



รายงานวิจัยฉบับสมบูรณ์

โครงการ การปรับปรุงประสิทธิภาพด้านกำลังงานและด้านการปรับตัวของวงจรกรอง
แบบขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดโดยการปรับอันดับของวงจรกรอง
และค่าช่วงก้ำวของการปรับตัว

โดย เจษฎา ชินรุ่งเรือง

สิงหาคม 2551

สัญญาเลขที่ RSA4580027

รายงานวิจัยฉบับสมบูรณ์

โครงการ การปรับปรุงประสิทธิภาพด้านกำลังงานและด้านการปรับตัวของวงจรกรอง
แบบขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดโดยการปรับอันดับของวงจรกรอง
และค่าช่วงก้ำวของการปรับตัว

เจษฎา ชินรุ่งเรือง
ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย

สนับสนุนโดยสำนักงานกองทุนสนับสนุนการวิจัย

(ความเห็นในรายงานฉบับนี้ เป็นของผู้วิจัย สกว. ไม่จำเป็นต้องเห็นด้วยเสมอไป)

กิตติกรรมประกาศ

โครงการวิจัยนี้ได้รับทุนสนับสนุนจากสำนักงานกองทุนสนับสนุนการวิจัย (สกว) กองทุนรัชดาภิเษกสมโภช จุฬาลงกรณ์มหาวิทยาลัย และทุนวิจัยจากโครงการเสริมสร้างความเชื่อมโยงระหว่างภาควิชาวิศวกรรมไฟฟ้าและภาคเอกชนทางด้านการวิจัยและพัฒนา

ผู้วิจัยขอขอบคุณภาควิชาวิศวกรรมไฟฟ้าที่ได้อนุญาตให้ใช้ห้องปฏิบัติการวิจัยกรรมวิธีสัญญาณดิจิทัลเป็นสถานที่ทำวิจัย โครงการวิจัยนี้สามารถสำเร็จลุล่วงไปได้ด้วยการช่วยเหลือของนายเมธา คงพูน นายสมบุรณ์ กลิ่นจันทร์กลิ่น และนายพลกฤษณ์ ทุนคำ ในการเขียนโปรแกรมภาษา Matlab เพื่อจำลองการทำงานของขั้นตอนวิธีวงจรกรองต่างๆ และภาษา VHDL และ Verilog เพื่อจำลองการทำงานของฮาร์ดแวร์ที่ใช้สร้างจริงวงจรกรองข้างต้น

บทคัดย่อ

รหัสโครงการ : RSA4580027

โครงการ : การปรับปรุงประสิทธิภาพด้านกำลังงานและด้านการปรับตัวของ
วงจรกรองแบบขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดโดยการปรับอันดับ
ของวงจรกรองและค่าช่วงก้าวของการปรับตัว

นักวิจัย : เจษฎา ชินรุ่งเรือง
ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย

E-mail Address : Chedsada.C@Chula.ac.th

ระยะเวลาโครงการ : 3 ปี และขออนุญาตขยายเวลาอีก 3 ปี

เครื่องช่วยฟังเป็นอุปกรณ์สำหรับช่วยเพิ่มประสิทธิภาพทางการได้ยินของผู้ที่มีปัญหาในการรับรู้ข่าวสารทางเสียง ปัญหาหลักอย่างหนึ่งที่เกิดขึ้นกับผู้ใช้เครื่องช่วยฟังคือ การเกิดเสียงรบกวนเนื่องมาจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง วิธีการแก้ปัญหาดังกล่าวที่มีประสิทธิภาพและใช้กันโดยทั่วไปคือ การใช้วงจรกรองปรับตัวสร้างสัญญาณเลียนแบบสัญญาณป้อนกลับเพื่อหักล้างกับสัญญาณป้อนกลับที่เกิดขึ้นจริง การปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวดังกล่าว โดยทั่วไปจะอาศัยขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมและกำหนดให้อันดับของวงจรกรองมีค่าคงที่ เนื่องจากขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบที่ไม่มีการปรับอันดับต้องมีการสูญเสียกำลังที่ค่อนข้างสูง ทำให้ไม่เหมาะที่จะใช้ในเครื่องช่วยฟังซึ่งใช้พลังงานจากแบตเตอรี่ เพื่อที่จะลดปัญหาเรื่องการสูญเสียกำลังงานสูงดังกล่าว โครงการวิจัยนี้ศึกษาทั้งระเบียบวิธีการปรับอันดับวงจรและการปรับค่าช่วงก้าว พร้อมทั้งศึกษาโครงสร้างวงจรกรองแบบที่ง่ายและเหมาะสมกับการสร้างจริงด้วยฮาร์ดแวร์

นอกจากงานวิจัยหลักข้างต้น โครงการวิจัยนี้ยังได้เพิ่มส่วนที่สองอันเป็นการประยุกต์ใช้หลักการกำลังสองต่ำสุดในการลดทอนสัญญาณรบกวน speckle ในภาพถ่าย synthetic aperture radar (SAR) และภาพอัลตราซาวนด์ทางด้านการแพทย์ โดยหลักการดังกล่าวนี้เป็นหลักการเดียวกับที่ใช้ในวงจรกรองแบบค่าเฉลี่ยกำลังสองต่ำสุด (LMS) จากการศึกษาวิธีออฟไลน์เพื่อปรับอันดับของวงจรกรอง LMS ผู้วิจัยพบว่าเราสามารถใช่วิธีการคำนวณค่าสถิติเฉลี่ยทางเวลาของค่าสัมประสิทธิ์ของวงจร LMS ดังกล่าวมาประยุกต์ใช้ในการประมวลสัญญาณภาพ SAR และภาพอัลตราซาวนด์ หลักการเฉลี่ยข้างต้นได้ถูกนำมาปรับปรุงวงจรกรองซาวิตสกี-โกเลย์สองมิติ (2-dimensional Savitzky-Golay filters) อันเป็นวงจรกรองแบบกำลังสองต่ำสุดประเภทหนึ่งซึ่งประมาณกลุ่มข้อมูลด้วยฟังก์ชันพหุนาม

คำหลัก : การลดทอนการป้อนกลับทางเสียง วงจรกรองปรับตัวแบบ LMS การปรับอันดับวงจรกรอง ค่าช่วงก้าว ภาพถ่าย SAR ภาพอัลตราซาวนด์

Abstract

Project Code : RSA4580027

Project Title : Improving Power and Adaptation Efficiency of the Least-Means-Square Adaptive Filtering Algorithm by Adjusting the Filter Order and the Adaptation Step-Size

Investigator : Chedsada Chinrungrueng
Department of Electrical Engineering, Chulalongkorn University

E-mail Address : Chedsada.C@Chula.ac.th

Project Period : 3 years with a 3-year extension

An hearing aid is an equipment used for amplifying the audio signal in order to enhance the hearing efficiency of people with hearing impairment. One major problem that hearing aid users usually encounter is screeching sound, which results from acoustic feedback in hearing aid. One effective and widely used solution for solving this problem is to employ an adaptive filter to produce signal for canceling out the acoustic feedback signal. Usually, the traditional Least-Mean-Square (LMS) algorithm is employed to adapt the coefficients of the adaptive filter of fixed order. As such traditional LMS algorithm consumes large amount of power, making it uneconomical to employ in hearing aids which are driven by battery. To reduce such power consumption problem, this research studies methods for adapting the filter order and the step-size. The research also investigates architectures that are simple enough for hardware implementation.

In addition to the main research above, this research also adds the supplementary part, which is the application of the least squares principle in reducing the speckle noise in synthetic aperture radar (SAR) and medical ultrasound images. The least squares principle is the same one as used by the LMS algorithm. In investigating optimization techniques for adapting the LMS filter order, we find that the averaging the LMS filter coefficients over time can be applied to process SAR and ultrasound images. This time-averaging principle is used to improve the 2-dimensional Savitzky-Golay filter--the least-sqaure-type filter that aims to fit the data with a given polynomial function.

Keywords : Acoustic feedback cancellation, LMS adaptive filter, adaptive tap, step-size, SAR image, ultrasound images.

สารบัญ

	หน้า
ปกใน.....	I
กิตติกรรมประกาศ.....	II
บทคัดย่อภาษาไทย.....	III
บทคัดย่อภาษาอังกฤษ.....	IV
สารบัญ.....	V
 บทที่	
1 บทนำ.....	1
2 การปรับอันดับของวงจรกรองแบบพลวัต.....	4
3 วงจรกรองอันดับคงที่.....	19
4 วงจรกรองแบบปรับอันดับ.....	37
5 การวัดประสิทธิภาพด้านพลังงานของวงจร.....	53
6 ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign พร้อมการปรับช่วงก้าวแบบพลวัต.....	66
7 การจำลองการทำงานของวงจรกรอง LMS แบบ Sign-Sign บนคอมพิวเตอร์.....	72
8 การสร้างจริงวงจรกรองปรับตัวแบบ Sign-Sign LMS.....	92
9 การทดสอบและสรุปผลการวิจัยของวงจรกรองปรับตัวแบบ Sign-Sign LMS.....	123
10 การประยุกต์ใช้หลักวงจรกรองกำลังสองต่ำเพื่อใช้ลดสัญญาณรบกวนแบบ speckle	131
11 การปรับปรุงวงจรกรองซาวีตกี-โกเลย์สำหรับลดทอนสัญญาณรบกวน Speckle บนภาพอัลตราซาวนด์.....	146
12 การขยายผลวงจรกรองซาวีตกี-โกเลย์แบบแอนไอโซทรอปิก.....	187
 รายการอ้างอิง.....	199
ผลลัพธ์จากโครงการวิจัยที่ได้รับทุนจาก สกว.....	202
ภาคผนวก.....	203

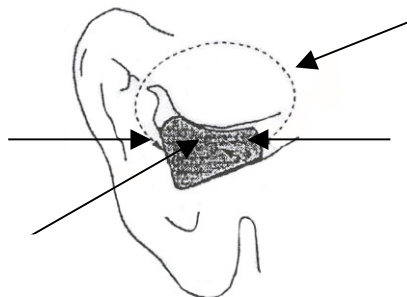
บทที่ 1

บทนำ

1.1 ความเป็นมาของงานวิจัย

การป้อนกลับทางเสียงในเครื่องช่วยฟัง เกิดจากการที่สัญญาณเสียงย้อนกลับจากลำโพงของเครื่องช่วยฟังไปยังไมโครโฟนของเครื่องช่วยฟังอีกครั้ง โดยสัญญาณเสียงที่ย้อนกลับจากลำโพงไปยังไมโครโฟนจะเรียกว่า สัญญาณป้อนกลับ และเส้นทางที่สัญญาณเสียงป้อนกลับเดินทางย้อนกลับจากลำโพงไปยังไมโครโฟน ซึ่งมีด้วยกันหลายเส้นทางจะเรียกรวมกันว่า เส้นทางป้อนกลับ (Feedback Path) เนื่องจากเส้นทางป้อนกลับทำให้เกิดการลดทอนสัญญาณ และเครื่องช่วยฟังโดยปกติจะทำหน้าที่ขยายสัญญาณ ดังนั้น ถ้าสัญญาณที่วนผ่านเส้นทางป้อนกลับถูกลดทอนด้วยอัตราที่น้อยกว่าอัตราขยายของเครื่องช่วยฟัง จะทำให้สัญญาณป้อนกลับถูกขยายขึ้นทุก รอบที่มีการป้อนกลับจนทำให้สัญญาณป้อนกลับ ใหญ่ขึ้นจนกลายเป็นเสียงหอนในที่สุด ดังแสดงในรูปที่ 1.1

การแก้ปัญหาการเกิดเสียงหอนเนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง มีแนวทางในการแก้ปัญหาหลัก ๆ อยู่ 2 แนวทาง แนวทางแรกเป็นการพยายามป้องกันไม่ให้สัญญาณป้อนกลับทางเสียงกลายเป็นเสียงหอน ส่วนแนวทางที่สองเป็นการพยายามลดสัญญาณป้อนกลับ เนื่องจากแนวทางแรกเป็นการป้องกันไม่ให้สัญญาณป้อนกลับถูกขยายจนเกิดเสียงหอนแต่ก็ยังคงมีการป้อนกลับอยู่ และวิธีการนี้จะทำให้ผลตอบสนองเชิงความถี่ของเครื่องช่วยฟังเปลี่ยนไปทำให้สัญญาณเสียงที่ออกจากเครื่องช่วยฟังเพี้ยนไปจากเดิม วิธีการนี้จึงไม่เป็นที่นิยม



รูปที่ 1.1 การป้อนกลับทางเสียงในเครื่องช่วยฟัง

ในที่นี้เราจึงสนใจเฉพาะวิธีการแก้ปัญหาแบบที่สอง โดยการใช้วงจรกรองปรับตัวสร้างสัญญาณขึ้นมาหักล้างกับสัญญาณป้อนกลับ [1] วิธีการนี้แบ่งออกเป็น 2 แบบคือ แบบที่วงจรกรองปรับค่าพารามิเตอร์ตลอดเวลาเรียกว่าแบบปรับตัวอย่างต่อเนื่อง ส่วนแบบที่สองคือแบบที่วงจรกรองปรับตัวอย่างไม่ต่อเนื่อง โดยวงจรกรองจะปรับค่าพารามิเตอร์เฉพาะช่วงที่เสียงเงียบ (สัญญาณเสียงขาเข้ามีกำลังต่ำ) หรืออาจปรับตัวในช่วงที่จะเกิดเสียงหอน รวมทั้งอาจมีการปรับตัวเป็นรายคาบ

การปรับตัวทั้งสองแบบมีข้อดีและข้อด้อยที่ต่างกัน การปรับตัวแบบต่อเนื่องมีข้อดีคือสามารถติดตามการเปลี่ยนแปลงของสัญญาณป้อนกลับได้ตลอดเวลาแต่มีข้อเสียคืออาจมีสัญญาณรบกวนที่ใช้สำหรับเป็นสัญญาณกระตุ้นสำหรับปรับวงจรกรองอาจรบกวนผู้ใช้ได้ รวมถึงปัญหาทางด้านเสถียรภาพของวงจร ขณะที่การปรับตัวแบบไม่ต่อเนื่องมีข้อดีคือ สัญญาณรบกวนที่ใช้มีกำลังต่ำจึงไม่รบกวนผู้ใช้งานหูฟัง แต่มีข้อเสียคือไม่สามารถติดตามการเปลี่ยนแปลงของเส้นทางป้อนกลับได้ตลอดเวลา ซึ่งในที่นี้จะใช้วงจรกรองที่ปรับตัวแบบไม่ต่อเนื่อง เนื่องจากให้ความมีเสถียรภาพของวงจรที่ดี

ตามปกติอันดับของวงจรกรองจะครอบคลุมความยาวของผลตอบสนองเชิงเวลาของเส้นทางป้อนกลับ แต่เนื่องจากเส้นทางป้อนกลับทางเสียงนี้มีผลตอบสนองเชิงเวลาที่เปลี่ยนแปลงอยู่ตลอดเวลาซึ่งทำให้ความยาวของผลตอบสนองเชิงเวลาลดลง ดังนั้นการใช้วงจรกรองที่มีอันดับคงที่อาจทำให้วงจรกรองสูญเสียพลังงานไปในส่วนที่ไม่จำเป็นได้ ซึ่งการสูญเสียพลังงานในส่วนที่ไม่จำเป็นนี้อาจก่อให้เกิดความร้อนที่สูงมากเกินไปและอาจนำมาซึ่งความเสียหายแก่วงจรได้ นอกจากนี้ในเครื่องช่วยฟังซึ่งเป็นอุปกรณ์พกพาที่ใช้พลังงานจากแบตเตอรี่นั้นก็มีพลังงานที่จำกัด จึงต้องมีการพยายามลดปัญหาเรื่องการสูญเสียพลังงานลง ซึ่งสามารถแก้ปัญหาได้ในหลายระดับ [2] ยกตัวอย่างเช่น การลดการสูญเสียพลังงานในระดับเกตโดยการพยายามลดจำนวนสวิตช์ของลอจิกในแต่ละโหนดของวงจร การลดการสูญเสียพลังงานในระดับสถาปัตยกรรมของวงจรโดยการเลือกใช้โครงสร้างของวงจรที่เหมาะสมในการประยุกต์ใช้กับงานต่าง ๆ และการลดการสูญเสียพลังงานของวงจรในระดับขั้นตอนวิธีที่ใช้ในวงจรกรองโดยใช้ขั้นตอนวิธีที่มีประสิทธิภาพเพื่อลดจำนวนครั้งของการคำนวณลง

ในโครงการวิจัยนี้ ส่วนแรกจะสนใจการใช้ขั้นตอนวิธีในการปรับอันดับของวงจรกรอง รวมถึงการใช้โครงสร้างของวงจรกรองที่เหมาะสม เพื่อให้สามารถลดการสูญเสียพลังงานของวงจรในส่วนที่ไม่จำเป็นลงได้ รวมทั้งวัดประสิทธิภาพการใช้พลังงานของวงจรกรองที่ออกแบบขึ้นเปรียบเทียบกับวงจรกรอง LMS แบบที่ไม่มีการปรับอันดับ

สำหรับส่วนของโครงการวิจัยนี้จะเป็นการพัฒนาวิธีการปรับค่าช่วงก้าว (step-size) สำหรับวงจรกรอง LMS พร้อมทั้ง ทดสอบเปรียบเทียบประสิทธิภาพการปรับตัวและความซับซ้อนของการคำนวณซึ่งจะสะท้อนถึงประสิทธิภาพการใช้พลังงาน โครงการวิจัยนี้ได้พัฒนางจรกรอง LMS ที่มีค่าช่วงก้าวปรับขนาดตามระยะห่างจากจุดทำงานออฟติ멈 ทั้งนี้ เพื่อว่า วงจรกรองจะสามารถปรับตัวได้เร็วเมื่อค่าสัมประสิทธิ์ห่างจากจุดออฟติ멈 เนื่องจากช่วงก้าวมีขนาดใหญ่ และเมื่อค่าสัมประสิทธิ์เข้าใกล้ค่าออฟติ멈 ช่วงก้าวจะถูกปรับให้มีขนาดเล็ก ทั้งนี้ เพื่อให้ค่าสัมประสิทธิ์สามารถปรับตัวเข้าใกล้ค่าออฟติ멈 ได้ใกล้ยิ่งขึ้น กลไกดังกล่าวสามารถช่วยให้วงจรกรองปรับตัวได้เร็วขึ้นและแม่นยำขึ้น นอกจากนี้ ยังอนุญาตให้เราหยุดปรับค่าสัมประสิทธิ์เมื่อค่าช่วงก้าวมีขนาดเล็กมากจนไม่คุ้มที่จะไปปรับค่าสัมประสิทธิ์ ซึ่งเป็นอีกทางหนึ่งในการช่วยประหยัดพลังงานสูญเสีย

นอกจากนี้ ส่วนหลักอันเป็นเรื่องเกี่ยวกับการปรับปรุงประสิทธิภาพด้านการปรับตัวและการสูญเสียพลังงานของวงจรกรองปรับตัวแบบ LMS สำหรับปัญหาการลดทอนสัญญาณป้อนกลับในเครื่องช่วยฟัง โดยการปรับอันดับของวงจรกรองและปรับช่วงก้าวแล้ว โครงการวิจัยนี้ยังได้เพิ่มส่วนที่สองอันเป็นการประยุกต์ใช้หลักการกำลังสองต่ำสุด (Least Squares) ในการลดทอนสัญญาณรบกวน speckle ในภาพถ่าย synthetic aperture radar (SAR) และภาพอัลตราซาวนด์ทางด้านการแพทย์ หลักการที่ใช้ในวงจรกรองกำลังสองต่ำสุด (Least Squares Filter) จะพยายามทำให้กำลังสองของค่าความผิดพลาดระหว่างสัญญาณออกของวงจรกรองกับค่าสัญญาณอ้างอิงเข้ามีค่าต่ำ สุด โดยหลักการดังกล่าวนี้จะเป็นหลักการเดียวกับที่ใช้ในวงจรกรองแบบค่าเฉลี่ยกำลังสองต่ำสุด (LMS) จากการศึกษาเกี่ยวกับวิธีออฟติไมซ์เพื่อหาเฉลี่ยของค่าสัมประสิทธิ์ของวงจรกรอง LMS เมื่อเทียบกับเวลา สำหรับใช้ในการลดอันดับของวงจรกรองแบบ LMS ผู้วิจัยพบว่าเราสามารถใช่วิธีการคำนวณค่าสถิติเฉลี่ยทางเวลาของค่าสัมประสิทธิ์ของวงจรดังกล่าวมาประยุกต์ใช้ในการประมวลสัญญาณภาพแบบ SAR และภาพอัลตราซาวนด์ โดยได้นำหลักการข้างต้นมาปรับปรุงวงจรกรองซาวิตซกี-โกเลย์สองมิติ (2-dimensional Savitzky-Golay filters) อันเป็นวงจรกรองแบบกำลังสองต่ำสุดประเภทหนึ่งซึ่งประมาณกลุ่มข้อมูลด้วยฟังก์ชันพหุนาม

บทที่ 2

การปรับอันดับของวงจรกรองแบบพลวัต

เนื้อหาในบทนี้อธิบายถึงวงจรกรองปรับตัว FIR ในหัวข้อ 2.1 และระเบียบวิธีการปรับตัวของวงจรกรองตามระเบียบวิธีกำลังสองเฉลี่ยน้อยที่สุดในหัวข้อ 2.2 จากนั้นจะกล่าวถึงระเบียบวิธีที่ใช้ในการปรับอันดับของวงจรกรองปรับตัวแบบพลวัตเพื่อนำมาประยุกต์สร้างเป็นวงจรกรองปรับตัวแบบปรับอันดับในหัวข้อที่ 2.3

2.1 วงจรกรองปรับตัว FIR แบบจำลองของปัญหาการป้อนกลับทางเสียงในเครื่องช่วยฟัง

วงจรกรองปรับตัวถูกนำมาใช้กันอย่างแพร่หลาย เห็นได้จากการนำมาประยุกต์ใช้ในงานกรรมวิธีสัญญาณดิจิทัลด้านโทรคมนาคมและคอมพิวเตอร์จำนวนมาก เช่น การปรับแต่งช่องสัญญาณสื่อสาร (Channel Equalization) การกำจัดสัญญาณเสียงก้องในช่องสัญญาณสื่อสาร (Echo Cancellation) และการเข้ารหัสสัญญาณเสียงพูด (Speech Encoding)

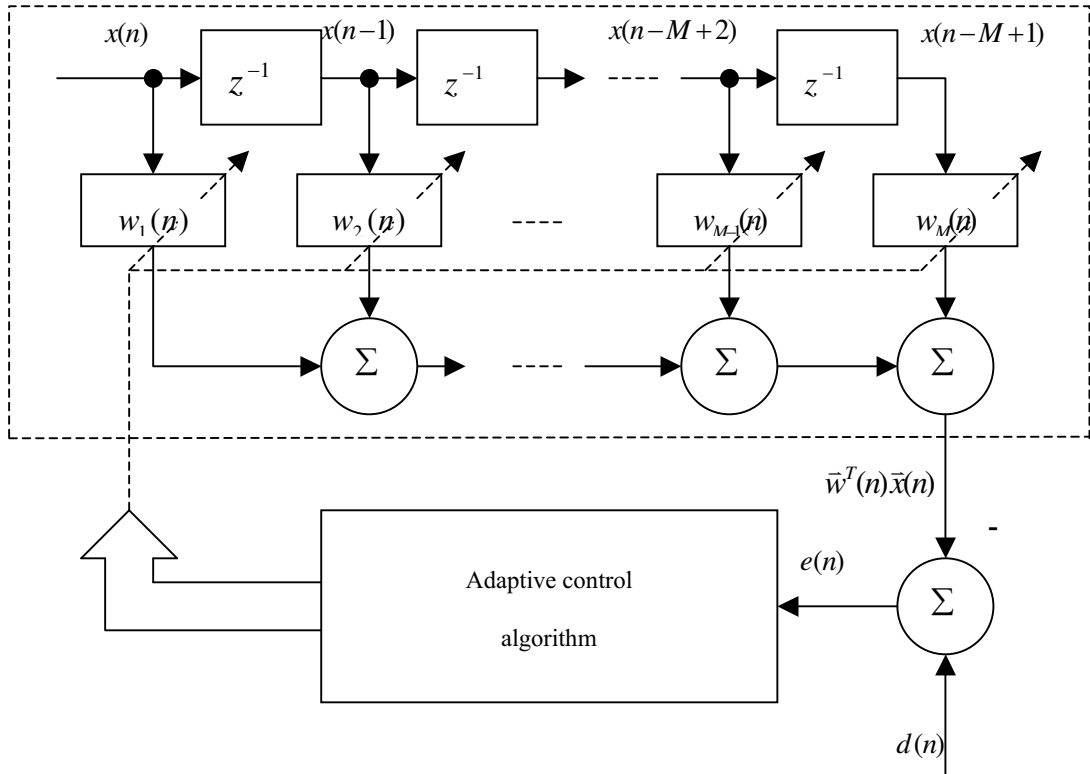
วงจรกรองปรับตัว (Adaptive Filter) คือ วงจรกรองที่สามารถปรับค่าสัมประสิทธิ์ (Tap Weight) ของวงจรกรองได้ กระบวนการทำงานจะเหมือนกับวงจรกรองทั่ว ๆ ไป ดังแสดงในรูปที่ 2.1 โดยทำการคูณค่าสัญญาณขาเข้า (Tap Input Vector) ที่เวลา n กับค่าสัมประสิทธิ์ของวงจรกรอง (Tap Weight Vector) ที่เวลา n เพื่อที่จะหาผลลัพธ์ที่ได้จากการประมาณ (Estimate) ที่เวลา n จากนั้นนำค่าผลลัพธ์ที่ได้ไปทำการลบออกจากค่าผลลัพธ์ที่ต้องการ (Desired Response) ที่เวลา n จะได้ค่าความผิดพลาด (Error) ที่เวลา n ซึ่งจะนำค่าความผิดพลาดดังกล่าวไปใช้ในขั้นตอนวิธีของการปรับค่าสัมประสิทธิ์ของวงจรกรองต่อไป สำหรับในส่วนของการปรับค่าสัมประสิทธิ์ของวงจรกรองนั้น จะนำเอาค่าความผิดพลาดที่เวลา n ที่คำนวณได้มาใช้ในการคำนวณ

กำหนดให้ $\vec{x}(n)$ คือ สัญญาณขาเข้า (Tap Input Vector) ที่เวลา n

$\vec{w}(n)$ คือ ค่าสัมประสิทธิ์ของวงจรกรอง (Tap Weight Vector) ที่เวลา n

$d(n)$ คือ ผลลัพธ์ที่ต้องการ (Desired Response) ที่เวลา n

$e(n)$ คือ ค่าความผิดพลาด (Error) ที่เวลา n



รูปที่ 2.1 โครงสร้างของวงจรกรองปรับตัวแบบทรานส์เวอร์

โดยที่

$$\bar{x}(n) = (x(n), \dots, x(n-M+1))^T \quad (2-1)$$

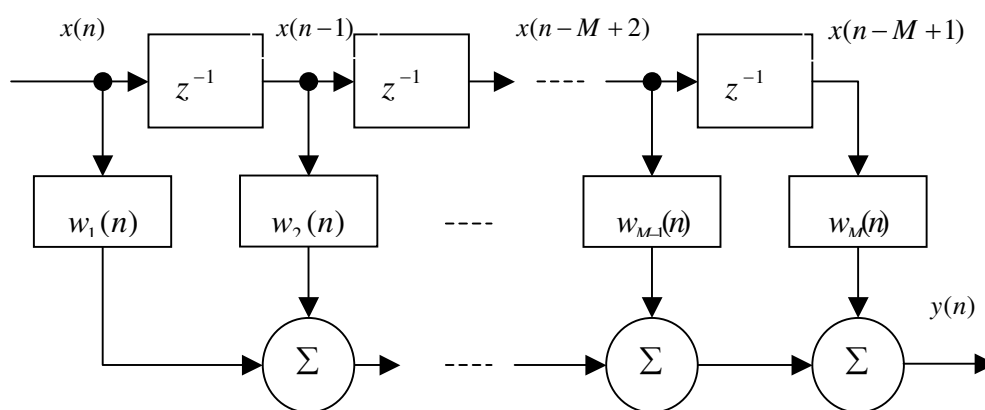
$$\bar{w}(n) = (w_1(n), \dots, w_M(n))^T \quad (2-2)$$

$$y(n) = \bar{w}^T(n) \bar{x}(n) \quad (2-3)$$

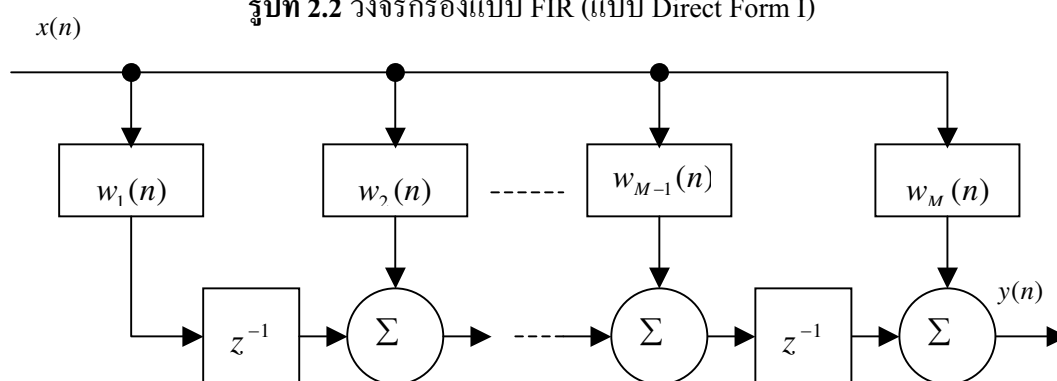
$$e(n) = d(n) - y(n) \quad (2-4)$$

โดยที่ M คือ อันดับของวงจรกรองปรับตัวและ $(.)^T$ คือ ตัวสลับเปลี่ยน (Transpose) ของเวกเตอร์ หรือของเมตริกซ์ สมการที่ (2-1) แสดงให้เห็นถึงสัญญาณขาเข้าที่เวลา n สมการที่ (2-2) แสดงให้เห็นถึงสัมประสิทธิ์ของวงจรกรองปรับตัวที่เวลา n สมการที่ (2-3) แสดงผลลัพธ์ที่ได้จากการคำนวณของวงจรกรอง ที่เวลา n และสมการที่ (2-4) แสดงให้เห็นถึงการคำนวณของค่าความผิดพลาดที่เวลา n ที่จะนำไปใช้ในกระบวนการปรับค่าสัมประสิทธิ์ (Adaptive Control Algorithm) ของวงจรกรองปรับตัวต่อไป

วงจรกรองที่นำมาใช้เป็นส่วนประกอบของวงจรกรองปรับตัวคือ วงจรกรองแบบ FIR (Finite Impulse Response) โดยทั่วไปวิธีการที่นำมาใช้ในการสร้างจริง ของวงจรกรองแบบ FIR มี 2 รูปแบบคือ แบบ Direct Form (DF) และแบบ Transposed Direct Form (TDF) ซึ่งทั้งสองรูปแบบมีข้อดีและข้อเสียที่แตกต่างกัน แบบ Direct Form จะใช้เวลาในการคำนวณสูงทำให้มีความเร็วในการคำนวณต่ำ เนื่องจากต้องเสียเวลาในการหาผลรวมของผลบวกในแต่ละ Tap ของวงจรกรอง แต่จะใช้พื้นที่ในการสร้างจริงที่น้อยทำให้มีการสูญเสียพลังงานน้อย ส่วนแบบ Transposed Direct Form จะใช้เวลาในการคำนวณต่ำทำให้ความเร็วในการคำนวณสูงกว่า แต่ต้องการพื้นที่ในการสร้างจริงที่มากกว่า เนื่องจากขนาดของ Register ที่นำมาใช้ในแต่ละ Tap จะต้องสามารถรองรับขนาดของผลบวกของวงจรกรองได้ ซึ่งจะทำให้ต้องสูญเสียพลังงานมากกว่าด้วย



รูปที่ 2.2 วงจรกรองแบบ FIR (แบบ Direct Form I)

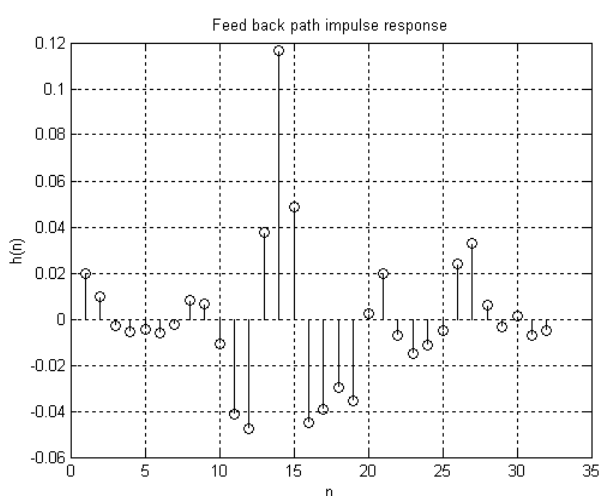


รูปที่ 2.3 วงจรกรองแบบ FIR (แบบ Transposed Direct Form)

รูปที่ 2.2 และ 2.3 แสดงรูปโครงสร้างของวงจรกรองแบบ FIR ในรูปแบบของ Direct Form และ Transposed Direct Form ตามลำดับ ซึ่งในที่วงจรกรองในที่นี้จะนำไปใช้ในเครื่องช่วยฟังซึ่งมีพื้นที่ของฮาร์ดแวร์จำกัด รวมทั้งความเร็วที่ใช้ในการประมวลผลเกี่ยวกับสัญญาณเสียงที่จะทำการวิเคราะห์เกี่ยวกับปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟังนั้นไม่สูงมาก ดังนั้น

การสร้างจริงวงจรกรองในโครงการวิจัยนี้จะใช้วงจรกรองแบบ FIR ในรูปแบบ Direct Form มาใช้ในการสร้างจริง

แบบจำลองของปัญหาแบบจำลองของปัญหาการป้อนกลับทางเสียงในเครื่องช่วยฟังจะใช้ผลตอบสนองต่ออิมพัลส์ (Impulse Response) ของเส้นทางป้อนกลับ [3] ที่หาได้จากผลตอบสนองทางความถี่ (Frequency Response) ของเส้นทางป้อนกลับที่วัดจริงจากเครื่องช่วยฟัง [4] โดยทำการสุ่มตัวอย่างจากผลตอบสนองทางความถี่ที่วัดได้ จากนั้นใช้การแปลงกลับดิสครีตไทม์ฟูริเยร์ (Inverse Discrete-Time Fourier Transform) ซึ่งจะได้ผลตอบสนองต่ออิมพัลส์ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ผลตอบสนองต่ออิมพัลส์ของเส้นทางป้อนกลับ

2.2 ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุด

วงจรกรองปรับตัวโดยทั่วไปจะนิยมใช้ระเบียบวิธีกำลังสองเฉลี่ยน้อยสุด (Least Mean Square : LMS) มาช่วยในการปรับค่าสัมประสิทธิ์ของวงจรกรอง เนื่องจากระเบียบวิธี LMS จะยังคงให้เสถียรภาพที่ดีเมื่อนำไปสร้างจริงเป็นฮาร์ดแวร์ รวมถึงความซับซ้อนในการสร้างจริงที่ไม่สูงทำให้ประหยัดพื้นที่ของฮาร์ดแวร์

ขั้นตอนการปรับปรุงค่าสัมประสิทธิ์ของวงจรกรองตามวิธีกำลังสองเฉลี่ยน้อยสุดจะอยู่ในรูปแบบของสมการที่ (2-5) และสมการที่ (2-6) ค่าช่วงก้าว μ ในสมการที่ (2-6) เป็นค่าพารามิเตอร์ที่สำคัญอีกตัวหนึ่งใช้สำหรับกำหนดอัตราเร็วในการปรับตัวของขั้นตอนวิธี โดยที่ค่าช่วงก้าวของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดจะถูกกำหนดให้เป็นค่าคงที่ ในกรณีที่ค่าช่วงก้าวมีขนาดใหญ่ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดจะมีอัตราการปรับตัวสูงแต่ค่าความถูกต้องในการปรับตัวที่สภาวะคง

ตัวจะต่ำ ในทางตรงกันข้ามถ้าค่าช่วงก้ำวมีขนาดเล็ก ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดจะมีอัตรา
การปรับตัวต่ำแต่ค่าความถูกต้องในการปรับตัวที่สภาวะคงตัวจะสูง

$$e(n) = d(n) - \bar{w}^T(n)\bar{x}(n) \quad (2-5)$$

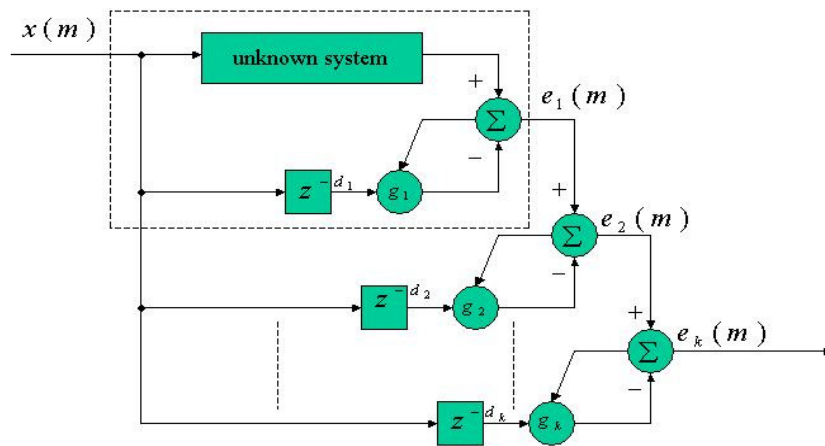
$$\bar{w}(n+1) = \bar{w}(n) + \mu \bar{x}(n)e(n) \quad (2-6)$$

2.3 ขั้นตอนวิธีการปรับอันดับของวงจรกรองแบบพลวัต

ถึงแม้วงจรกรองปรับตัวตามระเบียบวิธี LMS จะมีเสถียรภาพที่ดีในการสร้างจริงเป็น
ฮาร์ดแวร์ แต่ในการใช้งานกับช่องสัญญาณป้อนกลับทางเสียงในเครื่องช่วยฟังจริง ๆ นั้นช่อง
สัญญาณป้อนกลับทางเสียงมีการเปลี่ยนแปลงของผลตอบสนองเชิงเวลาตลอดเวลา จึงมีอันดับของ
วงจรกรองบางส่วนที่ทำงานโดยไม่จำเป็น เป็นเหตุให้ส่งผลต่อการรู้เข้าของวงจรกรอง และสูญเสีย
พลังงานเพื่อคำนวณในส่วนที่ไม่จำเป็นไป ดังที่ได้กล่าวในบทนำแล้วว่าการสูญเสียพลังงานในส่วน
ที่ไม่จำเป็นนี้อาจก่อให้เกิดความร้อนที่สูงมากเกินไปและอาจนำมาซึ่งความเสียหายแก่วงจรได้ นอก
จากนี้ในเครื่องช่วยฟังซึ่งเป็นอุปกรณ์พกพาที่ใช้พลังงานจากแบตเตอรี่นั้นก็มีพลังงานที่จำกัด ดังนั้น
จึงต้องมีการพยายามลดปัญหาเรื่องการสูญเสียพลังงาน โดยในที่นี้จะพิจารณาการแก้ปัญหาดังกล่าว
โดย

2.3.1 ขั้นตอนวิธีการปรับอันดับของวงจรกรองแบบ Adaptive Delay Filter

การปรับตัวของวงจรกรองตามระเบียบวิธี LMS ซึ่งมีหลักการคือพยายามทำให้ ค่ากำลัง
สองเฉลี่ยของผลต่างระหว่างสัญญาณขาออกที่ออกจากวงจรกรองกับสัญญาณขาออกที่ออกจากช่อง
สัญญาณนั้นต้องมีตัวแปรที่จะนำมาพิจารณาหลายตัว ซึ่ง Etter [5] เสนอความคิดว่าในกรณีที่ช่อง
สัญญาณมีผลตอบสนองเชิงเวลาส่วนใหญ่ที่เป็นศูนย์เราสามารถลดความซับซ้อนในการคำนวณลง
ได้โดยใช้วงจรกรองที่มีโครงสร้างแบบปรับอันดับได้ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 วงจรกรองปรับตัวแบบปรับอันดับตามระเบียบวิธี Adaptive Delay Filter

โดยสมมติให้

$x(m)$ แทนสัญญาณขาเข้าของวงจรกรองและช่องสัญญาณ ณ เวลา m

$e_i(m)$ แทนสัญญาณความผิดพลาดระหว่างสัญญาณขาออกที่ออกจากวงจรกรองกับสัญญาณขาออกที่ออกจากช่องสัญญาณ ณ ตำแหน่งสัมประสิทธิ์ที่ไม่เป็นศูนย์ตัวที่ i ณ เวลา m

g_i แทนอัตราขยายของวงจรกรอง ณ สัมประสิทธิ์ที่ไม่เป็นศูนย์ตัวที่ i

d_i แทนตำแหน่งของคี่เลขของสัมประสิทธิ์ที่ไม่เป็นศูนย์ตัวที่ i

c_i แทนขนาดของผลตอบสนองเชิงเวลาของช่องสัญญาณที่มีตำแหน่งตรงกับสัมประสิทธิ์ที่ไม่เป็นศูนย์ตัวที่ i

$\bar{R}_{xx}$ แทนอัตราสหสัมพันธ์ของสัญญาณขาเข้าของวงจรกรอง

i แทนตำแหน่งของสัมประสิทธิ์ของวงจรกรองตัวที่ i ที่มีค่าไม่เป็นศูนย์ โดยมีค่าตั้งแต่ 1,2,3,...,k

จากรูปที่ 2.5 จะสามารถหาได้ว่า

$$\overline{E}[e_i^2(m)] = K_1 + g_c^2 \left[\frac{1}{N} \sum_{m=0}^{N-1} x^2(m-d) \right] - K_2 \sum_{i=1}^k c_i \overline{R}_{xx}(d-d_i) \quad (2-7)$$

โดยที่

$$K_1 = \sum_{i=1}^k c_i^2 R_{xx}(0) + 2 \sum_{1 \leq i < j \leq k} c_i c_j R_{xx}(d_i - d_j)$$

$$K_2 = 2g_c$$

สมการ (2-7) ซึ่งให้เห็นพจน์แรกเป็นค่าคงที่ พจน์ที่สองมี ค่าลดลงเมื่อ ปรับค่าดีเลย์ขึ้นไปเรื่อย ๆ ส่วนพจน์ที่สามมีลักษณะคล้ายสัญญาณรบกวน ฉะนั้นค่า $E(e_i^2(m))$ ที่ประมาณขึ้นจากการปรับค่าดีเลย์ไปเรื่อย ๆ จะมีลักษณะเป็น กราฟเส้นตรงที่มีค่าความชันเป็นลบโดยค่อย ๆ มีค่าลดลงเมื่อดีเลย์มากขึ้น และมีสัญญาณรบกวนอยู่บนเส้นตรงดังกล่าว ดังนั้นวิธีการนี้ที่ Etter [5] นำเสนอ มีขั้นตอนคือ

ขั้นแรกพิจารณาระบบย่อยแรกของโครงสร้างตามรูปที่ 2.4 กำหนดค่าอัตราขยาย g_i ตามตัวค่าหนึ่งขึ้นมาโดยในที่นี้ให้มีความเท่ากับ g_c จากนั้นทำการประมาณค่า $E(e_i^2(m))$ จากการเปลี่ยนค่าดีเลย์ไปเรื่อย ๆ หลังจากนั้นจากสมการ (2-7) จะเห็นว่าถ้าเราเลือกตำแหน่งของดีเลย์ที่ตรงกับตำแหน่งที่ทำให้ค่าประมาณ $E(e_i^2(m))$ เบี่ยงเบนไปจากแนวโน้มของมันมากที่สุด จะได้ว่าตำแหน่งนั้นตรงกับตำแหน่งของผลตอบสนองเชิงเวลาที่มีค่าสูงสุด ซึ่งแนวโน้มของการประมาณ $E(e_i^2(m))$ สามารถหาได้จากการประมาณการเชิงเส้น เมื่อเราได้ตำแหน่งของดีเลย์มาแล้วเราก็จะทำการปรับอัตราขยายของดีเลย์ ณ ตำแหน่งนี้เพื่อให้ได้อัตราขยายใกล้เคียงความเป็นจริงโดยใช้ LMS อัลกอริทึม ซึ่งในบางกรณีหลังจากการปรับอัตราขยายอาจจะได้อัตราขยายที่ยังห่างจากค่าจริง ๆ อยู่ซึ่งค่าความผิดพลาดนี้จะถูกชดเชยอีกครั้งในการปรับครั้งต่อไป

หลังจากที่เราได้อัตราขยายและตำแหน่งของดีเลย์ที่ถูกต้องแล้ว ต่อไปก็ทำการลบส่วนนี้ออกจากเอาต์พุตของช่องสัญญาณ ซึ่งจะทำให้เหมือนกับว่าผลตอบสนองเชิงเวลาของช่องสัญญาณลดลงไปหนึ่งตัว ดังนั้นถ้าเราทำการประมาณค่า $E(e^2(m))$ ใหม่อีกครั้งต่อไปเราก็จะสามารถทำการประมาณค่าดีเลย์ตัวที่มีอัตราขยายมากที่สุดถัดจากตัวแรกได้ ทำต่อไปเรื่อย ๆ จนได้สัมประสิทธิ์ที่มีค่าไม่เป็นศูนย์ครบทุกตัว

ขั้นตอนวิธี Adaptive Delay Filter นี้จะให้การคำนวณที่น้อยกว่ารวมถึงให้อัตราการลู่เข้าของวงจรกรองที่ค่อนข้างเร็วเมื่อเทียบกับวงจรกรองแบบ LMS แบบที่ไม่มีการปรับอันดับ หากว่าช่องสัญญาณมีผลตอบสนองเชิงเวลาส่วนใหญ่ที่เป็นศูนย์มากเมื่อเทียบกับอันดับของวงจรกรอง ในทางตรงกันข้ามถ้าช่องสัญญาณมีจำนวนผลตอบที่ไม่เป็นศูนย์อยู่น้อยเมื่อเทียบกับอันดับของวงจร

รอง วิธีการนี้จะทำให้ผลที่แย่กว่าวิธีแบบธรรมดา นอกจากนี้วิธีการนี้ต้องกำหนดค่าพารามิเตอร์ให้เหมาะสม มิฉะนั้นจะทำให้การทำงานของวงจรไม่มีเสถียรภาพได้ง่าย

2.3.2 ขั้นตอนวิธีการปรับอันดับของวงจรกรองแบบ Adaptive Tap-Position Control

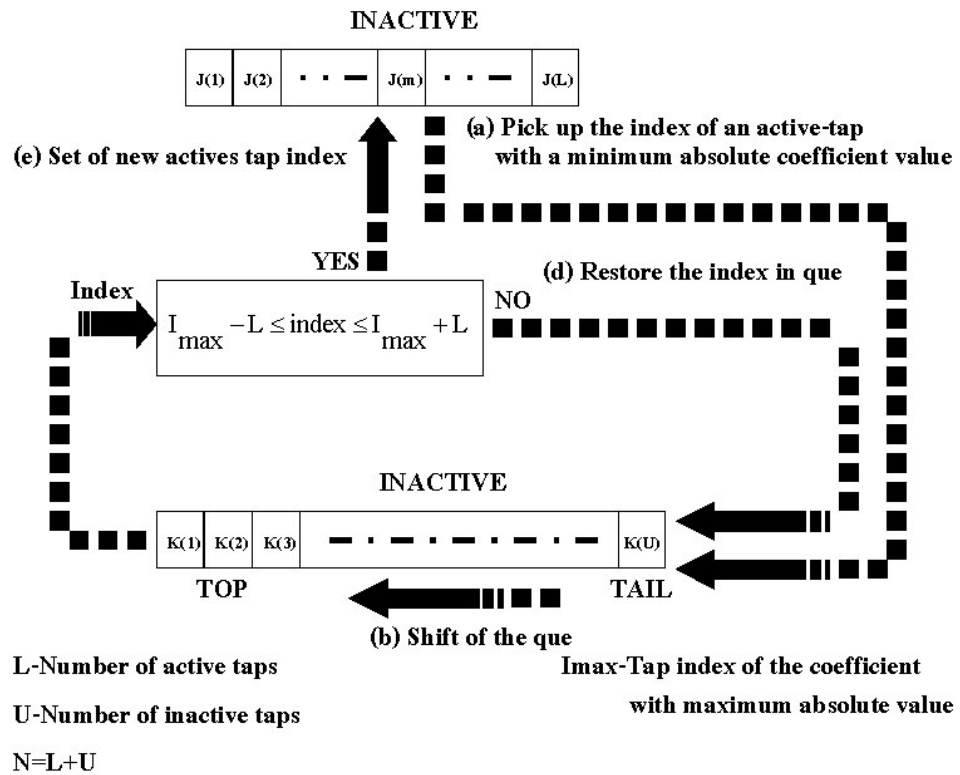
Sugiyama [6] ทำการแบ่งแท็ปทั้งหมดออกเป็นสองส่วน คือส่วนที่แอ็คทีฟแท็ปคือส่วนที่มีค่าสัมประสิทธิ์ และส่วนที่ไม่แอ็คทีฟที่ไม่มีค่าสัมประสิทธิ์ โดยแอ็คทีฟแท็ปที่มีค่าน้อยที่สุดจะถูกทำให้ไม่แอ็คทีฟ และเอาตำแหน่งของแท็ปตัวนั้น ไปเก็บไว้ในส่วนหางของส่วนที่ไม่แอ็คทีฟ หลังจากนั้นจะนำส่วนหน้าของแท็ปที่ไม่แอ็คทีฟมาทำให้แอ็คทีฟ

โดยมีหลักการทำงานสองขั้นตอน คือ

1. ขั้นแรกเรียกว่าส่วน Flat-Delay Estimation จะทำการประมาณ หาขอบเขตของ Dispersive Region

2. ขั้นตอนที่สองเรียกว่า Constrained Tap-Position Control ทำการปรับปรุงค่าแอ็คทีฟแท็ปหลังจากนั้นก็นำตำแหน่งของส่วนแอ็คทีฟแท็ปที่มีค่าน้อยที่สุดมาจำนวนหนึ่งและทำให้ไม่แอ็คทีฟโดยเอาไปต่อท้ายส่วนหางของส่วนที่ไม่แอ็คทีฟ และนำตำแหน่งของส่วนที่ไม่แอ็คทีฟที่อยู่ข้างหน้าออกมา ต่อไปก็จะพิจารณาจากเงื่อนไขว่าแต่ละตำแหน่งในส่วนที่ไม่แอ็คทีฟที่อยู่ภายใน Dispersive Region หรือไม่ถ้าอยู่ตำแหน่งนั้นก็จะถูกทำให้แอ็คทีฟแต่ถ้าไม่ ตำแหน่งนั้นก็จะถูกเอาไปต่อท้ายในส่วนที่ไม่แอ็คทีฟเหมือนเดิม ทำไปจนกว่าจะได้จำนวนแท็ปเท่ากับจำนวนแท็ปที่ถูกทำให้ไม่แอ็คทีฟ โดยแต่ละรอบในขั้นตอนที่สองนี้จะพิจารณาคุณค่าผิดพลาดจากการประมาณด้วยว่ามากน้อยแค่ไหน ถ้าค่าผิดพลาดที่เกิดจากการประมาณมีค่ามากเกินไปค่าหนึ่ง ก็จะอนุญาตให้มีการเลื่อน Dispersive Region ไป แล้วทำขั้นตอนที่สองใหม่ โดยกระบวนการดังกล่าวจะทำการวนซ้ำจนค่าผิดพลาดมีค่าน้อยที่จะสามารถยอมรับได้

วิธีการนี้ให้อัตราการลู่เข้าที่เร็วเมื่อเทียบกับวงจรกรองที่ใช้ระเบียบวิธี LMS ที่ไม่มีการปรับอันดับเมื่อนำไปใช้กับช่องสัญญาณที่มีผลตอบสนองเชิงเวลาส่วนใหญ่เป็นศูนย์มากเมื่อเทียบกับอันดับของวงจรกรอง แต่วิธีการนี้ต้องสร้างฮาร์ดแวร์ที่มีความซับซ้อนสูงเมื่อนำไปสร้างจริง



รูปที่ 2.6 ขั้นตอนวิธีการทำงานของระเบียบวิธี Adaptive Tap-Position Control

2.3.3 ขั้นตอนวิธีการปรับอันดับของวงจรกรองแบบ LMS Estimation via Structural Detection

Homer [7] ให้ค่าสัมประสิทธิ์ทุกตัว ของวงจรกรองแบบ FIR มีค่าเป็นศูนย์ ยกเว้น สัมประสิทธิ์จำนวนหนึ่งมีค่าไม่เป็นศูนย์ ซึ่งจำนวนสัมประสิทธิ์ที่ไม่เป็นศูนย์ดังกล่าวมีค่าน้อยเมื่อเทียบกับจำนวนแท็ปของวงจรกรอง และเรียกสัมประสิทธิ์ที่ไม่เป็นศูนย์ที่ใช้ในการประมาณช่องสัญญาณนั้นว่า “แอดคทีฟแท็ป” โดยสิ่งที่ใช้บอกกว่าแท็ปไหนบ้างที่เป็นแอดคทีฟแท็ปนั้นทำได้โดยการเปรียบเทียบกับ Threshold ค่าหนึ่ง

พิจารณาช่องสัญญาณไม่ทราบค่าที่โดยถือว่าช่องสัญญาณเป็นเชิงเส้นไม่เปลี่ยนแปลงตามเวลาและสามารถแทนได้ด้วยวงจรกรองแบบ FIR ที่มีความยาว $n-1$ แสดงได้โดย

$$\theta(z^{-1}) = \theta_0 + \theta_1 z^{-1} + \dots + \theta_{n-1} z^{-(n-1)} \quad (2-8)$$

โดย $u(k)$ เป็นสัญญาณขาเข้าของช่องสัญญาณที่ไม่ทราบค่า และวงจรกรอง โดยมีสัญญาณรบกวนเป็น $s(k)$ อยู่ในช่องสัญญาณ ส่วน $y(k)$ เป็นสัญญาณขาออกที่ออกจากช่องสัญญาณ ณ เวลา k ซึ่งมีค่าเท่ากับ

$$y(k) = U(k)^T \theta + s(k) \quad (2-9)$$

โดย $\theta = [\theta_0 \theta_1 \dots \theta_{n-1}]^T$

$$U(k) = [u(k) \ u(k-1) \ \dots \ u(k-n+1)]^T$$

และ $\bar{\theta} = [\bar{\theta}_0 \ \bar{\theta}_1 \ \dots \ \bar{\theta}_{n-1}]^T$ แทนสัมประสิทธิ์ของวงจรกรองปรับตัว

ก่อนที่จะแสดงให้เห็นถึงเหตุผลในการปรับจำนวนสัมประสิทธิ์ ค่าสัญญาณต่าง ๆ รวมถึงช่องสัญญาณต้องเป็นไปตามสมมติฐานต่อไปนี้

สมมติฐานข้อที่ 1. สัญญาณขาเข้าและสัญญาณรบกวนมีค่าเฉลี่ยเป็นศูนย์ กำลังจำกัด และค่าทางสถิติอันดับ 1 และ 2 ไม่เปลี่ยนแปลงตามเวลา

$$R = E(U(k)U(k)^T) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=k_0}^{k_0+N-1} U(k)U(k)^T \quad (2-10)$$

$$R_s(j) = E(s(k)s(k-j)) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=k_0}^{k_0+N-1} s(k)s(k-j) \quad (2-11)$$

สมมติฐานข้อที่ 2. Covariance Matrix R ขนาด $n \times n$ ของสัญญาณขาเข้ามีค่าเป็น Positive Definite

สมมติฐานข้อที่ 3. สัญญาณขาเข้าและสัญญาณรบกวนไม่มีความสัมพันธ์กัน ณ.ทุกเวลา

$$\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{k=k_0}^{k_0+N-1} s(k)u(k-j) = 0 \quad (2-12)$$

สมมติฐานข้อที่ 4. สัญญาณขาเข้า ไม่มีความสัมพันธ์กันตลอดเวลา

$$R = \sigma_u^2 I \quad (2-13)$$

โดย I คือเมทริกซ์เอกลักษณ์ขนาด $n \times n$

สมมติฐานข้อที่ 5. แบบจำลอง FIR ของช่องสัญญาณที่ไม่ทราบค่ามีเพียงสัมประสิทธิ์จำนวน $m < n$ ที่มีค่าไม่เป็นศูนย์

$$\theta(z^{-1}) = b_{t_1} z^{-t_1} + b_{t_2} z^{-t_2} + \dots + b_{t_m} z^{-t_m} \quad (2-14)$$

ในส่วนต่อไปจะแสดงให้เห็นว่าจะสามารถปรับปรุงการปรับค่าสัมประสิทธิ์ได้อย่างไรเมื่อใช้วิธีการปรับค่าสัมประสิทธิ์ของวงจรแบบ Active Tap Parameterization

พิจารณาสัญญาณที่ออกจากช่องสัญญาณ

$$y(k) = \theta(z^{-1})u(k) + s(k) = \sum_{i=1}^m b_{t_i} u(k - t_i) + s(k) \quad (2-15)$$

และสัญญาณที่ออกจากรองรับที่ใช้ประมาณ โดยถือว่าสัมประสิทธิ์ทุกตัวมีค่าไม่เป็นศูนย์

$$\bar{y}(k) = \bar{\theta}(z^{-1})u(k) = \sum_{i=1}^n \bar{\theta}_i u(k - i) \quad (2-16)$$

สิ่งที่ใช้วัดคุณภาพของการประมาณสามารถใช้สมการที่ (2-17) ซึ่งโดยสมมติฐานข้อที่ 4 จะได้ว่า

$$E[(y(k) - \bar{y}(k))^2] = [\sum_{i=1}^m (b_{t_i} - \bar{\theta}_{t_i})^2 + \sum_{i \neq t_i} \bar{\theta}_i^2] \sigma_u^2 + \sigma_s^2 \quad (2-17)$$

โดยค่า σ_u^2 และ σ_s^2 คือค่าความแปรปรวนของสัญญาณขาเข้า และสัญญาณรบกวนตามลำดับพร้อมกันนี้เรานิยามให้

$$\Delta \theta_i(N) = [\theta_i - \bar{\theta}_i(N)] \sqrt{N} \quad (2-18)$$

โดย N คือจำนวนของข้อมูลที่ใช้ในการประมาณ เนื่องจากอินพุตมีลักษณะเป็นสัญญาณรบกวนสีขาว และที่ค่า N มากจะได้ว่า

$$\Delta\theta_i(N) = AsN(0, \sigma_i) \quad (2-19)$$

$$\lim_{N \rightarrow \infty} E(\Delta\theta_i(N)\Delta\theta_j(N)) = \begin{cases} \sigma_i^2 & i = j \\ 0 & i \neq j \end{cases} \quad (2-20)$$

สำหรับตัวประมาณแบบ LS (Least Squares) มีคุณสมบัติ คือ

$$\sigma_i = \sqrt{\frac{\sigma_s^2 N}{\sigma_u^2 (N-i)}} \rightarrow \frac{\sigma_s}{\sigma_u} \text{ as } N \rightarrow \infty \quad (2-21)$$

ส่วนตัวประมาณแบบ LMS โดยใช้ช่วงก้าวก้าวในการปรับตัว (Stepsize) ที่เล็กพอ ผลคือ

$$\sigma_i = \sqrt{N\mu\sigma_s^2/2} \text{ as } N \rightarrow \infty \quad (2-22)$$

จะได้ว่า

$$\lim_{k \rightarrow \infty} E[(y(k) - \bar{y}(k))^2] = \sum_{i=1}^n \frac{\sigma_i^2 \sigma_u^2}{k} + \sigma_s^2 \quad (2-23)$$

จากสมการที่ (2-23) จะเห็นว่า Asymptotic Variance Error ที่เกิดจากการประมาณสัมประสิทธิ์ตัวที่ i คือ $\sigma_i^2 \sigma_u^2$ ขณะที่จากสมการที่ (2-17) หากเราละเลยหรือไม่ทำการประมาณสัมประสิทธิ์ตัวที่ t_i ซึ่งจะให้ค่า Bias Error คือ $b_{t_i}^2 \sigma_u^2$ ดังนั้นถ้าหากพบว่า $b_{t_i}^2 > \sigma_i^2$ ก็สมเหตุสมผลที่เราจะทำการประมาณค่าสัมประสิทธิ์ตัวที่ t_i และให้สัมประสิทธิ์เหล่านั้นเป็นแอ็คทีฟแท้ไป

ส่วนในเรื่องของอัตราการลู่เข้าสำหรับตัวประมาณแบบ LMS ภายในเงื่อนไขที่ว่าช่วงก้าวเล็ก และสมมติฐานที่ 1 ถึง 3 เป็นจริง จึงได้มีการเสนอ

$$C = \frac{2n\mu}{Tr(R^{-1})} \quad (2-24)$$

เพื่อใช้ในการวัดอัตราการลู่เข้า โดยที่ $Tr(.)$ คือ Trace และสำหรับสัญญาณขาเข้าที่เป็นสัญญาณรบกวนสีขาว C จะไม่ขึ้นกับความยาว n ของวงจรกรอง โดย $C = 2\mu\sigma_u^2$ ส่วนตัวประมาณแบบ NLMS มีช่วงก้าวคือ

$$\mu_{norm} = \mu/(n\sigma_u^2) \quad (2-25)$$

ดังนั้นสำหรับ สัญญาณขาเข้าที่เป็นสัญญาณรบกวนสีขาวจะได้อัตราการลู่เข้าคือ $C = 2\mu/n$ ซึ่งจะเห็นว่าอัตราการลู่เข้าของ NLMS ดีว่า LMS ถ้าหากอันดับของวงจรกรองไม่มากนักที่ Asymptotics Performance ของ NLMS จากสมการข้างล่าง จะเห็นว่าค่าที่ไม่ขึ้นอยู่กัอันดับของวงจรกรอง

$$\lim_{k \rightarrow \infty} E[(y(k) - \bar{y}(k))^2] = \frac{\mu\sigma_s^2}{2} + \sigma_s^2 \quad (2-26)$$

สรุปผลก็คือสำหรับกรณีของสัญญาณขาเข้าที่เป็นสัญญาณรบกวนสีขาวการปรับอันดับของวงจรให้ผลคือ สำหรับ LMS ทำให้ Asymptotic Performance ดีขึ้นแต่ไม่ได้ปรับปรุงอัตราการลู่เข้า ส่วน NLMS ปรับปรุงการลู่เข้าให้ดีขึ้นแต่ไม่ได้ส่งผลต่อ Asymptotic Performance

การพิจารณาว่าแท็ปใดบ้างที่เป็นแอ็คทีฟแท็ป ตามที่เสนอนั้นพิสูจน์โดยการพิจารณาค่าตำแหน่งที่สัมประสิทธิ์ของวงจรกรองมีขนาดที่ทำให้ Cost Function มีค่าน้อยที่สุดจากการประมาณโดยใช้แอ็คทีฟแท็ป ดังกล่าว ซึ่งสามารถทำได้โดยพิจารณา

$$X_N(j) = \frac{[\sum_{k=j+1}^N y(k)u(k-j)]^2}{\sum_{k=j+1}^N u^2(k-j)} \quad (2-27)$$

