (0.057625 0.11525 0.2305 0.461 0.922 1.844 1.688 | 1.376 0.752). We consider the result of division of 92.2 by 32 which the correct answer is 2.88125.

In the case that an error ε = 0.0842 is applied to the system, the results after a recovery process are (0.05795391 0.11590782 0.23181563 0.46363125 0.9272625 1.854525 1.70905 | 1.4181, 0.8362). It can be seen that the results are satisfied the properties of the continuous valued number system.

4. Conclusions

Formal foundation for addition subtraction and multiplication for continuous valued number system have been presented in [5]. In this work, we propose a division algorithm for continuous valued number system. An algorithm sequentially performs starting from the most significant digit, digit by digit. The result from the algorithm is still satisfied the CVNS properties. We show that the least significant digit error propagation can also be recovered if the error does not exceed the error tolerance.

References

[1] A. Avizienis, “Signed Digit Number representation for Fast Parallel Arithmetic,” IRE Trans. Electronic Computers, vol. 10, pp. 384-400, 1961.

[2] A. Saed, M. Ahmadi, G. A. Jullien and W.C. Miller, “Overlap Resolution: Continuous Valued Digits for Hybrid Architecture,” Proc. 40th Midwest Symp. Circuits and Systems, vol. 1, pp. 377-380, 1997.

[3] A. Saed, M. Ahmadi, G. A. Jullien and W.C. Miller, “Overlap Resolution: Arithmetic with Continuous Valued Digits in Hybrid Architecture,” Proc. 31st Asilomar Conf. Signals, Systems, and Computers, vol. 2, pp. 1188-1191, 1997.

[4] A. Saed, M. Ahmadi and G. A. Jullien, “Arithmetic with Signed Analog digits.” Proc. 14th IEEE Symp. computer Arithmetic (ARITH14), pp. 134-141, 1999.

[5] A. Saed, M. Ahmadi and G. A. Jullien, “A Number System with Continuous Valued Digits and Modulo Arithmetic,” IEEE Transactions on Computers, vol. 51, pp. 1294-1305, 2002.

[6] S. Srivanasont and A. Surarerks, “Modified Continuous Valued Number System,” The 2004 International Conference on Algorithmic Mathematics and Computer Science, 2004.

[7] S. Srivanasont and A. Surarerks, “Redundant Analog Number System…”, IEEE TENCON, 2004.

ประวัติผู้เขียนวิทยานิพนธ์

นายพิสุทธิ์ รัตนนัฏ เกิดเมื่อวันที่ 19 กรกฎาคม พ.ศ. 2518 เรียนจบการศึกษาระดับมัธยมศึกษาตอนปลายจากโรงเรียน ภ.ป.ร.ราชวิทยาลัยฯ อ.สามพราน จ.นครปฐม เข้ารับการศึกษาต่อที่มหาวิทยาลัยหอการค้าไทย ในคณะวิทยาศาสตร์ สาขาวิชาวิทยาการคอมพิวเตอร์ ภาควิชาวิทยาศาสตร์คอมพิวเตอร์ และสำเร็จการศึกษาในระดับปริญญาบัณฑิตในปี พ.ศ. 2540