ซึ่งแท็ปใดที่ Active ต้องสอดคล้องกับเงื่อนไข

$$X_N(j) > (\frac{\log N}{N}) \sum_{i=1}^N y^2(k) \quad (2-28)$$

เมื่อทราบวิธีการในการตรวจสอบแอ็คทีฟแท็ปแล้ว ต่อไปจะเป็นการประยุกต์ใช้วิธีการตรวจสอบแอ็คทีฟแท็ปกับวิธีการประมาณช่องสัญญาณแบบ LMS ซึ่งตามที่ถูกเสนอนี้มีอยู่มีอยู่สองวิธีคือ

1) วิธีการแรกในช่วงของตัวอย่างสัญญาณมีจำนวนน้อยกว่า N มาก ๆ โดยที่ N มีค่ามากกว่าจำนวนสัมประสิทธิ์ของวงจรกรอง มาก ๆ วงจรกรองจะทำการปรับค่าสัมประสิทธิ์แบบ LMS ตามปกติ เมื่อถึงสัญญาณสุ่มตัวที่ N หลังจากการกรองสัญญาณตามปกติ วงจรก็จะทำการตรวจสอบค่าสัมประสิทธิ์ของวงจรกรองว่าตัวไหนบ้างที่เป็นแอ็คทีฟแท็ป (เหมือนกับขั้นตอนที่ 1 ถึง 3 ของวิธีการแบบที่สอง) หลังจากนั้นปรับปรุงค่าสัมประสิทธิ์แบบ LMS โดยทำการปรับปรุงสัมประสิทธิ์เฉพาะตัวที่เป็นแอ็คทีฟแท็ป (ณ เวลา $k = N$) โดยใช้ค่าเริ่มต้นในการประมาณคือ

$$\bar{\theta}_j(N) = \frac{\sum_{k=j+1}^N y(k)u(k-j)}{\sum_{k=j+1}^N u(k-j)^2} \quad (2-29)$$

2) วิธีการที่สองคือในทุก ๆ ช่วงเวลาจะทำการตรวจสอบสัมประสิทธิ์ทุกตัวว่าเป็นแอ็คทีฟแท้ป หรือไม่ หลังจากนั้นทำการปรับค่าสัมประสิทธิ์ทุกตัวแบบ LMS โดยสัมประสิทธิ์ตัวที่ไม่แอ็คทีฟ จะมี Forgetting Factor (α) เป็นตัวถ่วงน้ำหนักอยู่และจะไม่มีการปรับปรุงค่าสัมประสิทธิ์ตัวดังกล่าว ซึ่งสามารถแสดงได้โดย

1) เลือก $\alpha \in [0,1)$ $q(k), f_j(0), x_j(0), \bar{\theta}_j(0) = 0, r_j(0) = r_0, j = 0,1,2,\dots,n-1$

2) ปรับปรุงค่า $q(k), f_j(j \leq k), r_j(j \leq k), x_j(j \leq k)$ ณ เวลา k โดย

$$f_j(k) = f_j(k-1) + y(k)u(k-j)^2 \quad (2-30)$$

$$r_j(k) = r_j(k-1) + u(k-j)^2 \quad (2-31)$$

$$x_j(k) = \frac{f_j^2(k)}{r_j(k)} \quad (2-32)$$

$$q(k) = q(k-1) + y(k)^2 \quad (2-33)$$

3) สร้างเมตริกซ์ $g(k)$ ที่เป็นหนึ่ง ณ ตำแหน่งที่แอ็คทีฟ และตำแหน่งที่ไม่แอ็คทีฟ เป็นศูนย์ จากเงื่อนไขที่แอ็คทีฟ คือ

$$x_j(k) > [q(k) \log k]/k \quad (2-34)$$

4) ปรับปรุง $\bar{\theta}_j(k)$ ณ เวลา k แล้วกลับไปขั้นตอนที่ 2

$$e(k) = y(k) - U(k)^T \bar{\theta}(k) \quad (2-35)$$

$$\bar{\theta}(k) = (\bar{\theta}_0(k), \bar{\theta}_1(k), \dots, \bar{\theta}_{n-1}(k))^T \quad (2-36)$$

$$\bar{\theta}_j(k+1) = \alpha^{1-g_j(k)} \bar{\theta}_j(k) + \mu e(k) g_j(k) u(k-j) \quad (2-37)$$

ระเบียบวิธี LMS Estimation via Structural Detection นี้จะให้อัตราการลู่เข้าของวงจรกรองที่ช้ากว่าระเบียบวิธี Adaptive Delay Filter และ Adaptive Tap-position Control แต่จะมีเสถียรภาพการทำงานของวงจรที่ดีกว่าเมื่อนำมาสร้างจริงเป็นฮาร์ดแวร์รวมถึงความซับซ้อนในการสร้างจริงเป็นฮาร์ดที่น้อยกว่า

เนื่องจากปัญหาการป้อนกลับทางเสียงที่กำลังพิจารณานั้น สัญญาณที่พิจารณามีความถี่ไม่สูง รวมถึงเครื่องช่วยฟังมีพื้นที่ของวงจรที่จำกัด ดังนั้นวงจรกรองที่ใช้ภายในเครื่องช่วยฟังต้องเป็นวงจรที่ไม่ใช้พื้นที่ของฮาร์ดแวร์มากนัก หากนำระเบียบวิธี Adaptive Delay Filter หรือ Adaptive Tap-position Control มาประยุกต์สร้างเป็นวงจรก็จะต้องใช้พื้นที่ของฮาร์ดแวร์ที่สูง รวมถึงความเร็วในการทำงานของวงจรที่สูงเกินความจำเป็น รวมถึงการทำงานของวงจรอาจมีเสถียรภาพที่ไม่ดี ดังนั้นโครงการวิจัยนี้จึงจะพิจารณาการนำระเบียบวิธี LMS Estimation via Structural Detection มาประยุกต์สร้างเป็นวงจรกรองแบบปรับอันดับเนื่องจากมีข้อดีคือสามารถสร้างให้ใช้พื้นที่ของฮาร์ดแวร์ที่ไม่สูง ความมีเสถียรภาพของวงจร และความเร็วในการทำงานของวงจรที่ไม่สูงมากเกินไป

บทที่ 3

วงจรกรองอันดับคงที่

ในบทนี้หัวข้อ 3.1 กล่าวถึงหลักการทำงานของวงจรกรองที่ใช้ในการกำจัดการป้อนกลับแบบไม่ต่อเนื่อง หัวข้อ 3.2 เป็นการอธิบายถึงหลักการทำงานของวงจรกรองแบบผสมระหว่างมัลติเพล็กซ์และ Direct Form และรายละเอียดโครงสร้างของวงจรในหัวข้อที่ 3.2.1 และ 3.2.2 ตามลำดับ

3.1 การทำงานของวงจรกรองแบบไม่ต่อเนื่องที่ใช้ในการกำจัดการป้อนกลับ

วิธีแก้ปัญหาลำโพงเสียงจากการป้อนกลับทางเสียงในเครื่องช่วยฟังที่เราสนใจ คือการแก้ปัญหาด้วยการใช้ตัวกำจัดการป้อนกลับ (Acoustic Feedback Canceller: AFC) โดยการใช้วงจรกรองปรับตัวสร้างสัญญาณเลียนแบบสัญญาณป้อนกลับ ซึ่งการทำงานของวงจรกรองเป็นแบบปรับค่าพารามิเตอร์ไม่ต่อเนื่อง การทำงานแบบนี้มีข้อดีกว่าการปรับค่าพารามิเตอร์แบบต่อเนื่องคือ การทำงานของวงจรมีเสถียรภาพที่ดีกว่าเนื่องจากวงจรจะทำการปรับค่าพารามิเตอร์เฉพาะตอนที่กำลังของสัญญาณขาเข้าของวงจรกรองมีค่าต่ำ และสัญญาณสะสมสร้างขึ้นมาเพื่อทำการปรับค่าพารามิเตอร์มีกำลังต่ำทำให้ไม่มีสัญญาณรบกวนผู้ใช้งานหูฟัง ต่อไปในโครงการวิจัยนี้จะเรียกววงจรกรองปรับตัวแบบไม่ต่อเนื่องที่ใช้ในการกำจัดการป้อนกลับว่า AFC

โดยสมมติให้ ณ เวลา n สัญญาณต่าง ๆ ภายในเครื่องช่วยฟังคือ

$s(n)$ คือสัญญาณขาเข้าของเครื่องช่วยฟัง กำหนดให้มีคุณสมบัติเป็นสัญญาณสุ่มแบบ White Gaussian ที่มีค่าเฉลี่ยเป็นศูนย์

$f(n)$ คือสัญญาณที่เกิดจากการป้อนกลับ

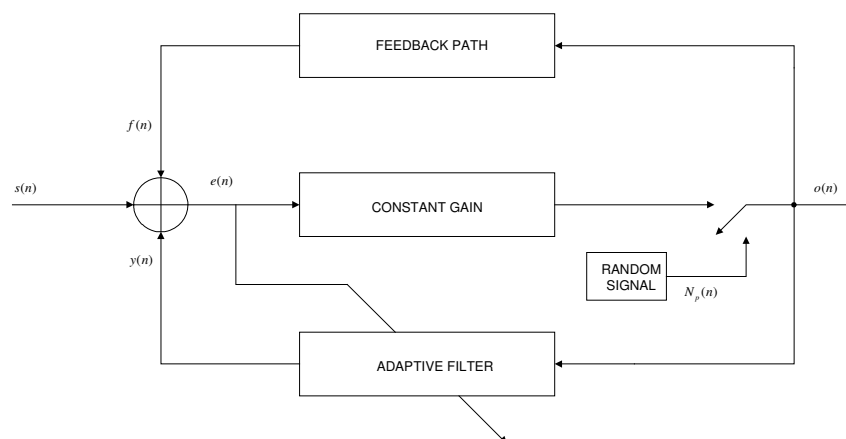
$y(n)$ คือสัญญาณขาออกของวงจรกรองปรับตัว

$o(n)$ คือสัญญาณขาออกของเครื่องช่วยฟัง

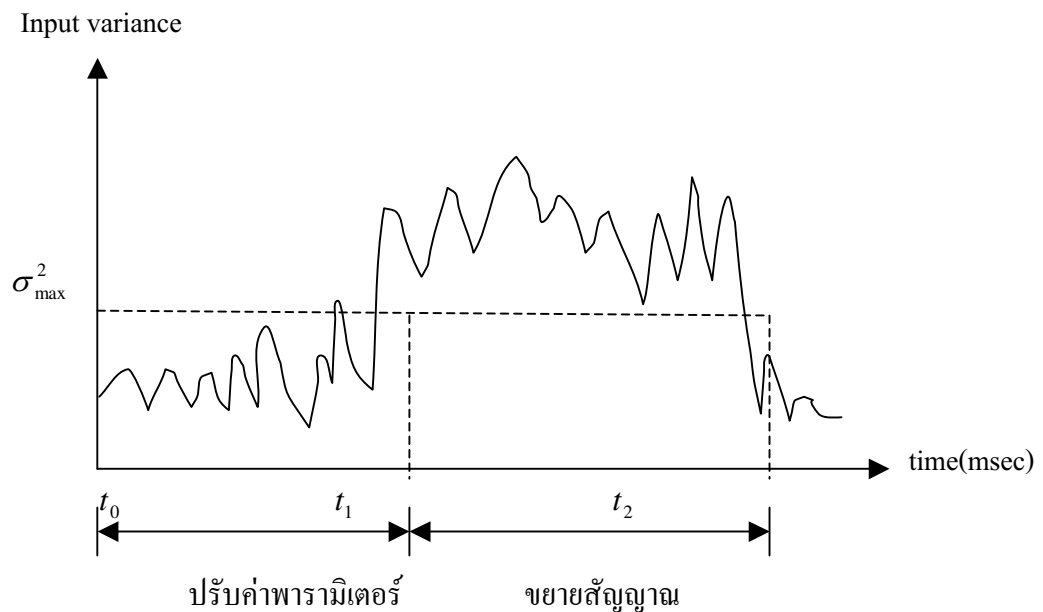
$e(n)$ คือความคลาดเคลื่อนระหว่างสัญญาณป้อนกลับกับสัญญาณขาออกของวงจรกรองปรับตัว

$N_p(n)$ คือสัญญาณสุ่มที่ใช้ในการปรับค่าพารามิเตอร์ของวงจรกรองกำหนดให้มีคุณสมบัติเป็นสัญญาณสุ่มแบบ White Gaussian ที่มีค่าเฉลี่ยเป็นศูนย์ และมีค่าความแปรปรวนเท่ากับ 0.1

พิจารณารูปที่ 3.1 ในรูปแสดงถึงโครงสร้างของ AFC ที่อยู่ภายในเครื่องช่วยฟังเพื่อสร้างสัญญาณขึ้นมัลดทอนสัญญาณป้อนกลับที่ถูกส่งผ่านกลับมาทางช่องสัญญาณป้อนกลับ หลักการทำงานของ AFC สามารถแสดงได้ดังรูปที่ 3.2 ซึ่งแสดงถึงความสัมพันธ์ระหว่างกำลังของสัญญาณขาเข้าของเครื่องช่วยฟังกับช่วงเวลาการทำงานของ AFC โดยมีหลักการทำงานดังนี้ ในช่วงที่กำลังของสัญญาณ $s(n)$ มีค่าสูงอยู่ (t_1 ถึง t_2) เครื่องช่วยฟังจะทำการขยายสัญญาณเสียงตามปกติให้ผลเป็นสัญญาณ $o(n)$ ซึ่งถูกป้อนกลับผ่านทางเส้นทางป้อนสัญญาณกลายเป็นสัญญาณ $f(n)$ เกิดขึ้นไปรบกวนสัญญาณเสียง $s(n)$ ที่ถูกขยายตามปกติ จากนั้นเมื่อกำลังของสัญญาณ $s(n)$ มีค่าต่ำ (t_0 ถึง t_1) ในที่นี้คือเมื่อแวลเรียนที่มีค่าน้อยกว่า 0.01 เครื่องช่วยฟังจะทำการตัดต่อวงจรไปที่วงจรสร้างสัญญาณสุ่มเพื่อสร้างสัญญาณ $N_p(n)$ ป้อนให้กับ AFC เพื่อใช้ในการปรับค่าพารามิเตอร์ ทำให้ได้สัญญาณ $y(n)$ ออกจาก AFC ไปหักล้างกับสัญญาณป้อนกลับ $f(n)$ ทำให้เกิดสัญญาณ $e(n)$ ซึ่ง AFC จะนำสัญญาณ $e(n)$ มาใช้ในการปรับค่าพารามิเตอร์เพื่อให้ ค่าความคลาดเคลื่อน $e(n)$ มีค่าต่ำที่สุดนั่นคือ AFC จะทำการปรับค่าพารามิเตอร์ไปจนมีลักษณะคล้ายกับเส้นทางป้อนกลับทำให้ได้ค่าสัญญาณ $y(n)$ ที่ใกล้เคียงกับสัญญาณ $f(n)$ หลังจากที AFC เข้าสู่สภาวะอยู่ตัว จะสามารถลดผลของสัญญาณป้อนกลับได้ เครื่องช่วยฟังก็จะตัดต่อวงจรไปที่วงจรขยายเสียงเพื่อทำการขยายเสียงต่อไป



รูปที่ 3.1 การใช้วงจรกรองปรับตัวสร้างสัญญาณลดทอนการป้อนกลับทางเสียงในเครื่องช่วยฟัง



รูปที่ 3.2 การทำงานในแต่ละช่วงของเครื่องช่วยฟังที่ขึ้นอยู่กับกำลังของสัญญาณขาเข้า

โดยทั่วไปวงจรกรองที่นำมาใช้มักจะเป็นวงจรกรอง FIR ที่มีโครงสร้างแบบ Direct Form และใช้ระเบียบวิธี LMS ในการปรับค่าพารามิเตอร์เนื่องจากให้เสถียรภาพที่ดี ดังที่ได้กล่าวถึงในบทที่ 2 ว่าวงจรกรองปรับตัว FIR ที่มีโครงสร้างแบบ Direct Form และปรับตัวตามระเบียบวิธี LMS นั้นโดยปกติจะประกอบด้วยสองส่วนคือส่วนที่ใช้กรองสัญญาณ (Filtering) และส่วนที่ทำการปรับปรุงค่าพารามิเตอร์ (Adaptation) ส่วนของการกรองสัญญาณ และการปรับค่าพารามิเตอร์ นั้นแต่ละแท็ปต้องการวงจรคูณและวงจรบวกอย่างละ 1 ตัวเนื่องจากวงจรคูณเป็นส่วนที่ต้องใช้เนื้อที่ของฮาร์ดแวร์มากแต่วงจรที่ใช้ในเครื่องช่วยฟังต้องเป็นวงจรที่ใช้เนื้อที่จำกัด ดังนั้นการใช้โครงสร้างแบบ Direct Form จึงไม่เหมาะสมสำหรับกรณีของเครื่องช่วยฟังซึ่งมีพื้นที่ฮาร์ดแวร์ที่จำกัด

3.2 วงจรกรองปรับตัว LMS แบบผสมระหว่างมัลติเพล็กซ์และ Direct Form

ปัญหาเรื่องพื้นที่ของฮาร์ดแวร์ในเครื่องช่วยฟังถ้าใช้โครงสร้างวงจรกรองแบบ Direct Form I ดังกล่าวสามารถแก้ไขได้โดยใช้โครงสร้างของวงจรกรองอีกลักษณะหนึ่งที่สามารถประหยัดพื้นที่ของฮาร์ดแวร์ได้ดีกว่า โครงสร้างดังที่ได้กล่าวมาแล้วนั้นก็คือ วงจรกรองที่มีโครงสร้างวงจรเป็นแบบมัลติเพล็กซ์ ภายในวงจรกรองลักษณะดังกล่าวจะใช้เพียงแฉ่ววงจรคูณและวงจรบวกอย่างละตัวเท่านั้น โดยในส่วนของการกรองสัญญาณ และการปรับค่าพารามิเตอร์จะมีเพียงวงจรคูณและ

วงจรดังกล่าวร่วมกันทำให้ลดเนื้อที่ของฮาร์ดแวร์ในการใช้วงจรคูณและวงจรบวกของแต่ละแท็ปไปได้ แต่โครงสร้างของวงจรลักษณะดังกล่าวยังมีข้อเสียคือ เมื่ออันดับของวงจรกรองมีค่าสูงขึ้นช่วงระยะเวลาการทำงานในแต่ละรอบของวงจรกรองก็จะสูงขึ้นตามไปด้วย ดังนั้นหากอันดับของวงจรกรองมีค่าสูงอาจส่งผลให้วงจรไม่สามารถประมวลผลได้ทันก่อนที่สัญญาณจะเข้ามาใหม่ได้ ดังนั้นเพื่อให้เกิดการ Trade Off กันระหว่างพื้นที่ฮาร์ดแวร์กับความเร็วการทำงานในแต่ละรอบของวงจรกรอง เราจึงเลือกใช้โครงสร้างของวงจรที่มีลักษณะผสมระหว่างโครงสร้างวงจรกรอง LMS แบบ Direct Form กับโครงสร้างแบบมัลติเพล็กซ์

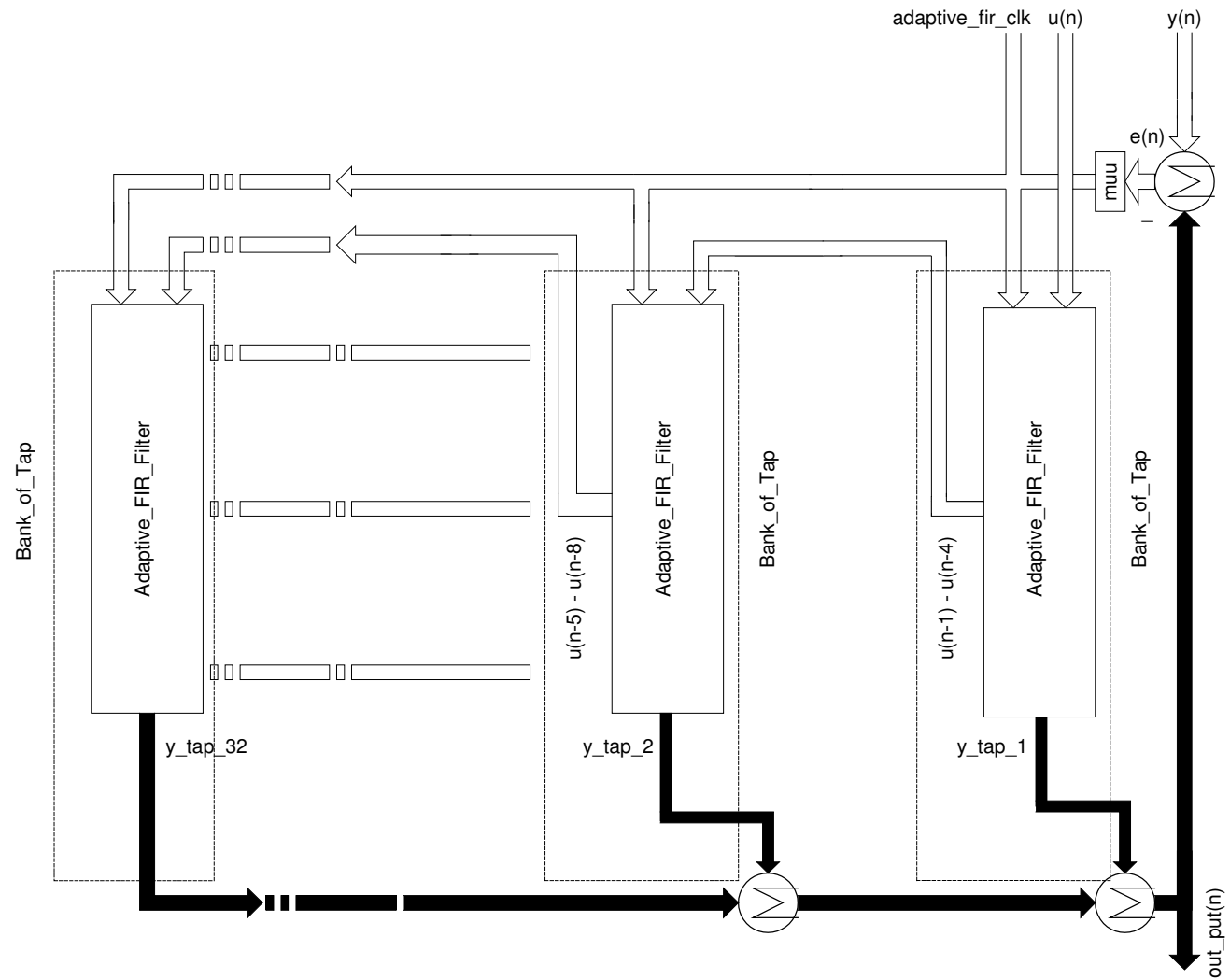
3.2.1 ขั้นตอนการทำงานของวงจรกรองปรับตัวแบบผสมมัลติเพล็กซ์และ Direct Form

วงจรกรองแบบผสมมัลติเพล็กซ์และ Direct Form ดังแสดงในรูปที่ 3.3 ภายในจะประกอบด้วยกลุ่มของวงจรกรองที่มีโครงสร้างแบบมัลติเพล็กซ์ซึ่งเรียกว่า Adaptive_FIR_Filter ซึ่งแต่ละกลุ่มมีจำนวน 4 แท็ป โดยจะเรียกกลุ่มของ 4 แท็ปดังกล่าวว่า Bank Of Taps ดังแสดงในรูปที่ 3.4 มีหลักการทำงานคือ การทำงานของวงจรจะแบ่งออกเป็นสองขั้นตอนคือขั้นตอนแรกวงจรจะทำหน้าที่กรองสัญญาณ (Filtering) จากนั้นในขั้นตอนที่สองวงจรจะตัดต่อโครงสร้างไปเพื่อทำการปรับปรุงค่าสัมประสิทธิ์ (Adaptation) พิจารณารูปสัญญาณนาฬิกาในรูปที่ 3.5

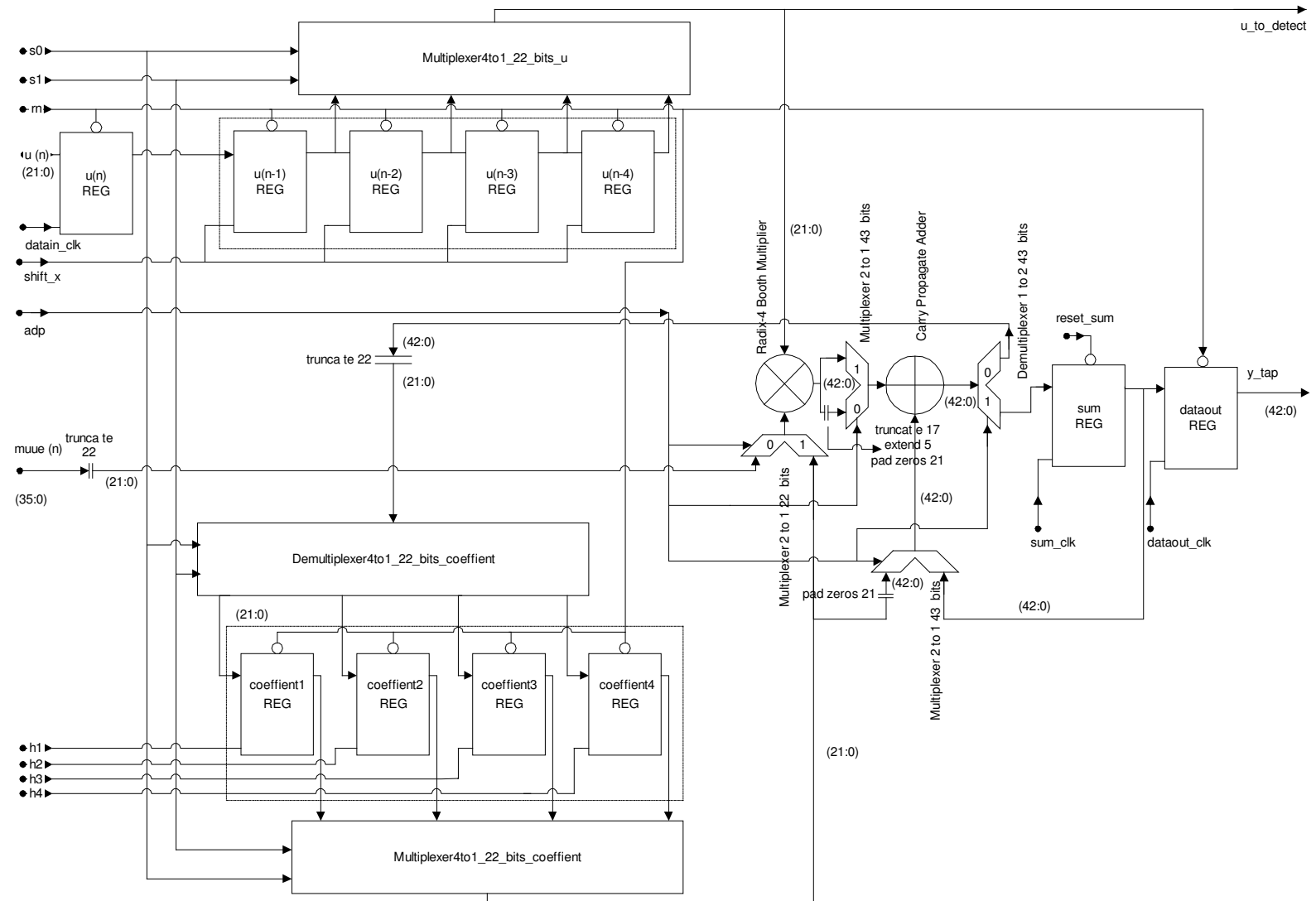
ในส่วนแรกคือส่วนของการกรองสัญญาณ ภายในวงจรกรองแบบมัลติเพล็กซ์แต่ละตัวเมื่อสัญญาณ Adp มีค่าเป็น 1 วงจรจะตัดต่อเป็นในส่วนของการกรองสัญญาณ จากนั้นจะมีสัญญาณ S0 และ S1 ทำการเลือกตำแหน่งของแท็ปเพื่อทำการเลือกข้อมูลแต่ละแท็ปกับค่าสัมประสิทธิ์ออกมาทำการคูณกันโดยมีสัญญาณ Sum_clk ป้อนให้กับ Sum_reg เพื่อใช้ในการสะสมค่าผลคูณแต่ละรอบจนได้ ค่าสัญญาณขาออกของวงจรออกมาและนำค่าสัญญาณขาออกของวงจรที่ได้จากวงจรกรองแบบมัลติเพล็กซ์ในแต่ละกลุ่มทั้งหมดมารวมกันเพื่อให้ได้สัญญาณขาออกของวงจรกรองทั้งหมด และขณะเดียวกันก็ทำการเลื่อนสัญญาณขาเข้าของวงจรกรองและค่าสัญญาณอ้างอิงค่าใหม่เข้ามา ณ.ขอบขาขึ้นของสัญญาณ Datain_clk และ Dataout_clk ตามลำดับ ซึ่งในขณะนี้จะได้ muu ออกมาด้วย ซึ่งผลที่ได้ก็คือการกรองสัญญาณนั่นเอง

หลังจากนั้นในส่วนที่สองก็จะเป็นส่วนของการปรับปรุงค่าสัมประสิทธิ์ โดยสัญญาณ Adp จะเปลี่ยนจาก 1 เป็น 0 วงจรก็จะตัดต่อโครงสร้างการทำงานเพื่อทำการปรับปรุงค่าสัมประสิทธิ์ จากนั้นสัญญาณ S0, S1 จะทำการเลือกตำแหน่งของสัมประสิทธิ์ที่ต้องการจะปรับปรุง หลังจากนั้นจะมีสัญญาณ ขอบขาขึ้นของ H1 ถึง H4 ป้อนให้กับบริจิสเตอร์ของสัมประสิทธิ์ ณ.ตำแหน่งที่ต้องการปรับปรุงค่า เพื่อทำการเก็บค่าสัมประสิทธิ์ที่ได้ทำการปรับปรุงแล้วไว้ใช้ในการกรองสัญญาณในรอบต่อไป

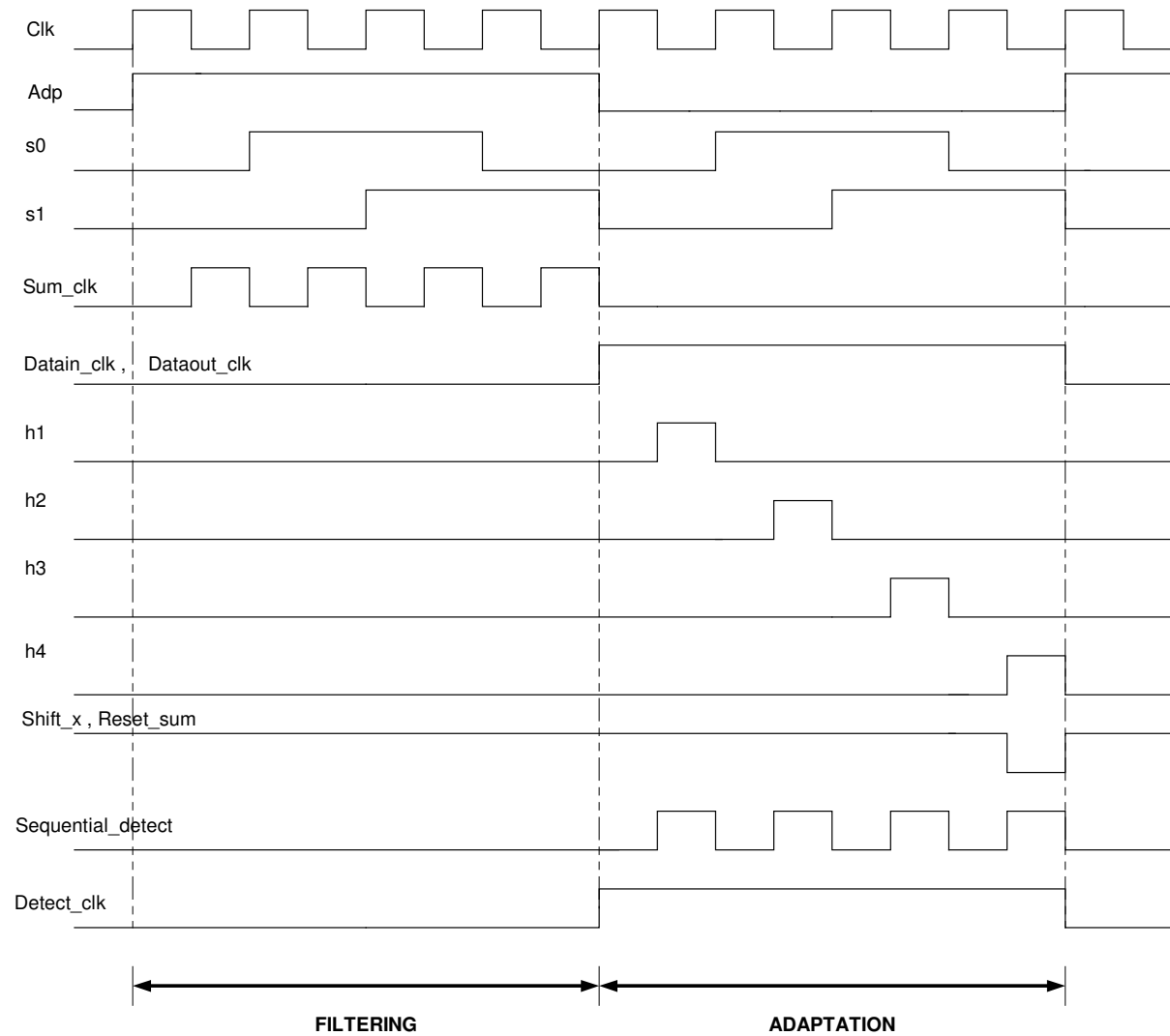
ข้อได้เปรียบอีกอย่างของโครงสร้างลักษณะนี้คือสามารถกำหนดจำนวน Bank of Taps ของวงจรกรอง (Pre-program) ก่อนการใช้งานเพื่อให้สามารถครอบคลุมขนาดของผลตอบสนองเชิงเวลาของช่องสัญญาณได้ โดยผู้ใช้สามารถเลือกจำนวนแท็ปได้เป็นจำนวนเท่าของสี่ โดยมีขนาดจำนวนแท็ปสูงสุดเท่ากับ $32 \times 4 = 128$ (สำหรับตัวอย่างวงจรที่แสดงนี้) แต่ถึงแม้จะมีความสะดวกในการกำหนดอันดับของวงจรกรองให้ครอบคลุมขนาดของผลตอบสนองเชิงเวลา แต่วงจรลักษณะนี้ก็ยังขาดความยืดหยุ่นในการใช้งานอยู่ เพราะถึงแม้จะสามารถกำหนดจำนวน Bank of Taps ได้ก็ตาม แต่จะไม่สามารถเพิ่มหรือลดจำนวน Bank of Taps ได้ขณะที่วงจรกำลังทำงาน ซึ่งบางครั้งช่องสัญญาณมีการเปลี่ยนแปลงทำให้ความยาวของผลตอบสนองเชิงเวลาเปลี่ยนไปและส่งผลให้จำนวน Bank of Taps ของวงจรกรองที่ถูกกำหนดขึ้นในตอนเริ่มต้นของการทำงาน อาจมีค่ามากกว่าหรือน้อยกว่าความยาวของผลตอบสนองเชิงเวลาที่เปลี่ยนไป ทำให้วงจรต้องทำงานในส่วนที่ไม่จำเป็นและสูญเสียกำลังไป หรือไม่สามารถทำงานตามที่ต้องการได้



รูปที่ 3.3 สถาปัตยกรรมของโครงสร้างวงจรผสมมัลติเพล็กซ์และ Direct Form



รูปที่ 3.4 สถาปัตยกรรมทางฮาร์ดแวร์ของส่วน Adaptive FIR Filter



รูปที่ 3.5 สัญญาณนาฬิกาที่ใช้ภายในวงจรกรองแบบปรับอันดับ

3.2.2 สถาปัตยกรรมของวงจรกรองปรับตัวที่มีโครงสร้างผสมมัลติเพล็กซ์และ Direct Form FIR

สถาปัตยกรรมของวงจรกรองปรับตัวที่มีโครงสร้างผสมระหว่างมัลติเพล็กซ์และ Direct Form FIR สามารถแสดงได้ดังรูปที่ 3.3 ซึ่งจะเห็นว่า โครงสร้างสถาปัตยกรรมโดยรวมจะเป็นการต่อ Cascade กันของ Bank of Taps ในรูปที่แสดง จำนวน Bank of taps มีเท่ากับ 32 โดยสถาปัตยกรรมของวงจรทั้งหมดจะแบ่งเป็น 3 ส่วนคือ

1) Bank of Taps ภายในเป็นวงจรส่วนที่เรียกว่า Adaptive FIR Filter เป็นวงจรกรองแบบ LMS ที่มีโครงสร้างแบบมัลติเพล็กซ์ซึ่งประกอบด้วยวงจรกรองและส่วนปรับปรุงค่าสัมประสิทธิ์ สำหรับตัวอย่างในที่นี้ แต่ละ Bank of Taps จะประกอบด้วยแท็ปจำนวน 4 แท็ป

2) วงจรบวกแบบ Carry Propagate Adder ที่ทำหน้าที่หาผลรวมของสัญญาณขาออกที่ออกจากวงจรกรองใน Bank Of Taps แต่ละช่วง และให้ผลเป็นสัญญาณขาออกจริง ๆ ที่ออกจากวงจรกรอง

3) วงจร muu ซึ่งเป็นวงจรสำหรับหาผลคูณระหว่างค่าความผิดพลาดกับค่าช่วงก้าวเพื่อนำไปใช้ในการปรับปรุงค่าสัมประสิทธิ์

สำหรับ Bank of Taps แต่ละตัวจะมีโครงสร้างตามรูปที่ 3.4 โดยวงจรดังกล่าวจะทำหน้าที่จำลองผลตอบสนองเชิงเวลาในแต่ละช่วง ซึ่งมีหลักการทำงานคือ สัญญาณขาเข้า $u(n)$ จะถูกเก็บไว้ในรีจิสเตอร์ของส่วน Adaptive FIR Filter ที่อยู่ภายใน Bank of Taps และทำการคูณกับค่าสัมประสิทธิ์ของวงจรกรองในแต่ละช่วงภายใน Adaptive FIR Filter จากนั้นผลคูณดังกล่าวในแต่ละ Bank of Taps ก็จะทำการรวมกันกลายเป็นสัญญาณ y_tap ซึ่งส่งออกมาเป็นสัญญาณขาออกของวงจร Bank of Taps โดย y_tap ของทุก Bank of Taps จะรวมกันเป็นสัญญาณ $output(n)$ ซึ่งสัญญาณขาออกดังกล่าวจะถูกนำไปลบจากสัญญาณอ้างอิง $y(n)$ กลายเป็นค่าสัญญาณผิดพลาด $e(n)$ และนำค่าผลคูณระหว่าง $e(n)$ กับค่าช่วงก้าว muu ป้อนให้กับ Adaptive FIR Filter อีกครั้งเพื่อทำการปรับปรุงค่าสัมประสิทธิ์ของวงจรกรอง หลังจากปรับปรุงค่าสัมประสิทธิ์เสร็จแล้ว ก็จะทำกรเลื่อนสัญญาณขาเข้าสู่ Bank of Taps เพื่อทำการกรองสัญญาณต่อไป

ในส่วนของวงจรกรอง Adaptive FIR Filter ที่อยู่ใน Bank Of Taps เป็นโครงสร้างของวงจรปรับตัว LMS แบบมัลติเพล็กซ์ซึ่งมีโครงสร้างตามรูปที่ 3.4 ประกอบด้วยส่วนต่าง ๆ ดังนี้คือ

- 1) ส่วนของรีจิสเตอร์ที่ใช้เก็บค่าสัญญาณขาเข้าของวงจรกรองในแต่ละแท็ป กับรีจิสเตอร์ที่ใช้เก็บค่าสัมประสิทธิ์ของวงจรกรอง
- 2) วงจรคูณแบบ Radix-4 Booth Multiplier และวงจรบวกแบบ Carry Propagate Adder ที่ใช้ทั้งในส่วนของการคำนวณสัญญาณออกจากวงจรกรอง และคำนวณการปรับปรุงค่าสัมประสิทธิ์
- 3) รีจิสเตอร์ที่ใช้ในการสะสมค่าผลคูณในแต่ละแท็ปเพื่อหาสัญญาณออกจากวงจรกรอง
- 4) วงจรมัลติเพล็กซ์ ที่ใช้ในการเปลี่ยนโครงสร้างของวงจรกรองเป็นส่วนของการปรับปรุงค่าสัมประสิทธิ์

3.3 โครงสร้างทางสถาปัตยกรรมของวงจรต่าง ๆ ภายในวงจรกรองแบบปรับอันดับ

3.3.1 วงจรคูณแบบ Radix-4 Booth encoder (Radix-4 Booth encoder Multiplier)

วงจรคูณแบบ Radix-4 Booth encoder ในที่นี้เป็นวงจรคูณแบบ Serial Multiplier ดังแสดงในรูปที่ 3.6 โดยมีหลักการทำงานตามขั้นตอนต่อไปนี้

- 1) เริ่มแรกวงจรจะทำการโหลดค่าตัวตั้ง (Multiplicand : Mcand) กับตัวคูณ (Multiplier : Mier) เข้ามาเก็บไว้รีจิสเตอร์ โดยสัญญาณ Load_Multiplier จะมีค่าเป็น 1 แล้วจากนั้นจะมีขอบขาขึ้นของ Load_Shift เพื่อโหลดค่า Mier เข้าไปเก็บไว้ในรีจิสเตอร์ของ Product ขณะที่พร้อมกันนี้รีจิสเตอร์ของ Product ตัวที่เหลือจะถูกรีเซ็ตให้มีค่าเป็น '0'
- 2) จากนั้นวงจรจะพิจารณาค่า 3 บิตสุดท้ายของผลคูณ (Product) ซึ่งแทนด้วย m_{i+1} , m_i , m_{i-1} โดยพิจารณาจากตารางที่ 3.1 ถ้าตรงกับเงื่อนไขดังต่อไปนี้จะทำงานในขั้นตอนต่อไปคือ
 - 1) ถ้ามีค่าเป็น '000' หรือ '111' ทำการเลื่อน Product ไปทางขวาเป็นจำนวน 2 บิต โดยสัญญาณ Add/Sub กับ Load_Multiplier จะมีค่าเป็น '0' และมีขอบขาขึ้นของสัญญาณ Shift 2 ครั้งเพื่อทำการเลื่อน Product ไปทางขวา 2 บิต
 - 2) ถ้ามีค่าเป็น '001' หรือ '010' ทำการบวก Mcand เข้ากับ Product แล้วเลื่อน Product ไปทางขวาเป็นจำนวน 2 บิต โดยสัญญาณ Add/Sub จะมีค่าเป็น '1' และ Cm มีค่าเป็น '0' เพื่อทำการบวกค่า Mcand เข้ากับ Product หลังจากทำการบวกเรียบร้อยแล้ว ก็จะมีขอบขาขึ้นสัญญาณ Load เพื่อทำการเก็บค่าผลบวกดัง

กล่าว จากนั้นสัญญาณ Add/Sub จะมีค่าเป็น '0' และมีขอบขาขึ้น 2 ครั้งของสัญญาณ Shift เพื่อทำการเลื่อน Product ไปทางขวา 2 บิต

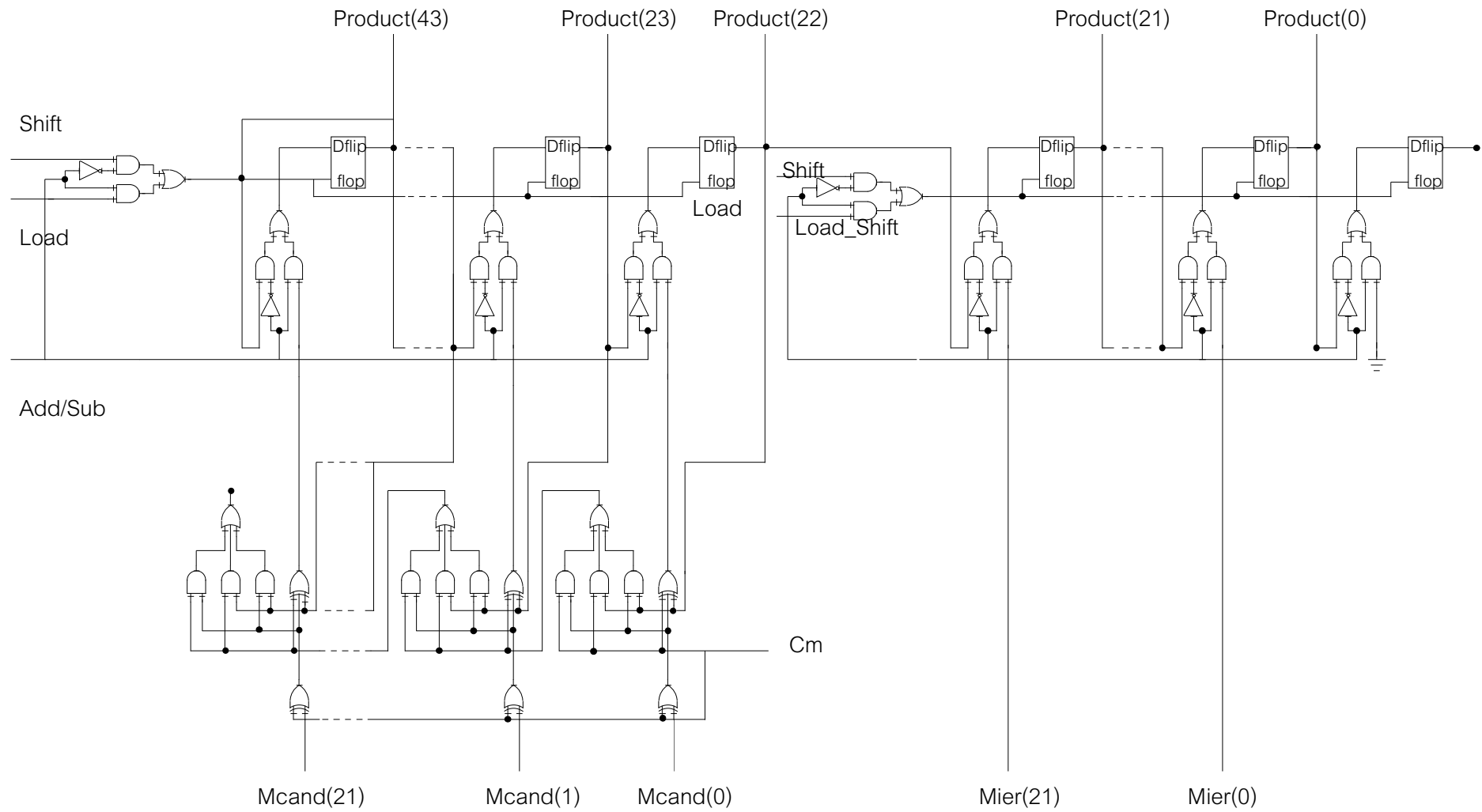
- 3) มีค่าเป็น '101' หรือ '110' ทำการลบ Mcand ออกจาก Product แล้วเลื่อน Product ไปทางขวาเป็นจำนวน 2 บิต โดยสัญญาณ Add/Sub จะมีค่าเป็น '1' และ Cm มีค่าเป็น '1' เพื่อทำการลบค่า Mcand ออกจาก Product หลังจากที่ทำ การบวกเรียบร้อยแล้ว ก็จะมีขอบขาขึ้นสัญญาณ Load เพื่อทำการเก็บค่าผลบวกดัง กล่าว จากนั้นสัญญาณ Add/Sub จะมีค่าเป็น '0' และมีขอบขาขึ้น 2 ครั้งของ สัญญาณ Shift เพื่อทำการเลื่อน Product ไปทางขวา 2 บิต
- 4) มีค่าเป็น '011' ทำการเลื่อน Product ไปทางขวาเป็นจำนวน 1 บิต แล้วบวกค่า Mcand เข้ากับ Product แล้วเลื่อน Product ไปทางขวาเป็นจำนวน 1 บิต โดย สัญญาณ Add/Sub จะมีค่าเป็น '0' แล้วมีขอบขาขึ้นของสัญญาณ Shift 1 ครั้ง เพื่อทำการเลื่อน Product ไปทางขวา 1 บิต จากนั้นสัญญาณ Add/Sub จะมีค่า เป็น '1' และสัญญาณ Cm มีค่าเป็น '0' เพื่อทำการบวกค่า Mcand เข้ากับ Product หลังจากที่ทำ การบวกเรียบร้อยแล้ว ก็จะมีขอบขาขึ้นของสัญญาณ Load เพื่อทำการเก็บค่าผลบวกดังกล่าว จากนั้นสัญญาณ Add/Sub จะมีค่าเป็น '0' และมีขอบขาขึ้นของสัญญาณ Shift เพื่อทำการเลื่อน Product ไปทางขวา 1 บิต
- 5) มีค่าเป็น '100' ทำการเลื่อน Product ไปทางขวาเป็นจำนวน 1 บิต แล้วลบค่า Mcand ออกจาก Product แล้วเลื่อน Product ไปทางขวาเป็นจำนวน 1 บิตโดย สัญญาณ Add/Sub จะมีค่าเป็น '0' แล้วมีขอบขาขึ้นของสัญญาณ Shift 1 ครั้ง เพื่อทำการเลื่อน Product ไปทางขวา 1 บิต จากนั้นสัญญาณ Add/Sub จะมีค่า เป็น '1' และสัญญาณ Cm มีค่าเป็น '1' เพื่อทำการลบค่า Mcand จาก Product หลังจากที่ทำ การลบเรียบร้อยแล้ว ก็จะมีขอบขาขึ้นของสัญญาณ Load เพื่อทำ การเก็บค่าดังกล่าว จากนั้นสัญญาณ Add/Sub จะมีค่าเป็น '0' และมีขอบขาขึ้น ของสัญญาณ Shift เพื่อทำการเลื่อน Product ไปทางขวา 1 บิต

- 3) กลับไปทำขั้นตอนที่ 2 จนครบ เป็นจำนวนครั้งหนึ่งครั้งของจำนวนบิตของ Mcand ก็จะได้ค่าผลคูณที่ทำการคูณเสร็จแล้ว

วงจรคูณแบบ Radix-4 Booth encoder แบบนี้ใช้พื้นที่ของฮาร์ดแวร์น้อยกว่าวงจรคูณแบบอะเรย์ และมีคาบการทำงานที่น้อยกว่าวงจรคูณแบบ Booth encoder แบบธรรมดา

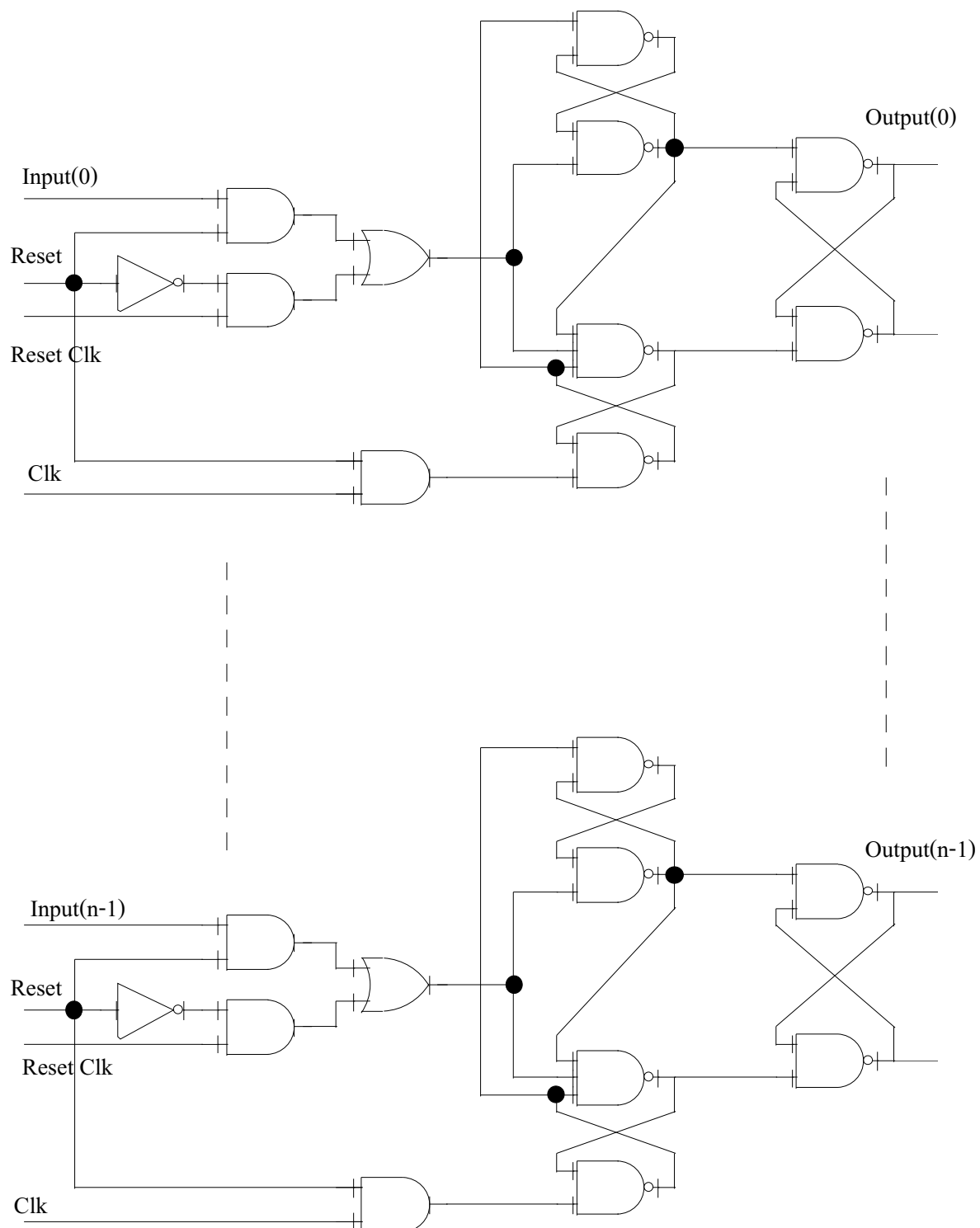
m_{i+1}	m_i	m_{i-1}	Status	Action
0	0	0	String of 0s	Shift by 2
0	0	1	End of string of 1s	Add
0	1	0	Single1	Add
0	1	1	End of string of 1s	Shift by 1, Add, Shift by 1
1	0	0	Begin string of 1s	Shift by 1, Subtract, Shift by 1
1	0	1	Single0	Subtract
1	1	0	Begin string of 1s	Subtract
1	1	1	Midstring of 1s	Shift by 2

ตารางที่ 3.6 ขั้นตอนการทำงานของวงจร Radix-4 Booth encoder



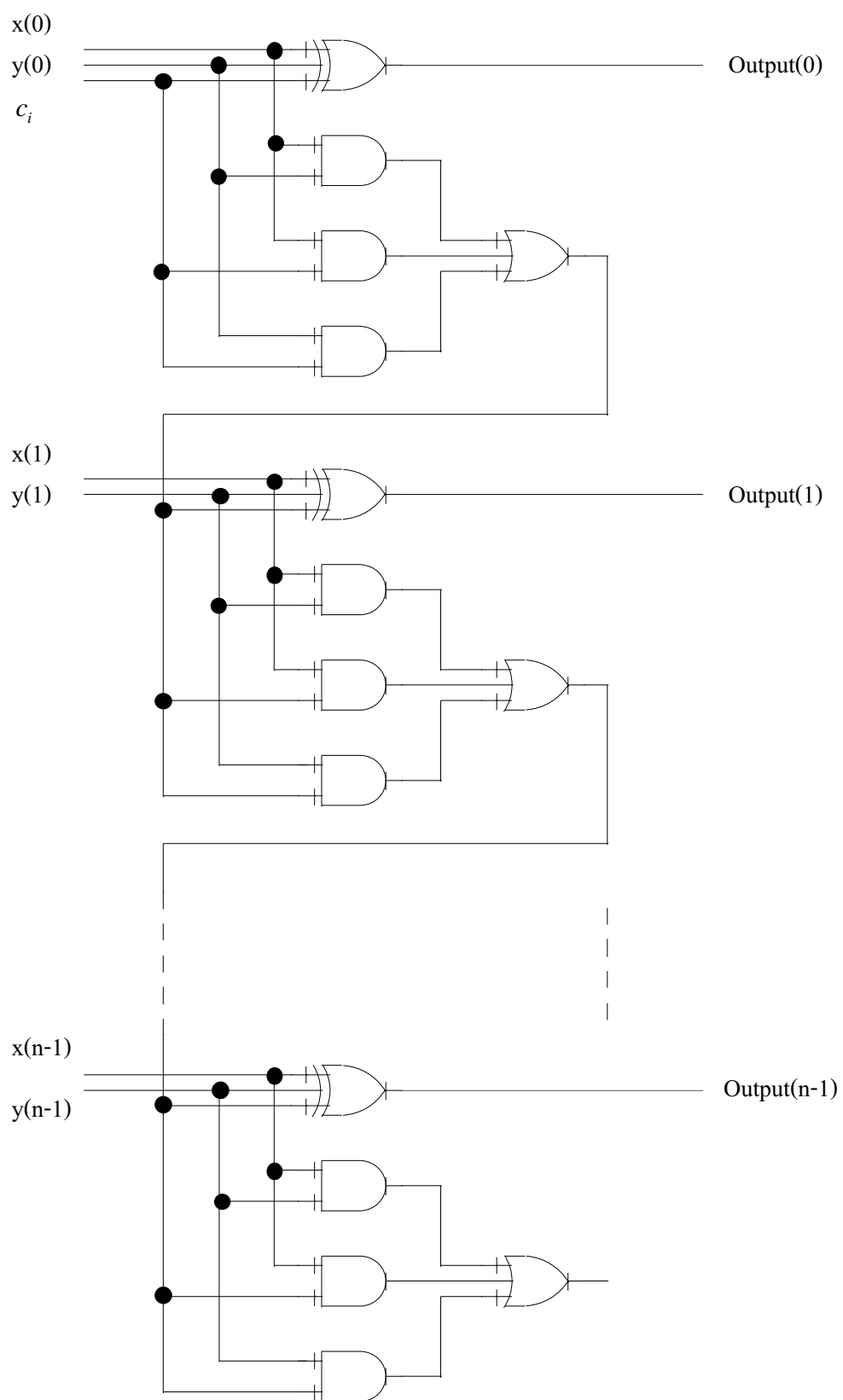
รูปที่ 3.6 สถาปัตยกรรมของวงจร Radix-4 Booth Multiplier

3.3.2 วงจร Synchronous Reset D flip-flop n บิต



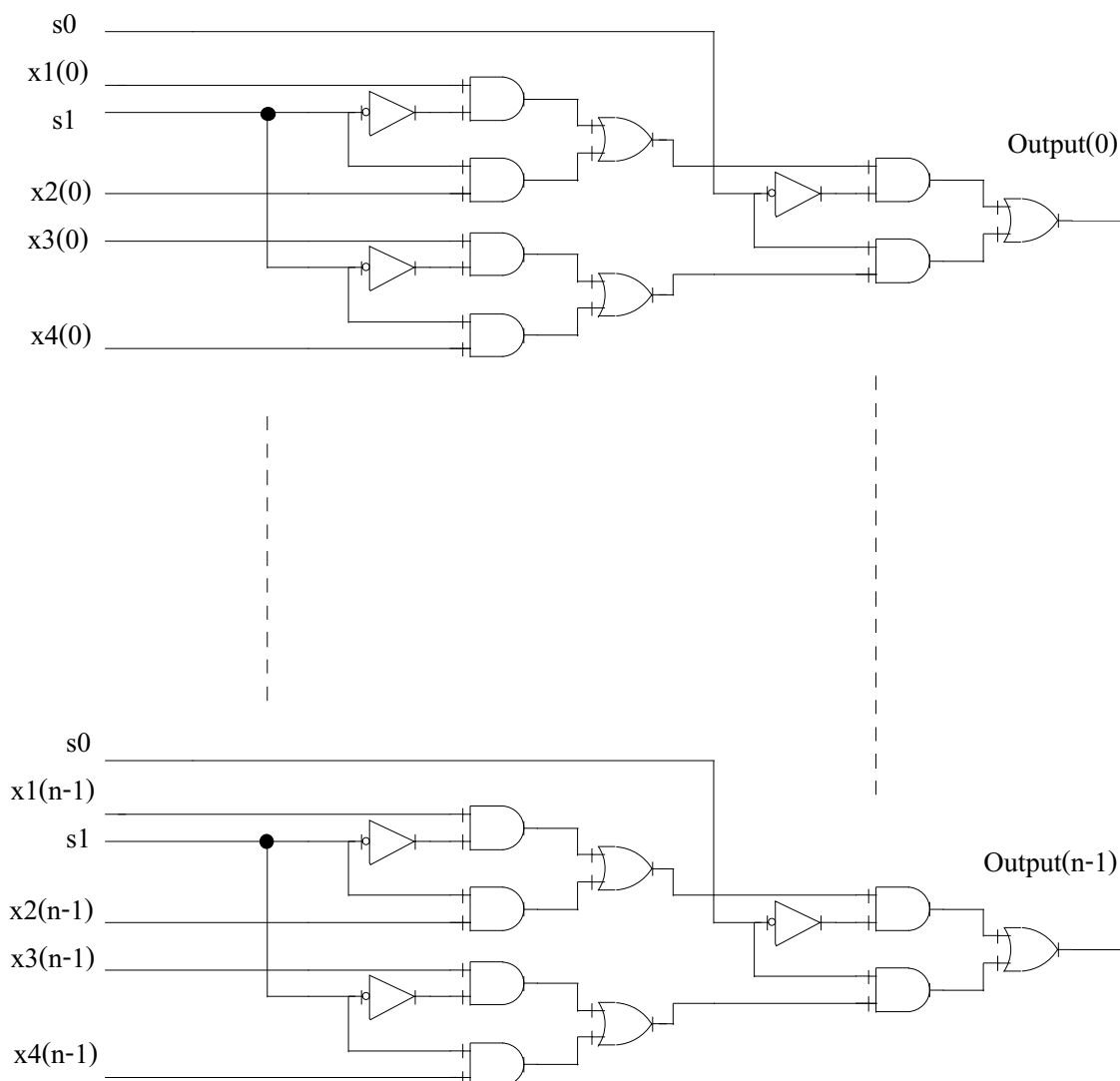
รูปที่ 3.7 สถาปัตยกรรมของวงจร Synchronous Reset D flip-flop n บิต

3.3.3 วงจรบวกแบบ Carry Propagate Adder n บิต



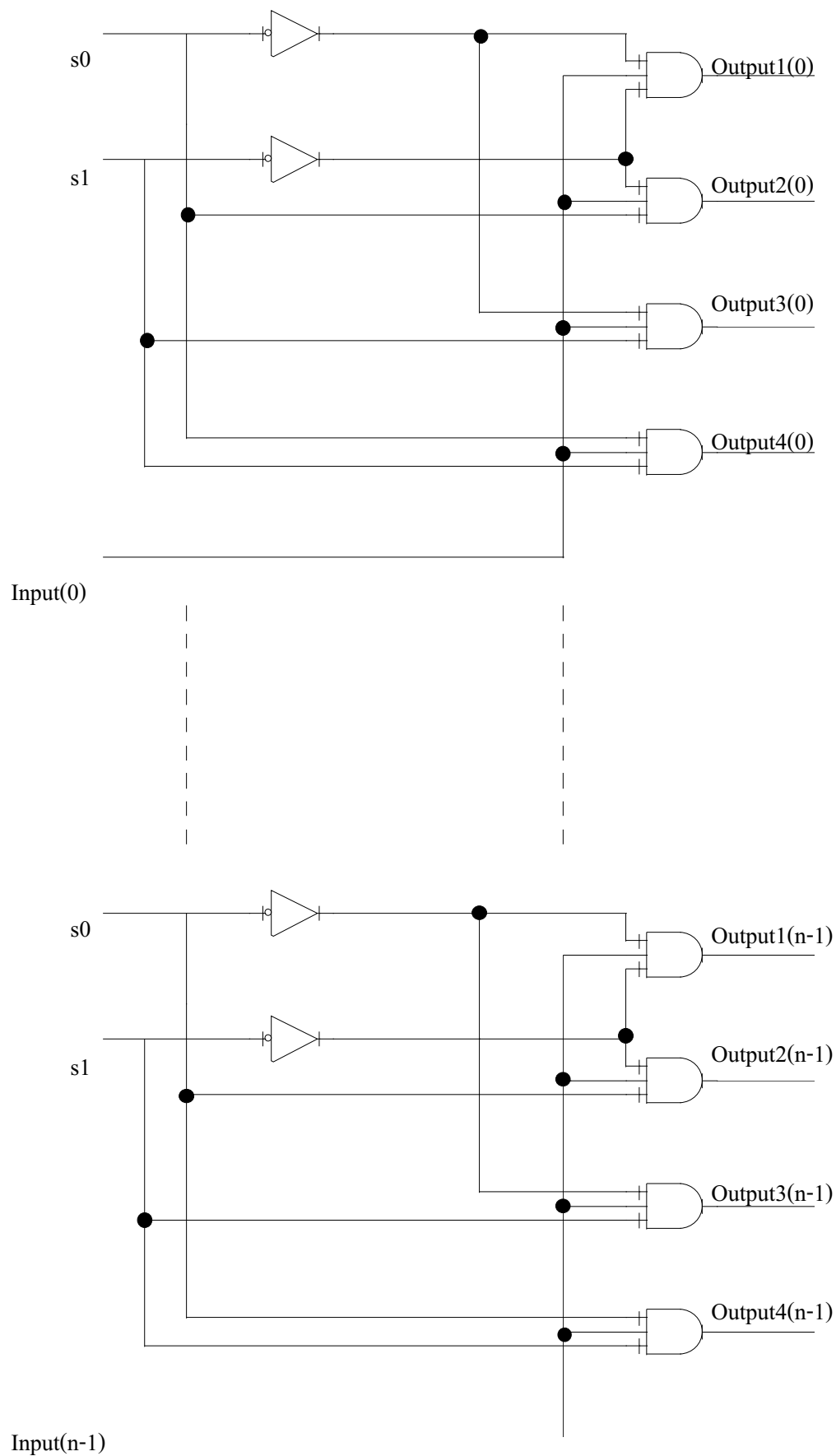
รูปที่ 3.8 สถาปัตยกรรมของวงจรบวกแบบ Carry Propagate Adder n บิต

3.3.4 วงจร Multiplexer 4 to 1 n บิต



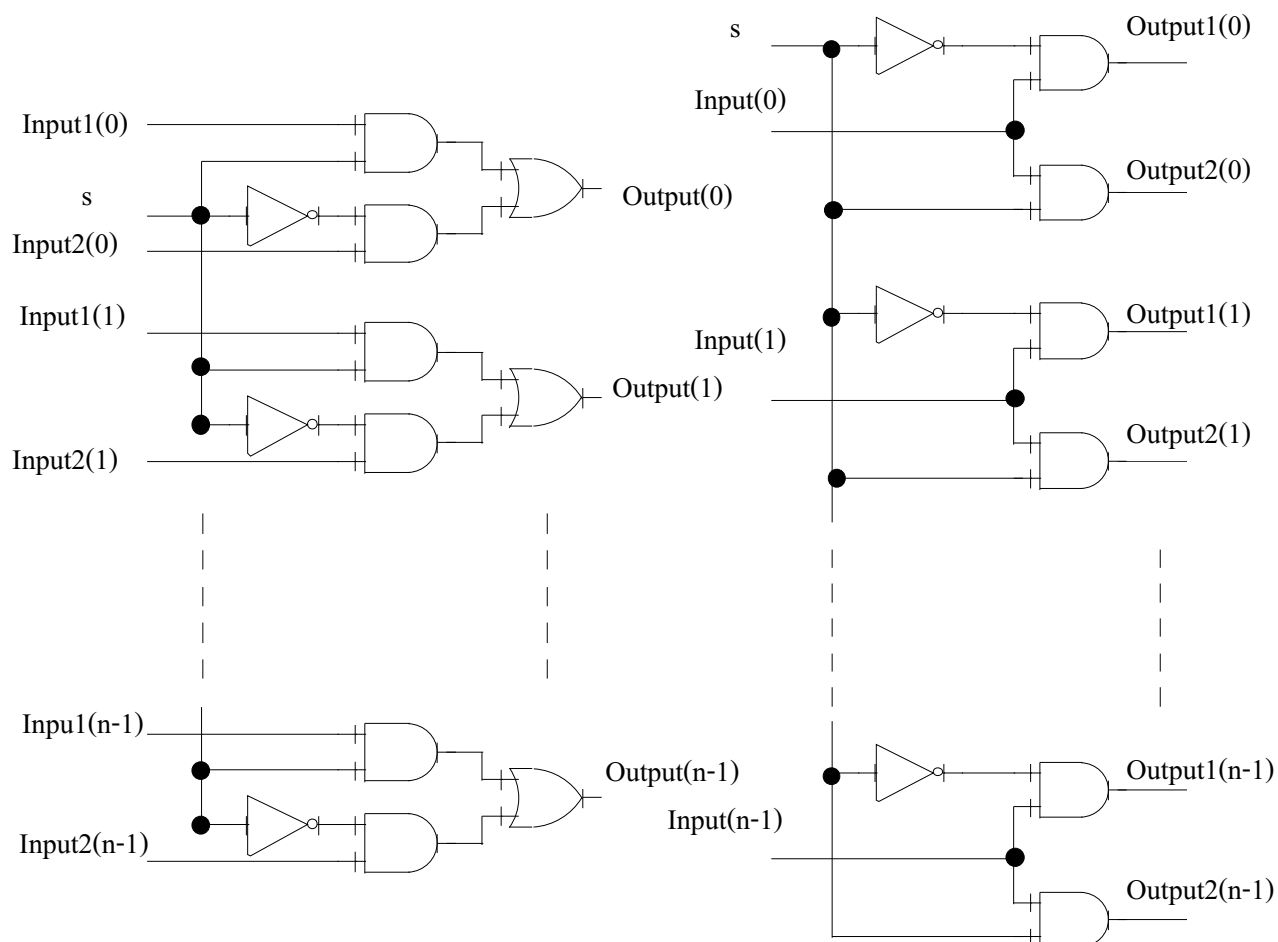
รูปที่ 3.9 สถาปัตยกรรมของวงจร Multiplexer 4 to 1 n บิต

3.3.5 วงจร Demultiplexer 1 to 4 n บิต



รูปที่ 3.10 สถาปัตยกรรมของวงจร Demultiplexer 1 to 4 n บิต

3.3.6 วงจร Demultiplexer 1 to 2 และวงจร Multiplexer 2 to 1 n บิต



รูปที่ 3.11 สถาปัตยกรรมของวงจร Demultiplexer 1 to 2 และวงจร Multiplexer 2 to 1 n บิต

บทที่ 4

วงจรกรองแบบปรับอันดับ

บทนี้เป็นการอธิบายถึงวงจรกรองแบบปรับอันดับ หัวข้อ 4.1 อธิบายขั้นตอนการทำงานของวงจรกรองแบบปรับอันดับ หัวข้อ 4.2 กล่าวถึงวงจรกรองแบบปรับอันดับ หัวข้อ 4.2.1 และหัวข้อ 4.2.2 อธิบายหลักการทำงาน และโครงสร้างของวงจรกรองแบบปรับอันดับ ตามลำดับ สุดท้ายหัวข้อ 4.3 อธิบายถึงการแทนตัวเลขของสัญญาณต่าง ๆ ภายในวงจร

เนื่องจากในเครื่องช่วยฟังซึ่งเป็นอุปกรณ์พกพานั้นต้องใช้พลังงานจากแบตเตอรี่ซึ่งมีพลังงานจำกัด ดังนั้นหากนำวงจรกรองแบบที่ไม่มีการปรับอันดับดังที่กล่าวไปในบทที่ 3 ไปใช้อาจทำให้สูญเสียพลังงานในส่วนที่ไม่จำเป็นรวมถึงอาจก่อให้เกิดปัญหาเรื่องความร้อนได้ ดังนั้นเพื่อให้การทำงานของวงจรมีความยืดหยุ่นตามลักษณะการเปลี่ยนแปลงของช่องสัญญาณรวมถึงลดปัญหาเรื่องการใช้พลังงาน ในโครงการวิจัยนี้จึงได้เพิ่มกลไกที่สามารถเพิ่มหรือลดอันดับของวงจรกรองได้ในระหว่างที่วงจรกรองยังคงทำงาน โดยนำระเบียบวิธี LMS Estimation via Structural Detection ที่ได้กล่าวไว้แล้วในบทที่ 2 มาประยุกต์ใช้ในการเพิ่มหรือลดอันดับของวงจรกรอง

4.1 ขั้นตอนการทำงานของวงจรกรองแบบปรับอันดับ

การทำงานของวงจรกรองปรับตัวของเครื่องช่วยฟังในที่นี้จะปรับแบบปรับตัวไม่ต่อเนื่อง คือจะปรับตัวเฉพาะตอนเสียงเงียบหรือสัญญาณเสียงขาเข้ามีกำลังต่ำเท่านั้น ถึงแม้การปรับตัวแบบไม่ต่อเนื่องแบบนี้มีข้อเสียคือไม่สามารถติดตามการเปลี่ยนแปลงของช่องสัญญาณป้อนกลับได้ตลอดเวลาเหมือนกรณีการปรับตัวต่อเนื่องแต่การปรับตัวแบบต่อเนื่องจะมีเสถียรภาพดีกว่าในกรณีของข้อจำกัดทางด้านการติดตามการเปลี่ยนแปลง ดังที่ได้กล่าวไว้แล้วในหัวข้อ 3.1 ว่าวิธีการปรับตัวแบบไม่ต่อเนื่องนี้มีข้อดีคือให้ความสมดุลในเรื่องความมีเสถียรภาพของการปรับตัวของวงจรกรอง กับประสิทธิภาพการทำงานของวงจรกรอง ส่วนในเรื่องความไม่ต่อเนื่องของการติดตามการเปลี่ยนแปลงของช่องสัญญาณนั้นเราสามารถแก้ไขได้โดยการเพิ่มความถี่ของการปรับค่าพารามิเตอร์

โดยในช่วงที่กำลังของสัญญาณขาเข้าของเครื่องช่วยฟังมีค่าต่ำ เครื่องช่วยฟังจะตัดวงจรเพื่อให้วงจรกรองทำการปรับตัวตามเส้นทางป้อนกลับ ในที่นี้ผลตอบสนองเชิงเวลาของช่องสัญญาณป้อนกลับทางเสียง $h(n)$ ที่กำลังพิจารณานั้นมีค่าขนาดสูงสุดอยู่ที่ 0.12 มีคิเลย์ของช่องสัญญาณเท่ากับ 50 และมีผลตอบสนองเชิงเวลาที่ไม่เป็นศูนย์เฉพาะช่วง $n = 51$ ถึง 83 เมื่อนำวง

จรกรองแบบปรับอันดับที่มีโครงสร้างแบบมัลติเพล็กซ์ ซึ่งสร้างให้มีอันดับเท่ากับ 128 โดยแบ่งเป็น Bank of Taps ภายในแต่ละ Bank of Taps มีจำนวน 4 แทป มาใช้กับช่องสัญญาณดังกล่าว

โดยอัตราการสุ่มเข้านี้สำหรับกรณีวงจรกรอง LMS และสัญญาณรบกวนกับสัญญาณขาเข้าของวงจรที่มีลักษณะเป็นสัญญาณรบกวนสีขาวจะให้อัตราการสุ่มเข้าตามสมการ

$$C = 2\mu\sigma_u^2 \quad (4.1)$$

โดยที่

C แทนอัตราการสุ่มเข้าของวงจร

μ แทนช่วงก้าวในการปรับค่าพารามิเตอร์

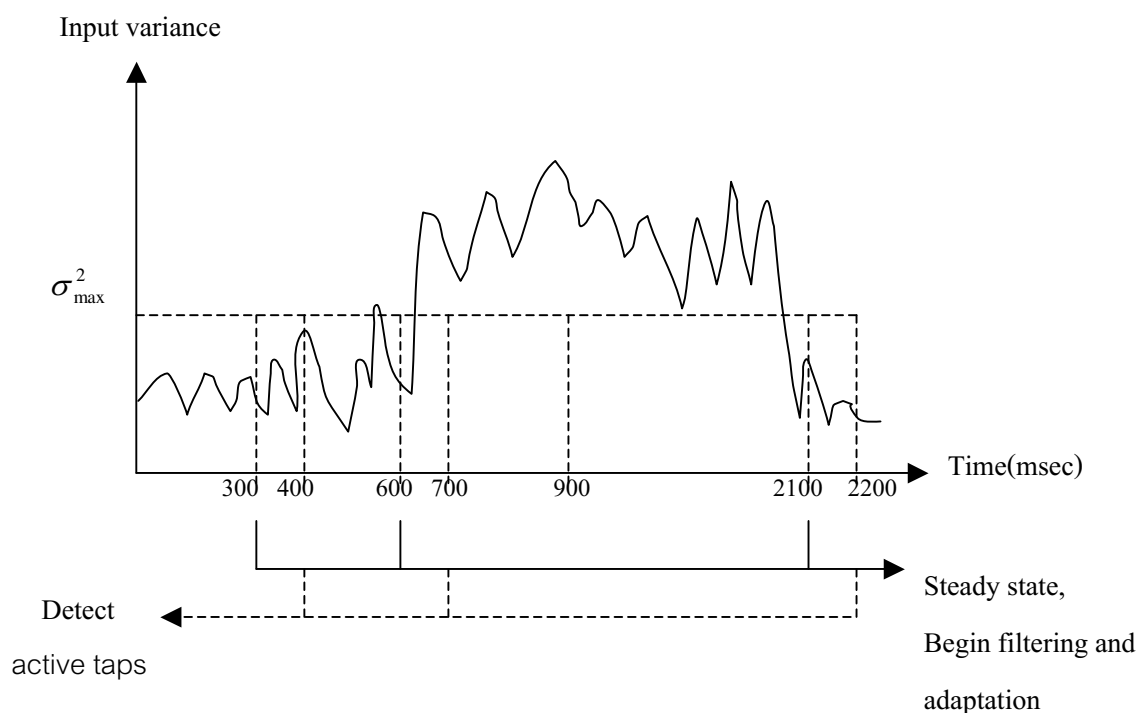
σ_u^2 แทนกำลังของสัญญาณขาเข้าของวงจรกรอง

จากสมการที่ (4.1) จะเห็นว่าอัตราการสุ่มเข้าของวงจรมีค่าขึ้นอยู่กับกำลังของสัญญาณขาเข้า เนื่องจากในที่นี้สัญญาณขาเข้าของวงจรคือสัญญาณสุ่ม N_p ซึ่งถูกกำหนดให้กำลังของสัญญาณคงที่เป็น 0.01 ดังนั้นไม่ว่าสัญญาณรบกวนรวมถึงช่องสัญญาณมีการเปลี่ยนแปลงแต่ก็ไม่ส่งผลต่ออัตราการสุ่มเข้าของวงจรเนื่องจากเรากำหนดให้กำลังของสัญญาณขาเข้ามีค่าคงที่ จากผลการจำลองการทำงานปรากฏว่า วงจรกรองจะเข้าสู่สภาวะอยู่ตัวเมื่อป้อนสัญญาณตัวอย่างเป็นจำนวน 15000 ตัวอย่าง เนื่องจากวงจรกรองที่นำมาใช้นี้มีอัตราการสุ่มสัญญาณ 50 kHz ดังนั้นวงจรกรองจะเข้าสู่สภาวะอยู่ตัวพร้อมสำหรับการนำไปใช้ขยายสัญญาณเมื่อวงจรกรองทำการปรับตัวผ่านไปแล้วเป็นเวลา 0.3 วินาทีนับตั้งแต่เริ่มต้นปรับตัว

นอกจากนี้ในส่วนที่เป็นวงจรตรวจสอบแอดิทิฟแท็ปจะสามารถตรวจสอบตำแหน่งของแอดิทิฟแท็ปได้ถูกต้องเมื่อป้อนสัญญาณตัวอย่างเป็นจำนวน 5000 ตัวอย่าง ดังนั้นวงจรกรองพร้อมที่จะหยุดการปรับตัวของแท็ปที่ไม่แอดิทิฟและวงจรตรวจสอบแอดิทิฟแท็ป เมื่อเวลาผ่านไปแล้วเป็นเวลา 0.1 วินาทีนับตั้งแต่เริ่มต้นปรับตัว โดยเวลาที่เหมาะสมสำหรับการตรวจสอบแอดิทิฟแท็ปนี้จะขึ้นอยู่กับค่า SNR (Signal to Noise Ratio) ของวงจรกรองโดยถ้าค่า SNR มีค่าไม่เปลี่ยนแปลงเวลาการตรวจสอบแอดิทิฟแท็ปก็จะไม่เปลี่ยนแปลง เนื่องจากเงื่อนไขการทำงานของวงจรกรองในที่นี้จะทำงานเมื่อกำลังของสัญญาณขาเข้าของเครื่องช่วยฟังมีค่าน้อยกว่า 0.01 และกำลังของสัญญาณสุ่มที่ป้อนให้กับวงจรกรองถูกกำหนดให้มีค่าคงที่ ดังนั้นเวลา 0.1 วินาทีที่ใช้ในการตรวจสอบแอดิทิฟแท็ปจึงเป็นกรณีที่แย่ที่สุดในการตรวจสอบแอดิทิฟแท็ป เนื่องจากยังกำลังของ

สัญญาณขาเข้าของเครื่องช่วยฟังมีค่าน้อยเท่าไร ก็จะทำให้ค่า SNR เพิ่มขึ้นทำให้เวลาที่ทำการตรวจสอบแอ็คทีฟแท็ปจะเร็วขึ้นไปด้วย

จากลักษณะดังกล่าวเราจะให้วงจรกรองทำการกรองสัญญาณและปรับตัวเป็นรายคาบในทุก ๆ 0.3 วินาที เพื่อให้วงจรกรองมีความพร้อมที่สุดสำหรับช่องสัญญาณเสียงป้อนกลับที่จะนำไปใช้งานในขณะนั้นหลักการทำงานคือวงจรกรองจะทำการปรับตัวเฉพาะตอนที่ยังไม่เกิดเสียงหอนหรือตอนที่เสียงเบา โดยในตอนเริ่มแรกวงจรกรองจะตรวจสอบกำลังของสัญญาณขาเข้า ถ้าขณะนั้นยังไม่เกิดเสียงหอนหรือเสียงเบา สัญญาณเสียงขาเข้าจะมีกำลังต่ำกว่าที่กำหนดไว้คือ $\sigma_{\max}^2$ ซึ่งในที่นี้ให้มีค่าเท่ากับ 0.1 เท่าของสัญญาณที่มีขนาดสูงสุด วงจรกรองก็จะเริ่มทำการกรองสัญญาณและปรับตัวพร้อมกันนั้นวงจรตรวจสอบแอ็คทีฟแท็ปก็จะทำการเก็บค่าสถิติของแต่ละแท็ปไปด้วยโดยการปรับตัวในช่วงแรกจะมีจำนวนแท็ปเท่ากับจำนวนแท็ปทั้งหมดจากนั้นเมื่อเวลาผ่านไป 0.1 วินาทีวงจรก็จะทำการตรวจสอบและตัดแท็ปที่ไม่แอ็คทีฟออกไปต่อไปจะใช้เฉพาะแท็ปที่เหลือทำการปรับตัวและกรองสัญญาณจนวงจรเข้าสู่สภาวะอยู่ตัวเมื่อเวลาผ่านไป 0.3 วินาที



รูปที่ 4.1 การทำงานในแต่ละช่วงของวงจรกรองปรับตัวที่ขึ้นอยู่กับกำลังของสัญญาณขาเข้า

เพื่อให้เห็นภาพลำดับขั้นตอนการทำงานของวงจรจะขอยกตัวอย่างรูปที่ 4.1 ซึ่งแสดงความสัมพันธ์ระหว่างกำลังของสัญญาณเสียงขาเข้ากับช่วงเวลาการทำงานของวงจรกรอง สมมติว่าวงจรเริ่มทำงาน ณ เวลา 300 msec ซึ่งขณะนั้นสัญญาณขาเข้ามีกำลังต่ำกว่า $\sigma_{\max}^2$ วงจรกรองตรวจสอบพบว่ากำลังของสัญญาณขาเข้าน้อยกว่าที่กำหนดไว้ก็จะทำการกรองสัญญาณและทำการปรับตัวพร้อมกับส่วนของวงจรตรวจสอบแอมพลิจูดที่เพิ่งทำงานไปเรื่อย ๆ จนถึงเวลา 400 msec เมื่อถึงเวลานี้วงจรกรองก็จะตัดแท็ปที่ไม่แอมพลิจูดพอกรวมถึงหยุดการทำงานของวงจรตรวจสอบแอมพลิจูดที่เพิ่ง จากนั้นจะทำการกรองสัญญาณและปรับตัวในครั้งต่อไปด้วยแท็ปที่แอมพลิจูดเท่านั้น เมื่อวงจรกรองทำการปรับตัวและเข้าสู่สถานะอยู่ตัว ณ เวลา 600 msec ก็จะหยุดการปรับตัวแล้วทำการตรวจสอบอีกครั้งว่าขณะนั้นกำลังของสัญญาณขาเข้ามีค่าสูงกว่า $\sigma_{\max}^2$ หรือไม่ถ้ายังต่ำกว่า ทุก 128 แท็ปของวงจรกรองก็จะเริ่มทำการกรองสัญญาณและปรับตัวใหม่อีกครั้ง แต่หากกำลังของสัญญาณขาเข้ามีค่าสูงกว่า $\sigma_{\max}^2$ ก็จะนำวงจรกรองในเฉพาะแท็ปที่เหลือที่ได้จากการปรับตัวในคาบที่ผ่านมาไปใช้ในการกรองสัญญาณป้อนกลับเลยโดยวงจรกรองไม่มีการปรับตัวต่อไปอีกซึ่ง ณ เวลา 600 msec จะเห็นว่ากำลังของสัญญาณเสียงขาเข้ายังต่ำกว่าค่าที่กำหนดอยู่ดังนั้น วงจรกรองจะทำการกรองสัญญาณและปรับตัวเหมือนคาบเวลาก่อนหน้านี้อีกครั้ง หลังจากนั้น ณ เวลา 900 msec วงจรกรองจะทำการตรวจสอบกำลังของสัญญาณขาเข้าอีกครั้งซึ่งจะเห็นว่ากำลังของสัญญาณเสียงขาเข้ามีค่ามากกว่าที่กำหนดไว้ดังนั้นจึงไม่มีการปรับตัว และนำเฉพาะแท็ปที่แอมพลิจูดที่เหลือจากการตรวจสอบในคาบที่ผ่านมาไปใช้ในการกรองสัญญาณอย่างเดียวในตลอดช่วงเวลา 900 ถึง 2100 msec ณ เวลา 2100 msec ปรากฏว่าสัญญาณขาเข้ามีค่าต่ำกว่า $\sigma_{\max}^2$ วงจรกรองก็จะกลับมาใช้ทุก 128 แท็ปเพื่อทำการปรับตัวและกรองสัญญาณอีกครั้ง

4.2 วงจรกรองปรับตัว LMS แบบผสมมัลติเพล็กซ์และ Direct Form ที่มีการปรับอันดับ

4.2.1 ขั้นตอนการทำงานของวงจรกรองปรับตัวแบบผสมมัลติเพล็กซ์และ Direct Form ที่มีการปรับอันดับ

วงจรกรองปรับตัวแบบผสมมัลติเพล็กซ์และ Direct Form ที่มีการปรับอันดับแบ่งขั้นตอนการทำงานออกเป็นสองช่วงคือ เริ่มแรกวงจรกรองจะทำการกรองสัญญาณและทำการปรับปรุค่าพารามิเตอร์ รวมถึงคำนวณค่า Threshold และค่าสถิติของแต่ละแท็ปเก็บไว้เพื่อทำการตรวจสอบแอมพลิจูดที่เพิ่งในขั้นตอนที่สอง จากนั้นในขั้นตอนที่สองวงจรกรองจะทำการตรวจสอบแอมพลิจูดที่เพิ่งโดยนำค่าสถิติของแต่ละแท็ปที่ได้จากการทำงานในช่วงแรกมาเปรียบเทียบกับค่า Threshold และตัดสินใจว่าแท็ปไหนที่มีค่าแอมพลิจูด โดยหลังจากนี้ในรอบต่อไปของการทำงานของวงจรกรองก็จะ

ใช้เพียงแค่แท็ปที่แอ็คทีฟเท่านั้นในการกรองสัญญาณและปรับปรุงค่าพารามิเตอร์ พิจารณา รูปที่ 3.5 และ รูปที่ 4.2 ถึง 4.5

การทำงานในช่วงแรกสัญญาณ Detect_clk และ Sequential_detect_clk ยังมีค่าเป็นศูนย์ วงจรกรองในส่วน Adaptive FIR Filter จะทำงานปรับตัวแบบ LMS ดังที่ได้กล่าวมาแล้ว ซึ่งขณะเดียวกันส่วน วงจร Threshold ก็จะคำนวณค่า Threshold ณ รอบการทำงานนั้นออกมา และในส่วนวงจร Tap Statistics ที่อยู่ภายในวงจร Detect Active Tap นั้นสัญญาณ S0,S1 จะถูกป้อนให้กับมัลติเพล็กซ์เซอร์เพื่อเลือก $u(n)$ ของแต่ละแท็ปในซึ่งในรูปคือสัญญาณ u_to_detect มาคูณกับ $y(n)$ และนำมารวมกับค่าสถิติ $f(n)$ ของแต่ละแท็ปเดิมเพื่อทำการหาค่าสถิติ $f(n)$ ของแต่ละแท็ปค่าใหม่ โดยค่าของ $f(n)$ ของแต่ละแท็ปเมื่อคำนวณเสร็จก็จะถูกเลื่อนไปเก็บไว้ในรีจิสเตอร์ของแต่ละแท็ปเมื่อมีขอบขาขึ้นของสัญญาณ h1, h2, h3, h4 ตามลำดับ ในรอบทำงานต่อไปก็จะทำการเก็บค่าสถิติในแต่ละแท็ปไว้ในลักษณะเดียวกันเพื่อนำค่าสถิติ $f(n)$ มาใช้ในการตรวจสอบแอ็คทีฟแท็ปในภายหลัง

จากนั้นเมื่อเวลาผ่านไประยะหนึ่งเมื่อวงจรเข้าสู่สภาวะอยู่ตัวแล้ว สัญญาณ Detect_clk จะมีค่าเป็น 1 วงจร Detect Active จะทำการตรวจสอบค่าสัมประสิทธิ์จะทำการตรวจสอบหาสัมประสิทธิ์ของวงจรกรองว่าสัมประสิทธิ์ตัวไหนบ้างที่ควรมีค่าไม่เป็นศูนย์โดยการเปรียบเทียบค่าสถิติ $f^2(n)$ ที่คำนวณได้จากวงจร Tap Statistics ของแต่ละแท็ปกับค่า Threshold จากนั้น ณ.แต่ละครั้งของขอบขาขึ้นของสัญญาณ Sequential_detect_clk เมื่อตรวจสอบพบว่าแท็ปนั้นแอ็คทีฟก็จะส่งสัญญาณ 1 ให้กับส่วนของวงจร Sequential Detect Active ทำการตรวจสอบภายใต้เงื่อนไขที่ว่า Bank of Taps ใดที่ประกอบด้วยสัมประสิทธิ์ที่ไม่แอ็คทีฟทั้งหมดค่าของสัญญาณ Detect_active ที่ออกจากวงจร Detect Active Tap จะมีค่าเป็นศูนย์ จากนั้นจะนำสัญญาณ Detect_active ไปทำให้สัญญาณขาออกของ Bank of Taps มีค่าเป็นศูนย์ เพราะถือว่าส่วนนี้ไม่มีผลต่อสัญญาณขาออกของวงจรกรอง จากนั้นในรอบต่อไป Adaptive FIR Filter ก็จะคำนวณในส่วน Bank of Taps ที่ประกอบด้วยสัมประสิทธิ์ตัวที่แอ็คทีฟเท่านั้น

4.2.2 สถาปัตยกรรมของวงจรกรองปรับตัวที่มีโครงสร้างผสมมัลติเพล็กซ์และ Direct Form ที่มีการปรับอันดับ

สถาปัตยกรรมของวงจรกรองปรับตัวที่มีโครงสร้างผสมมัลติเพล็กซ์และ Direct Form ที่มีการปรับอันดับที่ประยุกต์มาจากระเบียบวิธี LMS Estimation via Structural Detection สามารถแสดงได้ดังรูปที่ 4.2 โดยในส่วนโครงสร้างภายในของวงจรกรองแบบที่มีการปรับอันดับนี้จะประกอบด้วย 2 ส่วนคือ ส่วนแรกเป็นส่วนของวงจร Adaptive FIR Filter ที่อยู่ใน Bank Of Tap

เป็นโครงสร้างของวงจรปรับตัว LMS แบบผสมมัลติเพล็กซ์และ Direct Form เหมือนกับในบทที่ 3 และส่วนที่สองคือ ส่วนของวงจร Detect Active Tap ใช้ในการตรวจสอบแอมพลิจูดที่เก็บ โดยโครงสร้างหลักของวงจรทั้งหมดดังในรูปที่ 4.2 ประกอบด้วยส่วนต่าง ๆ ดังนี้คือ

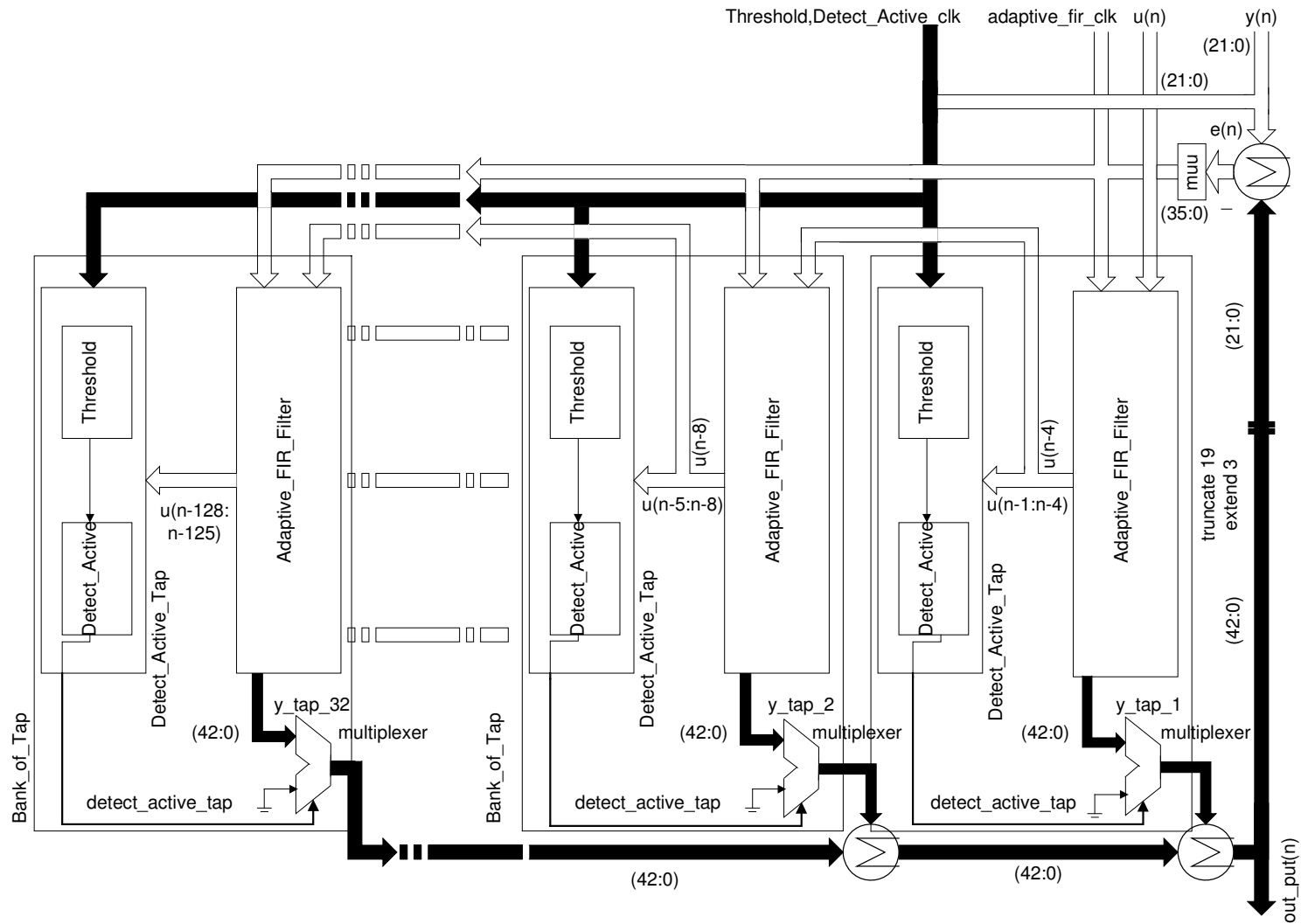
1) Bank Of Taps แบ่งเป็นสองส่วนคือ ส่วนแรกเรียกว่า Adaptive FIR Filter เป็นวงจรกรอง LMS ที่มีโครงสร้างแบบมัลติเพล็กซ์ตามที่กล่าวมาแล้วในบทที่ 3 ซึ่งประกอบด้วยวงจรกรองธรรมดาและส่วนปรับปรุงค่าสัมประสิทธิ์ จำนวน 4 แท้ปทำหน้าที่ในการกรองสัญญาณ ขณะที่ส่วนที่สองเรียกว่า Detect Active Tap ซึ่งเป็นวงจรที่ใช้ตรวจสอบค่าสัมประสิทธิ์แต่ละตัวว่าควรมีค่าเป็นศูนย์หรือไม่เป็นศูนย์ ซึ่งในส่วนของวงจร Detect Active Tap ยังแบ่งออกเป็น 2 ส่วน

1.1 ส่วนแรกคือส่วนที่เรียกว่า Detect Active ดังแสดงในรูปที่ 4.3 ใช้ในการคำนวณเก็บค่าสถิติของแต่ละแท้ปเพื่อนำไปเปรียบเทียบกับค่า Threshold เพื่อบ่งบอกว่าแท้ปนั้นแอมพลิจูดหรือไม่ โดยประกอบไปด้วยสองส่วนคือ ส่วนแรกเรียกว่า Tap Statistics ส่วนนี้ใช้ในการคำนวณค่าสถิติ $f^2(n)$ เพื่อทำการเปรียบเทียบกับค่า Threshold ประกอบด้วยวงจรมัลติเพล็กซ์ วงจรคูณแบบ Radix-4 Booth encoder วงจรบวกแบบ Carry Propagate Adder รีจิสเตอร์จำนวน 4 แท้ป และวงจรดีมัลติเพล็กซ์ ส่วนที่สองคือส่วน Sequential Detect Active ใช้ในการบอกว่าถ้าทุกแท้ปภายใน Bank of Taps มีค่าไม่แอมพลิจูดก็จะให้สัญญาณขาออกของ Bank of Taps มีค่าตามปกติ แต่ถ้ามีแท้ปใดแท้ปหนึ่งที่ไม่แอมพลิจูดจะให้สัญญาณขาออก มีค่าเป็นศูนย์

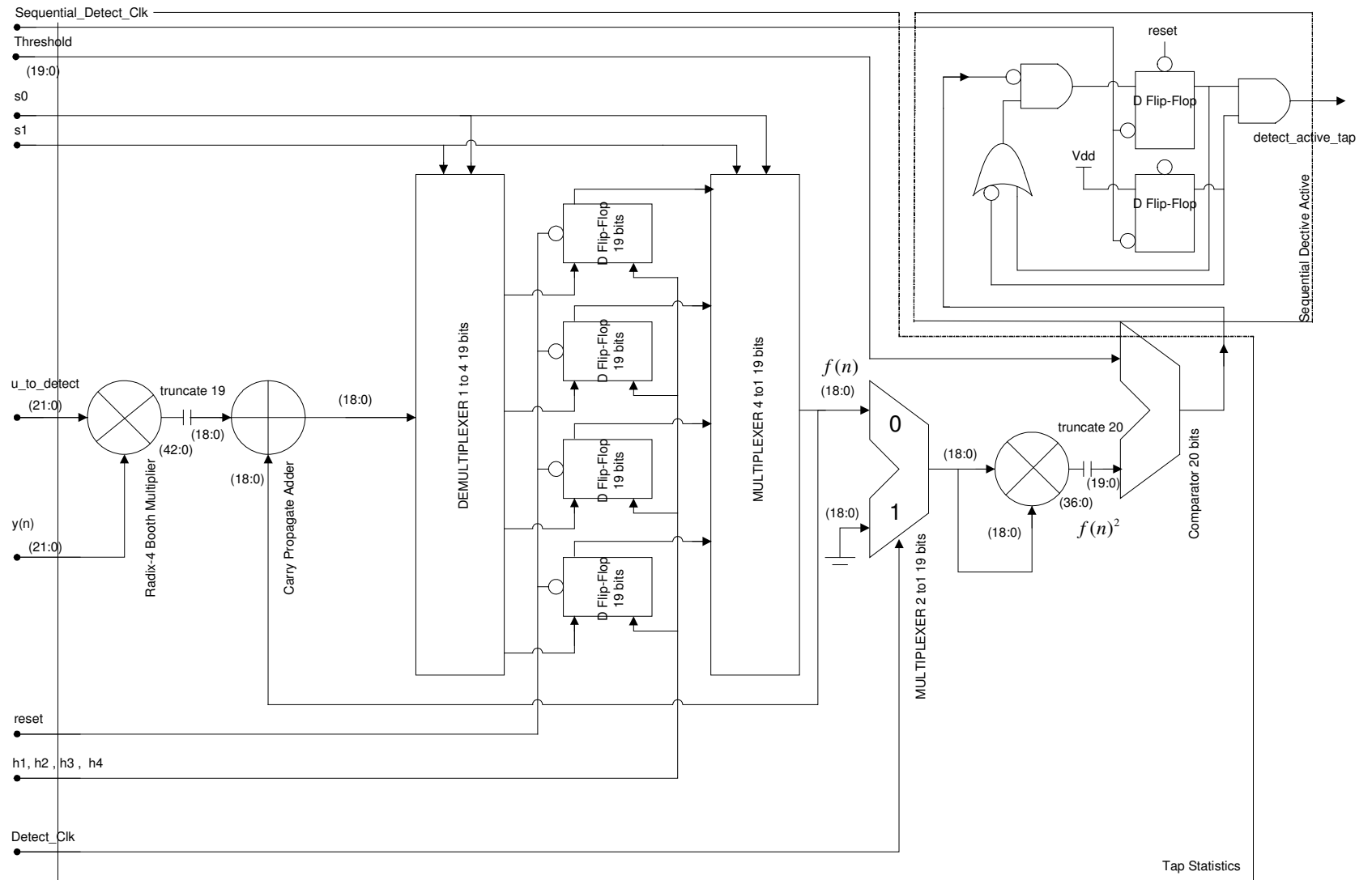
1.2 ส่วนที่สองคือส่วนที่เรียกว่า Threshold ดังแสดงในรูปที่ 4.4 ใช้ในการคำนวณค่า Threshold ส่งไปเปรียบเทียบกับค่าสถิติของแต่ละแท้ปในวงจร Detect Active เพื่อตรวจสอบว่าแท้ปใดบ้างที่แอมพลิจูด ประกอบด้วยวงจรคูณแบบ Radix-4 Booth encoder วงจรบวกแบบ Carry Propagate Adder และรีจิสเตอร์ รวมถึงวงจรที่ใช้ในการคูณค่า $f^2(n)$ กับค่า $\log(N)/N$ เพื่อให้ได้ค่า Threshold ออกมา

2) วงจรบวกที่ทำหน้าที่หาผลรวมของสัญญาณขาออกที่ออกจากวงจรกรองใน Bank Of Taps แต่ละช่วง และให้ผลเป็นสัญญาณขาออกจริง ๆ ที่ออกจากวงจรกรอง

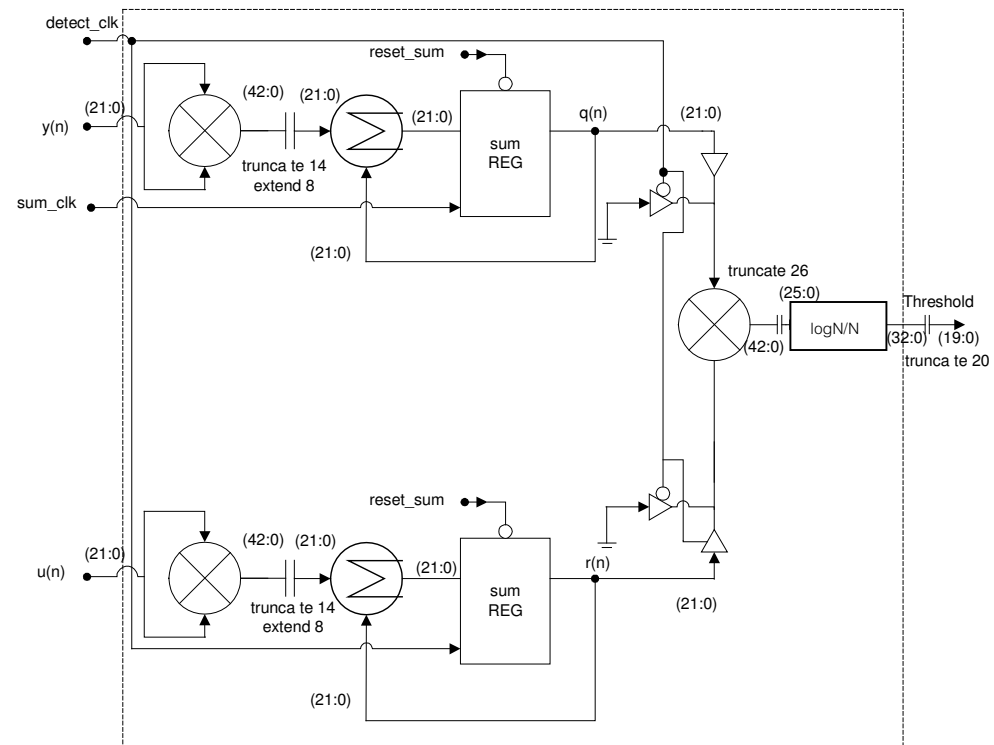
3) วงจรหาผลคูณระหว่างค่าความผิดพลาดกับค่าช่วงก้าวเพื่อนำไปใช้ในการปรับปรุงค่าสัมประสิทธิ์ในส่วนของวงจรกรอง Adaptive FIR Filter ที่อยู่ใน Bank Of Taps



รูปที่ 4.2 สถาปัตยกรรมของวงจรกรองแบบปรับอันดับ



รูปที่ 4.3 สถาปัตยกรรมของวงจร Detect Active Tap



รูปที่ 4.4 สถาปัตยกรรมของวงจร Threshold

4.3 การแทนค่าตัวเลขของสัญญาณต่าง ๆ ภายในวงจร

เนื่องจากแบบจำลองของวงจรการป้องกันการป้อนกลับทางเสียง (Feedback Acoustic Cancellor) ที่กล่าวมาข้างต้น เป็นการแทนสัญญาณต่าง ๆ ภายในวงจรโดยใช้เลขจำนวนจริง (Real Number) แต่ในการสร้างเป็นวงจรจริงบนฮาร์ดแวร์ (Hardware) นั้น การคำนวณภายในวงจรจะแทนสัญญาณ ด้วยค่าเลขฐานสองที่มีจำนวนบิตที่จำกัด โดยสัญญาณขาเข้าของวงจรที่ยังเป็นสัญญาณอะนาลอก จะถูกสุ่มค่าสัญญาณตัวอย่างและผ่านกระบวนการควอนไทซ์ (Quantize) หรือการทำให้เป็นสัญญาณดิจิทัลผ่านทางวงจร A/D converter ซึ่งทำหน้าที่เปลี่ยนสัญญาณอะนาลอกให้กลายเป็นสัญญาณดิจิทัล ซึ่งในที่นี้จะใช้การควอนไทซ์แบบ Uniform rounding ซึ่งสามารถแสดงได้ดังรูปที่ 4.5 และจะดำเนินการภายใต้ Fixed-point แบบ 2's complement โดยสัญญาณอะนาลอก $z \in \mathbb{R}$ จะถูกทำให้เป็นสัญญาณดิจิทัล $\bar{z}$ ซึ่งมีค่าเป็นจำนวนเท่าของระดับขั้นของการควอนไทซ์ q (Quantization step) ซึ่งสามารถแสดงได้ในรูปที่ 4.5

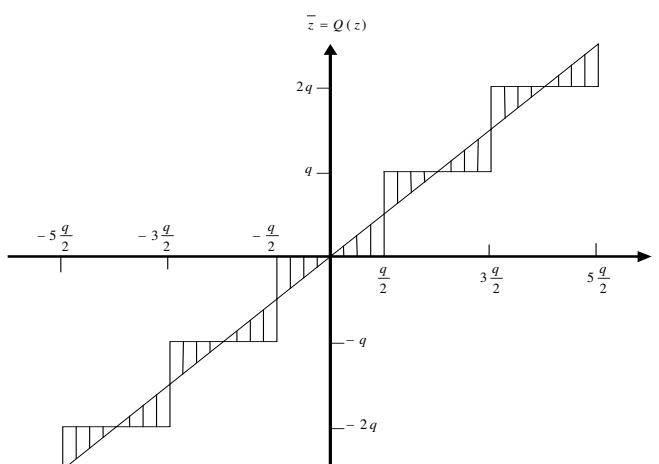
$$q = 2^{-b}$$

ซึ่ง b เป็นจำนวนเต็มบวกและเรียกว่า LSB (Least significant bit) และสามารถแสดง $\bar{z}$ ในรูป

$$\bar{z} = \text{sgn}(z) * \sum_{i=-b}^{B-1} z_i 2^i, \quad z_i = 0 \text{ or } 1$$

เรียก B ว่า MSB (Most significant bit) และลักษณะของสัญญาณหลังจากที่ถูกควอนไทซ์แล้วคือ

$\bar{z}$	z_B	z_{B-1}		z_0	z_{-1}		z_{-b}
scale	sign	2^{B-1}		2^0	2^{-1}		2^{-b}



รูปที่ 4.5 Uniform rounding quantization

ซึ่งกระบวนการดังกล่าวนี้ก็จะส่งผลทำให้ค่าสัญญาณที่เข้าวงจร มีค่าต่างจากสัญญาณอะนาล็อกเดิม นอกจากนี้การคำนวณค่าสัญญาณภายในวงจรเช่น การบวก การคูณ ภายใต้ Fixed Point Arithmetic ยังทำให้ผลลัพธ์ของการคำนวณมีความผิดพลาดเกิดขึ้นด้วย จากค่าผิดพลาดดังที่กล่าวนี้ ถ้าออกแบบวงจรโดยใช้จำนวนบิต ในการแทนสัญญาณต่าง ๆ ภายในวงจรไม่เหมาะสมก็อาจทำให้วงจรทำงานไม่มีเสถียรภาพได้เข้าเมื่อวงจรทำงานผ่านไปไ้ระยะเวลาหนึ่ง นอกจากนี้ถึงแม้จะใช้จำนวนบิต ที่ทำให้วงจรลู่เข้าได้แต่ก็อาจเป็นการใช้จำนวนบิต เกินความจำเป็นซึ่งมีผลทำให้วงจรมีขนาดใหญ่ได้ ดังนั้นจึงต้องมีการทดสอบเพื่อหาจำนวนบิต ที่เหมาะสมในการแทนสัญญาณต่าง ๆ ภายในวงจรโดยนำแบบจำลองของวงจรที่แทนสัญญาณต่าง ๆ ภายในวงจรด้วยจำนวนจริงมาปรับปรุงให้เป็น Fixed Point ซึ่งสามารถทำได้ดังนี้คือ

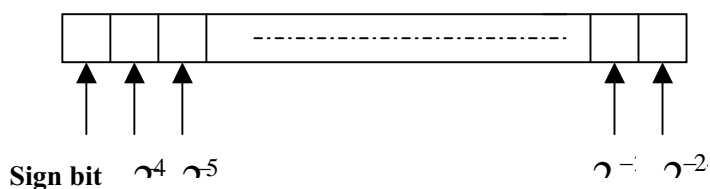
1) แทนการบวกและคูณแบบ Fixed Point

2) ทำการควอนไทซ์ค่าสัญญาณขาเข้าของวงจรกรอง ให้เป็นสัญญาณดิจิทัลโดยใช้ Uniform Rounding Quantization ซึ่งในที่นี้สำหรับสัญญาณต่าง ๆ จะถูกกำหนดจำนวนบิตที่ทราบค่าขึ้นมาแล้วทำการหารระดับขึ้นของการควอนไทซ์ q ที่เหมาะสมกับขนาดของสัญญาณนั้น ๆ ถ้าสัญญาณที่กำลังพิจารณามีขนาดมากที่สุดเป็น M จะสามารถหารระดับขึ้นของการควอนไทซ์ได้จาก $q = \log_2 \left(\frac{2^N - 1}{M} \right)$ จากนั้นจะพิจารณาหาจำนวนบิตและระดับขึ้นของการควอนไทซ์ที่เหมาะสมสำหรับสัญญาณต่าง ๆ โดยพิจารณาจากขนาดสูงสุดของสัญญาณแต่ละส่วนดังตารางที่ 4.1

ตารางที่ 4.1 ขนาดของสัญญาณภายในวงจรที่มีค่ามากที่สุด

สัญญาณภายในวงจร	ขนาดมากที่สุดของ สัญญาณ	จำนวนบิต (บิต)	ขนาด MSB	ขนาด LSB
สัญญาณขาเข้าของวงจรกรอง $u(n)$	1.3329	22	0	-20
สัญญาณอ้างอิง $y(n)$	1.2827	22	0	-20
สัมประสิทธิ์ของวงจรกรอง $w(n)$	0.0693	22	-4	-24
ผลคูณระหว่าง $u(n)$ กับ $w(n)$	0.0723	43	-3	-44
สัญญาณขาออกของ Bank of taps	0.0965	43	-3	-44
สัญญาณขาออกของวงจรกรอง	0.1092	43	-3	-44
ค่าความผิดพลาด $e(n)$	1.2377	22	0	-20
ผลคูณระหว่าง $u(n)$ $e(n)$ และค่า Step size	0.0013	43	-9	-50
สัญญาณ $r(n)$	996.5535	22	9	-11
สัญญาณ $q(n)$	1026.4	22	9	-11
ค่า Threshold	756.6897	33	9	-22
ผลคูณ $u(n)^2$ กับ $y(n)^2$ และ $u(n)$ กับ $y(n)$	1.376	43	1	-40
สัญญาณ $f(n)$	116.3594	19	6	-11

โดยอาศัยตารางที่ 4.1 ซึ่งแสดงค่าขนาดสูงสุดของสัญญาณต่าง ๆ เราสามารถหาจำนวนบิตที่เหมาะสมเพื่อให้ยังสามารถครอบคลุมขนาดของสัญญาณต่าง ๆ ได้โดยพิจารณาโครงสร้างของวงจรดังรูปที่ 4.2 ถึง 4.4 ประกอบ ได้ผลดังแสดงในรูปที่ 4.6 ซึ่งแสดงตัวอย่างของจำนวนบิต MSB และ LSB ของสัญญาณ $w(n)$



รูปที่ 4.6 การควอนไทซ์ของสัญญาณ $w(n)$

จากรูปจะเห็นว่าจำนวนบิตทั้งหมดของสัญญาณ $w(n)$ มีค่าเท่ากับ 22 บิต โดยที่บิตแรกคือตำแหน่งของ Sign Bit และถัดมาเป็นตำแหน่งของ MSB ซึ่งมีค่าเท่ากับ -4 และ ตำแหน่งสุดท้าย

เป็นตำแหน่งของ LSB ซึ่งมีค่าเท่ากับ 24 ในทำนองเดียวกันกับสัญญาณอื่น ๆ ในวงจรซึ่งสามารถแสดงจำนวนบิต MSB และ LSB ของสัญญาณได้ดังตารางที่ 4.1 โดยวิธีหาจำนวนบิตและการควอนไทซ์ของสัญญาณต่าง ๆ สามารถอธิบายได้ดังนี้

- สัญญาณขาเข้าของวงจรกรอง $u(n)$ ถูกกำหนดให้มีขนาด 22 บิต ใช้ระดับขั้นของการควอนไทซ์ เป็น 2^{-20}

- ค่าสัมประสิทธิ์ของวงจรกรอง $w(n)$ ถูกกำหนดให้มีขนาด 22 บิต ใช้ระดับขั้นของการควอนไทซ์ เป็น 2^{-24} โดยค่าสัมประสิทธิ์นี้จะอยู่ในขั้นตอนการปรับค่าสัมประสิทธิ์ก็就会被เพิ่มส่วนที่เป็นศูนย์เข้าไปใน LSB ทำให้มีขนาด 43 บิตเพื่อที่จะสามารถใช้งานร่วมกับขั้นตอนการกรองปกติได้

- สัญญาณอ้างอิง $y(n)$ ถูกกำหนดให้มีขนาด 22 บิตระดับขั้นของการควอนไทซ์ ที่เหมาะสมคือ 2^{-20} ซึ่งแสดงค่าได้ในช่วงเดียวกับ $u(n)$

- สัญญาณออกจากวงจรกรอง เป็นผลมาจากการรวมกันของผลคูณ $u(n)$ กับ $w(n)$ จึงมีขนาด 43 บิต และระดับขั้นของการควอนไทซ์ 2^{-44} และเนื่องจากต้องเอาสัญญาณขาออกของวงจรกรองไปทำการลบออกจากสัญญาณอ้างอิง $y(n)$ ดังนั้นก่อนที่จะเอาไปลบต้องทำให้ตำแหน่งทศนิยมของสัญญาณขาออกตรงกับตำแหน่งทศนิยมของสัญญาณอ้างอิงซึ่งทำได้โดยการนำ MSB 19 บิต แรกของสัญญาณขาออกของวงจรกรอง แล้วทำการขยาย signed bit ออกไปอีก 3 บิตเพื่อให้ได้จำนวนบิตเท่ากัน

- ค่าความผิดพลาด $e(n)$ ได้มาจากการรวมกันระหว่าง $u(n)$ กับ $y(n)$ จึง มีจำนวนบิต และระดับขั้นของการ ควอนไทซ์ เหมือนกับ $y(n)$

- ผลคูณ muu กับ $e(n)$ โดยในที่นี้ Step size มีค่า 0.0009 โดยใช้ระดับขั้นของการควอนไทซ์ 2^{-24} และใช้ขนาด 15 บิตซึ่งเขียนได้เป็น '01110101111011' โดยผลคูณจะมีขนาด 36 บิต และตัวควอนไทซ์ ขนาด 2^{-44}

- ผลคูณ muu กับ $e(n)$ นำค่าผลคูณ muu กับ $e(n)$ มา Truncate ให้เหลือ 22 บิต แรกเพื่อนำไปคูณกับ $u(n)$ ทำให้ได้ ผลคูณ จำนวน 43 บิต และระดับขั้นของการควอนไทซ์ 2^{-64} และเนื่องจากค่าผลคูณดังกล่าวนี้ต้องการนำไปบวกกับค่าสัมประสิทธิ์เพื่อทำการปรับค่าสัมประสิทธิ์ ดังนั้นจึงต้องมีขนาดและจุดทศนิยมตรงกับค่าสัมประสิทธิ์ ดังนั้นก่อนที่จะนำไปบวก

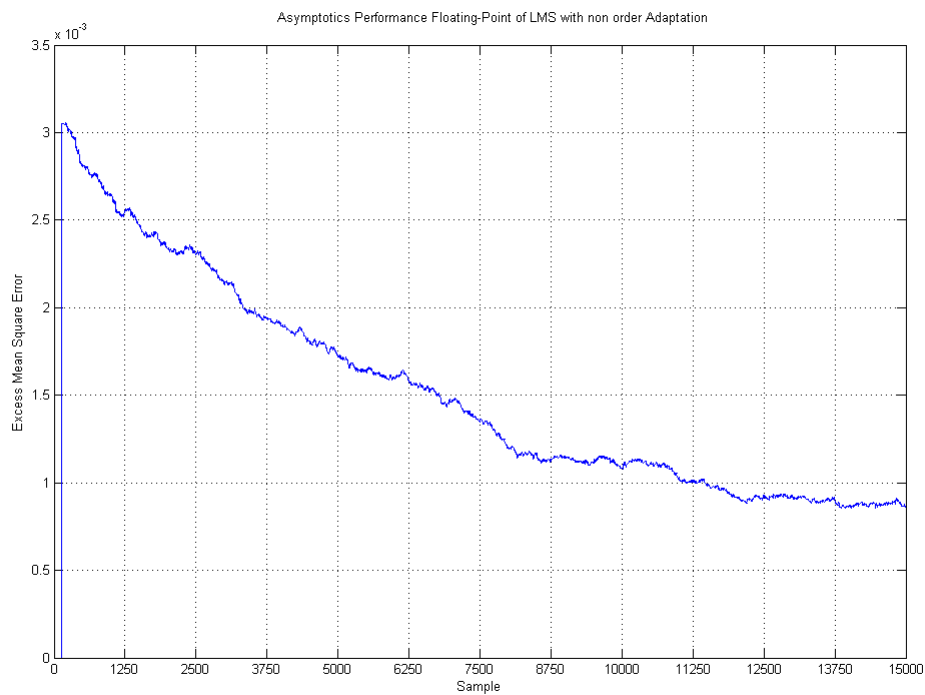
กับค่าสัมประสิทธิ์เพื่อทำการปรับปรุงจึงต้องนำผลคูณดังกล่าวมา Truncate เอา 17 บิตแรกและขยาย Sign bit ออกไปอีก 5 บิตและเพิ่มส่วนที่เป็นศูนย์เข้าไปอีก 21 บิต

- สัญญาณ $r(n)$ เนื่องจาก $r(n)$ เป็นผลบวกของ $u(n)$ ยกกำลังสองซึ่งมีจำนวน 42 บิต และระดับขั้นของการควอนไทซ์ 2^{-40} ขณะที่สัญญาณ $r(n)$ ซึ่งกำหนดให้มี 22 บิตและระดับขั้นของการควอนไทซ์ 2^{-11} นั้นมีจุดทศนิยมที่ไม่ตรงกับ $u(n)$ ยกกำลังสองดังนั้น จึงต้องนำค่า $u(n)$ ยกกำลังสองตัดมา 14 บิตและขยายไปอีก 8 บิต ก่อนที่จะเข้าวงจรบวกในการหา $r(n)$ เพื่อให้จำนวนบิตและจุดทศนิยมตรงกัน ส่วนสัญญาณ $q(n)$ ก็มีลักษณะเหมือนกับ $r(n)$

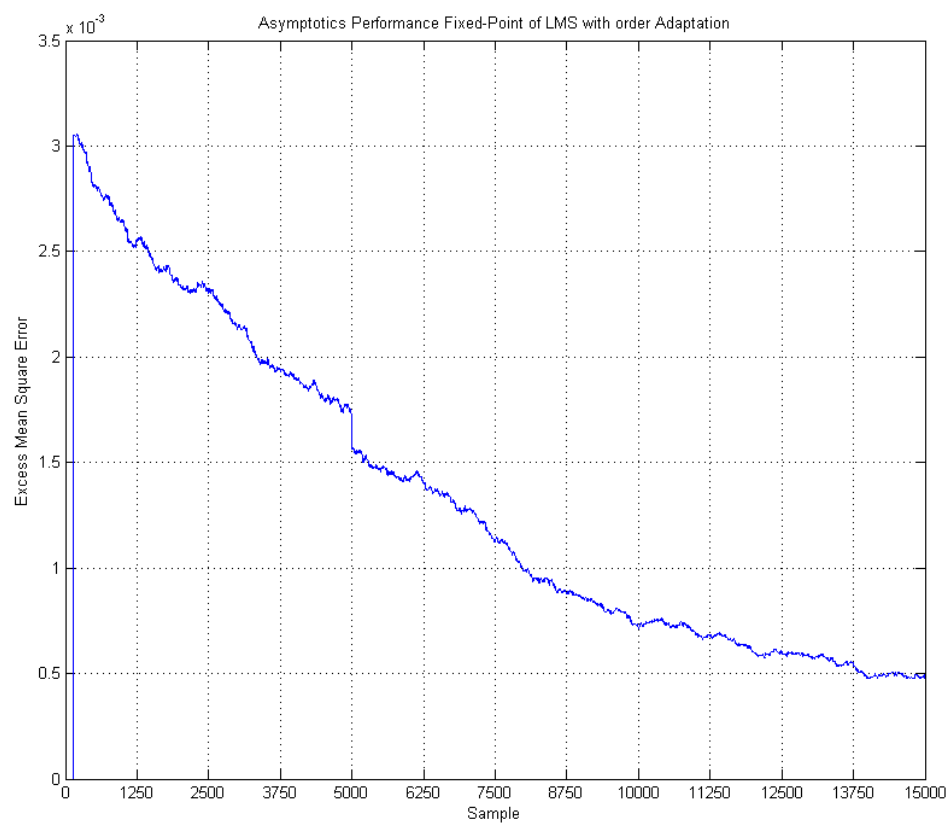
- ผลคูณระหว่าง $y(n)$ กับ $u(n)$ มีจำนวนบิตและระดับขั้นของการควอนไทซ์ ตรงกับ $u(n)$ ยกกำลังสองซึ่งต้องนำค่านี้นามาวกสะสมเพื่อให้ได้ $f(n)$ เนื่องจาก $f(n)$ ถูกกำหนดให้มีขนาด 19 บิตและมีระดับขั้นของการควอนไทซ์ 2^{-11} ดังนั้นเพื่อให้ผลคูณระหว่าง $y(n)$ กับ $u(n)$ มีจำนวนบิตและจุดทศนิยมตรงกับ $f(n)$ จึงต้องนำผลคูณระหว่าง $y(n)$ กับ $u(n)$ มา 14 บิตแรกและขยายไปอีก 5 บิต หลังจากที่ได้ค่า $f(n)$ มาแล้วต้องนำค่า $f(n)$ มายกกำลังสองเพื่อนำไปเปรียบเทียบกับ Threshold โดยค่า $f(n)$ ยกกำลังสองมีขนาด 37 บิตและมีระดับขั้นของการควอนไทซ์ 2^{-22} จากนั้นเพื่อให้สามารถนำค่า $f(n)$ ยกกำลังสองนี้ไปเปรียบเทียบกับ Threshold ได้ต้องตัดเอาค่า $f(n)$ ยกกำลังสองและเอา 20 บิตแรกไปใช้

- ผลคูณระหว่าง $r(n)$ และ $q(n)$ ให้ผลลัพธ์ 43 บิตและระดับขั้นของการควอนไทซ์ 2^{-22} ส่วน $\log_{10} N/N$ โดยในที่นี้ N เท่ากับ 5000 ซึ่งกำหนดให้ $\log_{10} N/N$ มีขนาด 8 บิตและระดับขั้นของการควอนไทซ์ 2^{-17} แสดงได้โดย “01100001” จากนั้นนำผลคูณระหว่าง $r(n)$ และ $q(n)$ ขนาด 43 บิตเอามาเฉพาะ 26 บิตแรกแล้วเอาไปคูณกับ $\log_{10} N/N$ จะให้ผลเป็นค่า Threshold ซึ่งมีขนาด 33 บิตและระดับขั้นของการควอนไทซ์ 2^{-22} แต่ค่า Threshold ดังกล่าวนี้นำไปทำการเปรียบเทียบกับค่า $f(n)$ ยกกำลังสองจึงต้องมีจำนวนบิตและจุดทศนิยมตรงกันดังนั้น ก่อนนำไปเปรียบเทียบจึงต้องนำค่า Threshold มา 16 บิตแรกและขยายไปอีก 4 บิต

หลังจากที่ได้วิเคราะห์หาจำนวนบิตและ ระดับขั้นของการควอนไทซ์ที่เหมาะสมของวงจรแล้วต่อไปจะทำการจำลองผลการทำงานของวงจรโดยเขียน Simulation จำลองการทำงานของวงจร บน โปรแกรม Matlab โดยการป้อนสัญญาณจำนวน 10,000 ค่าซึ่งได้ผลการทดลองตามรูปที่ 4.7 และ 4.8 ซึ่งแสดงความสัมพันธ์ระหว่าง Asymptotic Performance ของวงจรกับจำนวนสัญญาณตัวอย่างที่ป้อนให้กับวงจรจะเห็นว่าเมื่อเปรียบเทียบค่า Excess Mean Square Error กับค่าที่ทดลองจากการคำนวณแบบ Floating-Point กับ Fixed-Point จะเห็นว่าผลที่ได้มีค่าใกล้เคียงกัน



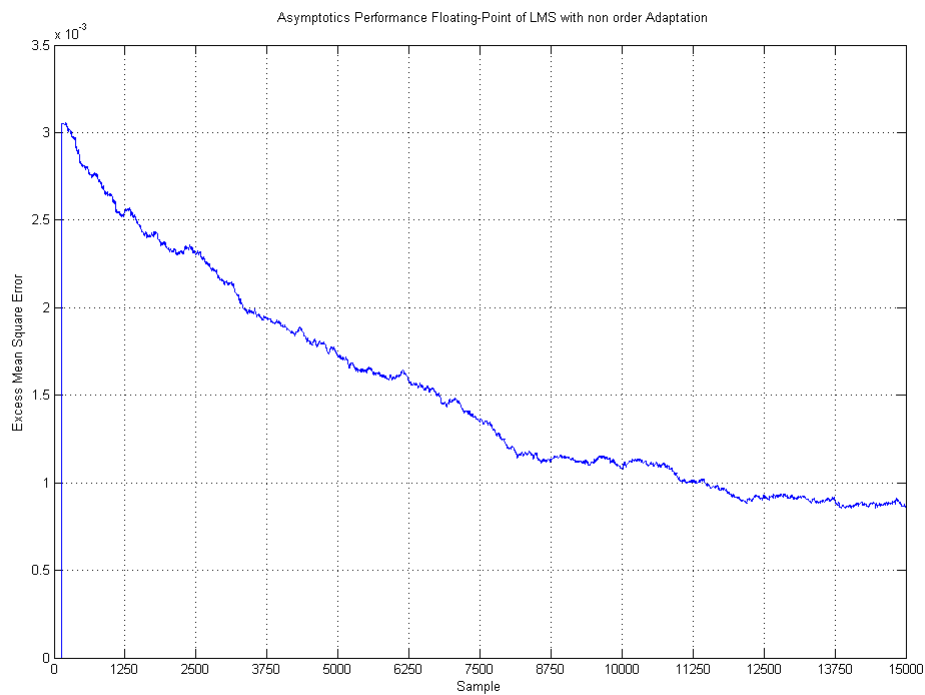
a)



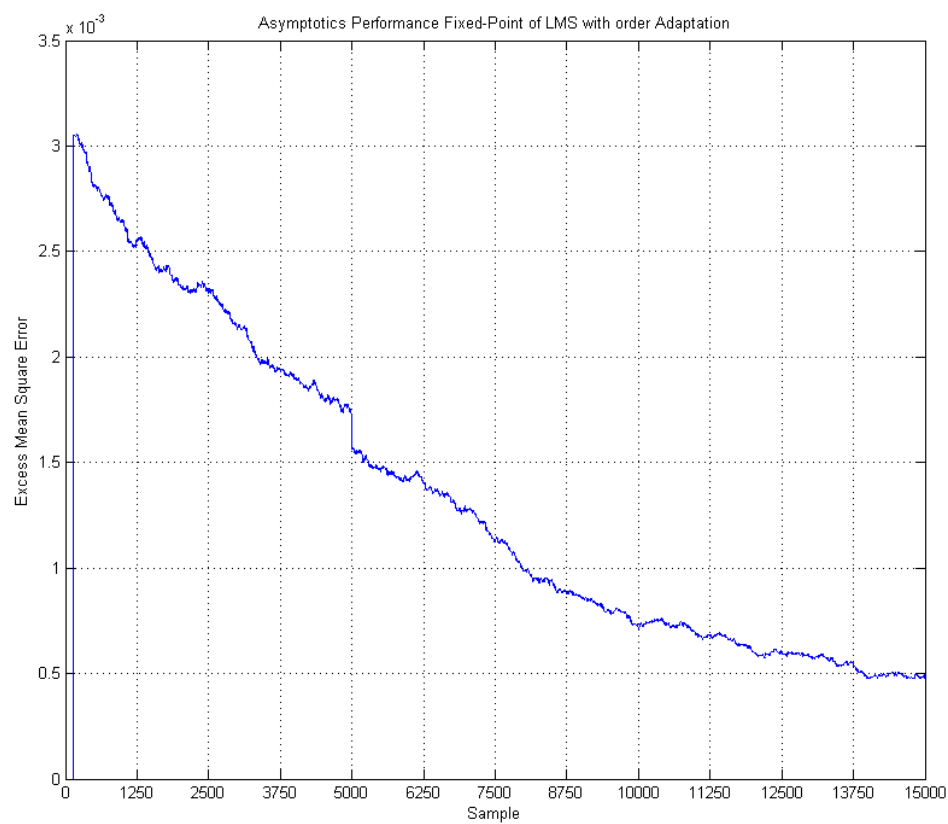
b)

รูปที่ 4.7 Asymptotics Performance แบบจำลอง Floating-Point ของวงจรกรอง LMS

a) แบบที่ไม่มีการปรับอันดับ b) แบบที่มีการปรับอันดับ



a)



b)

รูปที่ 4.8 Asymptotics Performance แบบจำลอง Fixed-Point ของวงจรกรอง LMS

a) แบบที่ไม่มีการปรับอันดับ b) แบบที่มีการปรับอันดับ

บทที่ 5

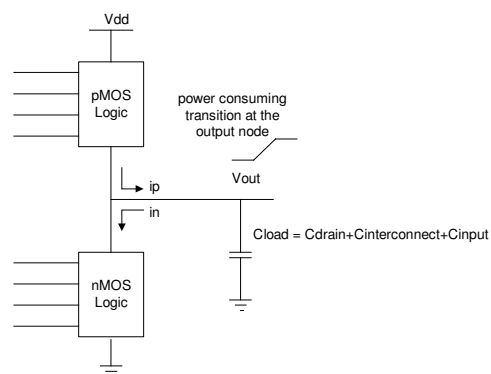
การวัดประสิทธิภาพด้านพลังงานของวงจร

บทนี้กล่าวถึงการสูญเสียพลังงานในวงจรดิจิทัลในหัวข้อที่ 5.1 หัวข้อ 5.2 กล่าวถึงวิธีการวัดจำนวนครั้งการเปลี่ยนแปลงของลอจิกภายในวงจร และหัวข้อ 5.3 กล่าวถึงวิธีการคำนวณประสิทธิภาพด้านการใช้พลังงานของวงจร และสุดท้ายหัวข้อ 5.4 แสดงผลการจำลองการทำงานและประสิทธิภาพทางด้านการใช้พลังงานของวงจรกรอบแบบปรับอันดับ

5.1 การสูญเสียพลังงานในวงจรดิจิทัล

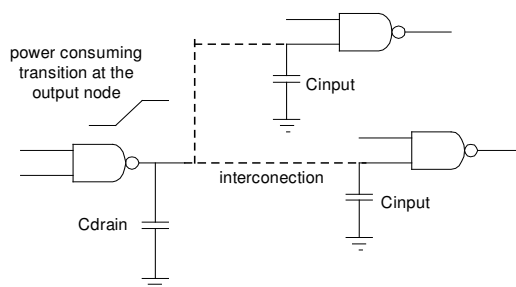
การสูญเสียพลังงานงานในวงจร CMOS ส่วนใหญ่มีสาเหตุมาจากเหตุผล 3 ประการคือ Leakage current, Short circuit current และ Switching current โดยการสูญเสียกำลังเนื่องจากเหตุผลประการสุดท้ายนี้เกี่ยวข้องกับการชาร์จตัวเก็บประจุของโหลด และกำลังที่สูญเสียในวงจรดิจิทัลส่วนใหญ่ก็มาจากส่วนนี้ ดังนั้นในที่นี้จะพิจารณาการสูญเสียกำลังเฉพาะสาเหตุส่วนนี้เท่านั้น

การสูญเสียพลังงานเนื่องจาก Switching current คือพลังงานที่สูญเสียที่โหนดเอาต์พุตของ CMOS เกิดเมื่อมีการเปลี่ยนแปลงค่าลอจิก ดังแสดงในรูปที่ 5.1 โดยพลังงานจะถูกดึงจากพาวเวอร์ซัพพลายในการชาร์จตัวเก็บประจุที่เอาต์พุต ในระหว่างการชาร์จนี้ค่าศักย์ของเอาต์พุตจะเปลี่ยนจาก 0 ไปเป็น V_{dd} และครึ่งหนึ่งของพลังงานที่ดึงมาจากพาวเวอร์ซัพพลายจะสูญเสียเป็นความร้อนใน pMOS ทรานซิสเตอร์ และพลังงานส่วนที่เหลือจะไปเก็บไว้ในตัวเก็บประจุ จากนั้นเมื่อศักย์ของเอาต์พุตถูกทำให้ลดจาก V_{dd} เหลือ 0 กระแสก็จะไหลจากตัวเก็บประจุไปยังกราวด์ และพลังงานส่วนที่เหลือก็จะสูญเสียเป็นความร้อนใน nMOS ทรานซิสเตอร์



รูปที่ 5.1 การสูญเสียพลังงานระหว่างการชาร์จและดิสชาร์จตัวเก็บประจุที่เอาต์พุต

ตัวเก็บประจุที่เอาต์พุตประกอบด้วยตัวเก็บประจุของทรานซิสเตอร์ (C_{drain}) ที่เอาต์พุตและตัวเก็บประจุที่อินพุตของเกตแต่ละตัวที่ถูกขับ (C_{input}) ดังแสดงในรูปที่ 5.2



รูปที่ 5.2 ตัวเก็บประจุของโหลดที่เอาต์พุต

ในการหารูปแบบความสัมพันธ์ที่เหมาะสมระหว่างการทำงานของวงจรกับการสูญเสียพลังงานนั้น ต้องสอดคล้องกับสมมติฐาน 2 ข้อ คือ

- 1) ตัวเก็บประจุของเกต CMOS แต่ละตัวมีเพียงแค่นี้ที่โหลดเอาต์พุตเท่านั้น
- 2) กระแสไหลจาก V_{dd} ไปยังตัวเก็บประจุที่เอาต์พุต (การเปลี่ยนจาก 0 ไป 1) หรือจากตัวเก็บประจุไปยังกราวด์ (การเปลี่ยนจาก 1 เป็น 0)

อาศัยสมมติฐาน 2 ข้อนี้นี้จะสามารถพิสูจน์ได้ว่าพลังงานที่สูญเสียในเกต CMOS คือ

$$P = \frac{1}{2} N C V_{\text{dd}}^2 \quad (5.1)$$

โดย

C คือ ตัวเก็บประจุที่เอาต์พุต

V_{dd} คือ ค่าศักย์ของพาวเวอร์ซัพพลายของเกต

N คือ จำนวนการเปลี่ยนแปลงที่เอาต์พุต (นับการเปลี่ยนแปลงจาก 0 เป็น 1 และการเปลี่ยนแปลงจาก 1 เป็น 0)

โดยทั่วไปแต่ละโหนดภายในวงจรดิจิทัลจะสามารถเกิดการเปลี่ยนแปลงนอกเหนือจากการเปลี่ยนสถานะของลอจิกตามการทำงาน นั่นคือการเปลี่ยนแปลงที่เรียกว่า Glitch โดยการเปลี่ยนแปลงนี้เป็นผลมาจากแต่ละเส้นทางในวงจรที่มีค่าดีเลย์ต่างกัน ทำให้เกิดการเปลี่ยนแปลงของสัญญาณที่ไม่ถูกต้องตามการทำงานจริง ซึ่งการเปลี่ยนแปลงลักษณะนี้ทำให้เกิดการสูญเสียพลังงานเช่นเดียวกับการเปลี่ยนแปลงของสัญญาณตามปกติด้วย ซึ่งลักษณะของการเกิด Glitch มีอยู่ 2 ลักษณะคือ

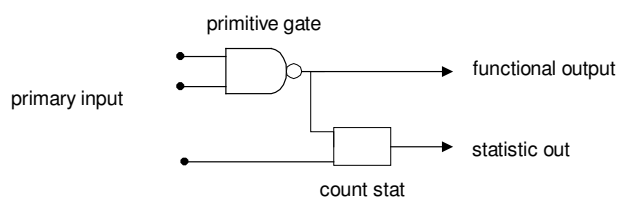
1) เมื่ออินพุตเปลี่ยนแปลงและทำให้เอาต์พุตเปลี่ยนจาก 0 ไปเป็น 1 แล้วเปลี่ยนแปลงกลับเป็น 0 อย่างเดิม ปรากฏการณ์นี้เราเรียกว่า glitch 0-1-0

2) เมื่ออินพุตเปลี่ยนแปลงและทำให้เอาต์พุตเปลี่ยนจาก 1 ไปเป็น 0 แล้วเปลี่ยนแปลงกลับเป็น 1 อย่างเดิม ปรากฏการณ์นี้เราเรียกว่า glitch 1-0-1

5.2 การวัดจำนวนครั้งการเปลี่ยนแปลงของลอจิกภายในวงจร

การประมาณการสูญเสียกำลังจะใช้หลักการเดียวกับโปรแกรมที่เขียนขึ้นด้วยภาษา HDL [8] โดยมีหลักการคือ โปรแกรมจะประกอบไปด้วยสองส่วนคือ ส่วนที่ใช้เป็นเกตปรกติ (Primitive gate) และส่วนที่ใช้ในการคำนวณการสูญเสียพลังงาน (Count stat) ดังแสดงในรูปที่ 5.3 โดยดีเลย์ของเอาต์พุตเกตถูกกำหนดให้มีค่าเป็นสัดส่วนโดยตรงกับจำนวน Fan Out ในส่วนของ Count stat จะทำการนับการเปลี่ยนแปลงของเอาต์พุตทั้งจาก 0 เป็น 1 และจาก 1 เป็น 0 หลังจากนั้นจะนำข้อมูลในส่วนนี้ไปทำการคำนวณการสูญเสียพลังงาน ในการนับการเปลี่ยนแปลงนั้นจะเริ่มนับจากเวลาที่สัญญาณอินพุตเปลี่ยนแปลงนับไปจนถึงระยะเวลาที่มากกว่าค่าดีเลย์ที่มากที่สุดของเส้นทางภายในวงจร เพื่อให้แน่ใจว่าเมื่อถึงเวลานั้นสัญญาณต่าง ๆ ภายในวงจรจะไม่มี การเปลี่ยนแปลงอีก

จากวิธีดังกล่าวเราสามารถนำมาประยุกต์ใช้เพื่อทำการประมาณการสูญเสียพลังงานของวงจรแบบปรับอันดับ กับ โครงสร้างของวงจรแบบที่ไม่มีการปรับอันดับ เพื่อนำมาเปรียบเทียบประสิทธิภาพทางด้านพลังงาน

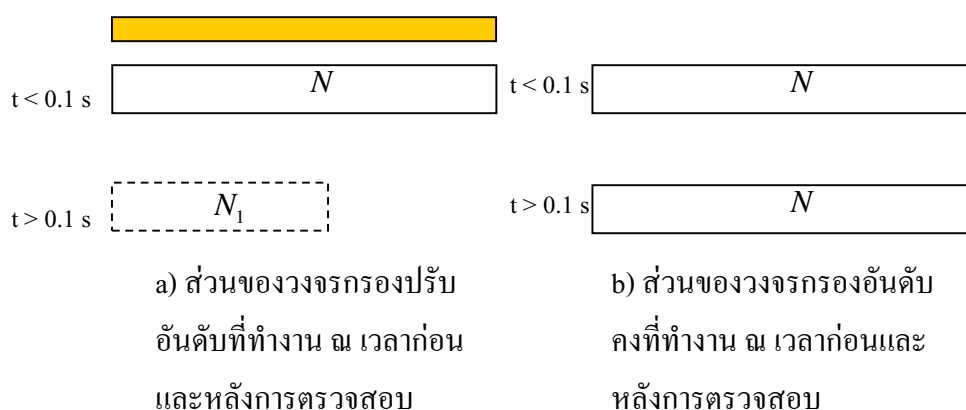


รูปที่ 5.3 ส่วนต่าง ๆ ภายในเกตที่ถูกเขียนให้สามารถวัดการสูญเสียพลังงานได้

5.3 การคำนวณประสิทธิภาพทางด้านพลังงานของวงจรกรองแบบปรับอันดับ

การวัดประสิทธิภาพการใช้พลังงานของวงจรกรองแบบปรับอันดับนี้จะทำโดยการวัดการใช้พลังงานของวงจรกรองแบบปรับอันดับตลอดช่วงระยะเวลาการทำงานของวงจรเปรียบเทียบกับการใช้พลังงานของวงจรกรองแบบที่ไม่มีการปรับอันดับในระยะเวลาการทำงานเท่ากัน ซึ่งสามารถพิจารณาได้จากการเปรียบเทียบส่วนต่าง ๆ ของวงจรที่ทำงานในระหว่างคาบการปรับตัวของวงจร

วงจรกรองแบบปรับอันดับนั้นแต่ละรอบของการปรับตัวของวงจรดังที่กล่าวไว้แล้วในบทที่ 4 จะใช้เวลา 0.3 วินาทีเพื่อเข้าสู่สภาวะอยู่ตัว โดยภายในคาบของการปรับตัวการทำงานจะแบ่งเป็น 2 ช่วงคือ ช่วงแรกจะมีทั้งส่วนของวงจรกรองและส่วนที่ใช้ในการตรวจสอบแอมพลิจูดที่ป้อนรวมทั้งส่วนของวงจร Threshold ที่ทำงานซึ่งส่วนของวงจรกรองและส่วนที่ใช้ในการตรวจสอบแอมพลิจูดที่ป้อนนี้ในตอนเริ่มแรกจะมีจำนวนเท่ากันกับจำนวนที่ป้อนทั้งหมดของวงจร จากนั้นในช่วงที่สองหลังจากการปรับตัวในช่วงแรกผ่านไป 0.1 วินาที ซึ่งเวลานี้จะสามารถตรวจสอบตำแหน่งของแอมพลิจูดที่ป้อนได้แล้วนั้น ก็จะทำการตัดการทำงานในส่วนที่ป้อนที่ไม่ใช่แอมพลิจูดและส่วนที่ใช้ในการตรวจสอบแอมพลิจูดที่ป้อนรวมถึงวงจร Threshold ออกไป เหลือไว้เฉพาะส่วนที่เป็นวงจรกรองเฉพาะที่ป้อนที่แอมพลิจูดเท่านั้นที่ยังทำงานอยู่ ขณะที่หากพิจารณาวงจรกรองแบบที่ไม่มีการปรับอันดับนั้นจะเห็นว่าอันดับของวงจรกรองจะคงที่ตลอดช่วงระยะเวลาการทำงานตั้งแต่เริ่มต้นจนเข้าสู่สภาวะอยู่ตัว ตลอดช่วงระยะเวลาการทำงาน 0.3 วินาที



รูปที่ 5.4 การทำงานของวงจรกรองแบบปรับอันดับและแบบไม่มีการปรับอันดับ

ระหว่างคาบการทำงานในขณะที่สัญญาณขาเข้ามีกำลังน้อยกว่า $\sigma_{\max}^2$

จากรูปที่ 5.4 แสดงให้เห็นถึงส่วนของวงจรที่ทำงานระหว่างคาบการปรับตัวของวงจรแบบปรับอันดับและไม่ปรับอันดับตามลำดับ ซึ่งส่วนของวงจรแบบปรับอันดับดังรูปที่ 5.4 a) จะเห็นว่า ณ.เวลาก่อนทำการตรวจสอบเอ็คทีฟแท็ป วงจรที่ทำงานมีอยู่สองส่วนคือ วงจรตรวจสอบเอ็คทีฟแท็ป (แสงเงา) และวงจรกรองปรับตัวจำนวน N แท็ป (ไม่แสงเงา) เมื่อเวลาผ่านไป 0.1 วินาทีก็จะทำการตรวจสอบเอ็คทีฟแท็ป หลังจากตรวจสอบแล้วก็จะมีเฉพาะส่วนที่เป็นวงจรกรองเฉพาะแท็ปที่เอ็คทีฟซึ่งมีจำนวน N_1 แท็ป (เส้นประ) เท่านั้นที่ทำงาน ขณะที่รูปที่ 5.4 b) วงจรกรองปรับตัวแบบที่ไม่มีการปรับอันดับนั้น ทุก N แท็ปจะทำงานตลอดเวลา จะเห็นว่าตอนที่ยังไม่ตรวจสอบเอ็คทีฟแท็ป วงจรกรองแบบปรับอันดับจะสูญเสียพลังงานมากกว่าวงจรกรองแบบที่ไม่มีการปรับอันดับในส่วนการทำงานของวงจรตรวจสอบเอ็คทีฟแท็ป หลังจากที่มีการตรวจสอบเอ็คทีฟแท็ปวงจรกรองแบบปรับอันดับจะสามารถประหยัดพลังงานได้เนื่องจากการทำงานเฉพาะส่วนแท็ปที่เอ็คทีฟเท่านั้น

จากลักษณะการสูญเสียพลังงานดังกล่าว เราจะสามารถวัดการใช้พลังงานในระหว่างคาบการทำงานปรับตัวของวงจรแบบปรับอันดับเปรียบเทียบกับวงจรกรองแบบที่ไม่มีการปรับอันดับได้ โดยใช้การวัดจำนวนครั้งของการเปลี่ยนแปลงของลอจิกภายในวงจรที่จำลองขึ้นบนภาษา Verilog แล้วนำมาคำนวณการสูญเสียพลังงานได้ แต่เนื่องจากอันดับของวงจรกรองที่นำมาจำลองการทำงานในที่นี้มีค่ามากรวมถึงหากวัดจำนวนการเปลี่ยนแปลงของลอจิกภายในวงจรทุกส่วนตลอดระยะเวลาการทำงานโดยตรงนั้นจะต้องใช้การคำนวณที่ซับซ้อนขณะที่โปรแกรมที่จำลองผลการทำงานสามารถคำนวณได้จำกัด ด้วยเหตุผลดังกล่าวการวัดการใช้พลังงานของวงจรทั้งหมดโดยตรงจึงเป็นไปได้ยาก ดังนั้นเพื่อให้โปรแกรมสามารถคำนวณหาจำนวนการเปลี่ยนแปลงของลอจิกภายในวงจรได้ จะใช้เงื่อนไข 2 ประการ คือ

1) การหาจำนวนการเปลี่ยนแปลงของลอจิกตลอดช่วงเวลาการทำงานของวงจรที่ป้อนตัวอย่างสัญญาณเข้าสู่สถานะอยู่ตัวนั้น จะทำโดยหาจำนวนการเปลี่ยนแปลงของลอจิกเฉลี่ยของวงจรต่อ 1 ตัวอย่างสัญญาณ โดยหาจากการป้อนสัญญาณตัวอย่างให้กับวงจรกรองเป็นจำนวน 5000 ตัวอย่างและนำค่าจำนวนการเปลี่ยนแปลงของลอจิกทั้งหมดมาเฉลี่ยต่อ 1 ตัวอย่างสัญญาณ จากนั้นนำจำนวนการเปลี่ยนแปลงของลอจิกเฉลี่ยของวงจรต่อ 1 ตัวอย่างสัญญาณดังกล่าวมาคูณกับจำนวนตัวอย่างสัญญาณทั้งหมดที่ป้อนให้กับวงจรคูณก็จะสามารถประมาณการเปลี่ยนแปลงของลอจิกทั้งหมดตลอดช่วงเวลาการทำงาน of วงจรได้

2) พิจารณาจำนวนการเปลี่ยนแปลงของลอจิกภายใน Bank of Taps แต่ละตัวที่อยู่ในเขตเอ็คทีฟแท็ป เนื่องจากจำนวนการเปลี่ยนแปลงของลอจิกภายใน Bank of Taps ที่ตรงกับตำแหน่ง

ผลตอบแทนเชิงเวลาของช่องสัญญาณป้อนกลับที่มีขนาดสูงสุดนั้นน่าจะมีจำนวนการเปลี่ยนแปลงของลอจิกที่มากที่สุดหรือมีการสูญเสียพลังงานมากที่สุดเมื่อเปรียบเทียบกับ Bank of Taps ณ ตำแหน่งอื่น ๆ ที่อยู่ในเขตแอ็คทีฟแท้ที่ตรงกับผลตอบแทนเชิงเวลาที่มีขนาดน้อยลงมา ดังนั้นเราจะคำนวณหาขอบเขตของจำนวนการเปลี่ยนแปลงของลอจิกของแต่ละ Bank of Taps ที่อยู่ในเขตแอ็คทีฟแท้ได้ด้วยจำนวนการเปลี่ยนแปลงของลอจิกของ Bank of Taps ที่ตรงกับตำแหน่งผลตอบแทนเชิงเวลาของช่องสัญญาณป้อนกลับที่มีขนาดสูงสุด ขณะที่แต่ละ Bank of Taps ที่อยู่ในเขตที่ไม่แอ็คทีฟนั้นจะใช้จำนวนการเปลี่ยนแปลงของลอจิกของ Bank of Taps ที่ตรงกับตำแหน่งที่ผลตอบแทนเชิงเวลาของช่องสัญญาณป้อนกลับที่มีกำลังของค่าสัมประสิทธิ์มากที่สุดเป็นตัวแทนของ Bank of Taps ที่เหลือ ดังนั้นในการหาจำนวนการเปลี่ยนแปลงของวงจรทุกส่วนเราจะแบ่งการวัดออกเป็น 2 ส่วนคือ

1) ส่วนแรกเป็นการวัดจำนวนการเปลี่ยนแปลงของลอจิกของแต่ละ Bank of Taps ที่ตรงกับตำแหน่งแท้ที่แอ็คทีฟ

2) ส่วนที่สองเป็นการวัดจำนวนการเปลี่ยนแปลงของลอจิกของแต่ละ Bank of Taps ที่ตรงกับตำแหน่งแท้ที่ไม่แอ็คทีฟ

จากเงื่อนไขดังกล่าวเราจะสามารถคำนวณประสิทธิภาพทางด้านพลังงานของวงจรแบบปรับอันดับกับวงจรแบบที่ไม่มีการปรับอันดับได้ โดยสมมติให้

m คือจำนวนตัวอย่างสัญญาณทั้งหมดที่ป้อนให้กับวงจรกรองก่อนการตรวจสอบแอ็คทีฟแท้

n คือจำนวนตัวอย่างสัญญาณทั้งหมดที่ป้อนให้กับวงจรกรองในระหว่างคาบของการปรับตัว

N คือจำนวนแท้ของวงจรกรองก่อนการตรวจสอบแอ็คทีฟแท้

N_1 คือจำนวนแท้ของวงจรกรองหลังการตรวจสอบแอ็คทีฟแท้

P_1 คือพลังงานที่สูญเสียเฉลี่ยต่อ 1 ตัวอย่างสัญญาณต่อ 1 แท้ ของวงจรกรองปรับตัว ตรงตำแหน่งแอ็คทีฟแท้ที่มีขนาดสูงสุด

P_2 คือพลังงานที่สูญเสียเฉลี่ยต่อ 1 ตัวอย่างสัญญาณต่อ 1 แท้ ของวงจรกรองปรับตัว ตรงตำแหน่งแท้ที่ไม่แอ็คทีฟ

P_3 คือพลังงานที่สูญเสียเฉลี่ยต่อ 1 ตัวอย่างสัญญาณต่อ 1 แท้ของวงจรตรวจสอบแอ็คทีฟแท้ ตรงตำแหน่งแอ็คทีฟแท้ที่มีขนาดสูงสุด

P_4 คือพลังงานที่สูญเสียเฉลี่ยต่อ 1 ตัวอย่างสัญญาณต่อ 1 แท้ปของวงจรตรวจสอบแเอ็คทีฟ แท้ป ตรงตำแหน่งแท้ปที่ไม่แเอ็คทีฟ

P_{TH} คือพลังงานที่สูญเสียตลอดระยะเวลาการป้อนตัวอย่างสัญญาณค่าของวงจร Threshold

$$Energy\ efficiency = \frac{P_2(N - N_1)(n - m) - P_3N_1m - P_4(N - N_4)m - P_{TH}}{P_1N_1n + P_2(N - N_1)n} \times 100\% \quad (5.2)$$

พิจารณาพจน์ที่เป็นเศษของสมการที่ (5.2) เทอมแรกของตัวเศษคือพลังงานทั้งหมดของ แท้ปที่ไม่แเอ็คทีฟที่สามารถประหยัดได้ตลอดช่วงเวลาหลังจากทำการตรวจสอบแเอ็คทีฟแท้ป เนื่องจากการหยุดการทำงานของแท้ปที่ไม่แเอ็คทีฟนั่นเอง เทอมที่สองและเทอมที่สามเป็นพลังงานที่สูญเสียไประหว่างการทำงานก่อนการตรวจสอบแเอ็คทีฟแท้ป ของวงจรตรวจสอบแเอ็คทีฟแท้ปในส่วน ของที่ตรงกับเขตแเอ็คทีฟและไม่แเอ็คทีฟตามลำดับ ดังนั้นพจน์ข้างบนที่ซึ่งเป็นผลต่างระหว่างพลังงานที่สามารถประหยัดได้ในเวลาหลังการตรวจสอบแเอ็คทีฟแท้ป กับพลังงานที่สูญเสียเพิ่มเติมจากวงจรรบกวนในเวลาก่อนการตรวจสอบแเอ็คทีฟแท้ป ก็คือพลังงานทั้งหมดที่วงจรสามารถประหยัดได้นับตั้งแต่เวลาเริ่มต้นทำงานจนถึงเวลาที่วงจรเข้าสู่สภาวะอยู่ตัวนั่นเอง

ส่วนที่พจน์ที่เป็นส่วนนั้น พจน์แรกและพจน์ที่สองเป็นพลังงานที่ใช้ตลอดระยะเวลาการทำงานตั้งแต่เริ่มต้นจนถึงเวลาที่วงจรเข้าสู่สภาวะอยู่ตัว ของวงจรกรองที่ตรงกับเขตแเอ็คทีฟและไม่แเอ็คทีฟตามลำดับ หรือเป็นพลังงานที่ต้องใช้ทั้งหมดของวงจรกรองที่ไม่ปรับอันดับนั่นเอง

แต่จากสมการดังกล่าวจำนวนสวิทช์หรือการสูญเสียพลังงานของแท้ปที่ไม่แเอ็คทีฟจะมีค่าน้อยกว่าจำนวนสวิทช์หรือการสูญเสียพลังงานของแท้ปที่แเอ็คทีฟ นั่นคือเราสามารถประมาณขอบเขตจำนวนการสวิทช์ของแท้ปที่ไม่แเอ็คทีฟได้ด้วยจำนวนสวิทช์ของแท้ปที่แเอ็คทีฟหรือ $P_4 < P_3$ และ $P_2 < P_1$ ดังนั้นเราสามารถคำนวณประสิทธิภาพทางพลังงานได้ใหม่จากสมการที่(5.3)

$$Energy\ efficiency \approx \frac{P_2(N - N_1)(n - m) - P_3Nm - P_{TH}}{P_1Nn} \times 100\% \quad (5.3)$$

5.4 การทดสอบหาการสูญเสียพลังงาน

จุดประสงค์ของการทดสอบนี้คือ หาจำนวนครั้งการเปลี่ยนแปลงของลอจิกทั้งหมดภายในแต่ละโหนดของวงจรแล้วนำมาคำนวณการสูญเสียพลังงาน โดยอาศัยวงจรที่เขียนจำลองการทำงานขึ้นบนภาษา Verilog ที่เกิดแต่ละตัวสามารถคำนวณการสูญเสียพลังงานได้ตามที่แสดงไว้ในหัวข้อ 5.2 โดยการทดสอบจะแยกออกเป็น 4 ส่วนดังนี้

1) ส่วนของวงจร Adaptive FIR Filter ที่ตรงกับตำแหน่งผลตอบสนองเชิงเวลาที่มีค่าสูงสุด ในที่นี้คือตำแหน่ง Bank of Taps ที่ 16 โดยมีสัญญาณอินพุตคือ $u(n)$ และ $muue(n)$

2) ส่วนของวงจร Adaptive FIR Filter ที่ตรงกับตำแหน่งตรงกับแท็ปที่ไม่แอ็คทีฟที่มีค่าแวลเรียนท์สูงสุด ในที่นี้คือตำแหน่ง Bank of Taps ที่ 21 โดยมีสัญญาณอินพุตคือ $u(n)$ และ $muue(n)$

3) ส่วนของวงจร Detect Active ที่ตรงกับตำแหน่งผลตอบสนองเชิงเวลาที่มีค่าสูงสุดในที่นี้คือตำแหน่ง Bank of Taps ที่ 16 โดยมีสัญญาณอินพุตคือ $y(n)$ และ u_to_detect

4) ส่วนของวงจร Threshold โดยมีสัญญาณอินพุตคือ $u(n)$ และ $y(n)$

โดยสัญญาณต่าง ๆ ที่ป้อนให้กับวงจรแต่ละส่วนนั้นนำมาจากผลการจำลองการทำงานของวงจรด้วย Fixed-Point บนโปรแกรม Matlab โดยสัญญาณอินพุตของวงจรในส่วนที่ 1 และ 3 จะนำมาจากอินพุตของแบบจำลอง Fixed-Point ของวงจรที่ตรงกับตำแหน่งแท็ปที่ 61 และสัญญาณอินพุตของวงจรในส่วนที่ 2 นำมาจากอินพุตของแบบจำลอง Fixed-Point ของวงจรที่ตรงกับตำแหน่งแท็ปที่ 81

กำหนดให้วงจรกรองที่ทำการทดสอบในแต่ละ Banks of Taps มีจำนวน 4 แท็ปและอันดับของวงจรกรองทั้งหมดมีจำนวน 128 แท็ป กำลังของสัญญาณขาเข้าของวงจรกรองและเส้นทางป้อนกลับทางเสียงมีค่าตามที่กำหนดไว้ในหัวข้อที่ 4.3 โดยทำการป้อนสัญญาณต่าง ๆ เป็นจำนวน 5000 ตัวอย่าง

ผลการทดสอบ

การแสดงผลการทดสอบจะแสดงด้วยตารางที่แสดงถึงจำนวน Transitions, Glitches, และ Power loss ของวงจรต่าง ๆ ภายในแต่ละส่วนที่ทำการทดสอบซึ่งแบ่งเป็น 4 ส่วน โดยผลที่ได้เป็นค่าเฉลี่ยต่อ 1 ตัวอย่างสัญญาณ

ตารางที่ 5.1 ผลการทดสอบการสูญเสียพลังงานของวงจร

Adaptive FIR Filter ที่ตรงกับตำแหน่ง Bank of Taps ที่ 21

	Average transition per iteration(ครั้ง)	Average glitch per iteration(ครั้ง)	Average power loss per iteration(วัตต์)
Carry propagate adder 43 bits	3914	4683	23730
Coefficient register 22 bits 4 taps	4135	1134	8815
Input register 22 bits 4 taps	392	0	703
Radix-4 booth multiplier 22 bits	77639	16306	181122
Dataout_reg, Sum_reg	7909	1831	15788
Multiplexer_4_to_1_22_bits_u, Multiplexer_4_to_1_22_bits_coefficient, Demultiplexer_1_to_4_22_bits_coefficient.	3159	446	4446
Multiplexer และ Demultiplexer ในการเปลี่ยนเส้นทางของวงจร	10684	831	24752
รวม	107840	25236	259360

ตารางที่ 5.2 ผลการทดสอบการสูญเสียพลังงานของวงจร

Adaptive FIR Filter ที่ตรงกับตำแหน่ง Bank of Taps ที่ 16

	Average transition per iteration(ครั้ง)	Average glitch per iteration(ครั้ง)	Average power loss per iteration(วัตต์)
Carry propagate adder 43 bits	3906	4683	24144
Coefficient register 22 bits 4 taps	4105	1134	8798
Input register 22 bits 4 taps	393	0	704
Radix-4 booth multiplier 22 bits	78193	16306	182871
Dataout_reg, Sum_reg	8139	1831	16243
Multiplexer_4_to_1_22_bits_u, Multiplexer_4_to_1_22_bits_coefficeint, Demultiplexer_1_to_4_22_bits_coefficient.	3158	446	4436
Multiplexer และ Demultiplexer ในการเปลี่ยนเส้นทางของวงจร	10920	831	25196
รวม	108820	25236	262400

ตารางที่ 5.3 ผลการทดสอบการสูญเสียพลังงานของวงจร

Detect Active ที่ตรงกับตำแหน่ง Bank of Taps ที่ 16

	Average transition per iteration(ครั้ง)	Average glitch per iteration(ครั้ง)	Average power loss per iteration(วัตต์)
Carry propagate adder 19 bits	1371	1309	10474
Coefficient register 19 bits 4 taps	817	381	1629
Radix-4 booth multiplier 22 bits	38889	7462	99961
Multiplexer_4_to_1_19_bits	383	10	516
Demultiplexer_1_to_4_19_bits	1242	283	1884
รวม	42704	9447	114470

ตารางที่ 5.4 ผลการทดสอบการสูญเสียพลังงานของวงจร Threshold

	Average transition per iteration(ครั้ง)	Average glitch per iteration(ครั้ง)	Average power loss per iteration(วัตต์)
Carry propagate adder 22 bits_1	365	335	2751
Radix-4 booth multiplier 22 bits_1	9708	1891	25070
Sum_reg 22 bits_1	740	293	1816
Carry propagate adder 22 bits_2	366	333	2745
Radix-4 booth multiplier 22 bits_2	9712	1892	25079
Sum_reg 22 bits_2	739	292	1812
รวม	21633	5038	59276

หมายเหตุ ค่าการสูญเสียพลังงานที่ได้นี้เป็นค่าการสูญเสียพลังงานในหน่วย Watt. ที่อุณหภูมิ 100°C

ด้วย ค่าตัวเก็บประจุของเอาต์พุตเมื่อมีเพียง 1 Fanout (C_0) และ V_{dd}^2

5.5 สรุปและวิเคราะห์ผลการทดสอบ

จากผลการทดสอบในตารางที่ 1 และ 2 ซึ่งเป็นส่วนของวงจรกรองที่ตรงกับตำแหน่งผลตอบสนองเชิงเวลา ณ ตำแหน่งที่แอ็คทีฟและไม่แอ็คทีฟตามลำดับนั้น จะเห็นว่าจำนวนการเปลี่ยนแปลงของลอจิกหรือค่าการสูญเสียพลังงานของส่วนของวงจรกรองที่ตรงกับตำแหน่งผลตอบสนองเชิงเวลา ณ ตำแหน่งที่แอ็คทีฟ มีค่ามากกว่าเนื่องจากขนาดของสัญญาณที่เกี่ยวข้องภายในวงจร ณ ตำแหน่งแอ็คทีฟนั้นมีขนาดที่มากกว่าขนาดของสัญญาณภายในวงจร ณ ตำแหน่งที่ไม่แอ็คทีฟนั้นคือสัญญาณจะมีการเปลี่ยนแปลงของจำนวนบิตที่มากกว่านั่นเอง

เมื่อนำค่าการสูญเสียพลังงานต่อ 1 ตัวอย่างสัญญาณที่ได้จากตารางที่ 5.1 ถึง 5.4 คือ

P_1 มีค่าเท่ากับ 262400 วัตต์

P_2 มีค่าเท่ากับ 259360 วัตต์

P_3 มีค่าเท่ากับ 114470 วัตต์

P_{TH} มีค่าเท่ากับ 59276 วัตต์

โดยค่าการสูญเสียพลังงานที่ได้นี้เป็นค่าการสูญเสียพลังงานในหน่วยวัตต์ ที่ถูกนอมอลไลซ์ด้วย ค่าตัวเก็บประจุของเอาต์พุตเมื่อมีเพียง 1 Fanout (C_0) และ V_{dd}^2 รวมกับผลที่ได้จากแบบจำลอง Fixed-Point คือจำนวน Bank of Taps ของวงจรกรองหลังจากการตรวจสอบแอ็คทีฟแท็ป N_1 มีค่าเท่ากับ 12 โดยที่จำนวน Bank of Taps ของวงจรกรองก่อนการตรวจสอบแอ็คทีฟแท็ป N มีค่าเท่ากับ 32 กำหนดให้ n มีค่าเท่ากับ 15000 และ m มีค่าเท่ากับ 15000 เนื่องจากจำนวนแท็ปที่เหลืออยู่หลังจากการตรวจสอบแอ็คทีฟแท็ปเรียบร้อยแล้วนั้นมีจำนวน 12 แท็ปจากจำนวนแท็ปเดิม 32 แท็ป ซึ่งก็น่าจะสามารถประหยัดพลังงานได้ 62.5% แต่จากการคำนวณหาค่าประสิทธิภาพการสูญเสียพลังงานของวงจรกรองแบบที่มีการปรับอันดับเปรียบเทียบกับแบบที่ไม่มีการปรับอันดับตามสมการที่ (5.3) จะให้ประสิทธิภาพการประหยัดพลังงานเท่ากับ 26% ซึ่งจะเห็นว่ามีค่าน้อยกว่า 62.5% ทั้งนี้มีสาเหตุเนื่องมาจาก ต้องมีพลังงานบางส่วนที่สูญเสียไปในส่วนของวงจร Detect_Active_Tap และวงจร Threshold วงจรกรองแบบปรับอันดับสามารถประหยัดพลังงานได้ประมาณ 26% เมื่อเทียบกับวงจรกรองแบบที่ไม่มีการปรับอันดับ เนื่องจากเมื่อถึงเวลาที่ทำการตรวจสอบแอ็คทีฟแท็ป วงจรกรองแบบปรับอันดับจะหยุดการทำงานของแท็ปที่ไม่แอ็คทีฟทำให้เหลือไว้เฉพาะแท็ปที่แอ็คทีฟเท่านั้นที่ยังคงทำงานต่อไป ซึ่งจำนวนแท็ปที่เหลืออยู่หลังจากการตรวจสอบ

แอดิทิฟแท้ป็นมีน้อยเมื่อเทียบกับอันดับของวงจรกรองทำให้สามารถประหยัการสูญเสียพลังงานลงได้เมื่อเทียบกับวงจรกรองแบบที่ไม่มีการปรับอันดับซึ่งทุกแท้ปของวงจรกรองยังค้งทำงานตลอดเวลา

การจำลองการทำงานบนคอมพิวเตอร์ด้วยภาษา Verilog เพื่อเปรียบเทียบประสิทธิภาพทางด้านการปรับตัวและการสูญเสียพลังงานของวงจรกรองข้างต้น แสดงให้เห็นว่า การทำงานของวงจรที่ทำงานเฉพาะตอนที่สัญญาณขาเข้าของเครื่องช่วยฟังมีกำลังต่ำ และอาศัยการปรับอันดับของวงจรด้วย LMS Estimation via Structural Detection และใช้โครงสร้างของวงจรกรอง FIR แบบกึ่งมัลติเพล็กซ์และ Direct Form สามารถประหยัการสูญเสียพลังงานลงได้เมื่อเทียบกับวงจรกรองแบบที่ไม่มีการปรับอันดับ ทั้งนี้ขึ้นอยู่กับจำนวนแอดิทิฟแท้ปที่เหลือหลังจากการตรวจสอบแอดิทิฟแท้ป คือถ้าจำนวนแท้ปที่เหลือมีน้อยเมื่อเทียบกับอันดับของวงจรกรองก็จะสามารถทำให้สามารถประหยัพลังงานได้ นอกจากนี้ ข้อมูลจากการจำลองการทำงานข้างต้น แสดงให้เห็นว่าการปรับอันดับของวงจรกรองจะช่วยให้ค่า Asymptotic Performance ของวงจรดีขึ้นแต่ไม่มีผลต่ออัตราการใช้ของวงจร นอกจากนี้อัตราการใช้ของวงจรมันขึ้นอยู่กับกำลังของสัญญาณรบกวนที่เรำป้อนให้กับวงจรกรอง และเวลาที่ใช้ในการตรวจสอบแอดิทิฟแท้ปขึ้นอยู่กับค่า SNR

ขั้นตอนการดำเนินงานวิจัยนี้พบว่ การที่จะจำลองการสูญเสียพลังงานของวงจรกรองทุกส่วนนั้นในทางปฏิบัติเป็นไปได้ยากเนื่องจากข้อจำกัดเรื่องขนาดของโปรแกรมที่มีขนาดใหญ่ ดังนั้นเพื่อให้สามารถประมาณการสูญเสียพลังงานได้ จึงต้องทำการจำลองการสูญเสียพลังงานของวงจรโดยแบ่งเป็นส่วนย่อย และนำมาคำนวณประสิทธิภาพการสูญเสียพลังงานเทียบกับวงจรกรองแบบที่ไม่มีการปรับอันดับตามที่ได้กล่าวไว้ข้างต้น อย่างไรก็ตาม วิธีการประมาณโดยการจำลองดังกล่าวก็ยังต้องใช้เวลานานเกินกว่าที่จะใช้ในทางปฏิบัติทั่วไป

ข้อจำกัดอีกประการหนึ่งของการประมาณพลังงานสูญเสียด้วยการจำลองบนคอมพิวเตอร์ด้วย Verilog ดังกล่าว คือ ข้อสมมติฐานที่ผู้วิจัยตั้งขึ้นเพื่อให้การจำลองสำหรับประมาณพลังงานการสูญเสียด้วยโปรแกรม Verilog สามารถเป็นไปได้นั้น ถึงแม้สมมติฐานดังกล่าวเป็นไปได้ในทางทฤษฎี แต่ในแง่ปฏิบัติ เรายังจะต้องทดสอบความถูกต้องเทียบกับพลังงานสูญเสียจริงในการปฏิบัติ แต่เนื่องจากผู้วิจัยไม่สามารถหาเครื่องมือสำหรับวัดพลังงานที่สูญเสียจริงในวงจรต้นแบบได้ ทำให้ผู้วิจัยไม่สามารถเปรียบเทียบกับค่าพลังงานที่สูญเสียไปจริงเทียบกับค่าพลังงานสูญเสียที่ได้จากการประมาณโดยการจำลอง งานวิจัยนี้จึงยังไม่สามารถสรุปได้ว่าค่าที่ได้จากการประมาณมีความถูกต้องมากน้อยเพียงใด

บทที่ 6

ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign พร้อมการปรับช่วงก้าวแบบพลวัต

เนื้อหาตั้งแต่บทที่ 1 ถึงบทที่ 5 จะเป็นการกล่าวถึงการประยุกต์ใช้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภทดั้งเดิม (traditional LMS algorithm) ผู้วิจัยพบว่าขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดดังกล่าวนี้มีความซับซ้อนที่ไม่เหมาะสมในการสร้างจริงบนฮาร์ดแวร์ในรูปแบบของวงจรรวม VLSI เพราะต้องการพื้นที่บนซิลิคอนสูงในการสร้างจริง เนื่องจากความซับซ้อนของขั้นตอนวิธี นอกจากนี้ความซับซ้อนดังกล่าวยังทำให้ต้องสูญเสียพลังงานในการประมวลผลของขั้นตอนวิธีสูงอีกด้วย เพื่อลดความซับซ้อนทางฮาร์ดแวร์ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดดังกล่าวในบทนี้จะนำเอาหลักการของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign มาประยุกต์ใช้ในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว แต่การลดความซับซ้อนของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ทำให้ประสิทธิภาพในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวด้อยลง ดังนั้น เพื่อเพิ่มประสิทธิภาพให้กับขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign นี้ ผู้วิจัยจึงนำเอาหลักการปรับช่วงก้าวแบบพลวัต (Dynamic Adaptation of Step Size) ที่มีความซับซ้อนในการคำนวณต่ำมาช่วยปรับปรุงประสิทธิภาพเพื่อให้เหมาะสมกับการสร้างจริงบนฮาร์ดแวร์

6.1 ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign

โครงการวิจัยนี้จะนำขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign มาใช้เพื่อแก้ปัญหาคำถามการป้อนกลับทางเสียงในเครื่องช่วยฟัง ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ดังกล่าวมีรูปแบบสมการดังต่อไปนี้

$$e(n) = d(n) - \bar{w}^T(n)\bar{x}(n) \quad (6-1)$$

$$\bar{w}(n+1) = \bar{w}(n) + \mu [\text{sgn}(\bar{x}(n)) \text{sgn}(e(n))] \quad (6-2)$$

โดยที่

$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \quad (6-3)$$

สมการข้างต้นชี้ให้เห็นว่าการประมวลผลของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ประกอบด้วยการบวกเพียง $M + 1$ ครั้งเท่านั้น ความซับซ้อนของการคำนวณสำหรับขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign จึงน้อยกว่าขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดมาก อย่างไรก็ตามการลดความซับซ้อนในการคำนวณของขั้นตอนวิธี ทำให้สมรรถนะการปรับตัวของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ลดลง เพื่อแก้ปัญหาดังกล่าว ผู้วิจัยจึงนำหลักการของการปรับช่วงก้าวแบบพลวัต (Dynamic Adaptation of Step Size) เข้ามาช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ให้มีประสิทธิภาพดีขึ้นและสามารถนำไปใช้งานจริงได้ ซึ่งในหัวข้อต่อไปจะกล่าวถึงหลักการปรับช่วงก้าวแบบพลวัตวิธีต่าง ๆ รวมถึงวิธีการเลือกการปรับช่วงก้าวแบบพลวัตมาใช้งานจริง

6.2 วิธีการปรับช่วงก้าวแบบพลวัต

ค่าช่วงก้าว (Step Size) ใช้สำหรับกำหนดอัตราเร็วและความถูกต้องในการปรับตัวของขั้นตอนวิธี ซึ่งในขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดจะถูกกำหนดให้เป็นค่าคงที่ ในกรณีที่ช่วงก้าวมีขนาดใหญ่และเล็กจะมีประสิทธิภาพที่แตกต่างกันดังที่กล่าวมาแล้วข้างต้น ดังนั้นจึงมีผู้นำเสนอวิธีการปรับช่วงก้าวแบบพลวัตขึ้นมา เพื่อให้สามารถใช้งานได้มีประสิทธิภาพ ซึ่งขนาดของช่วงก้าวจะถูกกำหนดให้มีขนาดใหญ่ในขณะที่อยู่ห่างจากสภาวะคงตัว เพื่อให้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดเข้าสู่ด้วยความรวดเร็ว เมื่อเข้าสู่สภาวะคงตัวแล้ว ขนาดของช่วงก้าวจะถูกลดลงเพื่อเพิ่มความถูกต้องในการประมาณค่าสัมประสิทธิ์ของวงจรกรอง ในที่นี้จะกล่าวถึงวิธีการปรับช่วงก้าวแบบพลวัต 4 วิธีด้วยกันคือ วิธีการปรับช่วงก้าวแบบ Squared Error วิธีการปรับช่วงก้าวแบบ Normalized Step-Size วิธีการปรับช่วงก้าวแบบ Cross Correlation และ วิธีการปรับช่วงก้าวแบบ Delta-Bar-Delta

6.2.1 วิธีการปรับช่วงก้าวแบบความผิดพลาดกำลังสอง

Hwong และ Johnston เสนอความคิดว่าควรปรับขนาดช่วงก้าวตามค่าความผิดพลาดกำลังสองเฉลี่ย [9] เมื่อค่าความผิดพลาดกำลังสองเฉลี่ยมีค่ามาก ค่าสัมประสิทธิ์ของวงจรกรองจะต่างจากค่าสัมประสิทธิ์ที่สภาวะออสซิลเลชันมาก จึงควรใช้ค่าช่วงก้าวขนาดใหญ่เพื่อให้ค่าสัมประสิทธิ์ของวงจรกรองเข้าสู่ค่าสัมประสิทธิ์ที่สภาวะออสซิลเลชันเร็วขึ้น ในทางตรงกันข้ามเมื่อค่าความผิดพลาดกำลังสองเฉลี่ยมีค่าน้อยแสดงว่าค่าสัมประสิทธิ์ของวงจรกรองใกล้เคียงกับค่าสัมประสิทธิ์ที่สภาวะออสซิลเลชันก็ควรให้ค่าช่วงก้าวมีขนาดเล็ก เพื่อให้การปรับค่าสัมประสิทธิ์ของวงจรกรองมีความถูกต้องมากขึ้น ซึ่งหลักการดังกล่าวมีรูปแบบสมการดังนี้

$$\mu'(n+1) = \alpha \mu(n) + \gamma e^2(n) \quad (6-4)$$

$$\mu(n+1) = \begin{cases} \mu_{\max}, & \mu'(n+1) > \mu_{\max} \\ \mu_{\min}, & \mu'(n+1) < \mu_{\min} \\ \mu'(n+1), & \text{others} \end{cases} \quad (6-5)$$

เมื่อ $e(n)$ คือ ค่าความผิดพลาดของวงจรกรองที่เวลา n , $0 < \alpha < 1$, $\gamma > 0$ และ $0 < \mu_{\min} < \mu_{\max}$ โดยที่ $\mu_{\min}, \mu_{\max}$ คือ ค่าคงตัวแทนขนาดช่วงก้ำวที่เล็กที่สุดและขนาดช่วงก้ำวที่ใหญ่ที่สุดตามลำดับ ซึ่งค่าดังกล่าวมีไว้เพื่อให้ขนาดของช่วงก้ำวอยู่ในช่วงที่สามารถส่งผลให้มีการปรับค่าสัมประสิทธิ์ของวงจรกรองได้ และมีไว้เพื่อป้องกันไม่ให้ขนาดของช่วงก้ำวมีขนาดใหญ่จนทำให้การปรับค่าสัมประสิทธิ์ของวงจรกรองขาดเสถียรภาพตามลำดับ

6.2.2 วิธีการปรับช่วงก้ำวแบบ Normalized Step-Size

Honig และ Messerschmitt บรรยายวิธีการปรับขนาดช่วงก้ำว โดยการหารขนาดช่วงก้ำวด้วยกำลังของสัญญาณขาเข้า [10] ซึ่งหลักการดังกล่าวมีรูปแบบสมการดังนี้

$$\mu(n) = \frac{a}{\sigma^2(n) + b} \quad (6-6)$$

เมื่อ $\mu(n)$ เป็น ขนาดช่วงก้ำวที่เวลา n โดยที่ a และ b คือ ค่าคงตัวที่เป็นค่าบวก (ค่าคงตัว b มีไว้เพื่อป้องกันไม่ให้ขนาดของช่วงก้ำวมีขนาดใหญ่เกินไป เมื่อกำลังของสัญญาณขาเข้ามีค่าน้อยมาก ๆ) และ $\sigma^2(n)$ คือ ค่าประมาณของกำลังของสัญญาณขาเข้าที่เวลา n ซึ่งนิยมใช้ค่าประมาณกำหนดดังสมการ

$$\sigma^2(n) = \alpha \sigma^2(n-1) + (1-\alpha)x^2(n) \quad (6-7)$$

โดยที่ α เป็นค่าคงตัวที่เป็นค่าบวกและน้อยกว่าหนึ่งและ $x(n)$ คือ สัญญาณขาเข้าของวงจรกรองที่เวลา n

6.2.3 วิธีการปรับช่วงก้ำวแบบ Cross Correlation

Karni และ Zeng ใช้ค่าสหสัมพันธ์ไขว้ (Cross Correlation) ระหว่างค่าความผิดพลาดกับสัญญาณขาเข้าของวงจรกรองเป็นตัวกำหนดขนาดช่วงก้ำว [11] ทั้งนี้เพราะมีทฤษฎีอยู่ว่าที่สถานะคงตัวค่าสหสัมพันธ์ไขว้ระหว่างค่าความผิดพลาดกับสัญญาณขาเข้าของวงจรกรองจะมีค่าเป็นศูนย์ ดังนั้นถ้าค่าสหสัมพันธ์ไขว้ระหว่างค่าความผิดพลาดกับสัญญาณขาเข้ามีค่ามาก แสดงว่าค่า

สัมประสิทธิ์ของวงจรกรองยังคงอยู่ห่างจากสถานะคงตัวจึงควรใช้ค่าของขนาดช่วงก้ำวที่มีขนาดใหญ่ และถ้าค่าสหสัมพันธ์ไขว้ระหว่างค่าความผิดพลาดและสัญญาณขาเข้ามีค่าน้อย ควรจะใช้ค่าของขนาดช่วงก้ำวที่มีขนาดเล็ก ซึ่งหลักการดังกล่าวมีรูปแบบของสมการดังนี้

$$C = \|e(n)\bar{x}(n)\|^2 \quad (6-8)$$

$$\mu(n+1) = \mu_{\max}(1 - \exp(-\alpha C)) \quad (6-9)$$

เมื่อ $e(n)$ คือ ค่าความผิดพลาดที่เวลา n และ $\bar{x}(n)$ คือ สัญญาณขาเข้าของวงจรกรองที่เวลา n โดยที่ α คือ ค่าคงตัวที่มากกว่าศูนย์

6.2.4 วิธีการปรับช่วงก้ำวแบบ Delta-Bar-Delta

Jacobs ได้เสนอความคิดในการปรับขนาดของช่วงก้ำวไว้ว่า ขนาดของช่วงก้ำวที่เหมาะสมกับค่าสัมประสิทธิ์ของวงจรกรองแต่ละตัวไม่จำเป็นต้องเท่ากัน [12] ค่าสัมประสิทธิ์แต่ละตัวควรมีขนาดช่วงก้ำวเป็นของตนเอง และขนาดช่วงก้ำวแต่ละตัวควรจะสามารถปรับเปลี่ยนค่าได้ โดยสังเกตว่าถ้าค่าสัมประสิทธิ์เพิ่มขึ้นติดต่อกันหรือลดลงติดต่อกันก็ควรที่จะเพิ่มขนาดช่วงก้ำว ในทางกลับกันถ้าค่าสัมประสิทธิ์ของวงจรกรองเพิ่มขึ้นและลดลงสลับกันก็ควรที่จะลดขนาดช่วงก้ำว เนื่องจากค่าสัมประสิทธิ์แต่ละตัวมีขนาดช่วงก้ำวเป็นของตนเอง ทำให้การปรับค่าสัมประสิทธิ์ไม่ใช้การปรับในทิศทางที่ตรงกันข้ามกับเกรเดียนต์ของค่าความผิดพลาดกำลังสองเฉลี่ย แต่เป็นการปรับค่าสัมประสิทธิ์ที่อาศัยการหาอนุพันธ์ย่อยของค่าความผิดพลาดกำลังสองเฉลี่ยเทียบกับค่าสัมประสิทธิ์แต่ละตัว ซึ่งหลักการดังกล่าวมีรูปแบบของสมการดังนี้

$$\mu_i(n+1) = \begin{cases} \mu_i(n) + k, & \bar{\delta}_i(n-1)\delta_i(n) > 0 \\ \mu_i(n) - \phi\mu_i(n), & \bar{\delta}_i(n-1)\delta_i(n) < 0 \\ \mu_i(n), & \text{others} \end{cases} \quad (6-10)$$

$$\bar{\delta}_i(n) = \alpha\bar{\delta}_i(n-1) + (1-\alpha)\delta_i(n) \quad (6-11)$$

$$\delta_i(n) = e(n)x_i(n) \quad (6-12)$$

โดยที่ k , ϕ และ α คือ ค่าคงตัวที่เป็นค่าบวก ($\alpha < 1$) และ $\delta_i(n)$ คือ ค่าประมาณของอนุพันธ์ย่อยของค่าความผิดพลาดกำลังสองเฉลี่ยเทียบกับค่าสัมประสิทธิ์ของวงจรกรองตัวที่ i ที่เวลา n ส่วน $\bar{\delta}_i(n)$ คือ ค่าเฉลี่ยของ $\delta_i(n)$

จากสมการที่ (6-10) หมายความว่าขนาดของช่วงก้ำตัวที่ i จะเพิ่มขึ้นครั้งละ k ถ้า $\bar{\delta}_i(n)$ มีเครื่องหมายเหมือนกับ $\delta_i(n)$ และขนาดช่วงก้ำตัวที่ i จะลดลง $\phi\mu_i(n)$ ทุก ๆ ครั้งที่ $\bar{\delta}_i(n)$ มีเครื่องหมายต่างกับ $\delta_i(n)$ ทั้งนี้การที่เพิ่มขนาดของช่วงก้ำขึ้นแบบเชิงเส้น แต่ลดขนาดของช่วงก้ำลงแบบเอกซ์โพเนนเชียล เพื่อเป็นการป้องกันไม่ให้น้ำหนักของช่วงก้ำมีขนาดใหญ่เกินไป หรือเพิ่มขึ้นเร็วเกินไป เพราะการลดขนาดของช่วงก้ำแบบนี้ทำให้น้ำหนักของช่วงก้ำสามารถลดลงได้อย่างรวดเร็ว อีกทั้งยังแน่ใจได้ว่าขนาดของช่วงก้ำจะเป็นบวกตลอดเวลา

ตารางที่ 6.1 เปรียบเทียบความซับซ้อนในการคำนวณของวิธีการปรับช่วงก้ำแบบต่าง ๆ

วิธีการปรับขนาด ช่วงก้ำ	Squared Error	Normalized Step-Size	Cross Correlation	Delta-Bar-Delta
การบวก	1 ครั้ง	2 ครั้ง	$M-1$ ครั้ง	$M+(M/3)$ ครั้ง
การลบ	-	1 ครั้ง	1 ครั้ง	$1+(M/3)$ ครั้ง
การคูณ	3 ครั้ง	3 ครั้ง	$M+4$ ครั้ง	$4M+(M/3)$ ครั้ง
การหาร	-	1 ครั้ง	-	-
การเปรียบเทียบ	2 ครั้ง	-	-	M ครั้ง
ค่าคงตัว	4 ตัว	3 ตัว	2 ตัว	3 ตัว
ตัวแปร	-	1 ตัว	-	M ตัว
ฟังก์ชัน Exponential	-	-	1 ครั้ง	-

จากวิธีการปรับช่วงก้ำแบบพลวัตทั้ง 4 วิธีที่นำเสนอมาแล้วข้างต้นนี้ สามารถสรุปกระบวนการคำนวณของทั้ง 4 วิธีได้ดังตารางที่ 6.1 เนื่องจากจุดประสงค์ในการนำขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign มาประยุกต์ใช้ก็เพื่อลดความซับซ้อนของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุด ดังนั้นการเลือกวิธีการปรับช่วงก้ำแบบพลวัตที่จะนำมาใช้ในการปรับปรุงประสิทธิภาพ

ภาพของขั้นตอนวิธีกำลังสองเหลี่ยน้อยสุดประเภท Sign-Sign ก็ควรจะเป็นวิธีการปรับช่วงก้ำวที่ไม่ซับซ้อน ทั้งนี้เพื่อให้ฮาร์ดแวร์ที่ใช้สร้างจริงขั้นตอนวิธีกำลังสองเหลี่ยน้อยสุดประเภท Sign-Sign พร้อมด้วยกลไกในการปรับค่าช่วงก้ำวมีความซับซ้อนไม่สูงเกินไป งานวิจัยนี้จึงเลือกใช้วิธีการปรับช่วงก้ำวแบบ Squared Error ที่นำเสนอโดย Kwong และ Johnston [9] มาใช้ในการสร้างจริงวงจรกรองปรับตัวที่ใช้สำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง

การจำลองการทำงานของวงจรกรอง LMS แบบ Sign-Sign บนคอมพิวเตอร์

เนื้อหาในบทนี้กล่าวถึงการจำลองการทำงานของวงจรกรองปรับตัวที่ใช้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign โดยมีวัตถุประสงค์เพื่อวิเคราะห์ห้ออกแบบฮาร์ดแวร์สำหรับสร้างวงจรกรองเพื่อแก้ปัญหาการเกิดเสียงฮอนเนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง สำหรับการจำลองในบทนี้ ผู้วิจัยจะใช้แบบจำลองของปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟังแบบไม่ต่อเนื่องตามที่อธิบายในหัวข้อ 3.1 (แผนภาพกล่องดังแสดงในรูปที่ 3.1) ซึ่งการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวจะทำการปรับค่าสัมประสิทธิ์เฉพาะช่วงที่เสียงเบามาก (ดังหลักการที่แสดงในรูปที่ 3.2) การปรับค่าสัมประสิทธิ์ของวงจรกรองในช่วงดังกล่าวจะเข้าสู่สภาวะออปติ멈ได้ง่ายกว่าการปรับค่าสัมประสิทธิ์ในช่วงที่มีเสียงดัง สำหรับแบบจำลองของช่องสัญญาณในปัญหาการป้อนกลับทางเสียงในเครื่องช่วยฟังจะใช้ผลตอบสนองต่ออิมพัลส์ (Impulse Response) ของเส้นทางป้อนกลับดังในรูปที่ 2.4

การจำลองบนคอมพิวเตอร์ข้างต้นนี้จะถูกนำไปใช้ในการวิเคราะห์และพิจารณาในสองรูปแบบ คือ ทั้งการประมวลผลแบบ Floating Point Arithmetic และ Fixed Point Arithmetic ตามลำดับด้วยโปรแกรม Matlab หลังจากนั้นจึงนำผลที่ได้จากการจำลองแบบ Fixed Point Arithmetic มาวิเคราะห์เพื่อกำหนดรายละเอียด (Resolution) จำนวนบิตของข้อมูลในแต่ละขั้นตอนที่จะใช้ในการสร้างจริงของวงจรกรองปรับตัวบน FPGA ด้วยภาษา VHDL

7.1 การจำลองผลโดยใช้การคำนวณแบบ Floating Point Arithmetic

ตามที่ได้กล่าวมาแล้วก่อนหน้านี้ที่ว่า โครงการวิจัยนี้จะทำการสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงฮอนเนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ซึ่งจะนำเอาขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign มาประยุกต์ใช้เพื่อลดความซับซ้อนของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุด ซึ่งในแต่ละรอบของการสุ่มตัวอย่างต้องใช้ในการคำนวณที่ประกอบด้วยการคูณ $M + 1$ ครั้ง และการบวก $M + 1$ ครั้ง ให้เหลือเพียงการบวก $M + 1$ ครั้งเท่านั้น นอกจากนี้ในงานวิจัยนี้จะนำเอาการปรับช่วงก้ำวแบบพลวัตมาช่วยเพิ่มประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ซึ่งจะใช้วิธีการปรับช่วงก้ำวแบบ Squared Error ที่มีหลักการไม่ซับซ้อนมากนัก การจำลองการทำงานในหัวข้อนี้จะแบ่งเป็น 2 ส่วน ดังนี้ คือ การจำลองเพื่อเปรียบเทียบประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมกับขั้นตอน

วิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign และใน ส่วนที่ 2 การจำลองการทำงานจะแสดงให้เห็นถึงการนำวิธีการปรับช่วงก้าวแบบพลวัตมาใช้เพื่อปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign

7.1.1 การจำลองการทำงานของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมและขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign

จากแบบจำลองของปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟังแบบไม่ต่อเนื่อง ดังแสดงในรูปที่ 3.1 ซึ่งจะทำให้การปรับค่าสัมประสิทธิ์เฉพาะช่วงที่เสียงเบามาก (สัญญาณขาเข้ามีกำลังต่ำ) โดยที่สัญญาณต่าง ๆ ที่จะนำมาใช้ในการวิเคราะห์และพิจารณาจะมีข้อกำหนดตามหัวข้อที่ 3.1 คือ

สัญญาณสุ่ม $N_p(n)$ ที่ช่วยในการปรับตัวของวงจรกรองเป็นสัญญาณสุ่มแบบ White Gaussian มีค่าเฉลี่ยเป็นศูนย์และมีค่าความแปรปรวนเท่ากับ 0.1 นั่นคือ

$$\sigma_{N_p(n)}^2 = 0.1 \quad \text{และ} \quad \sqrt{\sigma_{N_p(n)}^2} = 0.3162278 \quad (7-1)$$

โดยทั่วไป 3 เท่า ของส่วนเบี่ยงเบนมาตรฐาน(Standard Deviation) ทั้งทางด้านบวกและลบของค่าเฉลี่ยจะครอบคลุมขนาดของสัญญาณสุ่มได้ร้อยละ 99.73 ของสัญญาณทั้งหมด เพราะฉะนั้น เราจะกำหนดให้ขนาดของสัญญาณสุ่มอยู่ในวิสัย

$$-0.94868 < N_p(n) < 0.94868 \quad (7.2)$$

ในส่วนของสัญญาณขาเข้า $s(n)$ ที่นำมาพิจารณาจะทำการพิจารณาในช่วงที่สัญญาณขาเข้ามีกำลังต่ำ ซึ่งสัญญาณขาเข้าเป็นสัญญาณแบบ White Gaussian มีค่าเฉลี่ยเป็นศูนย์และมีค่าความแปรปรวนเท่ากับ 0.1 เท่าของสัญญาณสุ่ม $N_p(n)$ นั่นคือ สัญญาณขาเข้า $s(n)$ มีค่าความแปรปรวนเท่ากับ 0.01

คุณสมบัติของสัญญาณขาเข้า $s(n)$ ของเครื่องช่วยฟังมีคุณสมบัติเหมือนกับสัญญาณสุ่ม $N_p(n)$ ซึ่งในแบบจำลองนี้จะทำการพิจารณาเฉพาะช่วงที่เสียงเบามาก (สัญญาณขาเข้ามีกำลังต่ำ) คือ พิจารณาเฉพาะช่วงที่สัญญาณขาเข้า $s(n)$ มีความแปรปรวนเท่ากับ 0.01 นั่นคือ

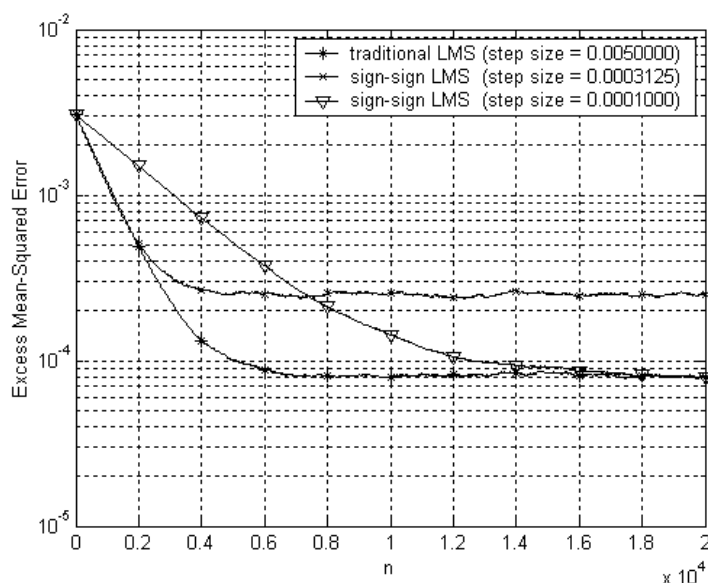
$$\sigma_{s(n)}^2 = 0.01 \quad \text{และ} \quad \sqrt{\sigma_{s(n)}^2} = 0.1 \quad (7-3)$$

เช่นเดียวกันกับกรณีของ $N_p(n)$ เรากำหนดให้ขนาดของสัญญาณ $s(n)$ มีขนาด

$$-0.3 < s(n) < 0.3 \quad (7-4)$$

คุณสมบัติของสัญญาณสุ่ม $N_p(n)$ และสัญญาณขาเข้า $s(n)$ ที่กล่าวมาแล้วข้างต้นนี้จะถูกนำมาใช้ในการวิเคราะห์และพิจารณาในการจำลองผลการทำงานเชิงโปรแกรมที่ใช้การคำนวณแบบ Floating Point Arithmetic การจำลองผลการทำงานเชิงฮาร์ดแวร์ที่ใช้การคำนวณแบบ Fixed Point Arithmetic และการสร้างจริงวงจรกรองปรับตัวบน FPGA

รูปที่ 7.1 แสดงให้เห็นถึงผลที่ได้จากการใช้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมเปรียบเทียบกับขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ในการแก้ปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟัง การจำลองการทำงานดังกล่าวอาศัยโปรแกรม Matlab และใช้การคำนวณแบบ Floating Point Arithmetic สำหรับวงจรกรองปรับตัวแบบทรานส์เวอร์อ์อันดับ 32 อันดับของวงจรกรองนี้ถูกกำหนดให้มีขนาดเท่ากับอันดับของผลตอบสนองต่ออิมพัลส์ของเส้นทางป้อนกลับในรูปที่ 2.4



รูปที่ 7.1 ผลการจำลองของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมเปรียบเทียบกับขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign สำหรับปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟัง

รูปกราฟข้างต้นแสดงค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกิน (Excess Mean-Squared Error) ที่ตำแหน่งเวลา n ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมเปรียบเทียบกับขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ค่าช่วงก้าวของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมกำหนดให้เป็น $(\mu = 0.005)$ และของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-

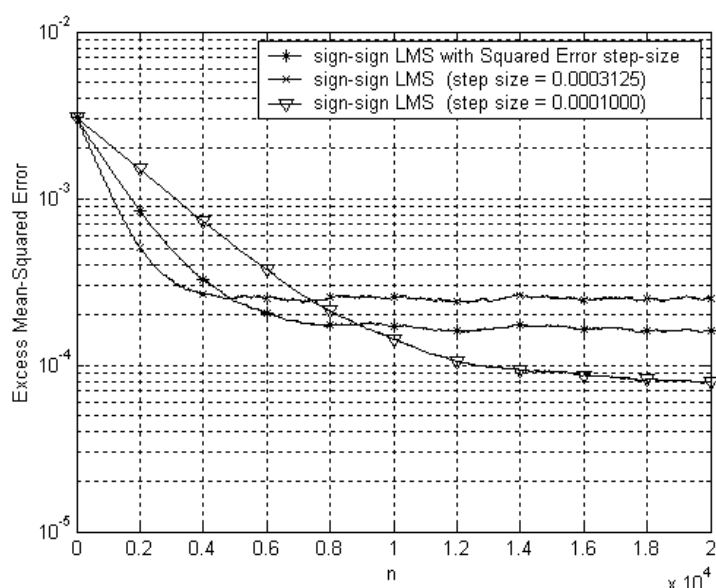
Sign กำหนดให้เป็น $\frac{1}{16}$ และ $\frac{1}{50}$ ของค่าช่วงก้าวของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมตามลำดับ กราฟแต่ละเส้นเป็นผลเฉลี่ยที่ได้จากการจำลองทั้งหมด 100 การทดลองที่เป็นอิสระจากกัน

จากเส้นกราฟดังกล่าว จะเห็นว่าค่าช่วงก้าวที่ใช้ในขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ทั้งสองกรณีมีอัตราการปรับตัวที่แตกต่างกัน ในกรณีแรกที่ค่าช่วงก้าวมีค่าเท่ากับ 1 ใน 16 ของค่าช่วงก้าวของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิม อัตราการปรับตัวจะสูงแต่ความถูกต้องที่สภาวะคงตัวจะต่ำ อัตราการลดลงของค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกินของทั้งสองขั้นตอนวิธีประเภท Sign-Sign มีค่าใกล้เคียงกันกับอัตราของขั้นตอนวิธีแบบดั้งเดิมเมื่อ $n < 2000$ ส่วนค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกินขั้นตอนวิธีประเภท Sign-Sign สูงเป็นประมาณ 3 เท่าของค่าของขั้นตอนวิธีแบบดั้งเดิม สำหรับกรณีที่สองที่ค่าช่วงก้าวมีค่าเท่ากับ 1 ใน 50 ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิม อัตราการปรับตัวจะต่ำแต่ความถูกต้องที่สภาวะคงตัวจะสูง โดยค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกินจะใกล้เคียงกันค่าของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมเมื่อ $n > 16,000$

จากรูปที่ 7.1 แสดงให้เห็นอย่างชัดเจนแล้วว่า ผลของการลดความซับซ้อนในการคำนวณของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดมาเป็นขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ทำให้ประสิทธิภาพในการปรับตัวของขั้นตอนวิธีลดลง เพื่อเพิ่มประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign จึงได้นำเอาการปรับช่วงก้าวแบบพลวัตแบบ Squared error มาประยุกต์ใช้ ผลการจำลองของการนำเอาวิธีการปรับช่วงก้าวแบบ Squared error มาช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign จะแสดงให้เห็นในหัวข้อต่อไป

7.1.2 การจำลองผลของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign โดยใช้การปรับช่วงก้าวแบบพลวัต

ในหัวข้อนี้จะทำการจำลองผลทางคอมพิวเตอร์ โดยการนำเอาวิธีการปรับช่วงก้าวแบบพลวัตแบบ Squared error มาช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign เพื่อใช้ในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว เพื่อใช้สำหรับแก้ปัญหาการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง



รูปที่ 7.2 ผลการจำลองของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign เปรียบเทียบกับ ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ที่ใช้การปรับช่วงก้าวแบบ Squared error สำหรับปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟัง

จากผลการจำลองทางคอมพิวเตอร์ในรูปที่ 7.2 จะเห็นว่าเส้นกราฟของค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกินในช่วงแรก of ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ที่ใช้การปรับช่วงก้าวแบบ Squared error มาช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธี จะมีค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกินสูงกว่าขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ที่ใช้ค่าช่วงก้าวเท่ากับ 1 ใน 16 ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดเล็กน้อยแต่ที่สภาวะคงตัวจะต่ำกว่าประมาณ 1.5 เท่า และเมื่อเปรียบเทียบกับเส้นกราฟของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ที่มีค่าช่วงก้าวเท่ากับ 1 ใน 50 ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุด ค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกินของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ที่ใช้การปรับช่วงก้าวแบบค่าผิดพลาดกำลังสองจะมีค่าสูงกว่าเล็กน้อยแต่จะเข้าสู่สภาวะคงตัวเร็วกว่ามาก (ประมาณหลังจาก 10000 ครั้งของการสุ่มตัวอย่าง) สำหรับค่าคงตัวของวิธีการปรับช่วงก้าวแบบค่าผิดพลาดกำลังสองที่นำมาใช้มีค่าดังนี้คือ α มีค่าเท่ากับ 0.69, γ มีค่าเท่ากับ 0.005, $\mu_{\min}$ มีค่าเท่ากับ 0.0001 และ $\mu_{\max}$ มีค่าเท่ากับ 0.0005 โดยที่กราฟแต่ละเส้นเป็นผลเฉลี่ยที่ได้จากการจำลองทั้งหมด 100 การทดลองที่เป็นอิสระจากกัน

จากรูปที่ 7.2 แสดงให้เห็นว่า การนำเอาวิธีการปรับช่วงก้าวแบบพลวัตมาใช้ เพื่อปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign สามารถช่วยให้ประสิทธิภาพของขั้นตอนวิธีดังกล่าวดีขึ้น ดังนั้นสามารถสรุปได้ว่าการนำเอาวิธีการปรับช่วงก้าวแบบ

Squared error มาใช้ เพื่อช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign ที่นำมาใช้แทนขั้นตอนวิธีกำลังสองเลี่ยน้อยสุด เพื่อลดความซับซ้อนของขั้นตอนวิธี ทำให้ได้ผลลัพธ์ที่ดีขึ้นจริง เมื่อเปรียบเทียบกับการใช้ขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign แต่เพียงอย่างเดียว

จากผลการจำลองการทำงานของแบบจำลองการแก้ปัญหการเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังแบบไม่ต่อเนื่อง ในส่วนของการจำลองที่ใช้การคำนวณแบบ Floating Point Arithmetic ถึงแม้ว่าประสิทธิภาพของขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign จะดีกว่าขั้นตอนวิธีกำลังสองเลี่ยน้อยสุด แต่เมื่อนำเอาการปรับช่วงก้ำวแบบพลวัตเข้ามาใช้เพื่อช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign ทำให้ประสิทธิภาพที่ได้ดีขึ้น แต่ความซับซ้อนในการคำนวณยังคงน้อยกว่าขั้นตอนวิธีกำลังสองเลี่ยน้อยสุด ซึ่งเป็นสิ่งที่แสดงให้เห็นถึงประสิทธิภาพในการนำเอาขั้นตอนวิธีดังกล่าวและวิธีการปรับช่วงก้ำวแบบค่าผิดพลาดกำลังสองไปประยุกต์ใช้งานจริง

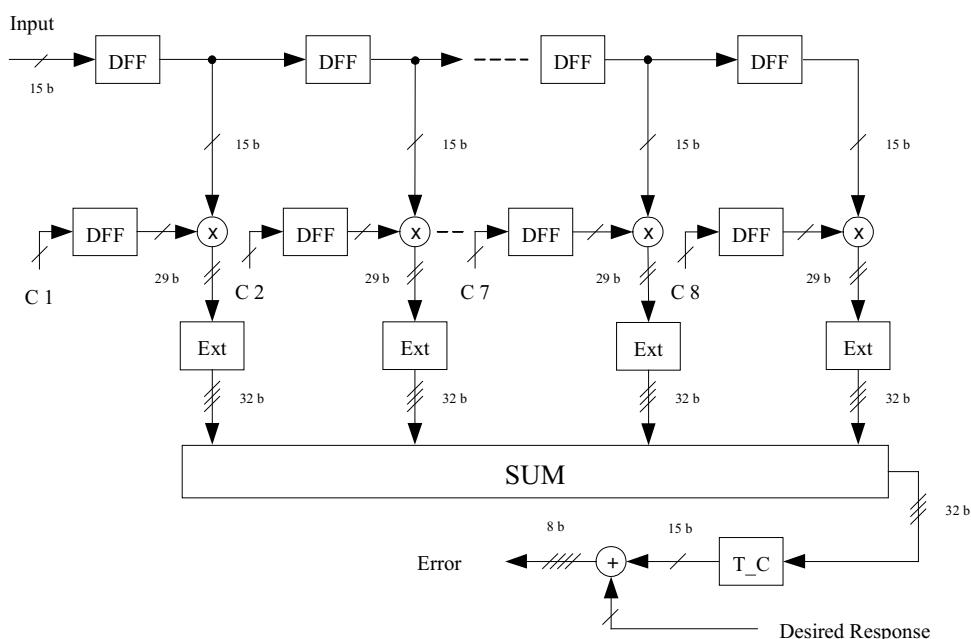
7.2 รายละเอียดการออกแบบบิตข้อมูล

จากผลการจำลองทางคอมพิวเตอร์ในหัวข้อที่ 7.1 แสดงให้เห็นแล้วว่าสามารถนำเอาขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign ที่ใช้การปรับช่วงก้ำวแบบพลวัตด้วย Squared error มาช่วยในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวช่วยเพิ่มประสิทธิภาพในการทำงานอยู่ในระดับที่ดี ถึงแม้ว่าประสิทธิภาพจะดีกว่าขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดแบบดั้งเดิมก็ตาม แต่ก็ทำให้ความซับซ้อนของขั้นตอนวิธีลดลงอย่างมาก ซึ่งจะเป็นประโยชน์ในการสร้างจริงที่จะช่วยลดความซับซ้อนในการสร้างจริง รวมถึงลดพื้นที่ในการสร้างจริงและยังส่งผลถึงการใช้นำงานที่จะลดลงในการปรับลดความซับซ้อนของขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดลง

อย่างไรก็ตามผลการจำลองทางคอมพิวเตอร์ที่ผ่านมาใช้การคำนวณแบบ Floating Point Arithmetic ในการประมวลผลของแต่ละขั้นตอน แต่เนื่องจากในระบบการทำงานจริงของฮาร์ดแวร์ การประมวลผลในขั้นตอนต่าง ๆ จะใช้การคำนวณในระดับบิตซึ่งจะใช้การคำนวณในรูปแบบของเลขฐานสองเพื่อประหยัดฮาร์ดแวร์ ค่าของสัญญาณที่ใช้ในการคำนวณจะแทนอยู่ในรูปแบบ Fixed Point ดังนั้นการจำลองผลการทำงานที่ถูกต้องควรจะทำจำลองผลในรูปแบบเชิงฮาร์ดแวร์ โดยที่การประมวลผลในแต่ละขั้นตอนจะต้องทำการประมวลผลในรูปแบบของเลขฐานสองแบบ Fixed Point Arithmetic เพื่อจำลองผลการทำงานที่ได้ซึ่งจะเกิดขึ้นจริง

สำหรับการวิเคราะห์ในการออกแบบการทำงานของวงจรกรองปรับตัวที่จะทำการสร้างจริงที่ใช้การคำนวณแบบ Fixed Point Arithmetic ซึ่งจะกล่าวถึงต่อไป จะแบ่งเป็น 2 ส่วน คือ ส่วนแรกจะทำการวิเคราะห์เกี่ยวกับวงจรกรอง และส่วนที่สองจะทำการวิเคราะห์ในส่วนของการปรับค่าสัมประสิทธิ์

7.2.1 รายละเอียดคร่าวๆของค่าสัมประสิทธิ์ในส่วนต่างๆของวงจรกรองปรับตัว



รูปที่ 7.3 รายละเอียดของวงจรกรองปรับตัวที่ใช้ในการสร้างจริง

รูปที่ 7.3 แสดงวงจรกรองปรับตัวที่จะนำมาใช้ในการสร้างจริง เพื่อใช้แก้ปัญหาการเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ซึ่งทำการออกแบบจากแบบจำลองของปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟังแบบไม่ต่อเนื่องในรูปที่ 3.1 โดยที่การสร้างจริงวงจรกรองปรับตัวจะทำการสร้างจริงบน FPGA ซึ่งมีข้อจำกัดทางด้านพื้นที่ จำนวน Gate และทรัพยากรต่างๆ ของชิป FPGA ดังนั้นการสร้างจริงวงจรกรองปรับตัวจะทำการสร้างจริงวงจรกรองปรับตัวอันดับ 8 เพื่อให้สามารถนำวงจรที่ออกแบบด้วยภาษา VHDL มาทำการสร้างจริงและสามารถใช้งานได้จริงบน FPGA

ส่วนประกอบต่างๆ ในรูปที่ 7.3 ประกอบด้วย D ฟลิปฟล็อป สัญญาณขาเข้าหรือสัญญาณสุ่ม วงจรคูณ ตัวขยายจำนวนบิตข้อมูล (Ext) วงจรบวก (SUM) ส่วน Truncate ข้อมูลและ Complement (T_C) จากนั้นก็เป็นส่วนของสัญญาณที่ต้องการ (Desired Response) และค่าความผิดพลาด (Error) จากแบบจำลองในรูปที่ 7.3 Desired Response จะประกอบด้วยสัญญาณสามสัญญาณ

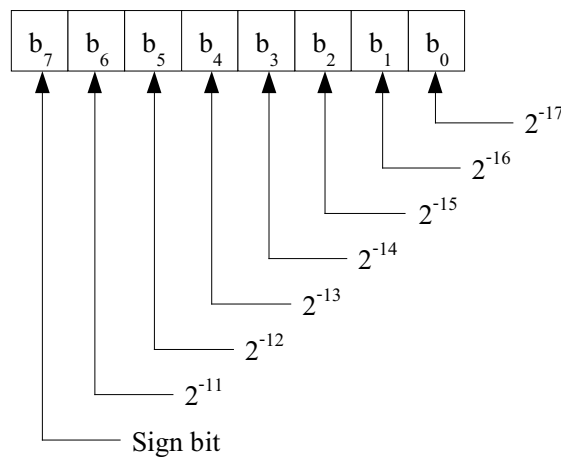
คือ สัญญาณเสียงที่เกิดจากการป้อนกลับรวมกับสัญญาณขาเข้า $s(n)$ และหักล้างกับสัญญาณที่สร้างจากวงจรกรองปรับตัวทำให้ได้ค่าความผิดพลาดที่เวลา n ใด ๆ

การออกแบบวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงหอนที่จะใช้ในการสร้างจริงบน FPGA ในที่นี้จะใช้แบบจำลองของปัญหาการลดการป้อนกลับทางเสียงในเครื่องช่วยฟังแบบไม่ต่อเนื่องเป็นต้นแบบในการสร้างจริง โดยจะเริ่มพิจารณาที่สัญญาณต่าง ๆ ที่ได้ทำการกำหนดคุณสมบัติเอาไว้แล้ว เริ่มต้นจากสัญญาณสุ่ม $N_p(n)$ ที่ป้อนให้กับวงจรกรองปรับตัวสำหรับในรูปที่ 7.3 แทนสัญญาณสุ่มด้วยสัญญาณอินพุต (Input) ซึ่งมีคุณสมบัติดังที่กล่าวมาแล้วก่อนหน้านี้ คือ มีความแปรปรวนเท่ากับ 0.1 ซึ่งจะทำให้สามารถหาส่วนเบี่ยงเบนมาตรฐานของสัญญาณได้ สุดท้ายสามารถประมาณค่าขนาดของสัญญาณสูงสุดและต่ำสุดได้ ซึ่งข้อกำหนดดังกล่าวมีส่วนในการพิจารณาจำนวนบิตที่จะนำมาใช้ในการแทนขนาดของสัญญาณ สัญญาณที่ถูกป้อนเข้าวงจรกรองปรับตัวจะถูกนำไปคูณเข้ากับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว เนื่องจากค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่จะทำการสร้างจริงสำหรับแก้ปัญหการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังจะใช้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign และวิธีการปรับช่วงก้ำวแบบ Squared error ซึ่งวิธีการปรับช่วงก้ำวดังกล่าวมีสมการดังนี้

$$\mu'(n+1) = \alpha \mu(n) + \gamma e^2(n) \quad (7-5)$$

$$\mu(n+1) = \begin{cases} \mu_{\max}, & \mu'(n+1) > \mu_{\max} \\ \mu_{\min}, & \mu'(n+1) < \mu_{\min} \\ \mu'(n+1), & \text{Others} \end{cases} \quad (7-6)$$

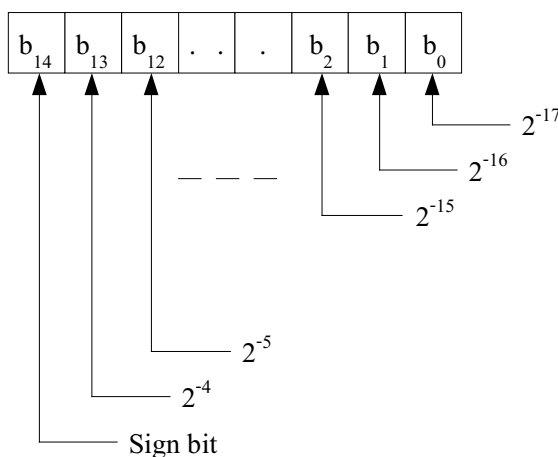
เนื่องจากค่าคงตัวในสมการที่ (7-5) และสมการที่ (7-6) ที่จะนำมาใช้ในการสร้างจริงวงจรกรองปรับตัว จะใช้ค่าเดียวกับค่าคงตัวที่ใช้การจำลองผลเชิงโปรแกรมที่ใช้การคำนวณแบบ Floating Point Arithmetic ซึ่งมีค่าคงตัวดังนี้ คือ α มีค่าเท่ากับ 0.69, γ มีค่าเท่ากับ 0.005, $\mu_{\max}$ มีค่าเท่ากับ 0.0005 และ $\mu_{\min}$ มีค่าเท่ากับ 0.0001 จากการจำลองผลทางคอมพิวเตอร์ค่าช่วงก้ำวของวิธีการปรับช่วงก้ำวแบบ Squared error สามารถแทนได้ด้วยเลขฐานสองจำนวน 8 บิต ซึ่งมีรายละเอียดดังนี้ ที่ตำแหน่งซ้ายสุดแทนตำแหน่งของบิตเครื่องหมาย (Sign bit) ตำแหน่งที่ 2 จากทางซ้ายแทนตำแหน่งของบิตที่มีค่าสูงสุด (Most Significant Bit: MSB) ซึ่งมีค่าประจำตำแหน่ง คือ 2^{-11} และบิตที่อยู่ตำแหน่งทางขวาสุดเป็นบิตที่มีค่าต่ำสุด (Least Significant Bit: LSB) ซึ่งมีค่าประจำตำแหน่งคือ 2^{-17} สำหรับรายละเอียดในตำแหน่งอื่น ๆ สามารถดูได้จากรูปที่ 7.4



รูปที่ 7.4 รายละเอียดบิตข้อมูลของค่าช่วงก้าว (Step Size)

ในกรณีที่ค่าช่วงก้าวมีค่าสูงสุดสามารถแทนค่าด้วยเลขฐานสองได้เท่ากับ 01000001_2 (มีค่าเป็นเลขจำนวนจริงเท่ากับ $2^{-11} + 2^{-17} = 0.00049591$) ส่วนกรณีที่ค่าช่วงก้าวมีค่าต่ำสุดสามารถแทนค่าด้วยเลขฐานสองได้เท่ากับ 00001101_2 (มีค่าเป็นเลขจำนวนจริงเท่ากับ $2^{-14} + 2^{-15} + 2^{-17} = 0.000099182$) ซึ่งจากผลของการที่ค่าช่วงก้าวค่าอยู่ในช่วง 2^{-11} ถึง 2^{-17} รวมกับผลของค่าคงตัวของเส้นทางป้อนกลับที่นำมาใช้ในการสร้างจริงวงจรกรองปรับตัวอันดับ 8 ที่มีค่าอยู่ในช่วง -0.0475 ถึง 0.1167 (คัดเลือกจากผลตอบสนองต่ออิมพัลส์ของเส้นทางป้อนกลับที่ n เท่ากับ 11 ถึง 18 ตามรูปที่ 2.4) ทำให้จำเป็นต้องใช้ค่าสัมประสิทธิ์ของวงจรกรองปรับตัว (Tap Weight) แทนด้วยเลขฐานสองจำนวน 15 บิต เพื่อให้สามารถครอบคลุมได้ถึงค่าสูงสุดและต่ำสุดของค่าช่วงก้าว ที่ใช้หลักการปรับค่าช่วงก้าวแบบ Squared error

จากการที่ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวถูกออกแบบให้แทนค่าด้วยเลขฐานสองจำนวน 15 บิต ซึ่งรายละเอียดของแต่ละบิตแสดงให้เห็นดังรูปที่ 7.5 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-4} และที่ตำแหน่ง LSB เท่ากับ 2^{-17} จากผลที่ได้ทำให้ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวสามารถแทนค่าต่ำสุดได้เท่ากับ -0.125 และสูงสุดเท่ากับ 0.12499



รูปที่ 7.5 รายละเอียดบิตข้อมูลของค่าสัมประสิทธิ์ของวงจรกรองปรับตัว

จากการที่ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวต้องใช้เลขฐานสองจำนวน 15 บิต เพื่อความสะดวกในการคำนวณ ดังนั้นสัญญาณขาเข้าของวงจรกรองปรับตัว (สัญญาณสุ่ม) ควรจะแทนค่าด้วยจำนวนบิตของข้อมูลที่เท่ากัน เพราะฉะนั้นจึงแทนสัญญาณขาเข้าของวงจรกรองปรับตัวด้วยจำนวนบิตเท่ากับ 15 บิต ซึ่งรายละเอียดของแต่ละบิตจะมีลักษณะใกล้เคียงกับรูปที่ 7.5 ต่างกันที่ค่าประจำตำแหน่งของแต่ละหลัก โดยที่ค่าสูงสุดของสัญญาณขาเข้ามีค่าประจำหลักเท่ากับ 2^{-1} (ตำแหน่งที่ 2 จากซ้าย) และค่าต่ำสุดมีค่าประจำหลักเท่ากับ 2^{-14} (ตำแหน่งขวาสุด) จากคุณสมบัติของสัญญาณสุ่มที่มีค่าความแปรปรวนเท่ากับ 0.1 และเนื่องจากการใช้เลขฐานสองจำนวน 15 บิต ในการแทนค่าสัญญาณสุ่ม ทำให้สามารถแทนค่าต่ำสุดได้เท่ากับ -1 และสูงสุดเท่ากับ 0.99994

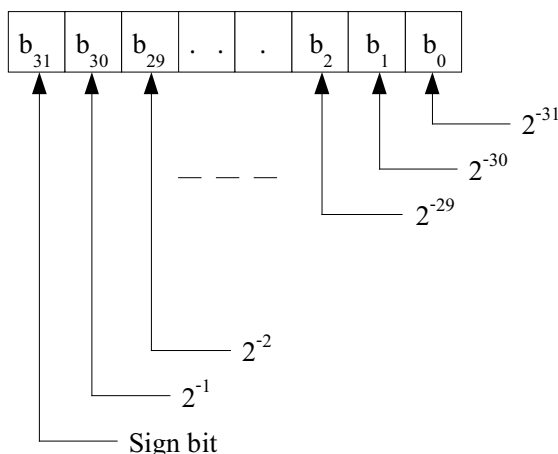
จากการที่ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวและสัญญาณขาเข้าของวงจรกรองปรับตัวใช้เลขฐานสองจำนวน 15 บิต แทนค่าเลขจำนวนจริง เพื่อที่จะทำการประมวลโดยใช้การคำนวณในรูปแบบ Fixed Point Arithmetic ซึ่งเป็นการจำลองการทำงานของระบบในรูปแบบของฮาร์ดแวร์ ซึ่งค่าสูงสุดของการคำนวณจะเกิดขึ้นในกรณีที่ทั้งค่าสัมประสิทธิ์ของวงจรกรองปรับตัวและสัญญาณขาเข้าของวงจรกรองปรับตัวมีค่าเท่ากับ 100000000000000₂ ซึ่งจะได้ผลคูณดังนี้

$$(100000000000000_2) * (100000000000000_2) = 01000000000000000000000000000000_2$$

จากการคำนวณในกรณีที่ค่าทั้งสองมีค่าเป็นลบต่ำสุดทั้ง 2 ค่าจะให้ผลลัพธ์ที่เป็นค่าบวกสูงสุด ซึ่งต้องใช้จำนวนบิตของเลขฐานสองแทนค่าดังกล่าวเป็นจำนวน 30 บิต กรณีอื่น ๆ สามารถใช้จำนวนบิตข้อมูลเพียง 29 บิตก็เพียงพอ ซึ่งโอกาสที่จะเกิดขึ้นน้อยมาก ดังนั้นในการออกแบบจึงแทนผลลัพธ์ของผลคูณทั้งสองดังนี้

การบวกก็จะทำงานไปตามปกติ เนื่องจากสัญญาณที่เข้าสู่ระบบได้ทำการแทนค่าด้วยเลขฐานสองที่มีค่าเป็นไปได้อันบวกและลบเรียบร้อยแล้ว โดยใช้หลักการ 2's Complement เพื่อให้่ายในการรวมสัญญาณ

เมื่อผ่านกระบวนการบวกเรียบร้อยแล้ว ข้อมูลที่ได้จะมีขนาด 32 บิต ซึ่งมีรายละเอียดของบิตแสดงให้เห็นดังรูปที่ 7.7 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-1} และที่ LSB เท่ากับ 2^{-31}

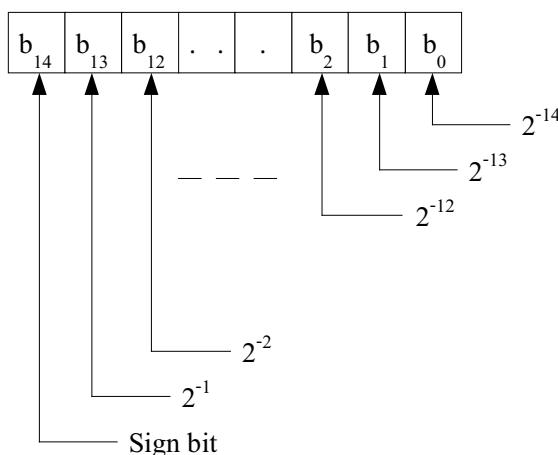


รูปที่ 7.7 รายละเอียดบิตข้อมูลที่สัญญาณขาออกของวงจรกรอง

เนื่องจากระบบที่ทำการออกแบบ ให้ขนาดของ Desired Response เป็นเลขฐานสองจำนวน 15 บิต ดังนั้นในขั้นตอนต่อไปจะนำเอาผลลัพธ์ที่ได้จากวงจรกรองปรับตัว ซึ่งมีขนาด 32 บิต โดยนำเอา 15 บิตแรกทางด้าน MSB มาใช้งาน ส่วนที่เหลือ 17 บิตทางด้าน LSB ตัดทิ้ง จากนั้นก็นำเอาข้อมูลทั้ง 15 บิต ไปผ่านกระบวนการ 2's Complement ก่อนที่จะนำไปรวมกับ Desired Response

การออกแบบวงจรกรองปรับตัวอันดับ 8 ที่จะทำการสร้างจริงนี้ จะทำการรวมสัญญาณขาเข้า $s(n)$ ซึ่งมีค่าความแปรปรวนเท่ากับ 0.1 เท่าของสัญญาณสุ่ม $N_p(n)$ กับสัญญาณเสียงที่ผ่านเส้นทางป้อนกลับก่อน เพื่อความสะดวกในการพิจารณาหาค่าความผิดพลาด ซึ่งในที่นี้จะแทนผลรวมของสัญญาณทั้งสองด้วย Desired Response โดยที่ค่าสูงสุดและต่ำสุดที่ได้ของ Desired Response มีขนาดเท่ากับ 0.68507 ทั้งทางด้านบวกและลบ ดังนั้นในการออกแบบจะใช้เลขฐานสองจำนวน 15 บิตแทนค่าดังกล่าว ซึ่งคิดจากขนาดของสัญญาณสูงสุดที่ผ่านเส้นทางป้อนกลับรวมกับขนาดของสัญญาณขาเข้า $s(n)$ สูงสุดของแบบจำลองที่เป็นได้

ซึ่งรายละเอียดของบิตข้อมูลของ Desired Response และผลลัพธ์ของวงจรกรองปรับตัวที่ผ่านกระบวนการ 2's Complement แสดงให้เห็นดังรูปที่ 7.8 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-1} และที่ LSB เท่ากับ 2^{-14}



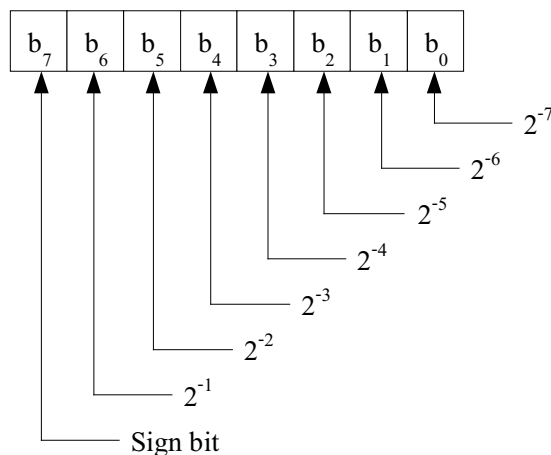
รูปที่ 7.8 รายละเอียดบิตข้อมูลของ Desired Response

เมื่อจำนวนบิตข้อมูลของทั้ง Desired Response และผลรวมของวงจรกรองปรับตัวที่ผ่านกระบวนการ 2's Complement เท่ากันแล้ว ต่อไปก็ทำการบวกข้อมูลทั้งสองเข้าด้วยกัน ก็จะได้ค่าความผิดพลาดที่มีขนาด 15 บิต สำหรับการออกแบบที่จะใช้ในการสร้างจริงจะใช้จำนวนบิตของค่าความผิดพลาดเท่ากับ 8 บิต โดยจะทำการเลือก 8 บิตทางด้าน MSB มาใช้ในการคำนวณในขั้นตอนของการปรับค่าช่วงก้ำวที่จะนำไปใช้ในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวต่อไป นอกจากนี้ยังนำไปใช้ในการประเมินประสิทธิภาพที่ได้ของขั้นตอนวิธีอีกด้วย ซึ่งในการจำลองนี้จะแสดงให้เห็นในรูปของค่าความผิดพลาดกำลังสองเฉลี่ยส่วนเกิน รายละเอียดของบิตข้อมูลของค่าความผิดพลาดแสดงให้เห็นดังรูปที่ 7.9 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-1} และที่ LSB เท่ากับ 2^{-7} จากผลที่ได้ในแบบจำลองแสดงให้เห็นว่าค่าความผิดพลาดจะมีค่าสูงสุดที่ 0.99609 และมีค่าต่ำสุดที่ -1 เมื่อแทนด้วยเลขฐานสองจะได้ 01111111_2 และ 10000000_2 ตามลำดับ

ค่าความผิดพลาดที่คำนวณได้จะถูกนำไปใช้ในวิธีการปรับช่วงก้ำวแบบค่าผิดพลาดกำลังสอง และ Sign bit ของค่าความผิดพลาดจะถูกนำไปใช้ในการพิจารณาในการปรับค่าสัมประสิทธิ์ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign

ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่สำคัญ ๆ เพื่อใช้ในการออกแบบ ได้นำเสนอมาครบแล้ว ในส่วนต่อไปก็จะกล่าวถึงค่าสัมประสิทธิ์ในส่วนของขั้นตอนวิธีการปรับค่าสัมประสิทธิ์

ซึ่งประกอบด้วยขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign และหลักการปรับช่วงก้ำวแบบค่าผิดพลาดกำลังสอง



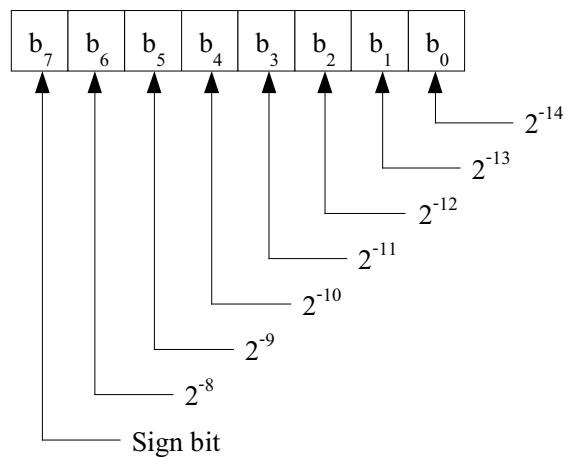
รูปที่ 7.9 รายละเอียดบิตข้อมูลของค่าความผิดพลาด $e(n)$

7.7.2 รายละเอียดของค่าสัมประสิทธิ์ในส่วนต่าง ๆ ของขั้นตอนวิธีการปรับค่าสัมประสิทธิ์

จากสมการของวิธีการปรับค่าช่วงก้ำวแบบค่าผิดพลาดกำลังสองในสมการที่ (7-5) และสมการที่ (7-6) ซึ่งมีค่าคงตัวดังนี้ คือ α มีค่าเท่ากับ 0.69, γ มีค่าเท่ากับ 0.005, $\mu_{\max}$ มีค่าเท่ากับ 0.0005 และ $\mu_{\min}$ มีค่าเท่ากับ 0.0001 ซึ่งค่าช่วงก้ำวสูงสุดและต่ำสุดของวิธีการปรับช่วงก้ำวแบบค่าผิดพลาดกำลังสอง สามารถแทนได้ด้วยเลขฐานสองจำนวน 8 บิต ดังแสดงในรูปที่ 7.4

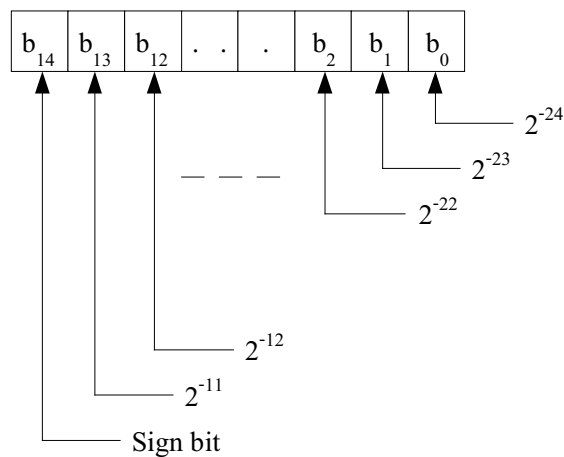
ค่าคงตัว α ซึ่งมีค่าเท่ากับ 0.69 สามารถแทนได้ด้วยเลขฐานสองจำนวน 8 บิต ซึ่งเท่ากับ 01011000₂ ซึ่งมีค่าประจำตำแหน่งเช่นเดียวกับค่าความผิดพลาดในรูปที่ 7.9 ส่วนค่าช่วงก้ำวที่ค่าสูงสุด เมื่อแทนเป็นเลขฐานสองจำนวน 8 บิต จะมีค่าเท่ากับ 01000001₂ ในกรณีที่ค่าช่วงก้ำวมีค่าต่ำสุด จะมีค่าเท่ากับ 00001101₂ ซึ่งรายละเอียดของบิตข้อมูลแต่ละตำแหน่งสามารถดูได้จากรูปที่ 7.4 ส่วนค่า γ ซึ่งมีค่าเท่ากับ 0.005 สามารถแทนได้ด้วยเลขฐานสองจำนวน 8 บิต ซึ่งเท่ากับ 01010010₂ โดยที่รายละเอียดและค่าประจำตำแหน่งของแต่ละบิตแสดงดังรูปที่ 7.10 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-8} และที่ LSB เท่ากับ 2^{-14}

การปรับค่าช่วงก้ำวแบบค่าผิดพลาดกำลังสอง ในสมการที่ (7-5) ค่าช่วงก้ำวใหม่ที่เวลา $n+1$ จะเท่ากับผลรวมของ ผลคูณของ α กับค่าช่วงก้ำวเก่าที่เวลา n และผลคูณของ γ กับค่าความผิดพลาดกำลังสอง



รูปที่ 7.10 รายละเอียดบิตข้อมูลของค่า γ

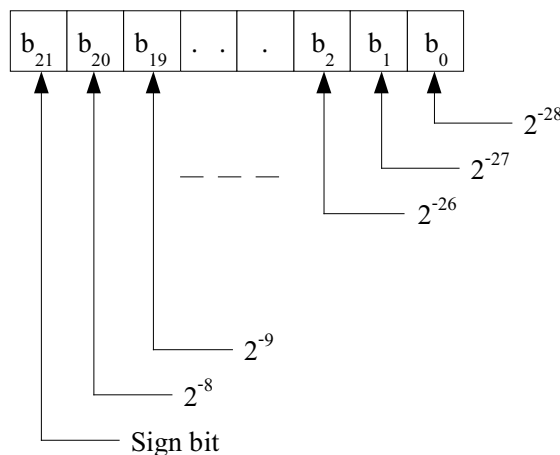
ผลคูณของ α กับค่าช่วงก้ำวเก่าที่เวลา n เมื่อแทนด้วยเลขฐานสองจะมีขนาดเท่ากับ 15 บิต รายละเอียดและค่าประจำหลักในแต่ละบิตของค่าช่วงก้ำวคูได้จากรูปที่ 7.4 ส่วนรายละเอียดและค่าประจำหลักในแต่ละบิตของ α เหมือนกับรูปที่ 7.9 เมื่อค่าทั้งสองคูณกันเสร็จเรียบร้อยแล้วจะได้เลขฐานสองจำนวน 15 บิต ซึ่งรายละเอียดและค่าประจำตำแหน่งในแต่ละบิตแสดงดังรูปที่ 7.11 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-11} และที่ LSB เท่ากับ 2^{-24}



รูปที่ 7.11 รายละเอียดบิตข้อมูลผลคูณของ α กับค่าช่วงก้ำวเก่าที่เวลา n

ผลคูณของ γ กับค่าความผิดพลาดกำลังสอง จะต้องทำการคูณค่าความผิดพลาดกำลังสองก่อนแล้วจึงคูณด้วยค่า γ การคูณค่าความผิดพลาดกำลังสองจะได้ผลลัพธ์เป็นเลขฐานสองจำนวน 15 บิต ซึ่งจะได้ค่าประจำหลักต่าง ๆ เหมือนกับรูปที่ 7.8 จากนั้นจึงนำค่า γ ซึ่งเป็นค่าคงตัวคูณกับค่าความผิดพลาดกำลังสองจะได้ผลลัพธ์สูงสุดอยู่ในเลขฐานสองจำนวน 22 บิต ซึ่งมีราย

ละเอียดและค่าประจำตำแหน่งดังรูปที่ 7.12 ซึ่งมีค่าประจำตำแหน่งที่ MSB เท่ากับ 2^{-8} และที่ LSB เท่ากับ 2^{-28}



รูปที่ 7.12 รายละเอียดบิตข้อมูลผลคูณของ γ กับค่าความผิดพลาดกำลังสอง

ในขั้นตอนที่จะนำเอาผลคูณทั้งสองมาบวกกัน จะต้องทำการเลือกบิตข้อมูลของผลคูณระหว่าง γ และค่าความผิดพลาดกำลังสอง จากที่ผลคูณดังกล่าวมีค่าประจำตำแหน่งดังรูปที่ 7.12 จะทำการเลือกข้อมูลในตำแหน่งที่มีค่าประจำหลักเท่ากับ 2^{-11} ถึง 2^{-24} มาใช้ โดยที่จะกำหนดให้ Sign bit เป็น 0 เพื่อให้มีขนาดเท่ากับผลคูณของ α กับค่าช่วงก้ำวเก่าที่เวลา n จากนั้นทำการบวกกัน และทำการเลือกเอาเฉพาะ 8 บิต ทางด้าน MSB มาพิจารณากับค่าช่วงก้ำวในสมการที่ (7-6) จากนั้นก็สามารถนำค่าช่วงก้ำวดังกล่าวไปใช้ในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวได้

ในส่วนของการปรับค่าสัมประสิทธิ์ของวงจรกรองที่เวลา $n+1$ จะนำเอาค่าช่วงก้ำวที่คำนวณได้จากวิธีค่าผิดพลาดกำลังสองมารวมกับค่าสัมประสิทธิ์ของวงจรกรองที่เวลา n ตามสมการของขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign ซึ่งมีรูปแบบสมการดังนี้

$$\tilde{w}(n+1) = \tilde{w}(n) + \mu [\text{sgn}(\tilde{x}(n)) \text{sgn}(e(n))] \quad (7-7)$$

ในการคำนวณค่าสัมประสิทธิ์ของวงจรกรองที่ใช้ขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign จะทำการตรวจสอบ Sign bit ของค่าความผิดพลาดและสัญญาณขาเข้าของวงจรกรอง ถ้า Sign bit ของค่าทั้ง 2 มีค่าเหมือนกันก็จะทำการขยายค่าช่วงก้ำวให้มีขนาด 15 บิต โดยทำการเพิ่มเลข 0 เข้าไปข้างหน้าค่าช่วงก้ำวอีก 7 ตัว ส่วนในกรณีที่ค่า Sign bit มีค่าแตกต่างกัน ก็จะนำเอาค่าช่วงก้ำวที่หามาได้ไปผ่านกระบวนการ 2's Complement จากนั้นก็จะเพิ่มเลข 1 เข้าไปข้างหน้าอีก 7 ตัว เมื่อทำการปรับขนาดค่าช่วงก้ำวให้มีค่าประจำตำแหน่งให้เหมือนกับค่าสัมประสิทธิ์

ของวงจรกรองปรับตัวเรียบร้อยแล้ว จากนั้นทำการรวมค่าสัมประสิทธิ์ของวงจรกรองที่เวลา n กับค่าช่วงก้ำกึ่งขยายเป็น 15 บิต ก็จะได้ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่เวลา $n+1$

ในหัวข้อ 7.2 ที่ผ่านมาได้แสดงให้เห็นถึงรายละเอียดในการวิเคราะห์และพิจารณาการแทนค่าสัญญาณต่าง ๆ ด้วยเลขฐานสองเพื่อให้สามารถนำไปใช้ในการคำนวณในรูปแบบ Fixed Point Arithmetic เพื่อจำลองการทำงานในรูปแบบของฮาร์ดแวร์ซึ่งจะนำไปใช้ในการสร้างจริง ในส่วนสุดท้ายของหัวข้อนี้จะแสดงให้เห็นถึงค่าต่าง ๆ ที่เป็นไปได้ที่มีขนาดสูงสุดทั้งทางด้านบวกและลบ

ตารางที่ 7.1 สรุปขนาดของสัญญาณต่าง ๆ ภายในวงจรกรองปรับตัว

สัญญาณ	ขนาดของสัญญาณ
สัญญาณขาเข้าวงจรกรองปรับตัว $x(n)$	ใช้เลขฐานสอง 15 บิต ค่าต่ำสุดเท่ากับ $100000000000000_2 = -1$ ค่าสูงสุดเท่ากับ $011111111111111_2 = 0.99994$
ค่าสัมประสิทธิ์ของวงจรกรองปรับตัว $w(n)$	ใช้เลขฐานสอง 15 บิต ค่าต่ำสุดเท่ากับ $100000000000000_2 = -0.125$ ค่าสูงสุดเท่ากับ $011111111111111_2 = 0.12499$
สัญญาณขาออกของวงจรกรองปรับตัว $y(n)$	ใช้เลขฐานสอง 15 บิต ค่าต่ำสุดเท่ากับ $100000000000000_2 = -1$ ค่าสูงสุดเท่ากับ $011111111111111_2 = 0.99999$
Desired Response $d(n)$	ใช้เลขฐานสอง 15 บิต ค่าต่ำสุดเท่ากับ $101010000101000_2 = -0.68507$ ค่าสูงสุดเท่ากับ $010101111011000_2 = 0.68507$

ตารางที่ 7.1 (ต่อ) สรุปรูปขนาดของสัญญาณต่าง ๆ ภายในวงจรกรองปรับตัว

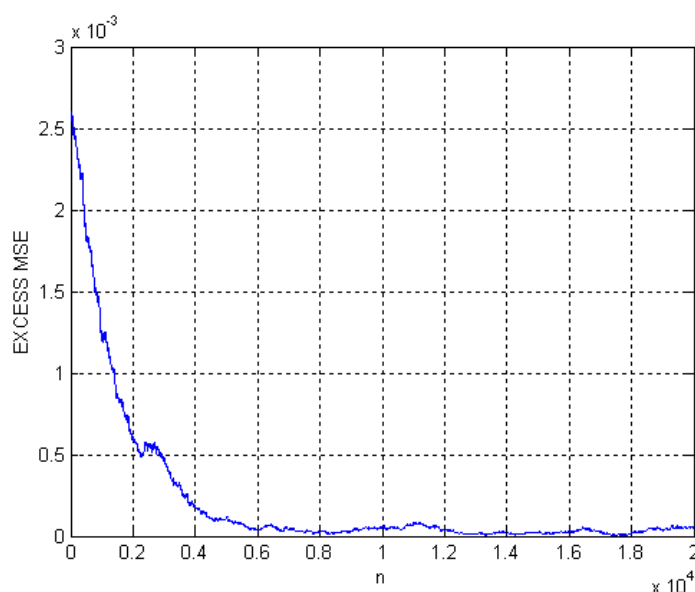
ค่าความผิดพลาด $e(n)$	ใช้เลขฐานสอง 8 บิต ค่าต่ำสุดเท่ากับ $10000000_b = -1$ ค่าสูงสุดเท่ากับ $01111111_b = 0.99609$
ค่าช่วงก้าว μ $\mu_{\min}$ $\mu_{\max}$	ใช้เลขฐานสอง 8 บิต ค่าต่ำสุดเท่ากับ $00001101_b = 0.000099182$ ค่าสูงสุดเท่ากับ $01000001_b = 0.00049591$
ค่าคงตัว α	ใช้เลขฐานสอง 8 บิต มีค่าเท่ากับ $01011000_b = 0.6875$
ค่าคงตัว γ	ใช้เลขฐานสอง 8 บิต มีค่าเท่ากับ $01010010_b = 0.005004882$

ทั้งนี้ค่าประจำตำแหน่งของค่าสัมประสิทธิ์ต่าง ๆ ที่เป็นเลขฐานสองสามารถดูได้จากเนื้อหาที่กล่าวมาแล้วข้างต้น ค่าสัมประสิทธิ์ต่าง ๆ ในตารางที่ 7.1 จะใช้ค่าเดียวกับการคำนวณในรูปแบบ Floating Point Arithmetic มาเปลี่ยนเป็นเลขฐานสอง ซึ่งการกำหนดค่าต่าง ๆ จะพยายามให้มีค่าใกล้เคียงกับค่าสัมประสิทธิ์ที่ใช้ในการคำนวณในรูปแบบ Floating Point Arithmetic ให้มากที่สุด เพื่อให้ผลการจำลองที่ได้จากการคำนวณในรูปแบบ Fixed Point Arithmetic สามารถเปรียบเทียบกับผลการจำลองที่ใช้การคำนวณในรูปแบบ Floating Point Arithmetic ได้ หากการจำลองผลของแบบจำลองที่ใช้ค่าสัมประสิทธิ์แทนด้วยเลขฐานสองและใช้การคำนวณในรูปแบบ Fixed Point Arithmetic ได้ผลการจำลองที่มีประสิทธิภาพดีจะทำให้สามารถประเมินผลที่จะเกิดขึ้นจริง เมื่อนำเอาข้อกำหนดต่าง ๆ ที่ใช้การคำนวณในรูปแบบ Fixed Point Arithmetic ไปสร้างจริงเป็นวงจรด้วยภาษา VHDL

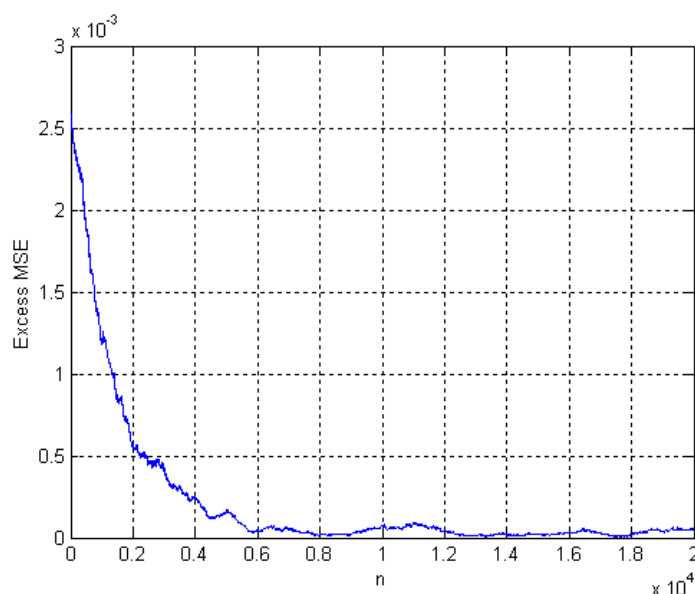
7.3 การจำลองผลโดยใช้การคำนวณแบบ Fixed Point Arithmetic

จากผลการจำลองของแบบจำลองการแก้ปัญหาก็เกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังแบบที่มีการปรับค่าสัมประสิทธิ์แบบไม่ต่อเนื่อง ที่ใช้การคำนวณในรูปแบบ Floating Point Arithmetic ให้ผลลัพธ์ของการจำลองดังที่แสดงให้เห็นแล้วในหัวข้อ 7.1 แต่เนื่องจากการทำงานจริงในระบบต่าง ๆ ไม่สามารถประมวลผลด้วยตัวเลขซึ่งเป็นจำนวนจริงได้ ดังนั้นการจำลองการทำงานของระบบที่จะเกิดขึ้นจริง เมื่อนำไปสร้างเป็นฮาร์ดแวร์ที่ใช้ค่าสัมประสิทธิ์ต่าง ๆ แทนด้วยเลขฐานสอง โดยใช้การคำนวณในรูปแบบ Fixed Point Arithmetic จึงเป็นสิ่งจำเป็นที่จะแสดงให้เห็นถึงประสิทธิภาพที่จะเกิดขึ้น เมื่อนำเอาแบบจำลองต่าง ๆ ที่ได้ออกแบบไว้ไปทำการสร้างจริง

เนื่องจากการสร้างจริงจะทำการสร้างจริงวงจรกรองปรับตัวอันดับ 8 ที่ใช้สำหรับแก้ปัญหาก็เกิดเสียงรบกวนเนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง เพื่อที่จะแสดงให้เห็นถึงประสิทธิภาพที่ได้ของการจำลองผลการทำงานของแบบจำลองที่ใช้การคำนวณในรูปแบบ Fixed Point Arithmetic และสามารถเปรียบเทียบกับผลที่ได้ของแบบจำลองที่ใช้การคำนวณในรูปแบบ Floating Point Arithmetic ได้ ดังนั้นในหัวข้อนี้จะทำการจำลองผลที่ได้ของวงจรกรองปรับตัวอันดับ 8 ที่ใช้สำหรับแก้ปัญหาก็เกิดเสียงรบกวนเนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง โดยใช้การคำนวณทั้ง 2 รูปแบบ



รูปที่ 7.13 ผลการจำลองของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign และการปรับช่วง
ก้าวแบบค่าผิดพลาดกำลังสอง ที่ใช้การคำนวณแบบ Floating Point Arithmetic



รูปที่ 7.14 ผลการจำลองของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign และการปรับช่วง
ก้าวแบบค่าผิดพลาดกำลังสอง ที่ใช้การคำนวณแบบ Fixed Point Arithmetic

จากผลที่ได้จากการจำลองในรูปที่ 7.13 และ 7.14 จะเห็นว่าผลที่ได้จากการจำลองที่ใช้การคำนวณในรูปแบบ Fixed Point Arithmetic ใกล้เคียงกับผลที่ได้จากการจำลองที่ใช้การคำนวณในรูปแบบ Floating Point Arithmetic ซึ่งผลการจำลองที่ได้ในรูปที่ 7.14 จะเป็นเครื่องยืนยันผลการทำงานที่จะเกิดขึ้นจริง เมื่อนำเอาขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ที่ใช้การปรับช่วงก้าวแบบพลวัตแบบ Squared error ไปทำการสร้างจริงในรูปแบบฮาร์ดแวร์ ซึ่งการสร้างจริงจะนำเสนอต่อไปในบทที่ 8

บทที่ 8

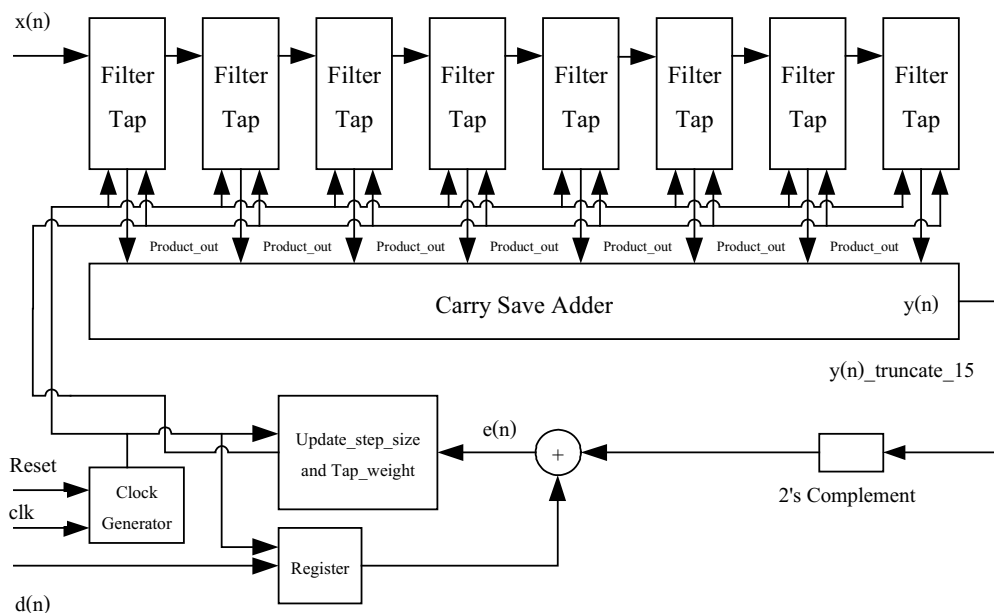
การสร้างจริงวงจรกรองปรับตัวแบบ Sign-Sign LMS

การสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหาก็เกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง จะใช้ผลที่ได้จากการจำลองเชิงฮาร์ดแวร์ที่ใช้การคำนวณในรูปแบบ Fixed Point Arithmetic ในบทที่ 7 เป็นพื้นฐานในการออกแบบส่วนประกอบต่าง ๆ ที่จะใช้ในการสร้างจริงบน FPGA ของบริษัท Xilinx รุ่น Spartan2 xc2s200-5pq208

เนื้อหาในบทนี้จะกล่าวถึงโครงสร้างทางสถาปัตยกรรมของวงจรปรับตัว ขั้นตอนการทำงานของวงจรกรองปรับตัว จากนั้นจะกล่าวถึงส่วนประกอบต่าง ๆ ที่นำมาใช้ในการสร้างจริงวงจรกรองปรับตัวรวมถึงระบบการทำงานของส่วนต่าง ๆ ต่อจากนั้นจะแสดงผลที่ได้จากการทำงานของวงจรกรองปรับตัวที่ผ่านการสังเคราะห์ทั้งในระดับ Behavioral และระดับ Logic

8.1 โครงสร้างทางสถาปัตยกรรมของวงจรกรองปรับตัว

โครงสร้างทางสถาปัตยกรรมของวงจรกรองปรับตัวที่จะใช้ในการสร้างจริงมีโครงสร้างดังแสดงดังรูปที่ 8.1



รูปที่ 8.1 โครงสร้างทางสถาปัตยกรรมของวงจรกรองปรับตัวที่ใช้ในการสร้างจริง

จากรูปที่ 8.1 แสดงโครงสร้างทางสถาปัตยกรรมของวงจรกรองปรับตัวที่ใช้ในการสร้างจริง สำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้ คือ Filter Tap วงจรบวก (Carry Save Adder, Carry Propagate Adder) วงจร 2's Complement วงจรกำเนิดสัญญาณนาฬิกา (Clock Generator) Register วงจรปรับค่าช่วงก้ำวและสัมประสิทธิ์ของวงจรกรอง

8.2 ขั้นตอนการทำงานของระบบ

การสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียง จะทำการสร้างจริงวงจรกรองอันดับ 8 ทั้งนี้เพื่อให้สามารถทำงานได้จริงบนทรัพยากรต่าง ๆ ที่มีจำกัด การสร้างจริงวงจรกรองปรับตัวจะทำการสร้างจริงบน FPGA ของบริษัท Xilinx รุ่น Spartan2 xc2s200-5pq208 โดยที่การสร้างจริงจะทำตามแบบจำลองที่ใช้การประมวลผลในรูปแบบ Fixed Point Arithmetic

ขั้นตอนการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง เริ่มต้นจากการรับข้อมูลเข้ามาในวงจรกรองปรับตัว ทำการประมวลในส่วนของวงจรกรอง หาค่าความผิดพลาด ทำการปรับค่าช่วงก้ำวและทำการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว ซึ่งการทำงานในขั้นตอนต่าง ๆ จะแสดงให้เห็นต่อไป

เริ่มต้นจากการคำนวณในแต่ละ Tap ของวงจรกรองปรับตัว เพื่อคำนวณหาผลลัพธ์ที่ได้จากกระบวนการกรองที่เวลา n

$$y(n) = w_1(n)x(n-1) + w_2x(n-2) + \dots + w_8x(n-8) \quad (8-1)$$

โดยที่

$y(n)$ คือ สัญญาณขาออกของวงจรกรองที่เวลา n

$d(n)$ คือ Desired Response ที่เวลา n (เป็นผลรวมจากสัญญาณเสียงขาเข้ากับสัญญาณเสียงที่ผ่านเส้นทางป้อนกลับ)

$e(n)$ คือ ค่าความผิดพลาดที่เวลา n

$w_i(n)$ คือ สัมประสิทธิ์ของวงจรกรอง Tap ที่ i ที่เวลา n ($1 \leq i \leq 8$)

$x(n-i)$ คือ สัญญาณขาเข้าของวงจรกรอง ณ Tap ที่ i ที่เวลา n ($1 \leq i \leq 8$)

$\mu(n)$ คือ ค่าช่วงก้าว (Step Size) ที่เวลา n

$y(n)_{\text{truncate_15}}$ คือ สัญญาณขาออกของวงจรกรองปรับตัว $y(n)$ ที่มีค่าเท่ากับ 15 บิต ทางด้าน MSB ของ $y(n)$

$y(n)_{\text{truncate_15_com}}$ คือ สัญญาณ $y(n)_{\text{truncate_15}}$ ที่ผ่านกระบวนการ 2's Complement

จากนั้นนำเอาค่าสัญญาณขาออกของวงจรกรอง $y(n)$ ที่มีขนาด 32 บิต ทำการเลือกบิต ข้อมูลทางด้าน MSB 15 บิต เก็บไว้ใน $y(n)_{\text{truncate_15}}$ จากนั้นนำผลลัพธ์ที่ได้ไปทำกระบวนการ 2's Complement และเก็บค่าไว้ใน $y(n)_{\text{truncate_15_com}}$ หลังจากนั้นนำค่าที่ได้ไปรวม กับค่า Desired Response $d(n)$ เพื่อหาค่าความผิดพลาด $e(n)$ ต่อจากนั้นนำค่าความผิดพลาด $e(n)$ และค่าช่วงก้าว $\mu(n)$ เพื่อนำไปคำนวณหาค่าช่วงก้าว $\mu(n+1)$ ที่จะใช้ในการคำนวณที่เวลา $n+1$ โดยใช้สมการที่ (8-2) และสมการที่ (8-3) ขั้นตอนสุดท้ายนำค่าช่วงก้าว $\mu(n)$ ค่า Sign bit ของค่าความผิดพลาด $e(n)$ Sign bit ของสัญญาณขาเข้า $x(n-i)$ ของวงจรกรอง ณ Tap ที่ i ที่ เวลา n และค่าสัมประสิทธิ์ของวงจรกรอง $w_i(n)$ เพื่อนำไปใช้คำนวณหาค่าสัมประสิทธิ์ของวงจรกรอง $w_i(n+1)$ ที่เวลา $n+1$ โดยใช้สมการที่ (8-4) และสมการที่ (8-5)

สมการของวิธีการปรับช่วงก้าวแบบค่าผิดพลาดกำลังสอง (Squared Error)

$$\mu'(n+1) = \alpha \mu(n) + \gamma e^2(n) \quad (8-2)$$

โดยที่

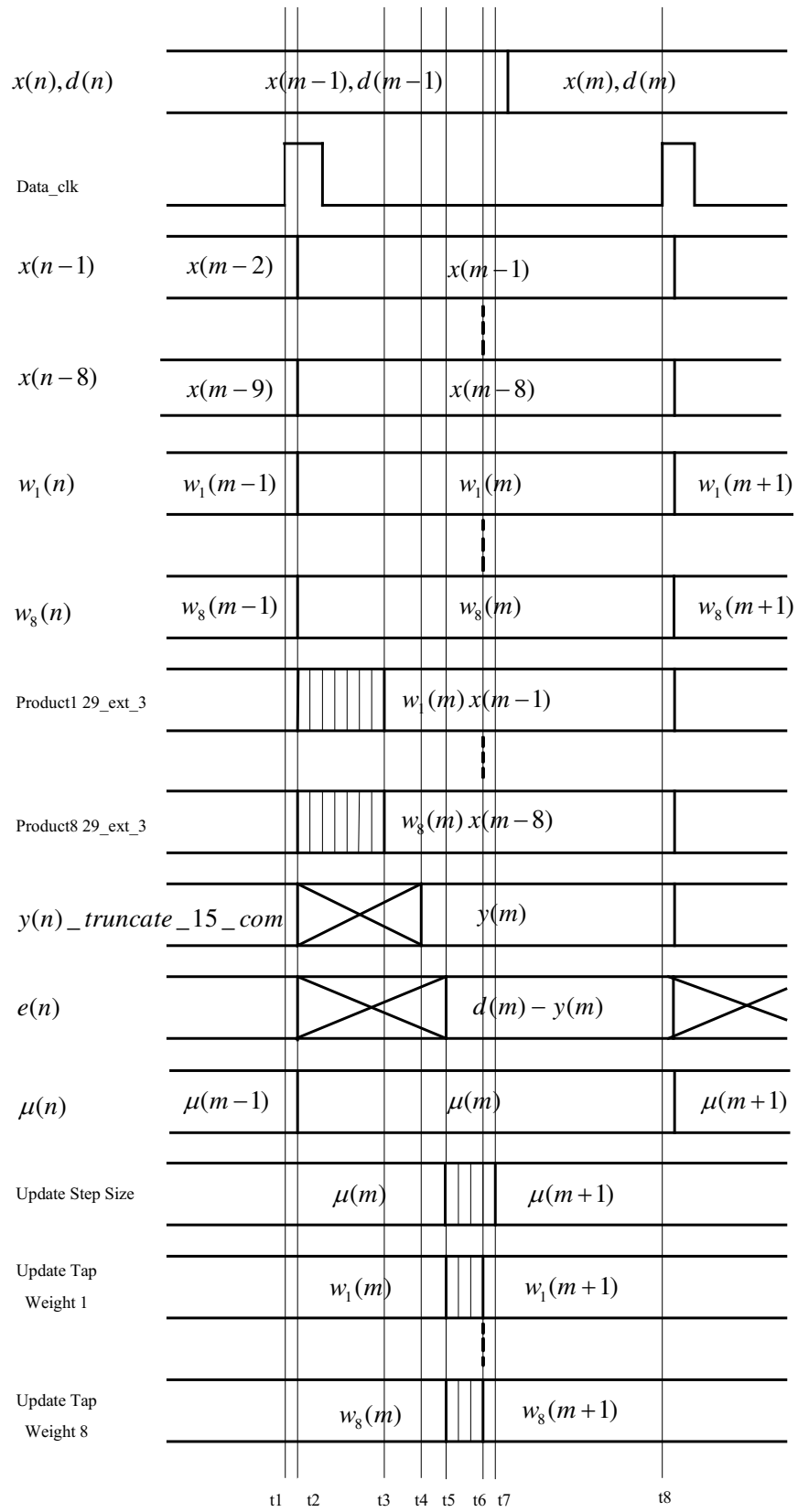
$$\mu(n+1) = \begin{cases} \mu_{\max}, & \mu'(n+1) > \mu_{\max} \\ \mu_{\min}, & \mu'(n+1) < \mu_{\min} \\ \mu'(n+1), & \text{Others} \end{cases} \quad (8-3)$$

สมการของขั้นตอนวิธีกำลังสองเลี่ยน้อยสุดประเภท Sign-Sign ณ Tap ที่ i ที่เวลา n

$$w_i(n+1) = w_i(n) + \mu(n) [\text{sgn}(x(n-i)) \text{sgn}(e(n))] \quad (8-4)$$

โดยที่

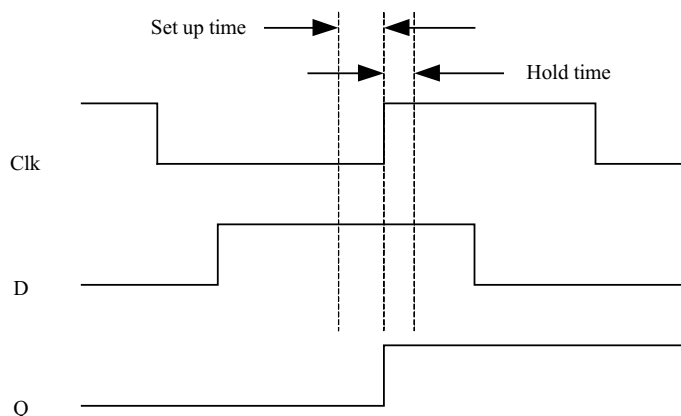
$$\text{sgn}(x) = \begin{cases} 1, x \geq 0 \\ -1, x < 0 \end{cases} \quad (8-5)$$



รูปที่ 8.2 Timing Diagram การทำงานของวงจรกรองปรับตัว

จากขั้นตอนการทำงานของวงจรกรองปรับตัวที่กล่าวมาแล้วข้างต้นสามารถแสดงขั้นตอนในแต่ละรอบของการทำงานได้ด้วย Timing Diagram ดังรูปที่ 8.2 โดยจะพิจารณาการทำงานของระบบที่เวลา $n = m$

สมมติให้ระบบเริ่มต้นการทำงานที่เวลา t_1 ซึ่งสัญญาณ data_clk มีค่าเป็น 1 ที่เวลาดังกล่าว ข้อมูลต่าง ๆ ที่ใช้ในระบบ เช่น $x(m-1)$, $d(m-1)$, $w_1(m)$ และ $\mu(n)$ เป็นต้น จะถูก Load เข้าไปเก็บไว้ใน Register ซึ่งจากรูปที่ 8.2 แสดงให้เห็นว่าจะเสร็จสิ้นกระบวนการ Load ข้อมูลที่เวลา t_2 ซึ่งช่วงเวลาระหว่าง t_1 ถึง t_2 จะเรียกว่า Clock to Q time คือ ระยะเวลาที่เริ่มเกิดการเปลี่ยนแปลงของสัญญาณ Clock จาก 0 เป็น 1 (สมมติให้ Register ทำงานที่ขอบขาขึ้น) เพื่อทำการรับข้อมูลทางด้าน Input จนกระทั่งข้อมูลดังกล่าวปรากฏทางด้าน Output ของ Register ทั้งนี้ข้อมูลที่เพิ่งป้อนเข้าสู่ Register จะต้องคำนึงถึงค่า Set up time และ Hold time [13] ของ Register ด้วย ข้อมูลต่าง ๆ ที่จะป้อนเข้าสู่ Register จะต้องมีความเสถียรภาพ (Stable) เป็นเวลานานกว่าค่า Set up time และค่า Hold time รวมกัน เพื่อให้ข้อมูลที่รับเข้าไปใน Register ถูกต้องดังแสดงในรูปที่ 8.3



รูปที่ 8.3 Set up time และ Hold time

จากรูปที่ 8.2 เมื่อทำการ Load ข้อมูลต่าง ๆ เข้าสู่ระบบเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นขั้นตอนการคูณค่าสัมประสิทธิ์ของวงจรกรองที่ Tap ต่าง ๆ กับสัญญาณขาเข้าของวงจรกรองที่เวลา m เท่ากับ $w_i(m)x(m-i)$ ซึ่งจะเป็นผลลัพธ์ที่ได้จากการคูณของวงจรกรองแต่ละ Tap จากนั้นนำผลคูณที่ได้ไปขยายจำนวนบิตเพิ่มทางด้าน MSB ของผลคูณ ให้มีค่าเหมือนกับ Sign bit ของผลคูณแต่ละ Tap กระบวนการทำงานของขั้นตอนเหล่านี้ใช้เวลาตั้งแต่ t_2 ถึง t_3 ต่อจากนั้นจะเป็นขั้นตอนของการรวมค่าผลคูณของวงจรกรองและทำการเลือกจำนวนบิตที่จะนำมาใช้และนำข้อมูลที่เลือกไปผ่านกระบวนการ 2's Complement เพื่อให้ได้ค่า $y(m)$ ซึ่งเป็นผลลัพธ์ของวงจรกรองที่ผ่านทั้งกระบวนการ Truncate 15 บิต และ 2's Complement เรียบร้อยแล้ว ซึ่งกระบวนการที่ผ่านมาใช้เวลา

ในการประมวลผลตั้งแต่เวลา t_3 ถึงเวลา t_4 หลังจากนั้นหาค่าความผิดพลาด ซึ่งจะได้ค่าความผิดพลาด $d(m) - y(m)$ ที่เวลา t_5 จากนั้นนำค่าความผิดพลาดที่ได้ไปคำนวณหาค่าช่วงก้าวเพื่อที่ใช้ในรอบต่อไปของการประมวลผล ในขณะเดียวกันก็ทำการคำนวณหาค่าสัมประสิทธิ์ที่จะนำไปใช้ในการประมวลผลในรอบต่อไปเช่นเดียวกัน ในที่นี้กำหนดให้การปรับค่าสัมประสิทธิ์ของวงจรกรองปรับใช้เวลาในการคำนวณตั้งแต่เวลา t_5 ถึง t_6 ส่วนการปรับช่วงก้าวใช้เวลาตั้งแต่ t_5 ถึง t_7 หลังจากเวลา t_6 และ t_7 จะได้ค่าสัมประสิทธิ์และค่าช่วงก้าวของวงจรกรองใหม่ตามลำดับ พร้อมทั้งจะนำค่าดังกล่าวไปใช้ในการประมวลผลรอบต่อไป จากรูปที่ 8.2 แต่ละรอบของการประมวลผลใช้เวลาตั้งแต่ t_1 ถึง t_7 ซึ่งจะเรียกเส้นทางที่ใช้เวลาในการประมวลผลตั้งแต่เริ่มต้นจนถึงสิ้นสุดกระบวนการว่า Critical Path ดังนั้นในแต่ละรอบของการสุ่มตัวอย่าง `data_clk` ของระบบต้องมีค่าเท่ากับหรือมากกว่า ค่าของ Critical Path เพื่อที่จะทำให้ระบบที่ออกแบบยังคงสามารถทำงานได้ถูกต้อง

8.3 การสร้างจริงส่วนประกอบต่าง ๆ ตามสถาปัตยกรรมของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง

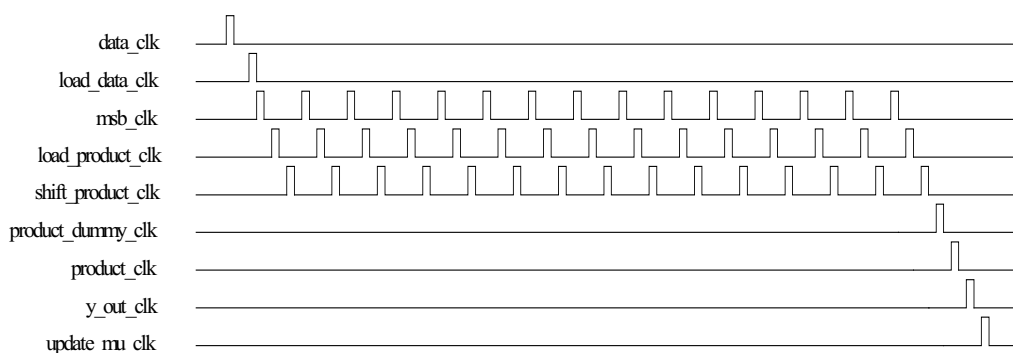
จากโครงสร้างทางสถาปัตยกรรมในรูปที่ 8.1 การสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง สามารถแบ่งได้เป็น 2 ส่วนหลัก ๆ คือ ส่วนของวงจรกรองและส่วนปรับค่าสัมประสิทธิ์ของวงจรกรอง

วงจรกรองประกอบด้วยส่วนประกอบที่สำคัญ คือ Filter Tap (ภายในประกอบด้วย Register และวงจรคูณแบบ Booth Encode) วงจรบวก (Carry Save Adder, Carry Propagate Adder) Register และวงจร 2's Complement ส่วนประกอบของวงจรในการปรับค่าช่วงก้าวและปรับค่าสัมประสิทธิ์ของวงจรกรองประกอบด้วย วงจรคูณแบบ Array วงจรบวกแบบ Carry Propagate Adder และวงจร 2's Complement นอกจากนี้ยังมีส่วนประกอบที่สำคัญที่ใช้ในการควบคุมการทำงานของระบบ คือ วงจรกำเนิดสัญญาณนาฬิกา (Clock Generator)

ในหัวข้อนี้จะกล่าวถึงส่วนประกอบต่าง ๆ ที่จะนำมาใช้ในการสร้างจริงวงจรกรองปรับตัว ซึ่งจะทำให้การออกแบบวงจรด้วยภาษา VHDL โดยจะอธิบายหลักการในการทำงานของส่วนประกอบต่าง ๆ หลังจากนั้นจะแสดงผลการทำงานของวงจรต่าง ๆ ที่ได้ทำการออกแบบด้วยภาษา VHDL ส่วนประกอบต่าง ๆ จะต้องผ่านการทดสอบว่าสามารถทำงานได้จริงบน FPGA ก่อนที่จะนำเอาส่วนประกอบต่าง ๆ มาประกอบรวมกันเพื่อที่จะทำการสร้างจริงเป็นวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังให้สามารถทำงานบน FPGA ได้จริง

8.3.1 ระบบ Clock ที่ใช้ควบคุมการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียง หอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง

การทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงหอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ที่จะทำการสร้างจริงจะต้องมีการทำงานที่เป็นระบบและมีเวลาในการทำงานที่แน่นอนเที่ยงตรงในแต่ละขั้นตอน เนื่องจากผลการทำงานในแต่ละขั้นตอนจะต้องสอดคล้องกัน ดังนั้นระบบ Clock ที่ใช้ควบคุมระบบจึงมีความสำคัญมาก ในหัวข้อนี้จะแสดงให้เห็นถึงระบบของสัญญาณ Clock ที่นำมาใช้งานจริงซึ่งถูกสร้างขึ้นจากวงจรกำเนิดสัญญาณนาฬิกา (Clock Generator) ที่ออกแบบด้วยภาษา VHDL



รูปที่ 8.4 ระบบของสัญญาณ Clock ที่ใช้ในการสร้างจริงวงจรกรองปรับตัว

จากรูปที่ 8.4 แสดงสัญญาณ Clock ต่าง ๆ ที่ทำการสร้างขึ้นมามาเพื่อใช้ควบคุมการทำงานของวงจรกรองปรับตัว โดยที่สามารถแบ่งสัญญาณ Clock ดังกล่าวได้เป็น 2 กลุ่ม คือ สัญญาณ Clock ที่ใช้งานในระบบของวงจร Booth Encode ซึ่งประกอบสัญญาณ Clock ต่าง ๆ ดังนี้ คือ load_data_clk, msb_clk, load_product_clk, shift_product_clk, product_dummy_clk และ product_clk

ส่วนสัญญาณ Clock ที่เหลือเป็นสัญญาณ Clock ที่ใช้ในการควบคุมการทำงานของระบบวงจรกรองปรับตัว ซึ่งใช้สำหรับควบคุมการรับส่งข้อมูล การปรับค่าช่วงก้วแบบค่าผิดพลาดกำลังสองและการปรับค่าสัมประสิทธิ์ของขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุคประเภท Sign-Sign ซึ่งประกอบด้วยสัญญาณ Clock ต่าง ๆ ดังนี้ data_clk, y_out_clk และ update_mu_clk

ในการทำงานของระบบที่ใช้งานจริง สัญญาณ Clock ที่ป้อนให้กับระบบควรมีเพียงสัญญาณเดียว เพื่อความมีเสถียรภาพของวงจรที่ออกแบบเมื่อนำไปใช้งานจริง เพราะถ้าหากระบบที่ออกแบบต้องใช้สัญญาณ Clock จากภายนอกหลายสัญญาณในการควบคุมการทำงาน เมื่อเวลาผ่าน

ไปหาความถี่ของสัญญาณใดสัญญาณหนึ่งเกิดการเปลี่ยนแปลง จะทำให้คาบเวลาในการทำงานของระบบบางส่วนเกิดการเปลี่ยนแปลงตามไปด้วย ซึ่งอาจส่งผลให้ระบบมีการทำงานที่ผิดพลาดได้ ดังนั้นในการออกแบบระบบของสัญญาณ Clock ที่จะนำไปใช้งานจริงด้วยภาษา VHDL จะใช้สัญญาณ Clock จากภายนอกเพียงสัญญาณเดียว โดยการนำวงจรนับ (Counter) มาใช้นับจำนวนของสัญญาณ Clock ที่ป้อนเข้าสู่ระบบ ซึ่งค่าที่นับได้จะเป็นตัวกำหนดการทำงานของขั้นตอนต่าง ๆ เช่น ค่าที่นับได้เท่ากับหนึ่งจะทำการรับข้อมูลจากภายนอกเข้าสู่ระบบ ค่าที่นับได้เท่ากับสองจะทำการประมวลผลตามขั้นตอนต่าง ๆ ในระบบ และค่าที่นับได้เท่ากับสามจะนำผลลัพธ์ที่ได้ออกสู่ส่วนแสดงผล เป็นต้น

การสร้างจริงสัญญาณ Clock ต่าง ๆ ที่กล่าวมาแล้วข้างต้น เพื่อใช้ในการควบคุมการทำงานของวงจรกรองปรับตัว อาศัยสัญญาณ Clock ที่ป้อนเข้าสู่ระบบซึ่งเป็นสัญญาณ Clock ที่มีความถี่เท่ากับ 10 MHz ซึ่งเป็นสัญญาณ Clock ที่มีคาบเวลาเท่ากับ 100 ns

วงจรกรองปรับตัวที่จะนำมาใช้ในการสร้างจริงออกแบบให้แต่ละรอบของการสุ่มตัวอย่างใช้เวลาเท่ากับ 12000 ns ในช่วงเวลาดังกล่าวจะทำการประมวลผลและส่งข้อมูลออกทางพอร์ตสัญญาณขาออก รวมถึงการปรับค่าสัมประสิทธิ์ของวงจรกรอง และการปรับค่าช่วงก๊าวเพื่อที่จะนำไปใช้ในการประมวลผลในรอบต่อไป จะเห็นว่าในแต่ละรอบของการสุ่มตัวอย่างใช้เวลาค่อนข้างมาก ซึ่งความถี่ในการสุ่มตัวอย่างเท่ากับ $1/12000 \text{ ns} = 83.33 \text{ KHz}$ การสร้างจริงวงจรกรองปรับตัวจะนำมาใช้กับสัญญาณเสียงซึ่งมีความถี่อยู่ที่ประมาณ 20 KHz จะเห็นได้ว่าความถี่ของการสุ่มตัวอย่างมีค่าสูงกว่าความถี่ของสัญญาณเสียง ดังนั้นวงจรที่ทำการออกแบบสามารถที่จะนำไปใช้งานกับสัญญาณเสียงได้

วงจรมานับจะทำการนับขอบขาขึ้นของสัญญาณ Clock ที่ป้อนเข้าสู่ระบบ ซึ่งระบบของวงจรกรองที่จะทำการสร้างจริงจะเริ่มนับนับตั้งแต่ 1 ถึง 120 และจะทำการ Reset ค่าของวงจรมานับ เมื่อถึงค่าสูงสุดที่ตั้งไว้ คือ 120 และจะเริ่มนับนับค่าดังกล่าวใหม่ ซึ่งขั้นตอนการทำงานของวงจรมานับจะทำงานดังกล่าวต่อไปเรื่อย ๆ เพื่อที่จะนำค่าที่นับได้ไปใช้ในการสร้างสัญญาณ Clock ต่าง ๆ ที่จะนำไปใช้ในระบบของวงจรกรองปรับตัว ค่าที่นับได้ของวงจรมานับที่นำมาใช้ในการสร้างสัญญาณ Clock ต่าง ๆ มีรายละเอียดดังนี้ คือ ค่าที่นับได้เท่ากับ 1 นำไปใช้สร้างสัญญาณ `data_clk` ค่าที่นับได้เท่ากับ 5, 11, 17, 23, 29, 35, 41, 47, 53, 59, 65, 71, 77, 83 และ 89 นำไปใช้สร้างสัญญาณ `msb_clk` ค่าที่นับได้เท่ากับ 120 นำไปใช้ในการ Reset ระบบของวงจรมานับ

สัญญาณ Clock ที่เหลือจะสร้างจากสัญญาณ data_clk และ msb_clk สัญญาณ Clock ที่สร้างจากสัญญาณ data_clk เป็นสัญญาณที่เกิดขึ้นเพียงครั้งเดียวในแต่ละรอบของการประมวลผล ซึ่งประกอบด้วยสัญญาณดังต่อไปนี้ คือ สัญญาณ load_data_clk สัญญาณ product_dummy_clk สัญญาณ product_clk สัญญาณ y_out_clk และสัญญาณ update_mu_clk

สัญญาณ Clock ทั้ง 5 สัญญาณ สร้างขึ้นโดยการนำเอาสัญญาณ data_clk ต่อเข้ากับ Register 3 ตัว, 94 ตัว, 96 ตัว, 98 ตัว และ 100 ตัว ตามลำดับ เพื่อใช้หน่วยเวลาในการเกิดขึ้นของสัญญาณดังกล่าว ส่วนที่เหลือเป็นสัญญาณ Clock ที่สร้างจากสัญญาณ msb_clk เป็นสัญญาณที่มีลักษณะเหมือนกับสัญญาณ msb_clk ซึ่งประกอบด้วยสัญญาณดังต่อไปนี้ คือ สัญญาณ load_product_clk และสัญญาณ shift_product_clk

การสร้างสัญญาณทั้งสองทำโดยนำเอาสัญญาณ msb_clk มาต่อกับ Register 2 ตัว และ 4 ตัว ตามลำดับเพื่อใช้หน่วยเวลาในการเกิดของสัญญาณดังกล่าวเช่นเดียวกัน

ที่ผ่านมาได้กล่าวถึงวิธีการในการสร้างสัญญาณ Clock ต่าง ๆ ที่จะนำมาใช้ในการควบคุมระบบการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาคาเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ที่จะทำการสร้างจริงบน FPGA ต่อไปจะอธิบายถึงหน้าที่ในการทำงานของสัญญาณ Clock ต่าง ๆ

สัญญาณ data_clk ใช้สำหรับ Load ข้อมูลต่าง ๆ เข้าสู่ระบบของวงจรกรองปรับตัว เช่น สัญญาณขาเข้าของวงจรกรอง สัญญาณที่ผ่านการป้อนกลับรวมกับสัญญาณเสียงที่มีกำลังต่ำ (เสียงเบามาก) ค่าช่วงก้าวและค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่ผ่านการ Update ในรอบที่ผ่านมาของการประมวลผล

สัญญาณ load_data_clk ใช้สำหรับ Load ค่าสัญญาณขาเข้าของวงจรกรองที่เวลา n และค่าสัมประสิทธิ์ของวงจรกรองที่เวลา n เข้าไปในวงจรคูณ Booth Encode

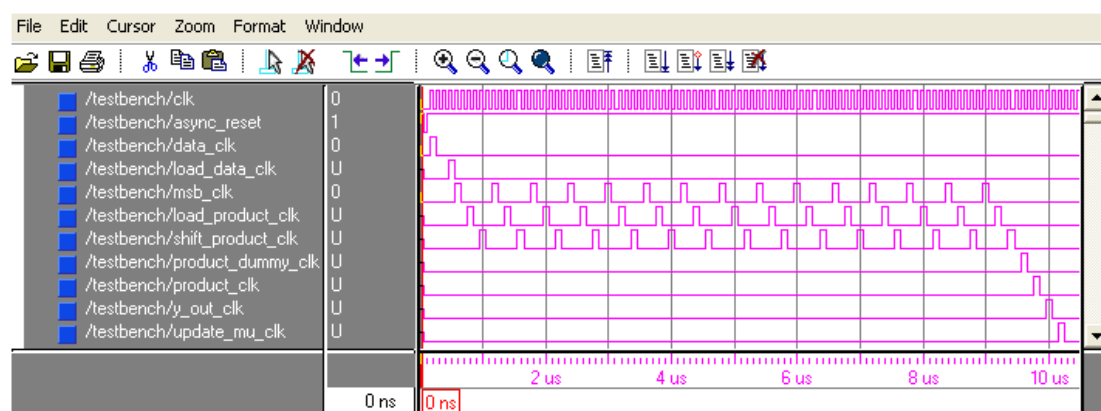
สัญญาณ msb_clk, load_product_clk และ shift_product_clk เป็นสัญญาณที่ใช้สำหรับกำหนดขั้นตอนในการทำงานของวงจร Booth Encode

สัญญาณ product_dummy_clk ใช้สำหรับพิจารณาจำนวนบิตของผลคูณ ที่เกิดจากวงจรคูณ Booth Encode

สัญญาณ `product_clk` ใช้สำหรับนำผลคูณจากวงจรคูณ Booth Encode ที่เสร็จเรียบร้อยแล้ว ออกสู่สัญญาณขาออกของวงจรคูณ Booth Encode

สัญญาณ `y_out_clk` ใช้สำหรับกำหนดช่วงเวลาในการนำข้อมูลที่ประมวลผลได้ของวงจรกรองปรับตัวออกสู่พอร์ตสัญญาณขาออก

สัญญาณ `update_mu_clk` ใช้สำหรับกำหนดช่วงเวลาในการปรับค่าสัมประสิทธิ์และค่าช่วงก้ำของวงจรกรอง โดยจะทำการปรับค่าดังกล่าวที่ขอขาขึ้นและขอขาลงของสัญญาณตามลำดับ

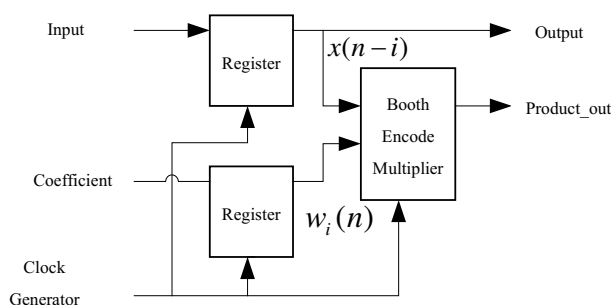


รูปที่ 8.5 สัญญาณ Clock ต่าง ๆ ที่ได้จากวงจรกำเนิดสัญญาณ Clock ที่ออกแบบด้วยภาษา VHDL

รูปที่ 8.5 แสดงสัญญาณ Clock ต่าง ๆ ที่สร้างขึ้นจากวงจรกำเนิดสัญญาณ Clock ซึ่งจะนำไปใช้ในการสร้างจริงบน FPGA โดยใช้ภาษา VHDL ในการออกแบบวงจร สัญญาณ Clock ต่าง ๆ ที่สร้างขึ้นมานี้จะนำไปใช้ในการควบคุมการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาคอนเวอร์ชันเสียง เนื่องจากมีการป้อนกลับทางเสียงในเครื่องช่วยฟังดังที่กล่าวมาแล้วข้างต้น

8.3.2 การสร้างจริง Filter Tap

จากโครงสร้างของสถาปัตยกรรมของวงจรกรองปรับตัวสำหรับแก้ปัญหาคอนเวอร์ชันเสียง เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังในรูปที่ 8.1 Filter Tap เป็นส่วนประกอบที่สำคัญของวงจรกรอง เพราะเป็นส่วนที่ใช้ในการประมวลผลของวงจรกรอง โดยจะทำการคูณค่าสัมประสิทธิ์ของวงจรกรองที่ Tap ต่าง ๆ กับค่าของสัญญาณขาเข้าที่ Tap ดังกล่าวที่เวลา n ใด ๆ Filter Tap แต่ละตัวประกอบด้วย Register 15 บิต แบบขนานและวงจรคูณ Booth Encode ดังแสดงในรูปที่ 8.6 ซึ่งต่อไปจะอธิบายขั้นตอนการออกแบบและหลักการทำงาน เพื่อที่จะทำการสร้างจริงส่วนประกอบทั้งสองด้วยภาษา VHDL



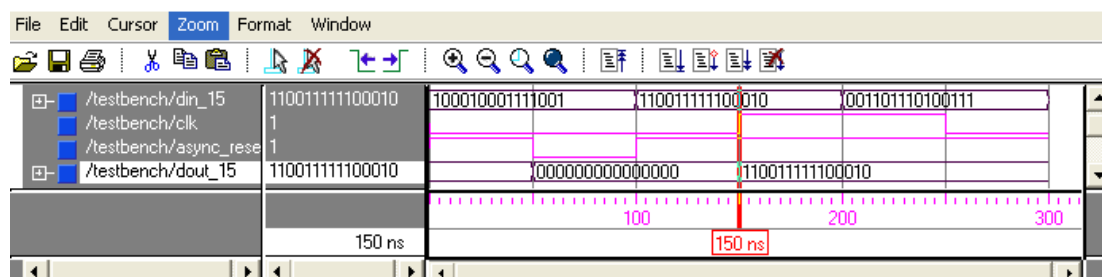
รูปที่ 8.6 รายละเอียดทางโครงสร้างของ Filter Tap

รูปที่ 8.6 แสดงรายละเอียดทางโครงสร้างของ Filter Tap ซึ่งมีส่วนประกอบที่สำคัญดังนี้ คือ Register ขนาด 15 บิต แบบขนาน 2 ตัว และวงจร Booth Encode การทำงานของ Filter Tap แต่ละตัว (ในที่นี้จะพิจารณา Tap ที่ i) จะทำการรับค่าสัญญาณขาเข้าและค่าสัมประสิทธิ์ ที่เวลา n ใด ๆ เข้ามาเก็บไว้ใน Tap ที่ i (สัญญาณ Input อาจจะเป็นสัญญาณที่เข้ามาสู่ระบบเป็นครั้งแรก หาก Filter Tap ดังกล่าวเป็น Tap ที่ 1 หรือเป็นสัญญาณที่ถูกส่งมาจาก Tap ก่อนหน้านี้) เมื่อ Register ได้รับค่าสัญญาณขาเข้าและค่าสัมประสิทธิ์ของวงจรกรองเรียบร้อยแล้ว ข้อมูลทั้งสองจะถูกส่งต่อเข้าไปในวงจร Booth Encode เพื่อคำนวณผลลัพธ์ของ Filter Tap กระบวนการทำงานทั้งหมดจะถูกกำหนดช่วงเวลาในการทำงานด้วยสัญญาณ Clock จากตัวกำเนิดสัญญาณ Clock

Register เป็นส่วนประกอบของวงจรกรองที่ใช้เก็บข้อมูล เพื่อให้คงค่าของสัญญาณต่าง ๆ ไว้จนกว่าจะมีการเปลี่ยนแปลงของสัญญาณ Clock เข้ามาเป็นตัวเปิด Gate ของ Register เพื่อรับข้อมูลใหม่ที่เข้ามาทางด้าน Input ของ Register ให้ไปปรากฏทางด้าน Output ของ Register ซึ่งช่วงเวลาในการทำงานดังกล่าวเรียกว่า Clock to Q time ส่วนการเปลี่ยนแปลงของสัญญาณ Clock ที่จะนำมาใช้ในการเปิด Gate ของ Register สามารถใช้ได้ทั้งขอบขาขึ้นและขอบขาลงของสัญญาณ Clock นอกจากสัญญาณ Clock ที่ใช้ควบคุมการรับข้อมูลของ Register แล้วในการใช้งานจะต้องมีการ Reset ระบบ ซึ่งจะต้องใช้สัญญาณที่ทำหน้าที่ Reset ระบบโดยเฉพาะ ในระบบวงจรกรองปรับตัวที่ทำการออกแบบจะเรียกสัญญาณดังกล่าวว่า ASYNC_RESET การสร้างจริง Register 15 บิต แบบขนานโดยการออกแบบวงจรด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.7 ซึ่ง Register ทั้งหมดที่จะทำการสร้างจริงด้วยภาษา VHDL ที่ จะนำมาใช้งานในวงจรกรองปรับตัวสำหรับแก้ปัญหาก็เกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังจะมีคุณลักษณะเหมือนกับ D Flip Flop ทั่วไป

จากรูปที่ 8.7 สามารถอธิบายหลักการทำงานของ Register ที่ออกแบบวงจรด้วยภาษา VHDL ได้ดังนี้ เมื่อสัญญาณ ASYNC_RESET มีการเปลี่ยนแปลงจาก 1 เป็น 0 จะเกิดการ Reset

ระบบของ Register ทำให้ค่าที่ Output ของ Register ถูก Reset ให้เป็น 0 ทั้ง 15 บิต และหลังจากนั้นเมื่อสัญญาณ ASYNC_RESET เปลี่ยนค่าจาก 0 เป็น 1 เรียบร้อยแล้ว เมื่อมีสัญญาณ Clock ที่ขอบขาขึ้น(ให้วงจรทำงานที่ขอบขาขึ้นของสัญญาณ Clock) เข้ามา Register ก็จะรับค่าของสัญญาณทางด้าน Input ไปปรากฏออกทางด้าน Output ของ Register ซึ่งการทำงานทุกครั้งต้องเริ่มต้นด้วยการ Reset การทำงานของระบบ เพื่อให้สามารถทำงานได้ถูกต้อง



รูปที่ 8.7 ผลการทำงานของ Register ที่ออกแบบวงจรด้วยภาษา VHDL

วงจรคูณ Booth Encode เป็นวงจรคูณที่จะนำมาใช้ในการคูณค่าสัญญาณขาเข้าของ Filter Tap กับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว สาเหตุของการเลือกใช้วงจรคูณแบบ Booth Encode เนื่องจากการสร้างจริงวงจรกรองปรับตัวจะทำการสร้างจริงบน FPGA ซึ่งมีข้อจำกัดทางด้านทรัพยากรดังที่ได้กล่าวมาแล้ว แต่สำหรับวงจรคูณแบบ Booth Encode จะใช้จำนวน Gate ในการสร้างจริงที่น้อยกว่าวงจรคูณที่ใช้กันทั่วไป เช่น วงจรคูณแบบ Array เป็นต้น ดังนั้นการสร้างจริงวงจรคูณในส่วน of Filter Tap ที่จำเป็นต้องใช้วงจรคูณหลายตัว จึงเลือกที่จะใช้วงจรคูณแบบ Booth Encode

ขั้นตอนวิธีการคูณของวงจรคูณ Booth Encode (Booth Algorithm) ที่ใช้ในการคูณเลขฐานสองซึ่งประกอบด้วย ตัวตั้ง (Multiplicand) และตัวคูณ (Multiplier) จำนวน n บิต มีขั้นตอนดังต่อไปนี้

ขั้นตอนที่ 1 กำหนดค่าเริ่มต้นให้ผลคูณ (Product) มีจำนวนบิตเท่ากับ $2n+1$ โดยที่ n บิตแรกทางด้านซ้ายมือกำหนดให้มีค่าเป็น 0 ทุกบิต ต่อจากนั้นกำหนดค่าให้กับ n บิตถัดมาให้มีค่าเท่ากับตัวคูณ (Multiplier) และบิตสุดท้ายทางด้านขวามือสุด (LSB) ให้มีค่าเท่ากับ 0

ขั้นตอนที่ 2 พิจารณา 2 บิต สุดท้ายของผลคูณ ที่ได้กำหนดไว้ในข้อ 1 หากมีค่าเท่ากับ 00 หรือ 11 ไม่มีการคำนวณของขั้นตอนวิธี, หากมีค่าเป็น 01 ให้ทำการบวกค่าตัวตั้งเข้ากับผลคูณที่ตำแหน่ง n บิตแรก และหาก 2 บิต สุดท้ายมีค่าเท่ากับ 10 ให้นำค่าตัวตั้งลบออกจากผลคูณที่ตำแหน่ง n บิตแรก

ขั้นตอนที่ 3 ทำการเลื่อนบิตของผลคูณไปทางขวา 1 บิต และแทนที่บิตทางซ้ายสุดของผลคูณด้วยค่าเดิม

ขั้นตอนที่ 4 กลับไปที่ขั้นตอนที่ 2 ทำซ้ำขั้นตอนต่าง ๆ จนครบ n ครั้ง จากนั้นตัดบิตสุดท้ายของผลคูณที่ได้ทิ้ง จะได้ผลคูณที่ถูกต้อง

ตัวอย่างการใช้ขั้นตอนวิธีการคูณแบบ Booth Encode คูณเลขฐานสองขนาด 4 บิต แสดงดังตารางที่ 8.1 กำหนดให้ตัวตั้ง (Multiplicand) มีค่าเท่ากับ 0010_2 และตัวคูณ (Multiplier) มีค่าเท่ากับ 0110_2 ซึ่งจะได้ผลคูณ (Product) เท่ากับ 0001100_2

ตารางที่ 8.1 ตัวอย่างการคำนวณของขั้นตอนวิธีการคูณแบบ Booth Encode

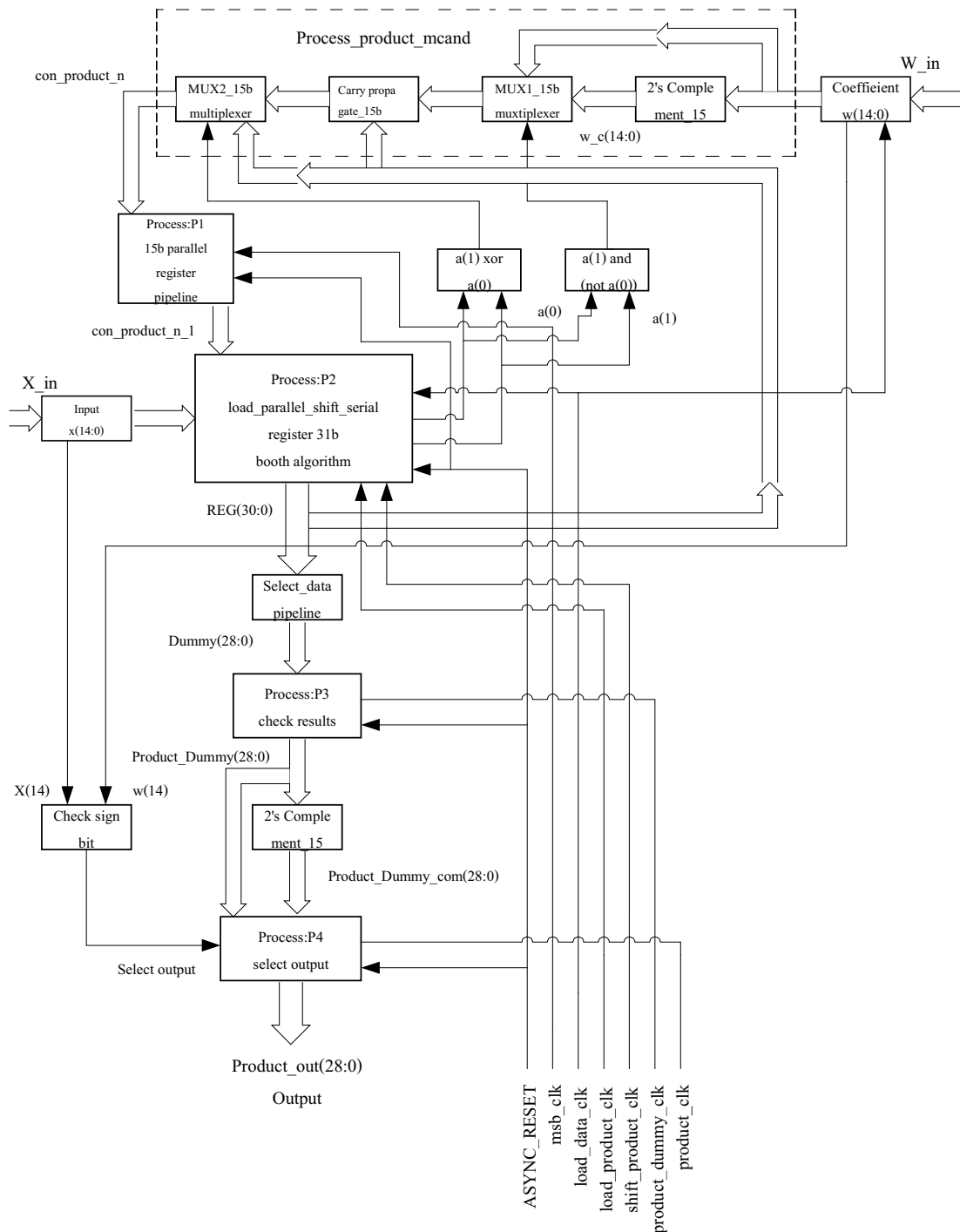
Iteration	Step	Multiplicand	Product
0	Initial Values	0010	0000 0110 0
1	00 => No Operation	0010	0000 0110 0
	Shift Right Product	0010	0000 0011 0
2	10 => Prod = Prod - Mcand	0010	1110 0011 0
	Shift Right Product	0010	1111 0001 1
3	11 => No Operation	0010	1111 0001 1
	Shift Right Product	0010	1111 1000 1
4	01 => Prod = Prod + Mcand	0010	0001 1000 1
	Shift Right Product	0010	0000 1100 0

จะเห็นว่าการคำนวณของขั้นตอนวิธีการคูณแบบ Booth Encode จะเป็นลำดับขั้นตอนที่ซ้ำ ๆ กัน แต่จะต้องมีการกำหนดช่วงเวลาในการทำงานให้ในแต่ละขั้นตอนให้แน่นอน ทำให้ต้องใช้สัญญาณ Clock ในการควบคุมการทำงานของวงจรให้ถูกต้องตามช่วงเวลาที่กำหนด

วงจร Booth Encode เมื่อทำการสร้างจริงจะช่วยประหยัดจำนวน Gate บนชิป FPGA ได้มาก แต่การคำนวณของขั้นตอนวิธีจะใช้เวลาในการคำนวณค่อนข้างสูง ซึ่งการคำนวณในแต่ละขั้นตอนจะใช้สัญญาณ Clock ต่าง ๆ ที่สร้างจากวงจรกำเนิดสัญญาณ Clock ที่ได้ทำการออกแบบวงจรด้วยภาษา VHDL ไว้ก่อนหน้านี้ ในส่วนของช่วงเวลาในการคำนวณ เมื่อขนาดของสัญญาณขาเข้าและค่าสัมประสิทธิ์ของ Filter Tap มีขนาดเท่ากับ 15 บิต ถึงแม้ว่าจะใช้เวลาในการคำนวณค่อนข้างมาก แต่เมื่อทำการคำนวณความถี่ของการสุ่มตัวอย่างในแง่รอบของการประมวลผลความถี่ของการสุ่มตัวอย่างยังคงสูงกว่าความถี่เสียง ดังนั้นสามารถใช้กับสัญญาณเสียงได้ ดังที่กล่าวมาแล้วในหัวข้อของวงจรกำเนิดสัญญาณ Clock

จากลำดับขั้นตอนการประมวลผลของขั้นตอนวิธีการ Booth Encode สามารถนำเอาหลักการดังกล่าวมาทำการสร้างจริงเป็นวงจร Booth Encode ได้ การสร้างจริงวงจร Booth Encode จะทำการสร้างจริงตามโครงสร้างที่ได้ออกแบบไว้ในรูปที่ 8.8

จากโครงสร้างของวงจร Booth Encode ในรูปที่ 8.8 สามารถอธิบายหลักการในการทำงานของวงจรได้ดังนี้ วงจร Booth Encode พร้อมทั้งจะทำงานเมื่อมีการ Reset ระบบ โดยใช้สัญญาณ `ASYNC_RESET` ซึ่งเป็นสัญญาณที่ใช้ Reset ระบบการทำงานของวงจรรองรับตัว เพื่อให้ระบบต่าง ๆ พร้อมทั้งจะทำงาน หลังจากมีการ Reset ระบบเรียบร้อยแล้ว การทำงานของวงจร Booth Encode จะทำงานในขั้นตอนต่อไปเมื่อมีสัญญาณ `load_data_clk` สัญญาณดังกล่าวจะ Load ค่าสัญญาณขาเข้าและค่าสัมประสิทธิ์ของ Filter Tap เข้าไปในวงจร Booth Encode ซึ่งการทำงานในขั้นตอนนี้จะอยู่ที่กระบวนการ P2 ในรูปที่ 8.8 ซึ่งเป็นขั้นตอนที่ 1 ของขั้นตอนวิธีการ Booth Encode เนื่องจากค่าสัญญาณขาเข้าและค่าสัมประสิทธิ์ของวงจรรองรับตัวใช้เลขฐานสองที่มีขนาด 15 บิต ดังนั้นต้องใช้ตัวแปรที่เก็บค่าในการคำนวณเท่ากับ 31 บิต โดยที่ครั้งแรกของการทำงาน 15 บิตแรกทางด้าน MSB จะมีค่าเป็น 0 ทั้ง 15 บิต, 15 บิต ต่อจากนั้นจะมีค่าเท่ากับสัญญาณขาเข้าและบิตสุดท้าย LSB จะมีค่าเริ่มต้นเท่ากับ 0 นอกจากการ Set ค่าเริ่มต้นดังกล่าวแล้วจะมีการตรวจสอบ Sign bit ของตัวคูณทั้ง 2 จำนวน เพื่อใช้ตรวจเช็คในขั้นตอนสุดท้าย นอกจากนี้ค่าสัมประสิทธิ์ของ Filter Tap จะถูกนำเข้าไปที่กระบวนการ `Process_product_mcan` เพื่อนำไปผ่านกระบวนการ 2's Complement ต่อจากนั้นจะทำตามขั้นตอนที่ 2 โดยการตรวจเช็ค 2 บิตสุดท้าย ซึ่งการทำงานของวงจรจะนำ 2 บิตสุดท้ายมาพิจารณาสำหรับเลือกค่าของสัมประสิทธิ์ระหว่างค่าของสัมประสิทธิ์ที่รับเข้ามาจากภายนอก หรือนำเอาค่าสัมประสิทธิ์ที่ผ่านกระบวนการ 2's Complement ไปใช้ สำหรับการบวกหรือลบ หรือไม่ต้องทั้งสองอย่าง ขึ้นอยู่กับการพิจารณาค่า 2 บิตสุดท้ายของผลลัพธ์ที่ได้ในขั้นตอนที่ 2



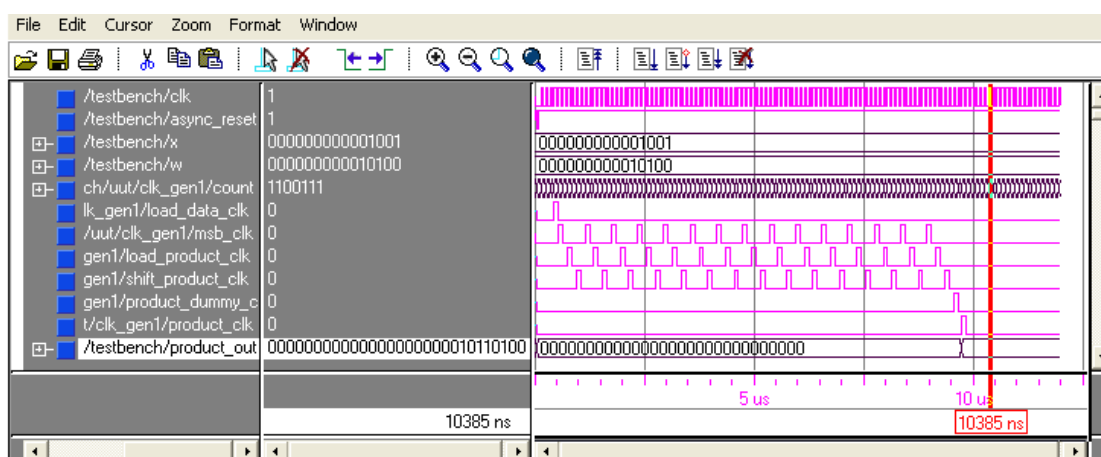
รูปที่ 8.8 โครงสร้างของวงจร Booth Encode

เมื่อทำตามขั้นตอนที่ 2 เสร็จเรียบร้อยแล้ว วงจรก็นำค่าที่คำนวณส่งต่อไปยังกระบวนการ P1 ซึ่งการทำงานมีลักษณะเหมือน Buffer เพื่อรอสัญญาณ `msb_clk` เมื่อมีสัญญาณ `msb_clk` เข้ามาก็จะทำการ Load ค่าที่คำนวณได้ไปเก็บไว้ในกระบวนการ P1 จากนั้นรอสัญญาณ `load_product_clk` เพื่อที่จะนำผลการคำนวณของ 15 บิต ทางด้าน MSB ที่ได้ไปเก็บไว้ที่ตัวแปรเริ่มต้นที่ตำแหน่ง 15

บิตแรกทางด้าน MSB ของตัวแปรที่มีขนาดเท่ากับ 31 บิต หลังจากนั้นรอนจนกว่าจะมีสัญญาณ shift_product_clk เพื่อเลื่อนค่าของตัวแปรตั้งต้นไปทางขวา 1 บิต ในการเลื่อนบิตข้อมูลดังกล่าว บิตข้อมูลทางด้านซ้ายสุด MSB จะต้องมียกเท่ากับค่าเดิมก่อนการเลื่อนบิตข้อมูล เมื่อวงจรทำงานถึงขั้นตอนนี้จะทำทั้งหมด 15 รอบ (เนื่องจากใช้เลขฐานสองขนาด 15 บิต แทนค่าสัญญาณขาเข้าและสัมประสิทธิ์ของ Filter Tap) จึงจะจบการทำงานของขั้นตอนที่ 3

หลังจากการทำงานของวงจรผ่านขั้นตอนที่ 3 จะทำการเลือกบิตที่จะนำมาใช้เป็นผลลัพธ์ของผลคูณที่ได้เก็บไว้ในตัวแปร Dummy ที่มีขนาดเท่ากับ 29 บิต จากนั้นตรวจเช็คความถูกต้องของผลลัพธ์ที่ได้ จากรูปจะอยู่ที่กระบวนการ P3 หลังจากนั้นรอสัญญาณ product_dummy_clk เพื่อที่จะส่งผลที่ได้ตรวจเช็คเรียบร้อยแล้วไปเก็บได้ในตัวแปร Product_dummy เพื่อที่จะเช็คผลลัพธ์กับค่า Sign bit ของสัญญาณขาเข้าและสัมประสิทธิ์ให้มีความถูกต้อง หากค่า Sign bit ของสัญญาณขาเข้าและสัมประสิทธิ์เหมือนกันจะใช้ค่าในตัวแปร Product_dummy เป็นผลลัพธ์ที่ต้องการ ในทางตรงกันข้ามจะใช้ค่า Product_dummy ที่ผ่านกระบวนการ 2's Complement เป็นผลลัพธ์ที่ต้องการ เมื่อทำการตรวจสอบความถูกต้องทุกอย่างเรียบร้อยแล้ว วงจรจะรอสัญญาณ product_clk เพื่อที่จะส่งผลลัพธ์ที่ต้องการออกสู่ทางด้าน Output ของวงจร Booth Encode เป็นการเสร็จสิ้นกระบวนการทำงานของวงจร Booth Encode

จากโครงสร้างของวงจร Booth Encode ดังแสดงในรูปที่ 8.8 เมื่อทำการออกแบบวงจรคุณดังกล่าวด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.9



รูปที่ 8.9 ผลการทำงานของวงจร Booth Encode ที่ออกแบบด้วยภาษา VHDL

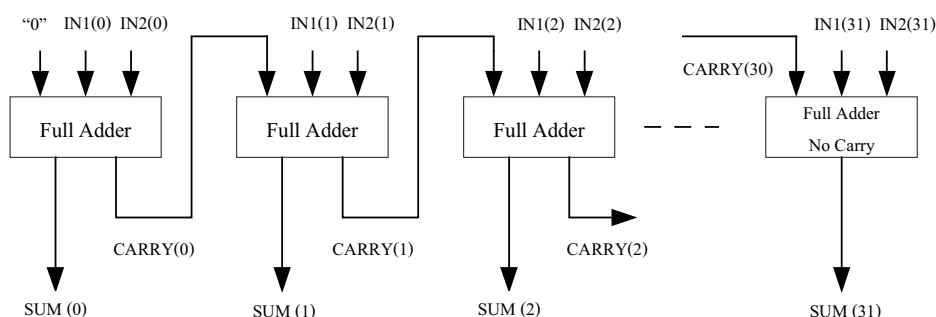
จากผลที่ได้จากการออกแบบวงจรด้วยภาษา VHDL ในส่วนของการสร้างจริง Filter Tap การทำงานของวงจรให้ผลลัพธ์ที่ถูกต้อง สามารถนำวงจรดังกล่าวไปใช้งานได้จริงในวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง

8.3.3 การสร้างจริงวงจรบวก

วงจรบวกที่จะนำมาใช้ในการสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง มี 2 แบบ คือ วงจรบวกแบบ Carry Propagate Adder และวงจรบวกแบบ Carry Save Adder

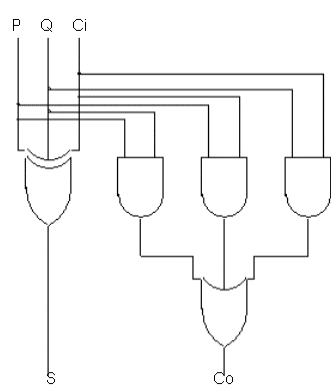
1. วงจรบวกแบบ Carry Propagate Adder

วงจรบวกแบบ Carry Propagate Adder เป็น วงจรบวกที่ใช้สำหรับรวมสัญญาณ 2 สัญญาณเข้าด้วยกัน ส่วนมากวงจรบวกแบบ Carry Propagate Adder ถูกนำมาใช้ในขั้นตอนสุดท้ายของการบวกในแต่ละกระบวนการ เนื่องจากต้องใช้เวลาในการคำนวณค่อนข้างสูง เพราะการบวกแบบ Carry Propagate Adder การบวกจะเริ่มที่บิตที่มีค่าต่ำสุด การบวกแต่ละครั้งจะสร้างตัวทด (Carry) ซึ่งตัวทวดังกล่าวจะถูกนำไปเป็น Input ของการบวกในบิตต่อไปด้วย ดังนั้นการบวกจะต้องใช้เวลาตั้งแต่การบวกบิตแรกที่มีค่าต่ำจนถึงบิตสุดท้ายที่มีค่าสูงสุด สำหรับวงจรกรองที่จะทำการสร้างจริงในการบวกค่าผลคูณที่ผ่านเพิ่มจำนวนบิตเรียบร้อยแล้วจะต้องใช้วงจรบวกแบบ Carry Propagate Adder ขนาด 32 บิต ซึ่งหลักการบวกของวงจรบวกแบบ Carry Propagate Adder จะเริ่มที่บิตที่มีค่าต่ำสุด โดยนำค่าของสัญญาณทั้งสองสัญญาณที่บิต 0 (IN1(0) และ IN2(0)) ป้อนเข้าวงจรบวกแบบ Full Adder ดังแสดงในรูปที่ 8.10 ที่ตำแหน่งดังกล่าว ตัวทดที่ถูกป้อนเข้าวงจรบวกแบบ Full Adder จะถูกกำหนดให้มีค่าเท่ากับศูนย์



รูปที่ 8.10 วงจรบวกแบบ Carry Propagate Adder

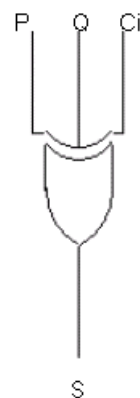
เมื่อวงจรบวกแบบ Full Adder ทำการรวมค่าสัญญาณที่บิต 0 ของสัญญาณทั้งสองเรียบร้อยแล้ว จะได้ผลลัพธ์ 2 ค่า คือ ผลรวมที่บิต 0 (SUM(0)) และตัวทดทที่บิต 0 (Carry(0)) ผลรวมที่บิตดังกล่าวจะถูกส่งต่อไปที่ Output ของวงจรบวก ส่วนตัวทดทจะถูกป้อนเป็นตัวทดทเข้าไว้กับการบวกในบิตต่อไป กระบวนการดังกล่าวจะเกิดขึ้นต่อไปจนถึงการบวกในบิตที่ 30 เมื่อถึงการบวกในบิตที่ 31 ซึ่งเป็นบิตสุดท้าย วงจรบวกที่นำมาใช้จะเป็นวงจรบวกแบบ Full Adder No Carry ซึ่งจะคำนวณหาเฉพาะค่าผลบวกในบิตที่ 31 เท่านั้น ซึ่งผลลัพธ์ที่ได้ของวงจรบวกแบบ Carry Propagate Adder ขนาด 32 บิต แสดงให้เห็นดังรูปที่ 8.10



$$S = P \text{ XOR } Q \text{ XOR } Ci$$

$$Co = (P \text{ AND } Q) \text{ OR } (Q \text{ AND } Ci) \text{ OR } (P \text{ AND } Ci)$$

(ก)

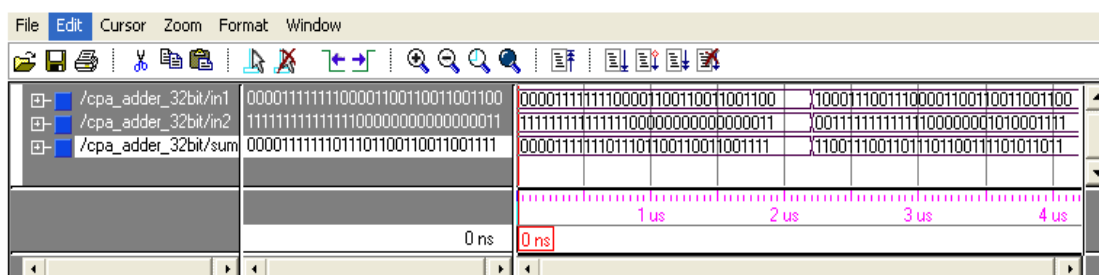


$$S = P \text{ XOR } Q \text{ XOR } Ci$$

(ข)

รูปที่ 8.11 วงจรบวกแบบ (ก) Full Adder (ข) Full Adder No Carry

รูปที่ 8.11 แสดงส่วนประกอบของวงจรบวกแบบ Carry Propagate Adder ซึ่งประกอบด้วยวงจรบวกแบบ Full Adder และ Full Adder No Carry สำหรับผลการทำงานของวงจรบวกแบบ Carry Propagate Adder เมื่อทำการออกแบบวงจรดังกล่าวด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.12

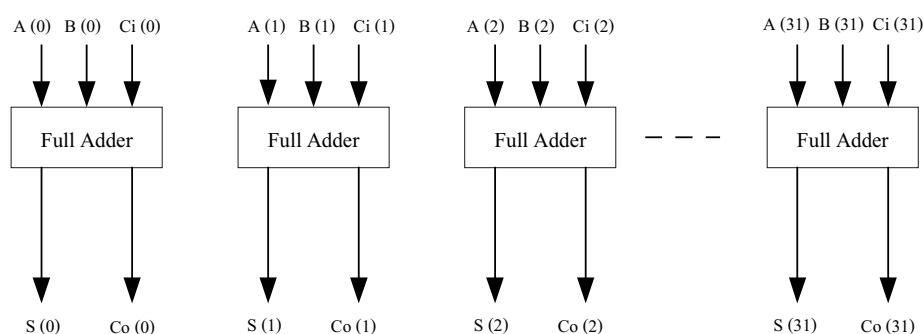


รูปที่ 8.12 ผลการทำงานของวงจรบวกแบบ Carry Propagate Adder ที่ออกแบบด้วยภาษา VHDL

2. วงจรบวกแบบ Carry Save Adder

วงจรบวกแบบ Carry Save Adder เป็นวงจรบวกที่นิยมนำมาใช้ในการรวมค่าสัญญาณที่มีจำนวนมาก เพื่อความรวดเร็วในการประมวลผล สำหรับการใช้งานจริงจะนำเอาวงจรบวกแบบ Carry Save Adder มาใช้ในการคำนวณจนกระทั่งถึงขั้นตอนในการบวกครั้งสุดท้ายจึงจะใช้การบวกแบบ Carry Propagate Adder

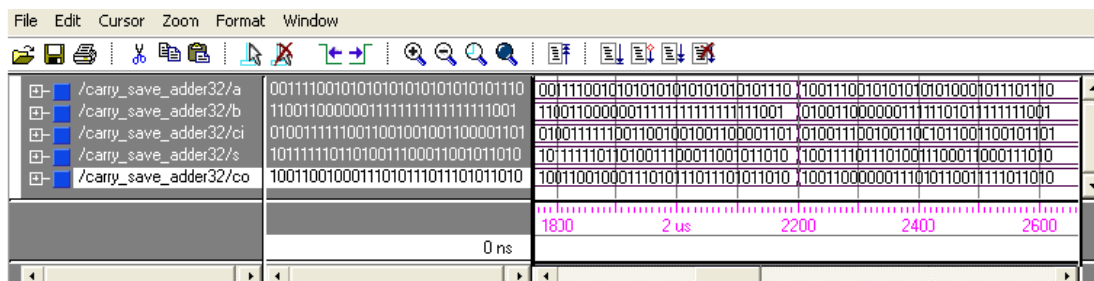
โครงสร้างของวงจรบวกแบบ Carry Save Adder ขนาด 32 บิต ที่จะทำการสร้างจริงเพื่อนำไปใช้ในการประมวลผลของวงจรกรองปรับตัว จะนำเอาวงจรบวกแบบ Full Adder มาใช้ในการประมวลผลดังแสดงในรูปที่ 8.13



รูปที่ 8.13 วงจรบวกแบบ Carry Save Adder

วงจรบวกแบบ Carry Save Adder ที่จะทำการสร้างจริงมีโครงสร้างตามรูปที่ 8.13 การทำงานของวงจรจะต้องป้อนสัญญาณ Input ให้กับวงจรบวกแบบ Carry Save Adder 3 สัญญาณ ส่วนประกอบภายในของวงจรจะใช้วงจรบวกแบบ Full Adder ซึ่งการประมวลผลภายในแต่ละบิตจะเป็นอิสระจากกัน จากรูปที่ 8.13 สัญญาณ Input ที่ป้อนให้กับวงจรบวกแบบ Carry Save Adder คือ สัญญาณ A, B และ C_i ซึ่งมีขนาด 32 บิต โดยที่การคำนวณภายในจะแยกการคำนวณของแต่ละบิตออกจากกัน การประมวลผลในบิตที่ 0 ผลบวก $S(0)$ และตัวทด $Co(0)$ ได้จาก $A(0)$, $B(0)$ และ $C_i(0)$ จนกระทั่งถึงบิตสุดท้าย ผลบวก $S(31)$ และตัวทด $Co(31)$ ได้จากการรวมกันของ $A(31)$, $B(31)$ และ $C_i(31)$ การประมวลผลในแต่ละบิตจะทำงานเหมือนกัน จากผลลัพธ์ที่ได้ของวงจรบวกแบบ Carry Save Adder จะได้ผลบวกและตัวทด ค่าของผลบวกสามารถนำไปใช้งานได้ทันที ส่วนค่าของตัวทดจะต้องทำการเลื่อนตำแหน่งไปทางด้าน MSB 1 บิต จากนั้นใส่ 0 เข้าไปในบิตที่มีค่าต่ำสุด LSB จึงจะสามารถนำค่าดังกล่าวไปใช้งานในกระบวนการต่อไปได้

จากหลักการในการทำงานของวงจรบวกแบบ Carry Save Adder เมื่อทำการออกแบบวงจรดังกล่าวด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.14

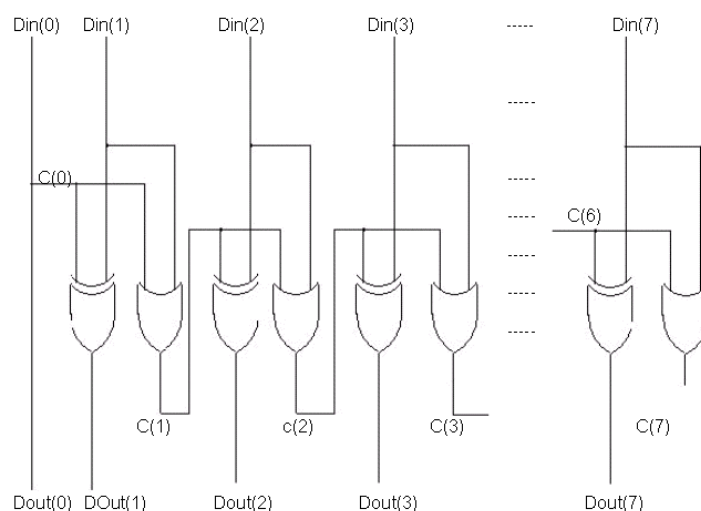


รูปที่ 8.14 ผลการทำงานของวงจรบวกแบบ Carry Save Adder ที่ออกแบบด้วยภาษา VHDL

8.3.4 การสร้างจริงวงจร 2's Complement

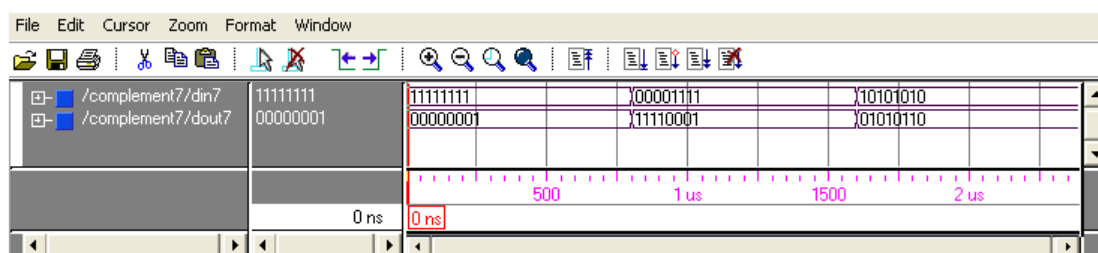
วงจร 2's Complement เป็นวงจรที่มีความสำคัญในการบวกและลบ ทำให้ผู้ออกแบบวงจรสามารถประยุกต์นำเอากระบวนการบวกมาใช้แทนการลบได้อย่างมีประสิทธิภาพ ซึ่งวงจรดังกล่าวจะเปลี่ยนค่าของสัญญาณขาออกให้มีค่าตรงกันข้ามกับสัญญาณขาเข้า โดยที่ขนาดของสัญญาณยังคงเดิม เช่น สัญญาณขาเข้าของวงจรเท่ากับ 01010100_2 (มีค่าเท่ากับ 84 ในเลขฐานสิบ) จะได้สัญญาณขาออกมีค่าเท่ากับ 10101100_2 (มีค่าเท่ากับ -84 ในเลขฐานสิบ) การออกแบบวงจรจะกำหนดตัวแปรขึ้นมา 3 ตัว คือ สัญญาณขาเข้า (Din) ตัวทด (C) และสัญญาณขาออก (Dout) ซึ่งตัวอย่างของวงจรที่จะนำเสนอนี้เป็นวงจร 2's Complement ขนาด 8 บิต ดังนั้นตัวแปรที่กำหนดขึ้นมามีในวงจรทั้ง 3 ตัว จะมีขนาดเท่ากับ 8 บิต

การทำงานของวงจรจะเริ่มตั้งแต่บิต 0 (LSB) ที่บิตดังกล่าวค่าของสัญญาณขาออกจะมีค่าเท่ากับค่าของสัญญาณขาเข้า ($Dout(0) = Din(0)$) และค่าของตัวทด $C(0)$ จะถูกกำหนดให้มีค่าเท่ากับ $Din(0)$ สำหรับค่าของสัญญาณขาออกในบิตต่อไปสามารถหาได้ดังนี้ นำค่าของสัญญาณในบิตดังกล่าวทำการ Exclusive Or กับตัวทดในบิตที่ผ่านมา ตัวอย่างเช่น $Dout(1) = Din(1) \oplus C(0)$ เป็นต้น สำหรับค่าของตัวทดสามารถหาได้จากการนำค่าของสัญญาณขาเข้าในบิตที่ต้องการหาค่าตัวทด มาทำการกระบวนการ Or กับตัวทดในบิตที่ผ่านมา ตัวอย่างเช่น $C(1) = Din(1) \vee C(0)$ สิ่งที่สำคัญในการทำงานของวงจร คือ ต้องกำหนดค่าเริ่มต้นทั้ง 2 ค่า คือ สัญญาณขาออกที่บิต 0 ($Dout(0)$) และตัวทดที่บิต 0 ($C(0)$) ให้มีค่าถูกต้องตามข้อกำหนด หลังจากนั้นจึงทำการคำนวณค่าสัญญาณขาออกและตัวทดที่เหลือในบิตต่อไป จากหลักการในการทำงานของวงจร 2's Complement ที่ผ่านมาสามารถเขียนเป็นรูปวงจรได้ดังรูปที่ 8.15



รูปที่ 8.15 วงจร 2's Complement

สำหรับผลการทำงานของวงจร 2's Complement เมื่อทำการออกแบบวงจรดังกล่าวด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.16



รูปที่ 8.16 ผลการทำงานของวงจร 2's Complement ที่ออกแบบด้วยภาษา VHDL

8.3.5 การสร้างจริงวงจรคูณแบบ Array

วงจรคูณแบบ Array เป็นวงจรคูณอีกแบบหนึ่งที่จะนำมาใช้ในการออกแบบวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง การสร้างจริงของวงจรกรองปรับตัวดังกล่าว จะใช้วงจรคูณแบบ Array ในส่วนของการปรับค่าช่วงก้ำวแบบค่าผิดพลาดกำลังสอง

การคูณแบบ Array จะมีลักษณะดังรูปที่ 8.17 ซึ่งแสดงตัวอย่างการคูณเลขฐานสองที่มีขนาด 8 บิต กำหนดให้ตัวตั้งเท่ากับ $X = x_7x_6x_5x_4x_3x_2x_1x_0$ ตัวคูณเท่ากับ $Y = y_7y_6y_5y_4y_3y_2y_1y_0$ และผลลัพธ์เท่ากับ Z ในการคูณจะคูณทีละบิต เริ่มจากบิตที่มีค่าต่ำสุด นำบิตที่มีค่าต่ำสุดของตัวคูณ y_0 คูณเข้ากับตัวตั้ง X ทุกบิต จะได้ผลคูณทั้งหมด 8 บิต ส่วนบิตที่เหลือด้านหน้าจะแทนด้วย 0 ทั้ง 8 บิต (เนื่องจากการคูณเลขฐานสอง 8 บิต 2 ตัวจะได้ผลลัพธ์มีขนาด 16 บิต) ซึ่งจะได้ผลคูณเป็นแถว

ที่ 1 ต่อจากนั้นจะใช้ ตัวคูณบิตถัดมา คือ y_1 เป็นตัวคูณ ผลคูณของบิตแรกจะต้องใส่ไว้ในตำแหน่งเดียวกับตัวคูณ y_1 เมื่อทำการคูณครบทั้ง 8 บิตแล้วในบิตที่เหลือจะแทนค่าด้วยเลข 0 ซึ่งผลลัพธ์ที่ได้จะอยู่ในแถวที่ 2 ทำตามขั้นตอนดังกล่าวจนครบทั้ง 8 บิตของตัวคูณ หลังจากนั้นทำการรวมค่าในแต่ละคอลัมน์ของผลคูณที่ได้แต่ละแถว จะได้ผลลัพธ์ $Z = z_{15}z_{14}z_{13}z_{12}z_{11}z_{10}z_9z_8z_7z_6z_5z_4z_3z_2z_1z_0$ ที่มีขนาดเท่ากับ 16 บิต

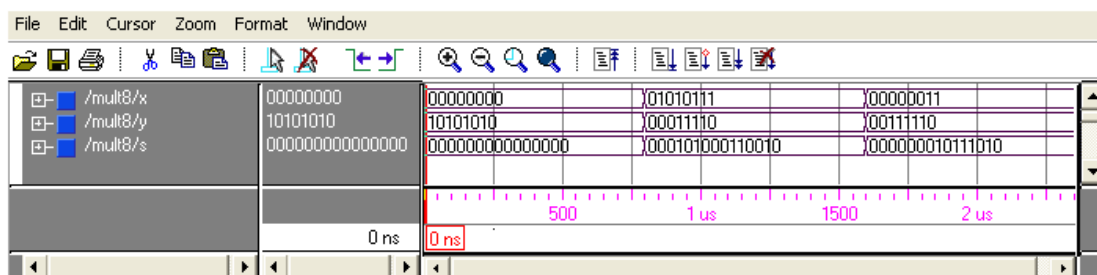
						x_7	x_6	x_5	x_4	x_3	x_2	x_1	x_0	x
						y_7	y_6	y_5	y_4	y_3	y_2	y_1	y_0	
0	0	---	0	0	0	x_7y_0	x_6y_0	x_5y_0	x_4y_0	x_3y_0	x_2y_0	x_1y_0	x_0y_0	}
0	0	---	0	0	x_7y_1	x_6y_1	x_5y_1	x_4y_1	x_3y_1	x_2y_1	x_1y_1	x_0y_1	0	
0	0	---	0	x_7y_2	x_6y_2	x_5y_2	x_4y_2	x_3y_2	x_2y_2	x_1y_2	x_0y_2	0	0	
														}
0	x_7y_7	---	x_3y_7	x_2y_7	x_1y_7	x_0y_7	0	0	0	0	0	0	0	
z_{15}	z_{14}	---	z_{10}	z_9	z_8	z_7	z_6	z_5	z_4	z_3	z_2	z_1	z_0	

รูปที่ 8.17 วิธีการคูณแบบ Array

จากรูปแบบการคูณแบบ Array ดังแสดงในรูปที่ 8.17 ในการออกแบบวงจรคูณดังกล่าวด้วยภาษา VHDL จะอาศัยการเลื่อนบิตในการคูณ หากบิตของตัวคูณที่นำมาพิจารณามีค่าเป็น 1 ให้นำตัวตั้งมาใส่ไว้ที่ผลคูณของบิตดังกล่าว โดยที่บิตที่มีค่าต่ำสุดของตัวตั้งต้องอยู่ที่ตำแหน่งเดียวกับบิตของตัวคูณ ส่วนบิตอื่น ๆ ของตัวตั้งจะถูกใส่ไว้ในตำแหน่งที่มีค่าสูงขึ้นไปทางด้าน MSB และบิตที่เหลือให้แทนด้วย 0 ทั้งหมด ในกรณีที่บิตของตัวคูณที่นำมาพิจารณามีค่าเป็น 0 ผลคูณในแถวดังกล่าวให้แทนค่าด้วย 0 ทั้งหมด เมื่อทำการคูณโดยวิธีดังกล่าวซึ่งใช้วิธีพิจารณาจำนวนบิตที่เป็นเลข 1 และเลข 0 ครบทุกบิตแล้วต่อจากนั้นก็ใช้วงจรบวกที่ได้กล่าวมาแล้วก่อนหน้านี้คำนวณหาผลลัพธ์ของผลคูณที่ได้ เพื่อความสะดวกในการคูณอาจจะกำหนดให้ตัวตั้งและตัวคูณมีค่าเป็นบวกทั้งสองตัวก็ได้โดยใช้วงจร 2's Complement เปลี่ยนค่าดังกล่าวให้มีค่าเป็นบวก แต่จะต้องเก็บค่า Sign bit ครั้งแรกของทั้ง 2 ตัวไว้พิจารณาด้วยคือ หากค่า Sign bit ของทั้ง 2 ตัวเหมือนกัน สามารถที่จะนำผลคูณดังกล่าวเป็นผลลัพธ์ได้ทันที แต่ถ้าค่าของ Sign bit มีค่าต่างกัน ผลคูณที่ได้ต้องนำไปผ่าน

กระบวนการ 2's Complement ด้วยวงจร 2's Complement ที่ออกแบบไว้อีกครั้งจึงจะสามารถนำผลคูณดังกล่าวไปเป็นผลลัพธ์ได้

เนื่องจากวงจรกรองปรับตัวที่จะทำการสร้างจริง จะนำเอาวงจรคูณแบบ Array มาใช้งานจริง จึงนำเอาวิธีการคูณแบบ Array ที่นำเสนอข้างต้นมาสร้างเป็นวงจรคูณแบบ Array ด้วยภาษา VHDL และสำหรับผลการทำงานของวงจรคูณแบบ Array ที่ทำการออกแบบวงจรด้วยภาษาดังกล่าว สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.18



รูปที่ 8.18 ผลการทำงานของวงจรคูณแบบ Array ที่ออกแบบด้วยภาษา VHDL

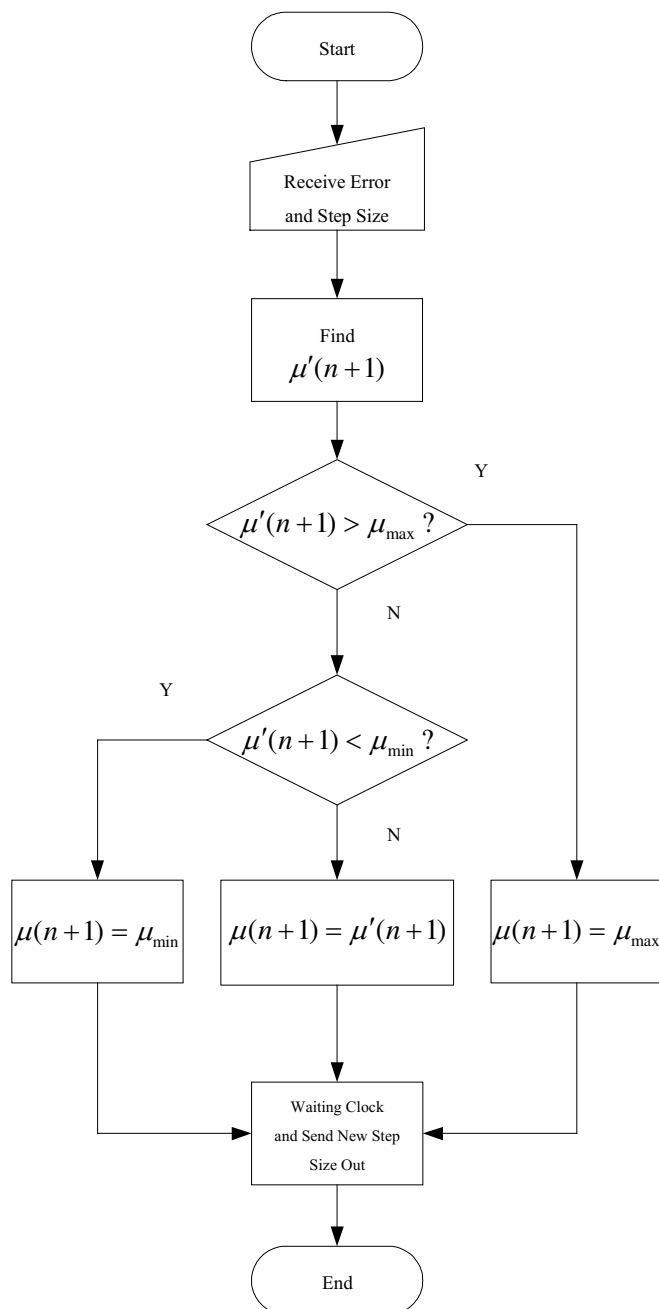
8.3.6 การสร้างจริงวงจรปรับค่าช่วงก้ำแบบค่าผิดพลาดกำลังสอง

วิธีการปรับช่วงก้ำแบบพลวัตที่นำมาใช้ในวิทยานิพนธ์นี้เป็นวิธีการปรับช่วงก้ำแบบค่าผิดพลาดกำลังสอง (Squared Error) ซึ่งมีสมการสำหรับการปรับค่าช่วงก้ำดังสมการที่ (8-2) และสมการที่ (8-3)

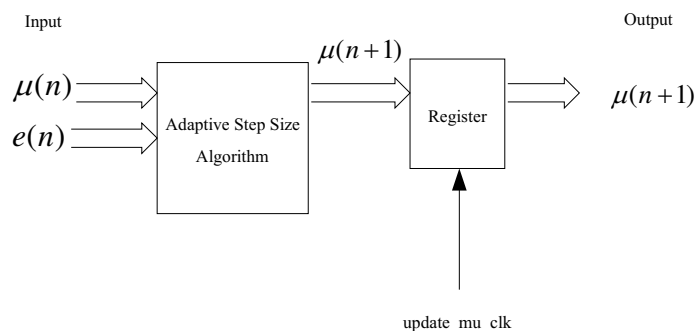
จากสมการที่ (8-2) $\mu'(n+1) = \alpha \mu(n) + \gamma e^2(n)$ การคำนวณค่าช่วงก้ำที่เวลา $n+1$ จะเริ่มจากการคำนวณหาผลคูณของค่าความผิดพลาดกำลังสอง $e^2(n)$ ในขณะเดียวกันก็จะหาค่าผลคูณของค่าช่วงก้ำที่เวลา n กับค่าคงตัว α จากนั้นจะหาค่าผลคูณของค่าคงตัว γ กับค่าความผิดพลาดกำลังสอง $e^2(n)$ โดยที่กระบวนการคูณที่ผ่านมาจะใช้วงจรคูณแบบ Array เมื่อทำการคูณเสร็จเรียบร้อยแล้ว ต่อจากนั้นจะทำกระบวนการบวกต่อไปตามสมการ โดยใช้วงจรบวกแบบ Carry Propagate Adder จากนั้นนำค่าช่วงก้ำที่คำนวณได้ไปเปรียบเทียบกับค่าช่วงก้ำที่มีค่าสูงสุดและต่ำสุด $\mu_{\max}$ และ $\mu_{\min}$ ตามลำดับ เพื่อหาค่าช่วงก้ำที่เหมาะสมตามสมการที่ (8-3) จากลำดับขั้นตอนในการคำนวณดังกล่าว สามารถเขียน Flowchart การทำงานของวงจรที่จะทำการออกแบบด้วยภาษา VHDL สำหรับใช้เป็นวงจรปรับค่าช่วงก้ำแบบค่าความผิดพลาดกำลังสอง ดังแสดงในรูปที่ 8.19

จาก Flowchart ในรูปที่ 8.19 สามารถอธิบายการทำงานของวงจรสำหรับการปรับค่าช่วงก้ำแบบค่าความผิดพลาดกำลังสอง ที่จะใช้ในการสร้างจริงได้ดังนี้ เมื่อวงจรกรองคำนวณค่าความ

ผิดพลาดเสร็จเรียบร้อยแล้ว วงจรปรับค่าช่วงก้ำวแบบค่าความผิดพลาดกำลังสองจะนำค่าความผิดพลาดและค่าช่วงก้ำวที่เวลา n เข้ามาในวงจรและเริ่มทำการคำนวณหาค่าช่วงก้ำวที่เวลา $n+1$ โดยใช้สมการที่ (8-2)

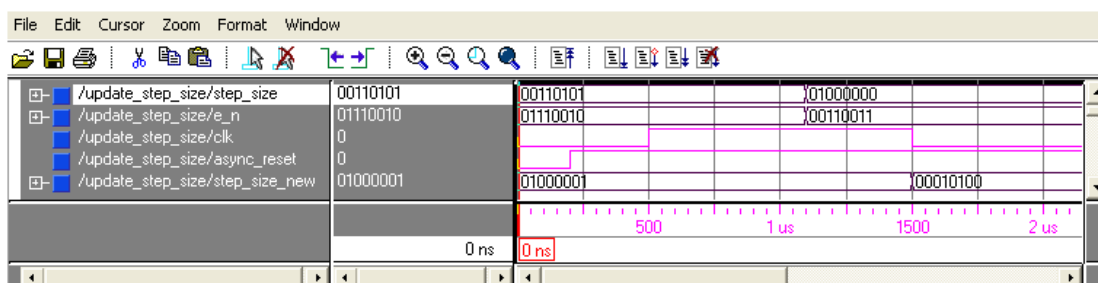


รูปที่ 8.19 Flowchart การทำงานของวงจรสำหรับปรับค่าช่วงก้ำวแบบค่าผิดพลาดกำลังสอง



รูปที่ 8.20 โครงสร้างของวงจรปรับค่าช่วงก้าวแบบค่าผิดพลาดกำลังสอง

เมื่อได้ผลลัพธ์เรียบร้อยแล้ว นำไปเปรียบเทียบกับค่าช่วงก้าวสูงสุดและต่ำสุดเพื่อตรวจสอบขนาดของช่วงก้าวให้อยู่ในช่วงที่ได้กำหนดไว้ หลังจากนั้นนำค่าช่วงก้าวที่คำนวณได้ไปเก็บไว้ใน Register โดยอาศัยการทำงานของขอบล่างของสัญญาณ update_mu_clk ดังแสดงในรูปที่ 8.20 หลังจากนั้นค่าช่วงก้าวที่คำนวณได้จะถูกส่งต่อไปยัง Register ตัวต่อไป เพื่อรอสัญญาณ data_clk เข้ามาจึงจะ Load ค่าช่วงก้าวที่คำนวณได้ไปใช้ในการประมวลผลในรอบต่อไป สำหรับผลการทำงานของวงจรปรับค่าช่วงก้าวแบบค่าความผิดพลาดกำลังสอง ที่ทำการออกแบบวงจรดังกล่าวด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรดังกล่าวที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.21

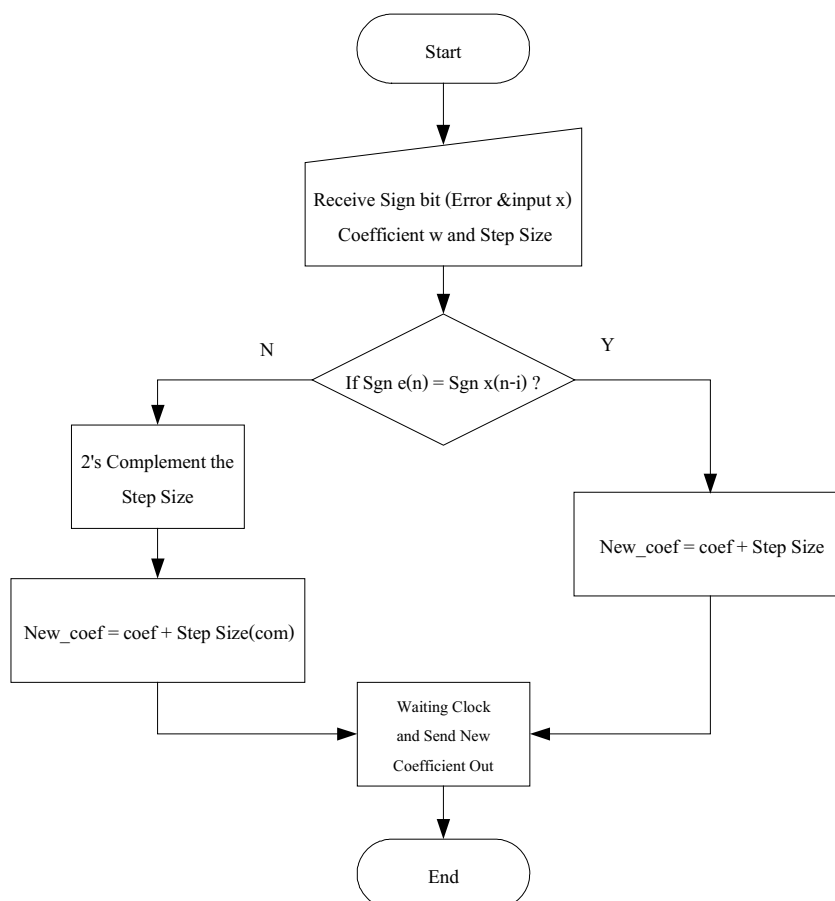


รูปที่ 8.21 ผลการทำงานของวงจรปรับค่าช่วงก้าวแบบค่าผิดพลาดกำลังสอง ที่ออกแบบด้วยภาษา VHDL

8.3.7 การสร้างจริงวงจรปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวโดยใช้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign

การสร้างจริงวงจรสำหรับปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว ซึ่งใช้ขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign ในการปรับค่าสัมประสิทธิ์ เริ่มต้นจากการพิจารณาสมการที่ (8-4) และสมการที่ (8-5)

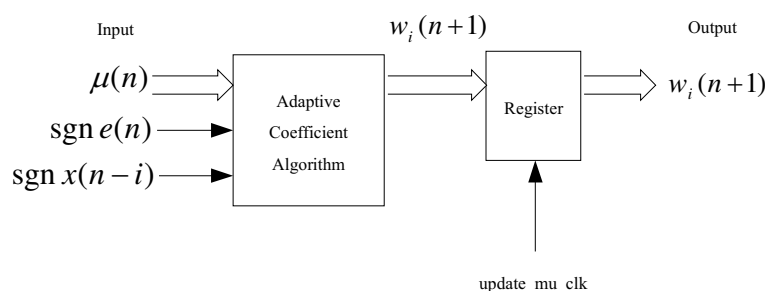
เริ่มพิจารณาจากสมการที่ (8-4) $w_i(n+1) = w_i(n) + \mu(n) [\text{sgn}(x(n-i)) \text{sgn}(e(n))]$ ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวใน Tap ที่ i ที่เวลา $n+1$ สามารถคำนวณหาได้จากค่าสัมประสิทธิ์ของวงจรกรองปรับตัวใน Tap ที่ i ที่เวลา n บวกหรือลบกับค่าช่วงก้าวที่เวลา n โดยจะพิจารณาค่า Sign bit ของสัญญาณขาเข้าของ Filter Tap ณ Tap i ที่เวลา n และค่า Sign bit ของค่าความผิดพลาดที่หาได้จากวงจรกรองปรับตัวที่เวลา n ถ้าค่า Sign bit ของทั้งสองค่าเหมือนกันจะทำการบวกในการปรับค่าสัมประสิทธิ์ของ Tap ที่ i โดยใช้วงจรบวกแบบ Carry Propagate Adder แต่ถ้าหากว่าค่า Sign bit ของทั้งสองต่างกันก็จะนำค่าช่วงก้าว ไปผ่านกระบวนการ 2's Complement ด้วยวงจร 2's Complement แล้วจึงนำค่าที่ได้ไปทำการบวกกันด้วยวงจรบวกแบบ Carry Propagate Adder แทนการลบ จากนั้นตอนในการคำนวณดังกล่าวสามารถเขียน Flowchart การทำงานของวงจรซึ่งจะออกแบบวงจรดังกล่าวด้วยภาษา VHDL ได้ดังรูปที่ 8.22



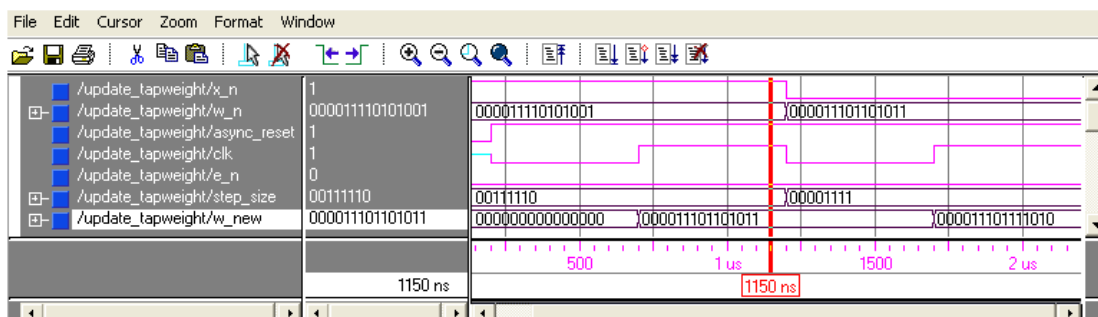
รูปที่ 8.22 Flowchart การทำงานของวงจรสำหรับปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว

จาก Flowchart การทำงานของวงจรสำหรับปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวในรูปที่ 8.22 วงจรจะเริ่มคำนวณค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่เวลา $n+1$ ในแต่ละ Tap เมื่อวงจรกรองปรับตัวได้ทำการคำนวณค่าความผิดพลาด $e(n)$ เสร็จเรียบร้อยแล้ว ค่า Sign bit ของค่า

ความผิดพลาด และค่า Sign bit ของค่าสัมประสิทธิ์ของวงจรกรองใน Tap ที่ i ที่เวลา n จะถูกนำมาพิจารณาในการปรับค่าสัมประสิทธิ์ของวงจรกรองใน Tap ที่ i ซึ่งการปรับค่าสัมประสิทธิ์ในแต่ละ Tap ของวงจรกรองปรับตัวจะทำไปพร้อม ๆ กัน และเป็นอิสระจากกันในการคำนวณแต่จะใช้ค่าช่วงก้ำที่เวลา n ค่าเดียวกัน ตามสมการที่ (8-4) ซึ่งในการคำนวณค่าสัมประสิทธิ์ในแต่ละ Tap ถ้าหากค่า Sign bit ของค่าความผิดพลาดและสัญญาณขาเข้าของ Filter Tap มีค่าเท่ากัน วงจรจะทำการปรับค่าสัมประสิทธิ์โดยการนำเอาช่วงก้ำที่รับเข้ามาทางด้าน Input โดยจะต้องนำค่าทั้งสองมาเรียงให้มีบิตตรงกันก่อน แล้วจึงทำการบวกกัน แต่ถ้ามีค่าต่างกันจะต้องนำค่าช่วงก้ำที่รับเข้ามาไปผ่านกระบวนการของวงจร 2's Complement จากนั้นจึงนำค่าทั้ง 2 มาเรียงให้มีบิตตรงแล้วจึงทำการบวก เมื่อทำการคำนวณค่าสัมประสิทธิ์ของแต่ละ Tap เสร็จเรียบร้อยแล้ว จะส่งค่าสัมประสิทธิ์ที่ได้ในแต่ละ Tap ไปเก็บไว้ใน Register ซึ่งจะใช้สัญญาณ `update_mu_clk` ที่ขอมาขึ้นในการรับค่าดังกล่าวมาเก็บไว้ใน Register และเมื่อมีสัญญาณ `data_clk` ค่าสัมประสิทธิ์ดังกล่าวก็จะถูก Load เข้าไปเก็บไว้ใน Register ที่ใช้เก็บค่าใน Filter Tap อีกครั้งหนึ่ง เพื่อนำไปใช้ในการประมวลผลในรอบต่อไป จากนั้นตอนในการทำงานของวงจรปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่ผ่านมาสามารถแสดงโครงสร้างของวงจรได้ดังรูปที่ 8.23



รูปที่ 8.23 โครงสร้างของวงจรปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวสำหรับ Tap ที่ i



รูปที่ 8.24 ผลการทำงานของวงจรปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว ที่ออกแบบด้วยภาษา

VHDL

สำหรับผลการทำงานของวงจรปรับค่าสัมประสิทธิ์ ที่ทำการออกแบบวงจรด้วยภาษา VHDL สามารถแสดงผลการทำงานของวงจรที่จะนำไปใช้ในการสร้างจริงได้ดังรูปที่ 8.24

ส่วนประกอบต่าง ๆ ที่ได้กล่าวมาแล้วทั้งหมด เป็นส่วนประกอบที่จะนำไปใช้ในการสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ตามโครงสร้างทางสถาปัตยกรรมของวงจรกรองปรับตัวในรูปที่ 8.1 ซึ่งส่วนประกอบต่าง ๆ ทั้งหมดได้ทำการออกแบบวงจรด้วยภาษา VHDL สำหรับส่วนประกอบอื่น ๆ ที่ไม่ได้นำเสนอไว้ ณ ที่นี้จะเป็นส่วนประกอบที่ทำหน้าที่เช่นเดียวกับส่วนประกอบหลัก ๆ ที่ได้นำเสนอไปแล้ว เช่น Register ขนาด 8 บิต วงจรบวกแบบ Carry Propagate Adder ขนาด 8 บิต และวงจรบวกแบบ Carry Propagate Adder ขนาด 15 บิต เป็นต้น ซึ่งรายละเอียดและหลักการทำงานต่าง ๆ จะเหมือนกับ Register ขนาด 15 บิต และวงจรบวกแบบ Carry Propagate Adder ขนาด 32 บิต ที่นำเสนอไปแล้วก่อนหน้านี้

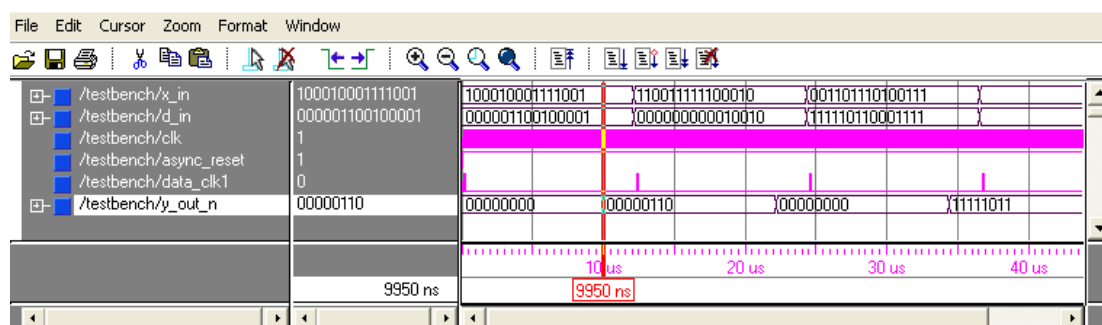
8.4 การสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง

การสร้างจริงส่วนประกอบต่าง ๆ ที่ได้ทำการออกแบบวงจรด้วยภาษา VHDL ซึ่งได้กล่าวมาแล้วก่อนหน้านี้ เป็นส่วนประกอบที่จะนำมาใช้เพื่อประกอบรวมกันเป็นวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง หลังจากที่ส่วนประกอบต่าง ๆ เช่น Register วงจรกำเนิดสัญญาณ Clock วงจรบวกแบบ Carry Save Adder วงจรคูณแบบ Booth Encode และวงจร 2's Complement เป็นต้น สามารถทำงานได้จริงแล้ว (สามารถทำงานในระดับ Logic ได้) จึงนำเอาส่วนประกอบต่าง ๆ ที่ได้ทำการออกแบบวงจรดังกล่าวด้วยภาษา VHDL มาประกอบรวมกันเป็นวงจรกรองปรับตัวตามโครงสร้างของสถาปัตยกรรมของวงจรกรองปรับตัว ที่ได้ทำการออกแบบไว้

หลังจากนำเอาส่วนประกอบต่าง ๆ ที่ทำการออกแบบวงจรด้วยภาษา VHDL มาประกอบรวมกันเป็นวงจรกรองปรับตัวสำเร็จเรียบร้อยแล้ว จะต้องทำการทดสอบวงจรดังกล่าวที่ได้ออกแบบไว้ โดยการเขียนวงจรที่ใช้สำหรับตรวจสอบการทำงานของระบบ หรือที่เรียกว่า Testbench กรอบวงจรกรองปรับตัวที่ได้ทำการออกแบบไว้ เพื่อตรวจสอบความถูกต้องในการทำงานของวงจร ซึ่งสิ่งสำคัญในการจำลองผลการทำงานของวงจรที่ออกแบบด้วยภาษา VHDL คือ การจำลองผลการทำงานจะทำบน โปรแกรม Xilinx ISE ซึ่งจะต้องจำลองผลการทำงานในระดับ Function (Behavioral) เพื่อตรวจสอบความถูกต้องในการทำงานของวงจรที่ได้ทำการออกแบบไว้ และจำลองผลการทำงานในระดับ Gate (Logic) เพื่อตรวจสอบผลการทำงานของวงจรที่จะเกิดขึ้นจริง เมื่อทำ

การ Download วงจรดังกล่าวที่ได้ออกแบบไว้ลงบน FPGA ซึ่งผลการจำลองจะรวมผลของ Delay Time ภายในระหว่าง Gate แต่ละตัวที่จะเกิดขึ้นจริงด้วย ถ้าผลของการจำลองการทำงานในระดับ Gate ยังคงถูกต้อง สามารถสรุปได้ว่า วงจรดังกล่าวที่ได้ทำการออกแบบด้วยภาษา VHDL สามารถใช้งานได้จริงและสามารถทำการสร้างจริงในรูปแบบของวงจรรวม VLSI ได้

การจำลองการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาคาเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง เพื่อตรวจสอบความถูกต้องในการทำงานของวงจรที่ออกแบบด้วยภาษา VHDL ว่าจะสามารถทำงานได้จริงก่อนที่จะนำเอาวงจรที่ออกแบบดังกล่าวไปใช้งานจริงบน FPGA จะต้องทำตามขั้นตอนดังนี้ เริ่มต้นจากการเปรียบเทียบผลการทำงานของวงจรกรองปรับตัวที่ทำการจำลองผลบนโปรแกรม Modelsim เปรียบเทียบกับการจำลองผลการทำงานของระดับ Function บนโปรแกรม Xilinx ISE เพื่อตรวจสอบความถูกต้องในการทำงานของวงจรที่ได้ออกแบบไว้

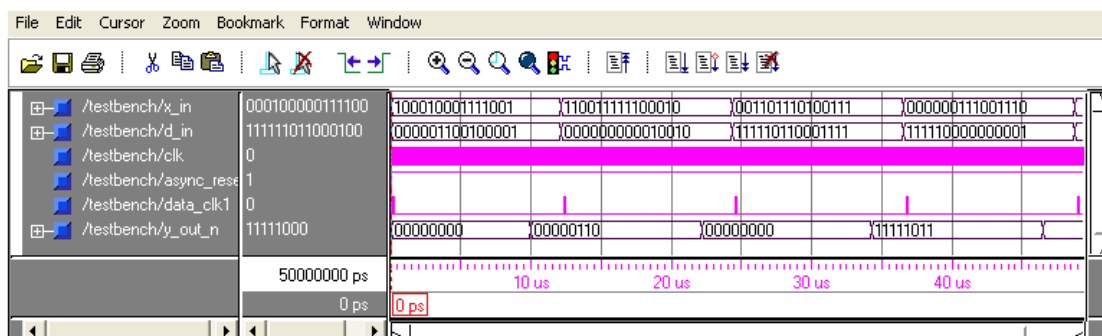


รูปที่ 8.25 ผลการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาคาเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง โดยใช้การจำลองผลการทำงานบนโปรแกรม Modelsim

จากรูปที่ 8.25 แสดงผลการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาคาเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง ซึ่งวงจรต่าง ๆ ที่เป็นส่วนประกอบของวงจรกรองปรับตัว ทำการออกแบบวงจรด้วยภาษา VHDL รูปที่ 8.25 แสดงผลการทำงานที่ได้จากการจำลองผลบนโปรแกรม Modelsim เพื่อใช้สำหรับตรวจสอบความถูกต้องในการทำงานของระบบในขั้นตอนต่าง ๆ ของการทำงาน นอกจากนี้ยังสามารถใช้เปรียบเทียบกับผลที่ได้จากการจำลองผลบนโปรแกรม Xilinx ISE ซึ่งในขั้นตอนแรกจะทำการเปรียบเทียบกับผลการทำงานของวงจรในระดับ Function (Simulate Behavioral VHDL Model)

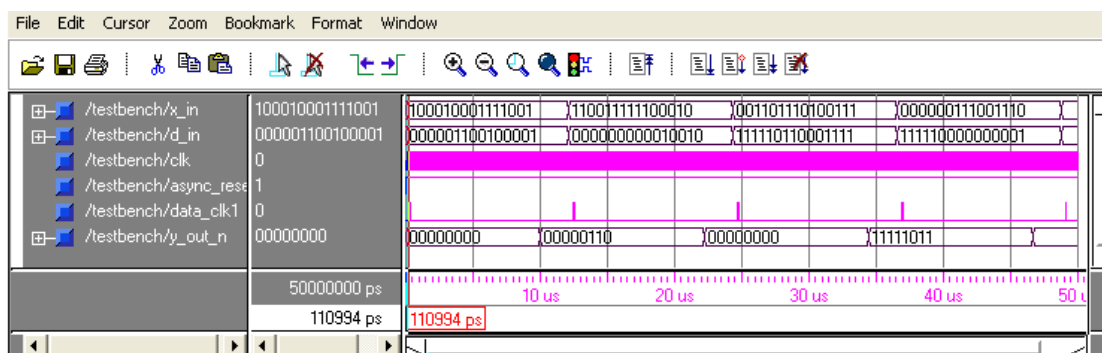
รูปที่ 8.26 แสดงผลการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาคาเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง โดยทำการจำลองผลบนโปรแกรม Xilinx ISE เพื่อ

ตรวจสอบความถูกต้องของการทำงานในระดับ Function เทียบกับผลที่ได้จากการจำลองการทำงานของวงจรกรองปรับตัวบน โปรแกรม Modelsim จากผลการจำลองการทำงานของวงจรกรองปรับตัวที่ได้ในรูปที่ 8.25 และ 8.26 เมื่อทำการตรวจสอบการทำงานของวงจรกรองปรับตัว โดยตรวจสอบค่าของสัญญาณต่าง ๆ ซึ่งจากผลที่ได้มีค่าเท่ากัน ดังนั้นสามารถสรุปได้ว่า การทำงานในระดับ Function ถูกต้อง



รูปที่ 8.26 ผลการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง โดยใช้การจำลองผลการทำงานในระดับ Function (Simulate Behavioral VHDL Model) บนโปรแกรม Xilinx ISE

เพื่อตรวจสอบการทำงานของวงจรกรองปรับตัวที่ได้ทำการออกแบบด้วยภาษา VHDL ให้สามารถนำวงจรที่ได้ออกแบบไว้ไปใช้งานจริงบน FPGA และสามารถนำไปสร้างจริงในรูปแบบของวงจรรวม VLSI จะต้องทำการจำลองผลในระดับ Gate (Simulate Post-Place & Route VHDL Model) ซึ่งผลการจำลองการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังในระดับ Gate ได้ผลการจำลองการทำงานดังรูปที่ 8.27



รูปที่ 8.27 ผลการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง โดยใช้การจำลองผลการทำงานในระดับ Gate (Simulate Post-Place & Route VHDL Model) บนโปรแกรม Xilinx ISE

จากผลการจำลองการทำงานในระดับ Gate ซึ่งรวมผลของ Delay Time ที่จะเกิดขึ้นในการทำงานของ Gate แต่ละตัวให้ผลลัพธ์ดังรูปที่ 8.27 เมื่อนำผลการจำลองที่ได้เทียบกับผลการจำลองในระดับ Function ปรากฏว่าผลการทำงานของวงจรถูกต้อง ดังนั้นทำให้สามารถสรุปได้ว่า วงจรกรองปรับตัวที่ทำการออกแบบวงจรด้วยภาษา VHDL เพื่อใช้แก้ปัญหาคาเกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง สามารถนำไปสร้างและใช้งานได้จริงบน FPGA รวมถึงสามารถนำไปสร้างในรูปแบบของวงจรรวม VLSI ได้จริง

บทที่ 9

การทดสอบและสรุปผลการวิจัยของวงจรกรองปรับตัวแบบ Sign-Sign LMS

วงจรกรองปรับตัวสำหรับแก้ปัญหาก็เกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังที่ได้ทำการออกแบบวงจรด้วยภาษา VHDL ในบทที่ 8 เมื่อนำเอาวงจรดังกล่าวไปทำการจำลองผลในระดับ Gate (Simulate Post-Place & Route VHDL Model) ผลการจำลองในการทำงานยังคงถูกต้องเหมือนกับการจำลองผลในระดับ Function (Simulate Behavioral VHDL Model) จากผลการจำลองดังกล่าวเป็นเครื่องยืนยันถึงผลการทำงานของวงจรกรองปรับตัวที่จะเกิดขึ้น (วงจรสามารถทำงานได้จริง) เมื่อนำเอาวงจรดังกล่าวไปทำการสร้างจริงบน FPGA หรือทำการสร้างจริงในรูปแบบของวงจรรวม VLSI

เนื่องจากการจำลองผลการทำงานของวงจรกรองปรับตัวที่ได้ทำการออกแบบวงจรด้วยภาษา VHDL ในบทที่ 8 เป็นการจำลองผลการทำงานบนคอมพิวเตอร์ ถึงแม้ว่าการจำลองผลการทำงานดังกล่าว จะเป็นเครื่องยืนยันการทำงานของวงจรมานำวงจรดังกล่าวไปใช้งานได้จริง รวมถึงนำวงจรดังกล่าวไปทำการสร้างจริงได้ แต่เพื่อแสดงให้เห็นถึงการทำงานของวงจรกรองปรับตัวที่จะเกิดขึ้นจริง เมื่อนำเอาวงจรดังกล่าวไปใช้งานจริง ดังนั้นเนื้อหาในบทนี้จะกล่าวถึงการทดสอบวงจรกรองปรับตัวที่ได้ทำการออกแบบไว้บน FPGA เพื่อทดสอบผลการทำงานของวงจรซึ่งจะทำให้สามารถสรุปได้ว่า วงจรกรองปรับตัวที่ได้ทำการออกแบบด้วยภาษา VHDL สามารถทำงานได้จริง

9.1 การทดสอบวงจรกรองปรับตัวสำหรับแก้ปัญหาก็เกิดเสียงรบกวน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังบน FPGA

การทดสอบการทำงานของวงจรกรองปรับตัว โดยจะทำการสร้างจริงบน FPGA เพื่อแสดงให้เห็นถึงผลการทำงานของวงจรที่จะเกิดขึ้นจริงบน FPGA ซึ่งสามารถแบ่งขั้นตอนในการทดสอบวงจรกรองปรับตัวบน FPGA ได้ดังนี้ เริ่มจากการทดสอบการทำงานของวงจรกรองปรับตัวที่ได้ทำการออกแบบด้วยภาษา VHDL บนโปรแกรม Xilinx ISE จากนั้นทำการกำหนด Input Port และ Output Port ของวงจรบน FPGA และทำการ Download วงจรที่ได้ออกแบบไว้ลงบน FPGA หลังจากนั้นทำการเลือกวิธีที่จะใช้ในการป้อนข้อมูลเข้าสู่วงจรกรองปรับตัวที่อยู่บน FPGA และทำการวัดสัญญาณ Output ต่าง ๆ บนขาของ FPGA ต่อไปจะกล่าวถึงรายละเอียดในการดำเนินการของแต่ละขั้นตอน

9.1.1 การทดสอบผลการทำงานของวงจรกรองปรับตัว โดยการจำลองผลการทำงานบนโปรแกรม Xilinx ISE

เมื่อทำการออกแบบวงจรกรองปรับตัวด้วยภาษา VHDL เสร็จเรียบร้อยแล้ว หลังจากทำการจำลองผลการทำงานของวงจรบนโปรแกรม Modelsim จนได้ผลการทำงานของวงจรถูกต้อง จากนั้นนำวงจรที่ได้ออกแบบไว้ไปทำการจำลองผลการทำงานบนโปรแกรม Xilinx ISE โดยที่วงจรดังกล่าวจะต้องสามารถจำลองผลการทำงานในระดับ Gate ได้ถูกต้องจึงจะดำเนินการทดสอบในขั้นตอนต่อไปได้ อย่างไรก็ตามต้องระวังในเรื่องของจำนวนสัญญาณ Clock ที่ใช้ในการควบคุมการทำงานของระบบ เนื่องจากการจำลองผลการทำงานบนคอมพิวเตอร์จะทำการเขียนวงจรที่เรียกว่า Testbench ทดสอบการทำงานของวงจร ดังนั้นการทำงานของวงจรกรองปรับตัวที่ทำการจำลองผลทั้งในระดับ Gate และในระดับ Logic ถ้าสัญญาณ Clock ที่ป้อนให้กับวงจรกรองปรับตัวมากกว่า 1 สัญญาณ จะยังคงให้ผลการจำลองที่ถูกต้อง เนื่องจากเป็นสัญญาณที่ป้อนมาจากภายนอกและจะมีเสถียรภาพของสัญญาณสูง (เป็นไปตามค่าที่กำหนดไว้) แต่ในความเป็นจริงถ้าจำนวนของสัญญาณ Clock มากกว่า 1 สัญญาณ จะทำให้การทำงานของระบบผิดพลาดได้เมื่อนำไปใช้งานจริง เพื่อแก้ปัญหาดังกล่าว ในการออกแบบวงจรกรองปรับตัวจะรับสัญญาณ Clock จากภายนอกเพียงสัญญาณเดียวและใช้วงจรกำเนิดสัญญาณ Clock สร้างสัญญาณ Clock อื่น ๆ ที่จำเป็นต้องใช้ขึ้นมาเอง ซึ่งจะทำให้ระบบมีเสถียรภาพในการทำงาน ถ้าวจรที่ออกแบบมีเสถียรภาพในการทำงานและผ่านการจำลองผลการทำงานในระดับ Gate เรียบร้อยแล้วสามารถดำเนินการทดสอบในขั้นตอนต่อไปได้อย่างมีประสิทธิภาพ

9.1.2 การกำหนด Input Port และ Output Port ของวงจรกรองปรับตัวและการ Download วงจรลงบน FPGA

เมื่อระบบที่ออกแบบมีเสถียรภาพในการทำงานและผ่านการจำลองผลในระดับ Gate เรียบร้อยแล้ว ต่อจากนั้นจะทำการกำหนดขาของ Input และ Output ของวงจรกรองปรับตัวลงบน FPGA โดยจะกระทำกระบวนการดังกล่าวบนโปรแกรม Xilinx ISE ซึ่งเรียกขั้นตอนดังกล่าวว่า Back-annotate Pin Locations (ขั้นตอนในการใช้โปรแกรม Xilinx ISE ดูได้จากภาคผนวก ก) หลังจากผ่านกระบวนการดังกล่าว โปรแกรม Xilinx ISE จะกำหนดขาที่เป็น Input Port และ Output Port ลงบน FPGA เพื่อใช้สำหรับรับข้อมูลเข้าและส่งข้อมูลออกจากวงจรกรองปรับตัวที่ตามลำดับโดยปกติแล้วขาต่าง ๆ ที่ถูกกำหนดโดยโปรแกรมจะสามารถใช้งานได้ ทั้งนี้ถ้าผู้ออกแบบต้องการเปลี่ยนตำแหน่งของขาบน FPGA ก็สามารทำได้ โดยเข้าไปที่กระบวนการ Edit Implementation Constraints (Constraints Editor) เพื่อแก้ตำแหน่งขาที่จะใช้เป็น Input Port และ Output Port

เมื่อกำหนดตำแหน่งขาของ FPGA ที่เป็น Input Port และ Output Port ของวงจรกรองปรับตัวเรียบร้อยแล้ว จะต้องทำการ Implement Design ใหม่อีกครั้ง ในกรณีที่มีการเปลี่ยนแปลงตำแหน่งขาของ FPGA แต่ถ้าใช้ตำแหน่งขาของ FPGA ที่กำหนดโดยโปรแกรมสามารถทำขั้นตอนต่อไปได้เลย จากนั้นเตรียมบอร์ด FPGA ให้พร้อม โดยการต่อสาย Download เข้ากับคอมพิวเตอร์ เสียบไฟเข้าบอร์ด FPGA กด Switch On เนื่องจากการ Download วงจรลงบน FPGA บอร์ดที่ใช้ทำงานจะต้องมีไฟเลี้ยงตลอดเวลาวงจรที่ Download จึงจะสามารถทำงานได้ หากไฟดับก็จะต้องทำการ Download วงจรลงบนบอร์ด FPGA ใหม่วงจรดังกล่าวจึงจะสามารถทำงานได้ต่อไป

ในขั้นตอนการ Download วงจรกรองปรับตัวที่ได้ทำการออกแบบไว้ลงบอร์ด FPGA จะต้องเข้าไปที่กระบวนการ Generate Programming File จากนั้นเข้าไปที่ Properties เพื่อกำหนดคุณสมบัติของ Start-Up Clock ให้เป็นแบบ JTAG Clock เนื่องจากการ Download วงจรลงบน FPGA จะใช้สาย JTAG ในการ Download จากนั้นเข้าไปที่กระบวนการ Configure Device (iMPACT) โปรแกรม Xilinx ISE ก็จะเริ่มกระบวนการ Download วงจรลงบน FPGA ถ้าสามารถ Download วงจรลงบน FPGA ได้สำเร็จ โปรแกรมก็จะแสดงผลว่าประสบความสำเร็จในการ Download วงจรลงบน FPGA แต่ถ้าไม่สามารถ Download วงจรลงบน FPGA ได้ก็จะแสดงข้อความดังกล่าวออกมา เมื่อทำการ Download วงจรลงบน FPGA ได้สำเร็จแล้วก็จะดำเนินการในขั้นตอนต่อไป

9.1.3 การป้อนข้อมูลเข้าสู่วงจรกรองปรับตัวบน FPGA

การทดสอบการทำงานของวงจรกรองปรับตัวจะต้องป้อนข้อมูลเข้าสู่ Input Port ของวงจรกรองปรับตัวที่ผ่านการ Download วงจรดังกล่าวลงบน FPGA เรียบร้อยแล้ว ซึ่งการป้อนข้อมูลที่เป็นข้อมูลขาเข้าของวงจรกรองปรับตัวสามารถทำได้ 3 วิธี วิธีแรกโดยการป้อนแรงดันเข้าที่ขาของ FPGA ที่ทำหน้าที่เป็น Input Port ซึ่งระดับแรงดันที่ป้อนด้วยแรงดันสูงจะแทนด้วย Logic High '1' ส่วนแรงดันต่ำแทนด้วย Logic Low '0' แต่เนื่องจากการทดสอบด้วยวิธีการดังกล่าวไม่สามารถทำการปรับเปลี่ยนข้อมูลในแต่ละบิตที่จะป้อนเข้าสู่ Input Port ของวงจรกรองปรับตัวได้พร้อมกันจึงไม่นำวิธีการดังกล่าวมาใช้ในการทดสอบ วิธีที่สองโดยการใช้เครื่อง Pattern Generator เป็นตัวป้อนข้อมูลให้กับ Input Port ของวงจรกรองปรับตัวบน FPGA ซึ่งวิธีที่สองนี้จะต้องมีความรู้ในการใช้งานเครื่อง Pattern Generator และวิธีที่สามโดยการเขียนวงจรด้วยภาษา VHDL ให้ทำหน้าที่เป็น ROM สำหรับเก็บข้อมูลที่จะป้อนให้กับ Input Port ของวงจรกรองปรับตัวบน FPGA สำหรับการเขียนวงจรจำลองที่ทำหน้าที่เป็น ROM เมื่อจะใช้งานสามารถทำได้ 2 แบบ คือ แบบแรกทำการ Download วงจรจำลองที่ทำหน้าที่เป็น ROM ลงบน FPGA ตัวเดียวกับวงจรที่จะทำการทดสอบ

และแบบที่สองใช้บอร์ด FPGA อีกหนึ่งบอร์ดสำหรับ Download วงจรจำลองที่ทำหน้าที่เป็น ROM ลงบน FPGA ซึ่งการทำงานของทั้ง 2 แบบจะเหมือนกัน

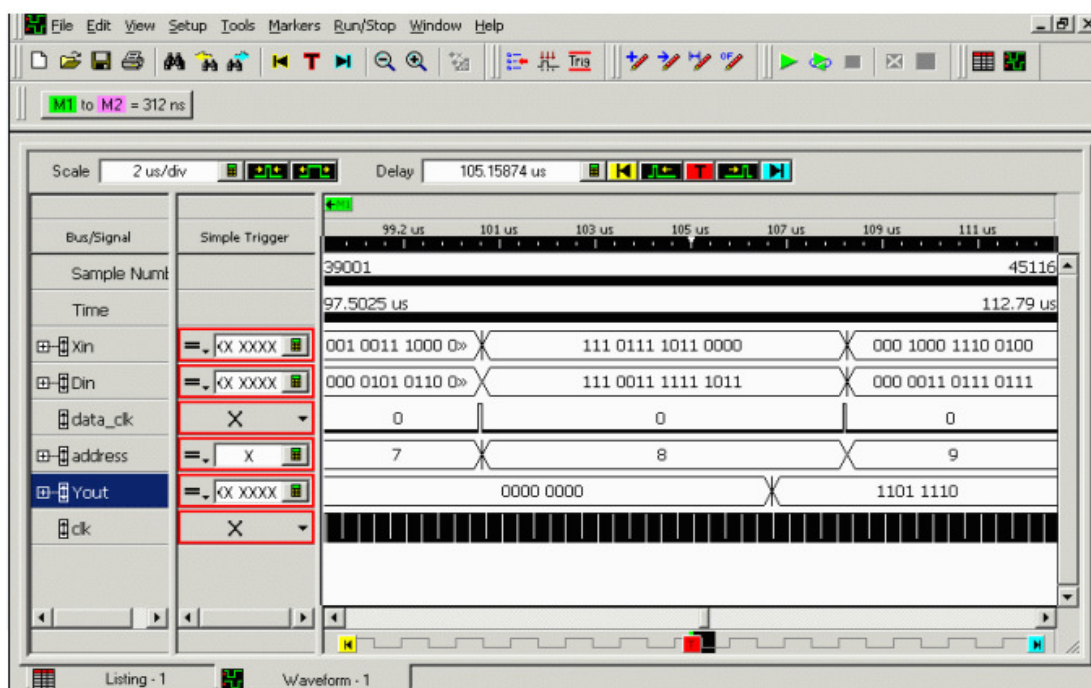
วิธีการที่นำมาใช้ในการป้อนข้อมูลเข้าที่ Input Port ของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA เพื่อทดสอบการทำงานของวงจรกรองปรับตัวสำหรับวิทยานิพนธ์นี้ จะใช้วิธีการเขียนวงจรจำลองที่ทำหน้าที่เป็น ROM ที่ใช้สำหรับป้อนข้อมูลให้กับวงจรกรองปรับตัว โดยจะใช้สัญญาณ Clock จากวงจรกำเนิดสัญญาณ Clock เป็นตัวควบคุมการเปลี่ยนแปลงข้อมูลภายในเพื่อที่จะส่งข้อมูลดังกล่าวออกมาที่ Input Port ของวงจรกรองปรับตัว ซึ่งจะรอสัญญาณ data_clk ของวงจรกำเนิดสัญญาณ Clock ในการ Load ข้อมูลต่าง ๆ เข้าสู่วงจรกรองปรับตัวต่อไป โดยจะทำการสร้างจริงวงจรกรองปรับตัวและวงจรที่ทำหน้าที่เป็น ROM ลงบน FPGA ตัวเดียวกัน

ส่วนสัญญาณ Clock ที่เป็นสัญญาณ Input ของวงจรกรองปรับตัวที่จะนำไปใช้ในการสร้างสัญญาณ Clock ต่าง ๆ เพื่อใช้ควบคุมการทำงานของวงจรกรองปรับตัว สามารถเลือกแหล่งกำเนิดสัญญาณ Clock ดังกล่าวได้ 3 แบบ คือ สร้างสัญญาณ Clock ดังกล่าวจากเครื่อง Pattern Generator หรือสร้างสัญญาณ Clock จากเครื่อง Signal Generator และใช้สัญญาณ Clock จาก Oscillator บนบอร์ด FPGA ซึ่งในการทดสอบการทำงานของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA สำหรับวิทยานิพนธ์นี้ใช้สัญญาณ Clock บนบอร์ด FPGA เป็นสัญญาณ Input ที่ป้อนให้กับวงจรกรองปรับตัว

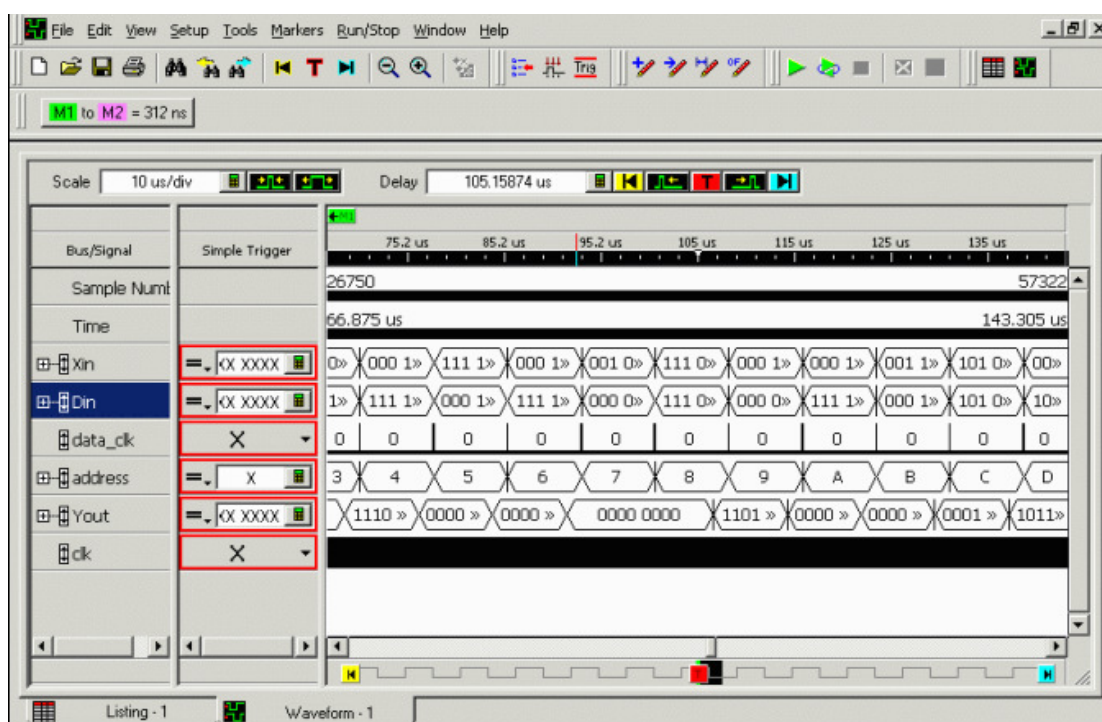
9.1.4 การตรวจสอบผลการทำงานของวงจรกรองปรับตัวบน FPGA

สำหรับการตรวจสอบผลการทำงานของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA สามารถวัดผลการทำงานของวงจรดังกล่าวได้ โดยใช้เครื่อง Logic Analyzer สำหรับตรวจวัดสัญญาณที่ต้องการตรวจสอบ ซึ่งตัวอย่างของสัญญาณที่วัดได้จริงในการทดสอบวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA แสดงให้เห็นดังรูปที่ 9.1 และรูปที่ 9.2

รูปที่ 9.1 แสดงผลการทำงานของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA โดยใช้เครื่อง Logic Analyzer วัดสัญญาณที่ขาของ FPGA ที่ทำหน้าที่เป็น Input Port และ Output Port ให้กับวงจรกรองปรับตัว โดยที่จะแสดงผลในรูปแบบของสัญญาณ ส่วนรูปที่ 9.2 แสดงผลการทำงานของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA โดยใช้เครื่อง Logic Analyzer วัดสัญญาณที่ขาของ FPGA ที่ทำหน้าที่เป็น Input Port และ Output Port ให้กับวงจรกรองปรับตัว โดยที่จะแสดงผลในรูปแบบของรายละเอียดของสัญญาณต่าง ๆ ที่ต้องการวัด



(ก)



(ข)

รูปที่ 9.1 ผลการทำงานของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA โดยแสดงรูปสัญญาณที่วัดได้จากเครื่อง Logic Analyzer (ก) Zoom In (ข) Zoom Out

File Edit View Setup Tools Markers Run/Stop Window Help

M1 to M2 = 312 ns

Sample Number	Time	Xin	Din	address
485720	1.2143000 ms	110 0111 1110 0010	000 0000 0001 0010	X
485721	1.2143025 ms	110 0111 1110 0010	000 0000 0001 0010	
485722	1.2143050 ms	110 0111 1110 0010	000 0000 0001 0010	
485723	1.2143075 ms	110 0111 1110 0010	000 0000 0001 0010	
485724	1.2143100 ms	110 0111 1110 0010	000 0000 0001 0010	
485725	1.2143125 ms	110 0111 1110 0010	000 0000 0001 0010	
485726	1.2143150 ms	110 0111 1110 0010	000 0000 0001 0010	
485727	1.2143175 ms	110 0111 1110 0010	000 0000 0001 0010	
485728	1.2143200 ms	110 0111 1110 0010	000 0000 0001 0010	
485729	1.2143225 ms	110 0111 1110 0010	000 0000 0001 0010	
485730	1.2143250 ms	110 0111 1110 0010	000 0000 0001 0010	
485731	1.2143275 ms	110 0111 1110 0010	000 0000 0001 0010	
485732	1.2143300 ms	110 0111 1110 0010	000 0000 0001 0010	
485733	1.2143325 ms	110 0111 1110 0010	000 0000 0001 0010	
485734	1.2143350 ms	110 0111 1110 0010	000 0000 0001 0010	
485735	1.2143375 ms	110 0111 1110 0010	000 0000 0001 0010	
485736	1.2143400 ms	110 0111 1110 0010	000 0000 0001 0010	
485737	1.2143425 ms	110 0111 1110 0010	000 0000 0001 0010	
485738	1.2143450 ms	110 0111 1110 0010	000 0000 0001 0010	

Listing - 1 Waveform - 1

(ก)

File Edit View Setup Tools Markers Run/Stop Window Help

M1 to M2 = 312 ns

Din	address	Yout	data_clk	clk
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0
1110 0010	000 0000 0001 0010	1	1111 1101	0

Listing - 1 Waveform - 1

(ข)

รูปที่ 9.2 ผลการทำงานของวงจรกรองปรับตัวที่ทำการสร้างจริงบน FPGA โดยแสดงรายละเอียดของข้อมูลที่วัดได้จากเครื่อง Logic Analyzer (ก) ส่วนที่ 1 (ข) ส่วนที่ 2

9.2 สรุปผลการทดสอบวงจรกรองปรับตัวบน FPGA

จากการทดสอบวงจรกรองปรับตัวที่ใช้สำหรับแก้ปัญหาการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟังที่ได้ทำการออกแบบด้วยภาษา VHDL และนำมาสร้างจริงบน FPGA จากผลการทดสอบซึ่งวัดผลการทำงานโดยใช้เครื่อง Logic Analyzer สามารถสรุปได้ว่า วงจรกรองปรับตัวที่ใช้สำหรับแก้ปัญหาการเกิดเสียงฮอน เนื่องจากการป้อนกลับทางเสียงในเครื่องช่วยฟัง โดยนำเอาขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign และอาศัยการปรับช่วงก้าวแบบพลวัตซึ่งใช้วิธีการปรับช่วงก้าวแบบค่าผิดพลาดกำลังสองมาช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัว สามารถทำงานได้จริงบน FPGA และสามารถทำการสร้างจริงในรูปแบบของวงจรรวม VLSI ได้

9.3 สรุปผลการวิจัยของวงจรกรองปรับตัวแบบ Sign-Sign LMS

งานวิจัยนี้นำเอาขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign มาประยุกต์ใช้ ในการปรับค่าสัมประสิทธิ์ของวงจรกรองปรับตัวที่ใช้สำหรับแก้ปัญหาการเกิดเสียงฮอนในเครื่องช่วยฟัง โดยใช้การปรับค่าช่วงก้าวแบบ Squared error มาช่วยปรับปรุงประสิทธิภาพของขั้นตอนวิธีดังกล่าว ทำให้สามารถลดความซับซ้อนของฮาร์ดแวร์ ทั้งนี้จะส่งผลให้สามารถประหยัดพื้นที่ในการสร้างจริงซึ่งจะทำให้ขนาดของเครื่องช่วยฟังมีขนาดเล็กกว่าการนำเอาขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดแบบดั้งเดิมมาทำการสร้างจริง และยังส่งผลต่อการสูญเสียพลังงานที่จะลดลงจากการลดความซับซ้อนของขั้นตอนวิธี ซึ่งจะช่วยให้อายุการใช้งานของแบตเตอรี่ในเครื่องช่วยฟังมีอายุการทำงานที่ยาวนานขึ้น และจะทำให้ผู้ใช้เครื่องช่วยฟังไม่ต้องเปลี่ยนแบตเตอรี่บ่อย ๆ

การสร้างจริงวงจรต่าง ๆ ที่เป็นส่วนประกอบของวงจรกรองปรับตัว เมื่อนำมาประกอบรวมกันสามารถทำงานได้อย่างมีประสิทธิภาพ เนื่องจากการออกแบบวงจรดังกล่าวด้วยภาษา VHDL ได้ทำการทดสอบวงจรต่าง ๆ ที่เป็นส่วนประกอบของวงจรกรองปรับตัวให้สามารถทำงานได้จริงก่อนนำมาใช้งาน นอกจากนี้ระบบในการทำงานของวงจรกรองปรับตัวยังถูกออกแบบมาให้สามารถใช้งานได้จริงและมีเสถียรภาพในการทำงาน เนื่องจากใช้สัญญาณ Clock จากภายนอกมาควบคุมการทำงานเพียงสัญญาณเดียว จากผลการทดสอบการทำงานของวงจรกรองปรับตัวสำหรับแก้ปัญหาการเกิดเสียงฮอน สามารถสรุปได้ว่า วงจรกรองปรับตัวดังกล่าวที่ทำการสร้างจริงด้วยภาษา VHDL สามารถทำงานได้จริงบน FPGA และสามารถนำไปสร้างจริงในรูปแบบของวงจรรวม VLSI เพื่อนำไปใช้งานในรูปแบบของฮาร์ดแวร์ได้จริง

ข้อจำกัดสำหรับการนำเอาขั้นตอนวิธีกำลังสองเฉลี่ยน้อยสุดประเภท Sign-Sign มาประยุกต์ใช้ในการสร้างจริงวงจรกรองปรับตัวสำหรับแก้ปัญหการเกิดเสียงหอน คือ จะทำให้ประสิทธิภาพในการปรับตัวของขั้นตอนวิธีลดลง เนื่องจากการประมวลผลที่ใช้ค่าน้ำหนักเป็นค่า Sign bit ของค่าความผิดพลาดและค่าของสัญญาณขาเข้าในแต่ละ Tap ไม่ใช่ค่าที่แท้จริง แต่อย่างไรก็ตาม ข้อจำกัดดังกล่าวเป็นผลมาจากการลดความซับซ้อนของฮาร์ดแวร์

บทที่ 10

การประยุกต์ใช้หลักวงจรรองกำลังสองต่ำในการลดสัญญาณรบกวนแบบ Speckle

10.1 ความเป็นมาของงานวิจัย

เนื้อหาในบทนี้เป็นการประยุกต์ใช้หลักการกำลังสองต่ำสุด (Least Squares) ในการลดทอนสัญญาณรบกวนแบบ speckle ในภาพ Synthetic Aperture Radar (SAR) และในภาพอัลตราซาวนด์ ทางด้านการแพทย์ หลักการที่ใช้ในวงจรรองกำลังสองต่ำสุด (Least Squares Filter) จะพยายามทำให้กำลังสองของค่าความผิดพลาดระหว่างสัญญาณออกของวงจรรองกับค่าสัญญาณขาเข้ามีค่าต่ำสุด โดยหลักการดังกล่าวนี้จะเป็นหลักการเดียวกับที่ใช้วงจรรองแบบค่าเฉลี่ยกำลังสองต่ำสุด (LMS) ในการศึกษาเกี่ยวกับวิธีออฟดิแม็กซ์เพื่อหาเฉลี่ยของค่าสัมประสิทธิ์ของวงจรรอง LMS เมื่อเทียบกับเวลา ซึ่งใช้ในการลดอันดับของวงจรรองแบบ LMS ผู้วิจัยพบว่าเราสามารถใช่วิธีการคำนวณค่าสถิติเฉลี่ยทางเวลาของค่าสัมประสิทธิ์ของวงจรรองดังกล่าวมาประยุกต์ใช้ในการประมวลสัญญาณภาพแบบ SAR และภาพอัลตราซาวนด์ โดยใช้หลักการข้างต้นมาปรับปรุงวงจรรองซาวิตกี-โกเลย์สองมิติ (2-dimensional Savitzky-Golay filters) อันเป็นวงจรรองแบบกำลังสองต่ำสุดประเภทหนึ่งซึ่งประมวลกลุ่มข้อมูลด้วยฟังก์ชันพหุนาม และเหมาะสำหรับใช้ลดทอนสัญญาณรบกวนแบบ Speckle ในภาพแบบ SAR และภาพอัลตราซาวนด์ [14] หัวข้อ 10.2 อธิบายหลักการของวงจรรองซาวิตกี-โกเลย์สองมิติ หัวข้อ 10.3 แสดงหลักการทางสถิติที่พัฒนาขึ้นสำหรับวงจรรองซาวิตกี-โกเลย์สองมิติที่ใช้ในการลดทอนสัญญาณรบกวนแบบ speckle บนภาพ SAR และบทที่ 11-12 แสดงการปรับปรุงหลักการวงจรรองซาวิตกี-โกเลย์สองมิติสำหรับลดทอนสัญญาณรบกวน speckle บนภาพ อัลตราซาวนด์ทางการแพทย์

10.2. วงจรรองซาวิตกี-โกเลย์สองมิติ

วงจรรองซาวิตกี-โกเลย์สองมิติได้ถูกพัฒนานำมาใช้ในการลดทอนสัญญาณรบกวนในภาพอัลตราซาวนด์ [14] เริ่มจากพิจารณาภาพขนาด $M \times N$ พิกเซลซึ่งแสดงได้ในรูปของอาร์เรย์ของกลุ่มข้อมูลสองมิติ $f(i, j)$, เมื่อ $i \in [1, \dots, M]$ และ $j \in [1, \dots, N]$. กำหนดให้หน้าต่างของชุดข้อมูลที่เรสนใจมีขนาด $(2L+1) \times (2L+1)$ โดยมีตำแหน่งศูนย์กลางอยู่ที่ (i, j) ดังสมการ

$$D_{i,j} = \{f(i+m, j+n) : -L \leq m \leq L, -L \leq n \leq L\} \quad (10-1)$$

ให้ $p_{i,j}$ เป็นฟังก์ชันพหุนามสองมิติ

$$p_{i,j}(m,n) = \sum_{s=0}^S \sum_{t=0}^T a_{i,j}(s,t) m^s n^t \quad (10-2)$$

โดย m และ n เป็นเลขจำนวนเต็มดังที่อ้างไว้ในสมการที่ (10-1), S และ T เป็นอันดับสูงสุดของ m และ n ตามลำดับ ระเบียบวิธีของวงจรรองชาวิสกี-โกเลย์สองมิติจะคำนวณค่าเอาต์พุตที่ตำแหน่ง (i,j) ด้วยการฟิตฟังก์ชันพหุนาม $p_{i,j}$ แบบกำลังสองน้อยสุด (Least square) ไปยังข้อมูล f แต่ละพิกเซลในหน้าต่าง $D_{i,j}$ และเลือกเอาต์พุตของวงจรรองจากฟังก์ชันพหุนามที่ตำแหน่ง $p_{i,j}(0,0)$ ซึ่งมีค่าเท่ากับค่าสัมประสิทธิ์ $a_{i,j}(0,0)$

วัตถุประสงค์ในการฟิตกลุ่มข้อมูลด้วยหลักการกำลังสองน้อยสุดในวงจรรองชาวิสกี-โกเลย์สองมิติ สามารถนิยามด้วย

$$\sum_{m,n \in D_{i,j}} w_{i,j}(m,n) \{f(i+m, j+n) - p_{i,j}(m,n)\}^2 \quad (10-3)$$

โดย $w_{i,j}$ คือฟังก์ชันถ่วงน้ำหนักแต่ละตำแหน่งในหน้าต่าง $D_{i,j}$ ในการที่เราจะใช้ฟังก์ชันพหุนามฟิตกลุ่มข้อมูลด้วยหลักการกำลังสองน้อยสุดให้มีประสิทธิภาพนั้น เราได้ขยายหลักการที่ได้พัฒนามาแล้วในระเบียบวิธีของวงจรรองชาวิสกี-โกเลย์หนึ่งมิติ [15] โดยการคำนวณค่าสัมประสิทธิ์ในการฟิตชุดของกลุ่มข้อมูลไว้เพียงครั้งเดียว จากนั้นจึงจัดรูปสมการให้สามารถหาค่าเอาต์พุตได้ในลักษณะผลรวมเชิงเส้น (Linear combination) จากหลักการนี้ได้นำประยุกต์ใช้ในวงจรรองชาวิสกี-โกเลย์สองมิติเช่นกัน [14]

ขั้นตอนในการหาค่า $p_{i,j}(0,0)$ ให้อยู่ในรูปของผลรวมเชิงเส้นนั้น เราจะเริ่มจากกำหนด $\vec{a}_{i,j}$ เป็นเวกเตอร์ที่ประกอบไปด้วยสัมประสิทธิ์ $a_{i,j}(s,t)$ ทุกตัวของฟังก์ชันพหุนาม $p_{i,j}$ ดังนิยามไว้ในสมการที่ (10-2) จากนั้นเราจะทำการจัดเรียงค่าสัมประสิทธิ์ $a_{i,j}(s,t)$ ให้อยู่ในรูปของเวกเตอร์โดยการเรียงลำดับอันดับตามค่าดัชนี r โดย $r \in [1, \dots, (S+1)(T+1)]$ ตามฟังก์ชันดัชนี

$$s(r) = \lfloor (r-1)/(T+1) \rfloor$$

$$t(r) = (r-1) \bmod (T+1)$$

โดย $\lfloor \cdot \rfloor$ เป็นฟังก์ชัน floor และ $\bmod(\cdot)$ เป็นฟังก์ชัน modulo ดังนั้นเราสามารถเขียน $\vec{a}_{i,j}$ ได้ในรูป

$$\vec{a}_{i,j} = (a(s(r), t(r)) : r = 1, \dots, (S+1)(T+1))^T \quad (10-4)$$

ส่วน

$$m(q) = \lfloor (q-1)(2l-1) - l + 1 \rfloor$$

$$n(q) = (q-1) \bmod (2l-1) - l + 1$$

เป็นฟังก์ชันดัชนีของข้อมูล $f(i+m, j+n)$ ในหน้าต่าง $D_{i,j}$ ซึ่งสามารถเขียนให้อยู่ในรูปของเวกเตอร์ฟอร์ม

$$\vec{f}_{i,j} = (f(i+m(q), j+n(q)) : q = 1, \dots, (2l-1)^2)^T \quad (10-5)$$

จากนิยามของ $\vec{a}_{i,j}$ และ $\vec{f}_{i,j}$ ในข้างต้นเราสามารถสร้างเมตริกซ์ A ได้จาก

$$A_{qr} = m(q)^{s(r)} n(q)^{t(r)} \quad (10-6)$$

ในขั้นตอนนี้เราสามารถสร้างฟังก์ชันพหุนามกำลังสองน้อยสุดที่ได้นิยามไว้ในสมการที่ (10-3) ขึ้นใหม่ได้ในรูป

$$\varepsilon_{i,j} = (A\vec{a}_{i,j} - \vec{f}_{i,j})^T W (A\vec{a}_{i,j} - \vec{f}_{i,j}) \quad (10-7)$$

โดยที่ผลเฉลยทั่วไปของ $\vec{a}_{i,j}$ ที่ถูกกำหนดด้วยสมการความผิดพลาดกำลังสองน้อยสุดสามารถแก้สมการหาค่าได้คือ

$$\vec{a}_{i,j} = (A^T W A)^{-1} (A^T W \vec{f}_{i,j}) \quad (10-8)$$

โดยค่าสัมประสิทธิ์ ณ ตำแหน่งศูนย์กลางของฟังก์ชันพหุนาม $p_{i,j}(0,0)$ จะมีค่าเท่ากับ $a_{i,j}(0,0)$ ซึ่งก็คือค่าสัมประสิทธิ์ตัวแรกของ $\vec{a}_{i,j}$ ในสมการที่ (10-8) และเราสามารถลดรูปผลเฉลยของวงจรกรองให้อยู่ในรูปของผลรวมเชิงเส้นได้เป็น

$$p_{i,j}(0,0) = \sum_{q=1}^{(2l-1)^2} \alpha_q f(i+m(q), j+n(q)) \quad (10-9)$$

โดยที่

$$\alpha_q = \left\{ (A^T W A)^{-1} (A^T W \vec{e}_q) \right\}_1 \quad (10-10)$$

เครื่องหมาย $\vec{e}_q$ ในสมการที่ (10-10) คือเวกเตอร์หนึ่งหน่วยซึ่งมีค่าเท่ากับหนึ่งตำแหน่ง q th ส่วน $\{\bullet\}_1$ คือสัญลักษณ์ที่ระบุว่าป็นค่าตัวแรกของเวกเตอร์ผลเฉลย $\vec{a}_{i,j}$ จากสมการที่ (10-9) แสดงถึงค่าสัมประสิทธิ์ที่มีอยู่ของ α_q ซึ่งสามารถจะช่วยให้เราสามารถคำนวณค่าของการประมาณฟังก์ชันพหุนามกำลังสองน้อยสุดบนกลุ่มของข้อมูล โดยอาศัยการคำนวณหาผลรวมเชิงเส้น (Linear combination) ของกลุ่มข้อมูล $f(i+m, j+n)$ กับสัมประสิทธิ์ $\alpha_1, \dots, \alpha_{2M+1}$ เท่านั้น โดยค่าสัมประสิทธิ์ α_q จะขึ้นอยู่กับค่าเมตริกซ์ A และค่าเมตริกซ์ A ที่สร้างขึ้นนั้นจะเป็นค่าคงที่สำหรับทุก ๆ ตำแหน่ง (i, j) จึงทำให้เราสามารถหาค่า α_q เพียงครั้งเดียวสำหรับการสร้างวงจรกรองซาวีตกี-โกเลย์นี้ ส่งผลให้กระบวนการดังกล่าวมีความรวดเร็วและมีความซับซ้อนน้อยลง

10.3 วงจรกรองผสมระหว่างวงจรกรองซาวีตกี-โกเลย์และวงจรกรองมัธยฐาน

วงจรกรองผสมระหว่างวงจรกรองซาวีตกี-โกเลย์และวงจรกรองมัธยฐานเป็นวงจรที่พัฒนาขึ้นสำหรับลดทอนสัญญาณรบกวนแบบ speckle บนภาพ SAR (Synthetic aperture radar) วงจรกรองผสมดังกล่าวเป็นวงจรกรองสองมิติที่พัฒนาขึ้นจากวงจรกรองหนึ่งมิติ FIR-median hybrid

(FMH) filter ซึ่งเป็นวงจรกรองที่ผสมระหว่างวงจรกรอง Finite Impulse Response (FIR) และวงจรกรองมัธยฐาน วงจรกรอง FMH เป็นวงจรกรองที่พัฒนาเริ่มแรกเพื่อจัดสัญญาณรบกวนแบบ Gaussian ในสัญญาณทางเวลาหนึ่งมิติ [16-17] เราเรียกววงจรกรองผสมระหว่างวงจรกรองซาวิตกี-โกเลย์และวงจรกรองมัธยฐานว่า วงจรกรองผสมซาวิตกี-โกเลย์และมัธยฐาน (Savitzky-Golay Median Hybrid (SGMH) filter [18]

10.3.1 วงจรกรองผสมซาวิตกี-โกเลย์และมัธยฐาน

วงจรกรองผสมซาวิตกี-โกเลย์และมัธยฐาน (SGMH) ประกอบวงจรกรองมัธยฐานหนึ่งตัว และกลุ่มของวงจรกรองซาวิตกี-โกเลย์สองมิติ วงจรกรองซาวิตกี-โกเลย์จะรับสัญญาณเข้าซึ่งเป็นเซตย่อยของสัญญาณเข้าของวงจรกรอง SGMH (ดังตัวอย่างกำหนดโดยสมการ (10-12)) และวงจรกรองมัธยฐานจะรับสัญญาณเข้าเป็นผลลัพธ์ที่คำนวณได้จากวงจรกรองซาวิตกี-โกเลย์ทุกตัวในกลุ่ม ผลลัพธ์ที่ได้จากวงจรกรองมัธยฐานซึ่งก็คือค่ามัธยฐานของผลลัพธ์ของวงจรกรองซาวิตกี-โกเลย์ทุกตัวจะถูกกำหนดให้เป็นผลลัพธ์ของวงจรกรอง SGMH (ดังสมการ (10-14)) การทำงานของวงจรกรอง SGMH ที่ใช้ลดทอนสัญญาณรบกวน speckle บนภาพ SAR สามารถแสดงเป็นขบวนการดังต่อไปนี้

พิจารณาภาพขนาด $M \times N$ พิกเซลซึ่งแสดงได้ในรูปของอาร์เรย์ของกลุ่มข้อมูลสองมิติ $f(i, j)$, เมื่อ $i \in [1, \dots, M]$ และ $j \in [1, \dots, N]$. กำหนดให้หน้าต่างของชุดข้อมูลที่เราสนใจมีขนาด $(2L+1) \times (2L+1)$ โดยมีตำแหน่งศูนย์กลางอยู่ที่ (i, j) ดังสมการ

$$D_{i,j} = \{f(i+m, j+n) : -L \leq m \leq L, -L \leq n \leq L\} \quad (10-11)$$

กำหนดชุดหน้าต่างประกอบด้วยหน้าต่างย่อย $L+1$ หน้าต่าง: $w_1(i, j), \dots, w_l(i, j), \dots, w_{L+1}(i, j)$; โดยที่

$$w_l(i, j) = \{f(i+m, j+n) : -l+1 \leq m \leq l-1, -l+1 \leq n \leq l-1\} \quad (10-12)$$

สำหรับแต่ละหน้าต่างย่อย กำหนดฟังก์ชันพหุนามสองมิติ $p_l(m, n)$ ดังสมการ

$$p_l(m, n) = \sum_{s=0}^S \sum_{t=0}^T a_l(s, t) m^s n^t \quad (10-13)$$

โดย m และ n เป็นเลขจำนวนเต็มดังที่อ้างไว้ในสมการที่ (10-11), S และ T เป็นอันดับสูงสุดของ m และ n ตามลำดับ ค่าสัมประสิทธิ์ $a_l(s, t)$ สามารถหาได้ด้วยการฟิตฟังก์ชันพหุนาม p_l แบบกำลังสองน้อยสุด (Least squares) ไปยังข้อมูล f แต่ละพิกเซลในหน้าต่าง $w_l(i, j)$ ผลลัพธ์ของวงจรกรอง SGMH ณ ตำแหน่ง (i, j) จะมีค่าเท่ากับ

$$y_{i,j} = \text{median}\{p_0(0,0), \dots, p_{L+1}(0,0)\} \quad (10-14)$$

ในการคำนวณค่า $p_l(0,0)$ เราสามารถใช้ระเบียบวิธีของวงจรรองขาวิสตี้-โกเลย์สองมิติ ดังอธิบายในหัวข้อ 10.2

10.3.2 ขั้นตอนการประเมินประสิทธิภาพของวงจรรอง

เพื่อทดสอบประสิทธิภาพของวงจรรอง SGMH ในการลดทอนสัญญาณรบกวน speckle บนภาพ SAR งานวิจัยนี้ได้ศึกษาผลลัพธ์ในด้านการลดทอนสัญญาณรบกวนและการคงสภาพเส้นขอบเมื่อนำไปตรวจจับเส้นขอบภาพ ของวงจรรอง SGMH กับภาพทดสอบซึ่งถูกรบกวนด้วย สัญญาณรบกวนที่มีแบบจำลองเช่นเดียวกับสัญญาณรบกวน Speckle และภาพ SAR จากฐานข้อมูล MSTAR งานวิจัยนี้ได้เปรียบเทียบผลที่ได้วงจรรอง SGMH กับวงจรรองที่เป็นที่นิยมใช้กัน สำหรับลดทอน speckle บนภาพ SAR ซึ่งประกอบไปด้วยวงจรรองมัชฌาน วงจรรองแบบ Lee และวงจรรองแบบ Frost

วงจรรองลดทอนสัญญาณรบกวน Speckle แบบมัชฌาน

วงจรรองลดทอนสัญญาณรบกวน Speckle แบบมัชฌานอาศัยหลักการประมาณค่าสถิติของค่ามัชฌาน โดยมีรูปแบบสมการของวงจรรองดังนี้

$$y_{i,j} = \text{median}\{f(i, j), \text{for all } (i, j) \text{ in } D\} \quad (10-15)$$

เมื่อ D เป็นเซตของค่าดัชนี m และ n ในหน้าต่างของวงจรรอง

วงจรรองลดทอนสัญญาณรบกวน Speckle แบบ Lee

วงจรรองลดทอนสัญญาณรบกวน Speckle แบบ Lee [19] อาศัยหลักการประมาณค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNR) จากค่าเชิงสถิติในย่านที่เราสนใจด้วยอัตราส่วนของค่าความแปรปรวนและค่าเฉลี่ยเพื่อแยกความแตกต่างระหว่างหมวดของโครงสร้างภาพ โดยมีรูปแบบสมการของวงจรรองดังนี้

$$y_{i,j} = \mu + \frac{\sigma_f^2}{\sigma_f^2 + \mu^2 \sigma_n^2} (f(i, j) - \mu) \quad (10-16)$$

เมื่อ $y_{i,j}$ เป็นพิกเซลเอาต์พุตใหม่ที่ได้จากการประมาณค่ากลุ่มข้อมูล, $f(i, j)$ เป็นค่าเดิมหรือเป็นค่า ณ ตำแหน่งศูนย์กลาง μ และ σ_f เป็นค่าเฉลี่ยและค่าเบี่ยงเบนมาตรฐานของพิกเซลข้อมูลในย่านที่เรากำลังสนใจ และท้ายสุด σ_n เป็นค่าเบี่ยงเบนมาตรฐานของสัญญาณรบกวน speckle

วงจรกรองลดทอนสัญญาณรบกวน Speckle แบบ Frost

วงจรกรองลดทอนสัญญาณรบกวน Speckle แบบ Frost [20] เป็นวงจรกรองที่พัฒนาขึ้นเพื่อลดทอนสัญญาณแบบควมในภาพที่ได้จากเรดาร์ และมีรูปแบบสมการดังนี้

$$y_{i,j} = \frac{\sum_D \alpha(m,n) f(i+m, j+n)}{\sum_D \alpha(m,n)} \quad (10-17)$$

เมื่อ D เป็นเซตของค่าดัชนี m และ n ในหน้าต่างของวงจรกรอง และ

$$\alpha(m,n) = \exp\left(-\frac{\beta \sigma_f^2}{\mu^2} (m^2 + n^2)^{1/2}\right) \quad (10-18)$$

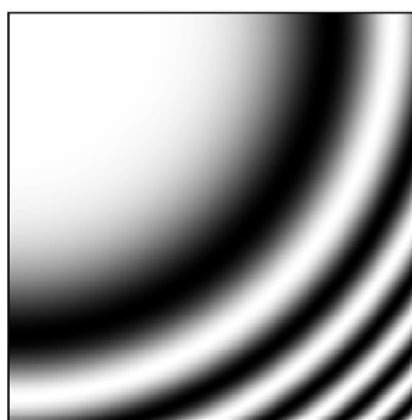
เมื่อ β เป็นอัตราหนักที่กำหนดความเร็วของค่า α ในการเข้าสู่ศูนย์

10.3.3 การทดสอบประสิทธิภาพของวงจรกรองกับภาพสังเคราะห์ทางคณิตศาสตร์

ภาพแรกที่ใช้ในการทดสอบประสิทธิภาพในการลดทอนสัญญาณรบกวนและการคงสภาพของของเส้นขอบเป็นภาพที่สร้างขึ้นจากสมการคณิตศาสตร์

$$f(i, j) = \cos(0.5 \times 10^{-8} (i^4 + 2i^2 j^2 + j^4)) \quad (10-19)$$

โยมีลักษณะดังแสดงในรูปที่ 10.1

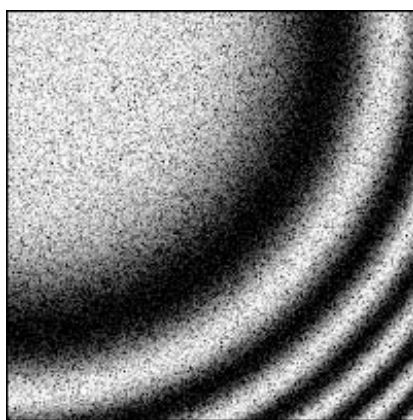


รูปที่ 10.1 ภาพที่ใช้ในการทดสอบ

ในการทดสอบจะนำภาพทดสอบดังกล่าวมาใส่สัญญาณรบกวนโดยคูณกับสัญญาณรบกวนเรย์ลีซึ่ง
เป็นแบบจำลองของสัญญาณรบกวน Speckle ดังสมการ

$$E[f_{image}] = E[f] \cdot E[N] \quad (10-20)$$

เมื่อ $E[\cdot]$ เป็นค่าคาดหวังหรือค่าเฉลี่ยเชิงสถิติ โดยสัญญาณรบกวนเรย์ลี N ที่เรารู้ค่ากับ
สัญญาณจริง f เป็นอิสระต่อกัน ดังนั้นเราจึงให้ค่าเฉลี่ยของสัญญาณรบกวนมีค่าเท่ากับหนึ่ง เพื่อ
ให้ค่าเฉลี่ยของภาพที่จะนำมาทดสอบมีค่าคงเดิม ส่งผลให้ค่าความแปรปรวนซึ่งมีค่าเท่ากับ
 $\sigma^2 = 0.2732$ ภาพทดสอบที่ถูกรบกวนด้วยสัญญาณรบกวนเรย์ลีแบบคูณจะมีลักษณะดังแสดงใน
รูป 10.2



รูปที่ 10.2 ภาพทดสอบที่ถูกรบกวนด้วยสัญญาณรบกวนเรย์ลีแบบคูณ

ในที่นี้ เราเลือกที่จะวัดประสิทธิภาพของวงจรกรองโดยใช้ค่าความผิดพลาดกำลังสองเฉลี่ย
แบบนอร์มัลไลซ์ (Normalized Mean Square Error: NMSE) เทียบกับขนาดหน้าต่างเป็นตัวแทน
ในการตัดสินใจหาค่าได้จากสมการ

$$NMSE = \frac{\sum_{\Omega} (f_{filtered} - f)^2}{\sum_{\Omega} (f_{image} - f)^2} \quad (10-21)$$

เมื่อ Ω เป็นกลุ่มของพิกเซลข้อมูลในภาพ ส่วน f_{image} เป็นค่าความเข้มของพิกเซลในภาพทดสอบ
ที่ถูกรบกวนด้วยสัญญาณรบกวน Speckle $f_{filtered}$ เป็นค่าความเข้มของพิกเซลในภาพที่ผ่านการ
กรองมาแล้ว และ f เป็นค่าความเข้มของพิกเซลข้อมูลในภาพทดสอบโดยไม่ถูกรบกวนด้วย
สัญญาณรบกวนใด ๆ

ในการวิเคราะห์การใช้งานร่วมกับวงจรตรวจจับเส้นขอบแบบเกรเดียนต์ เนื่องจากการหา
เส้นขอบในภาพอัตราขาวดำมักจะไม่มีประสิทธิภาพ ถ้าภาพดังกล่าวไม่เป็นไปตามสมมุติฐานที่
ตั้งไว้ เช่น สำหรับวงจรตรวจจับเส้นขอบที่ใช้หลักการของเกรเดียนต์ ซึ่งกำหนดจุดที่มีการเปลี่ยน

แปลงความเข้มสูงสุดเป็นตำแหน่งเส้นขอบ เมื่อนำมาใช้กับภาพที่มีสัญญาณรบกวนสูง ซึ่งไม่ได้มีขนาดความเข้มสูงสุดเฉพาะที่ตำแหน่งเส้นขอบตามที่ได้อ้างสมมุติฐานไว้ จึงได้เส้นขอบที่ไม่ต้องการเกิดขึ้นมากมาย ดังนั้นการใช้วงจรกรองที่เหมาะสมเพื่อนำมาลดทอนสัญญาณรบกวนในภาพ SAR และปรับปรุงภาพให้เข้าใกล้สมมุติฐานจึงมีความสำคัญ ดังนั้นในการวิเคราะห์การใช้งานร่วมกับวงจรตรวจจับเส้นขอบแบบเกรเดียนต์จะพิจารณาลักษณะเส้นขอบที่ได้หลังผ่านการกรองด้วยวงจรกรองต่าง ๆ เปรียบเทียบกัน

วงจรตรวจจับเส้นขอบที่เลือกใช้นั้นเป็นวงจรตรวจจับเส้นขอบของแคนนี่ [21] สาเหตุที่เลือกใช้วงจรตรวจจับเส้นขอบดังกล่าว เนื่องจากเป็นวงจรตรวจจับเส้นขอบที่มีโครงสร้างง่าย ใช้เวลาในการคำนวณต่ำ ได้รับการยอมรับว่ามีประสิทธิภาพในการตรวจจับเส้นขอบได้ดี และเป็นวงจรตรวจจับเส้นขอบที่ใช้เทคนิค Pre-emptive

ในการทดสอบการทำงานของวงจรกรองทั้งสี่นี้ วงจรกรอง SGMH วงจรกรองมัธยฐาน วงจรกรองแบบ Lee และวงจรกรองแบบ Frost จะถูกเขียนด้วยภาษา MATLAB และจำลองบนเครื่องคอมพิวเตอร์ Pentium 4 Processor, 1.6 GHz, RD-RAM 512 MB ในการประเมินประสิทธิภาพของแต่ละวงจรกรองนั้น การตั้งค่าพารามิเตอร์ต่าง ๆ เช่น ค่า-คงที่ในการปรับตัว อันดับของวงจรกรอง หรือ ขนาดหน้าต่างของวงจรกรองนั้นมีความสำคัญยิ่ง เพื่อให้วงจรกรองนั้น ๆ สามารถทำงานได้อย่างมีประสิทธิภาพ ดังนั้นจึงจะได้กล่าวถึงการตั้งค่า พารามิเตอร์ดังกล่าวของแต่ละวงจรกรองดังนี้

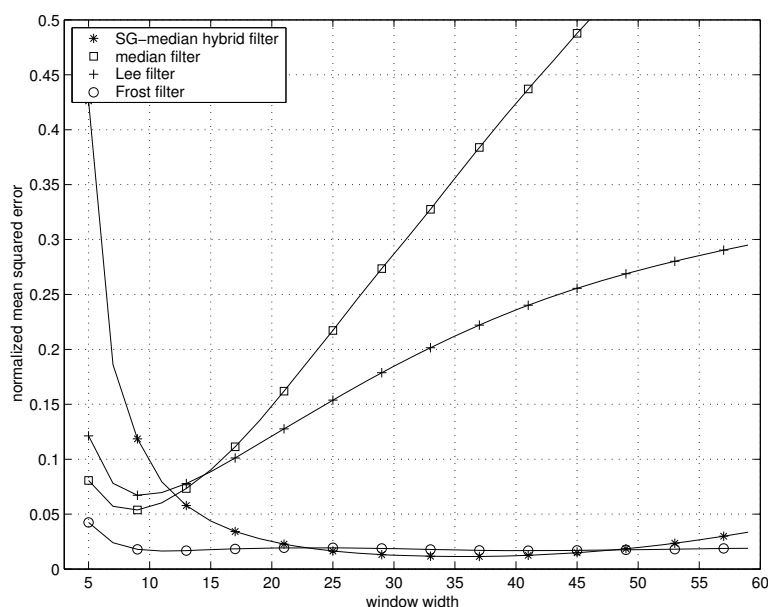
- วงจรกรอง SGMH จะใช้วงจรกรองซาวีตกี-โกเลย์สองมิติ มีค่าอันดับของฟังก์ชันพหุนาม S และ T เท่ากับ 2 ทั้งสองค่าเนื่องจากจะทำให้ได้ลักษณะการฟิตพื้นผิวแบบมีความโค้งและความชัน ซึ่งสามารถติดตามผลของสัญญาณที่ต้องการได้ดี จากการทดลองพบว่าหากใช้อันดับของฟังก์ชันพหุนามที่สูงกว่านี้ ไม่ได้ช่วยทำให้การกรองดีขึ้นเนื่องจาก จะมีความยืดหยุ่นในการฟิตกลุ่มข้อมูลสูงเกินไปจนเป็นเหตุให้ฟังก์ชันพหุนามวิ่งติดตามผลของสัญญาณรบกวนได้เร็วขึ้น

- วงจรกรองลดทอนสัญญาณรบกวน Speckle แบบ Lee ค่าสัมประสิทธิ์ในการปรับตัวของวงจรกรองนั้นขึ้นอยู่กับ SNR ของข้อมูลในย่านที่เราสนใจซึ่งสามารถคำนวณได้โดยตรง โดยเรากำหนดให้ σ_n^2 เท่ากับ 0.2732 ซึ่งเป็นค่าเบี่ยงเบนมาตรฐานของสัญญาณรบกวนเรย์ลี N

- วงจรกรองลดทอนสัญญาณรบกวน Speckle แบบ Frost กำหนดให้ β เท่ากับ 0.9 ซึ่งเป็นค่าที่ให้ค่า NMSE ต่ำสุด [18]

รูปที่ 10.3 แสดงค่า NMSE ของวงจรกรอง SGMH, วงจรกรองมัธยฐาน วงจรกรองแบบ Lee และวงจรกรองแบบ Frost ทดสอบกับภาพในรูปที่ 10.1 ในการทดสอบนี้ เส้นกราฟแต่ละเส้นได้จากการเฉลี่ยของ 10 การทดลอง จากรูปที่ 10.3 เมื่อขนาดหน้าต่างของวงจรกรองเพิ่มขึ้น ค่า NMSE ของทุกวงจรกรองจะมีลักษณะลดลงจนถึงระดับหนึ่งแล้วจะเพิ่มขึ้นอีกครั้ง การเพิ่มขึ้นของค่า NMSE นี้เกิดเนื่องจากการให้น้ำหนักกับพิกเซลอื่นมากเกินไปจึงทำให้เกิดไบแอสขึ้น ค่าที่ได้จึง

ต่างจากค่าก่อนใส่สัญญาณรบกวนมากและยังแตกต่างกันยิ่งขึ้นเรื่อย ๆ เมื่อปริมาณพิกเซลของข้อมูลที่นำมาพิจารณาเพิ่มมากขึ้น สำหรับวงจรกรอง SGMH เราพบว่า ค่า NMSE มีค่าต่ำสุดเท่ากับ 0.0114 เมื่อเปรียบเทียบกับค่า NMSE ต่ำสุดของวงจรกรองมัธยฐานที่ 0.0538 ค่า NMSE ต่ำสุดของวงจรกรองแบบ Lee ที่ 0.0672 และค่า NMSE ต่ำสุดของวงจรกรองแบบ Frost ที่ 0.0164 เมื่อพิจารณาเปรียบเทียบกราฟในรูปที่ 10.3 โดยรวม เราพบว่าค่า NMSE ของวงจรกรองมัธยฐานและของวงจรกรองแบบ Lee มีค่าต่ำสุดที่ หน้าต่างขนาด 9 และค่า NMSE จะเพิ่มขึ้นอย่างรวดเร็วเมื่อหน้าต่างเพิ่มขึ้น สำหรับวงจรกรองแบบ Frost มีค่าต่ำสุดที่หน้าต่างขนาด 11 และ NMSE มีค่าคงที่หลังจากนั้น แนวโน้มเดียวกันนี้จะพบในวงจรกรอง SGMH เช่นกัน โดยค่า NMSE จะต่ำสุดที่หน้าต่างขนาด 35 และอยู่เกือบคงที่หลังจากนั้น ปรากฏการณ์ที่ ค่า NMSE ของวงจรกรอง SGMH และวงจรกรองแบบ Lee มีค่าต่ำอยู่เป็นช่วงกว้างแสดงให้เห็นว่า วงจรกรองทั้งสองมีประสิทธิภาพเปลี่ยนแปลงตามต่อขนาดหน้าต่างน้อย อย่างไรก็ตาม จากการทดลอง เราพบว่า ประสิทธิภาพของวงจรกรองแบบ Frost จะขึ้นอยู่กับค่า β เป็นอย่างมาก

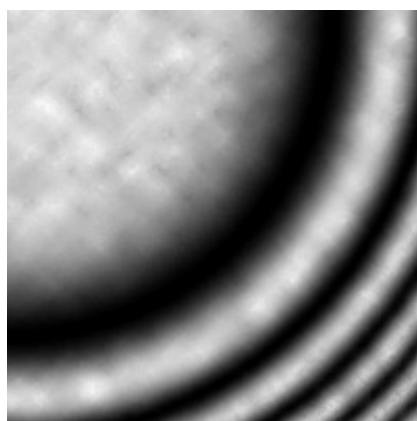


รูปที่ 10.3 ค่า NMSE ของวงจรกรองแต่ละชนิดตามขนาดหน้าต่าง ในการทดสอบนี้ เส้นกราฟแต่ละเส้นได้จากการเฉลี่ยของ 10 การทดลอง

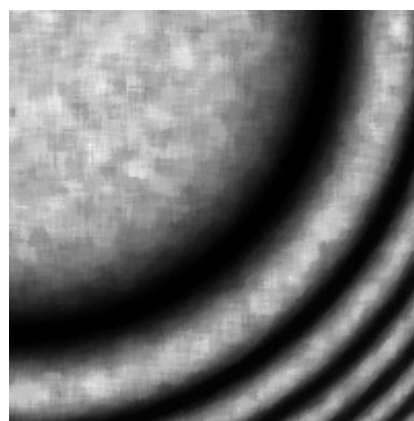
นอกจากนี้ต้องหมายเหตุไว้ว่าการให้ค่า NMSE สูงหรือต่ำไม่ได้บ่งบอกว่า เมื่อคำนวณเส้นขอบจากภาพหลังผ่านการกรองด้วยวงจรกรองจะได้เส้นขอบที่เรียบหรือต่อเนื่อง เพราะค่าดังกล่าวไม่ได้แสดงถึงความราบเรียบของข้อมูล ตัวอย่างของผลลัพธ์ของการทดสอบเมื่อเลือกใช้ขนาดหน้าต่างที่ทำให้ค่า NMSE ต่ำสุดได้แสดงไว้ในรูปที่ 10.4 และผลลัพธ์ของเส้นขอบในรูปที่ 10.5

รูปที่ 10.4 แสดงผลลัพธ์ของวงจรกรองแบบต่างๆ สำหรับขนาดหน้าต่างของแต่ละวงจรกรองที่ค่า NMSE ต่ำสุด (Minimum NMSE) รูปดังกล่าวแสดงให้เห็นว่า ผลลัพธ์จากวงจรกรอง SGMH และ

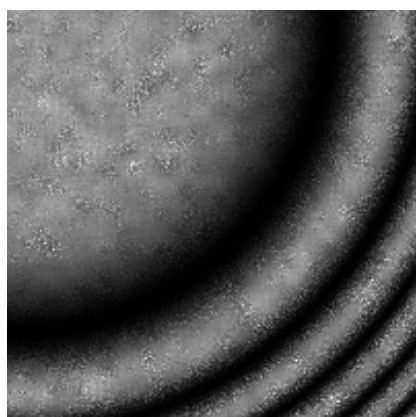
วงจรรองแบบ Frost มีลักษณะใกล้เคียงกัน สำหรับผลลัพธ์ของวงจรรองแบบ Lee เราจะเห็นว่ายังมีสัญญาณรบกวน speckle กระจายอยู่ทั่วไป สืบเนื่องจากบริเวณส่วนที่ควรจะเป็นสีขาวในบริเวณย่านความถี่สูงกลับเป็นสีเทา



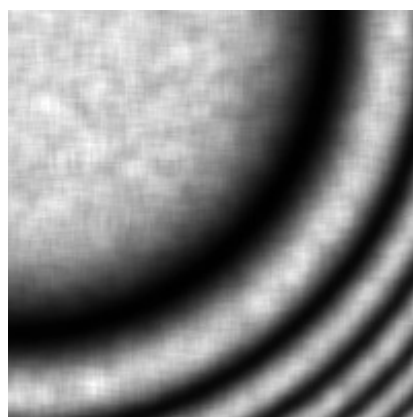
(ก)



(ข)



(ค)



(ง)

รูปที่ 10.4 ผลลัพธ์ของวงจรรองแต่ละชนิดที่ NMSE ที่ขนาดหน้าต่าง NMSE ต่ำสุด

(ก) วงจรรองแบบ SGMH ขนาดหน้าต่าง 35X35 พิกเซล

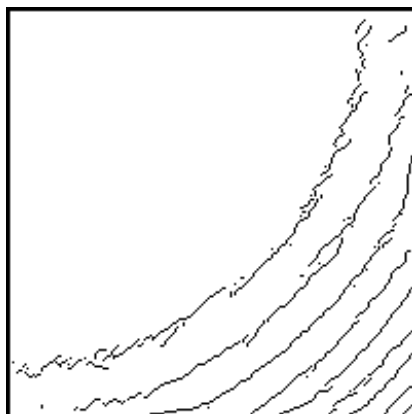
(ข) วงจรรองมัธยฐานขนาดหน้าต่าง 9x9 พิกเซล

(ค) วงจรรองแบบ Lee ขนาดหน้าต่าง 9X9 พิกเซล

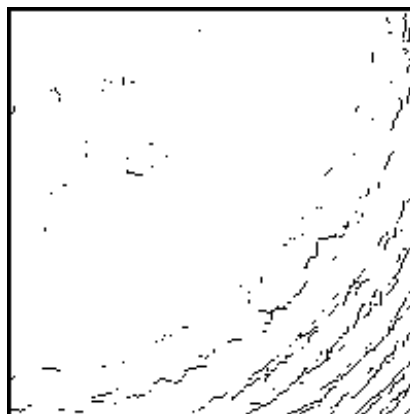
(ง) วงจรรองแบบ Frost ขนาดหน้าต่าง 11x11 พิกเซล

เพื่อเปรียบเทียบประสิทธิภาพของการคงสภาพเส้นขอบ ผลลัพธ์จากวงจรรองทั้งสี่จะถูกผ่านไปยังวงจรตรวจจับเส้นขอบของ Canny และผลที่ได้รับแสดงในรูปที่ 10.5 สังเกตว่าผลลัพธ์ที่

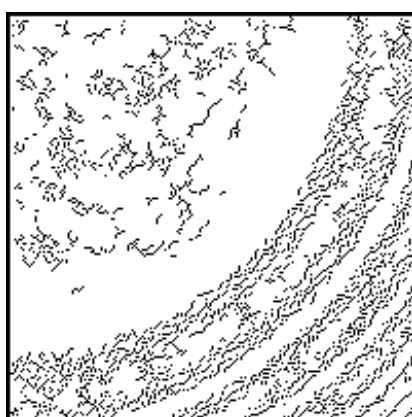
ได้จากวงจรรอง SGMH และวงจรรองแบบ Frost จะมีลักษณะใกล้เคียงกัน แต่เส้นขอบที่ได้จากกรณีของวงจรรอง SGMH จะเรียบและต่อเนื่องกว่าโดยมีสัญญาณรบกวนน้อยกว่า



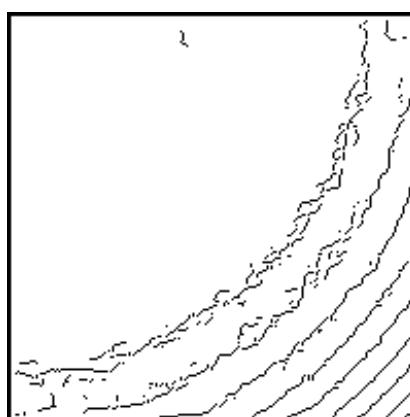
(ก)



(ข)



(ค)



(ง)

รูปที่ 10.5 เส้นขอบภาพหลังผ่านการกรองด้วยวงจรรองแต่ละชนิดในรูปที่ 10.4

ที่ให้ค่า NMSE ต่ำสุดด้วยวงจรตรวจจับเส้นขอบของแคนนี่

(ก) วงจรรองแบบ SGMH ขนาดหน้าต่าง 35X35 พิกเซล

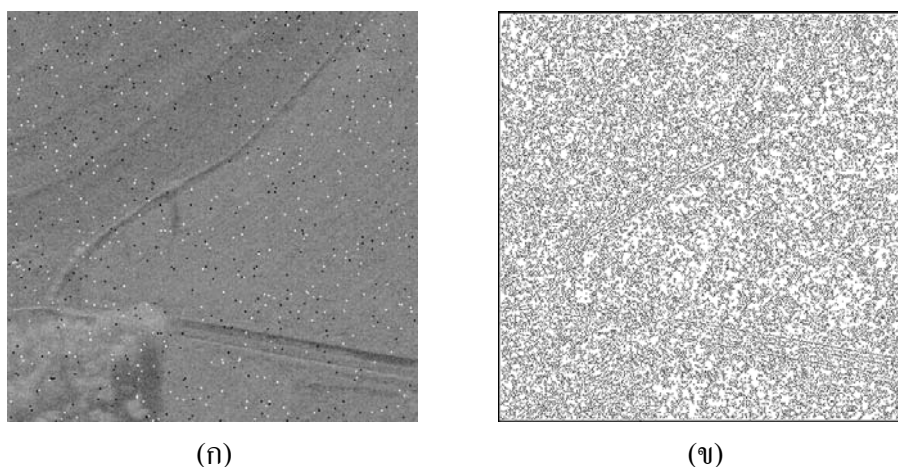
(ข) วงจรรองมัธยฐานขนาดหน้าต่าง 9x9 พิกเซล

(ค) วงจรรองแบบ Lee ขนาดหน้าต่าง 9X9 พิกเซล

(ง) วงจรรองแบบ Frost ขนาดหน้าต่าง 11x11 พิกเซล

10.3.4 การทดสอบกับภาพ SAR

ในการทดสอบการใช้งานของวงจรกรองแบบต่างๆกับภาพ SAR ของรูปท้องทุ่งจากฐานข้อมูล MSTAR นั้น จะทดสอบกับภาพในรูปที่ 10.6 (ก) ซึ่งประกอบด้วย 512X512 พิกเซล ในรูปที่ 10.6 (ข) แสดงผลกระทบของสัญญาณรบกวน speckle ในภาพ SAR เมื่อตรวจจับเส้นขอบภาพโดยไม่ผ่านการลดทอนสัญญาณรบกวนด้วยวงจรกรองใด ๆ จะเห็นได้ว่ามีเส้นขอบของสัญญาณรบกวนเกิดขึ้นอย่างมากมายและไม่สามารถแยกส่วนของเส้นขอบที่เราสนใจออกมาได้



รูปที่ 10.6 ตัวอย่างภาพ SAR ภาพเส้นขอบ

การลดทอนสัญญาณรบกวนในภาพ SAR เราต้องการให้ภาพราบเรียบขึ้นเพื่อให้การคำนวณหาเส้นขอบภาพไม่เกิดตำแหน่งที่ไม่ต้องการขึ้นมา ซึ่งโดยปกติจะเป็นผลกระทบของสัญญาณรบกวน เนื่องจากโดยปกติสัจฐานของภาพ SAR ที่เรานำมากรองสัญญาณรบกวนนั้นจะมีพื้นผิวของเนื้อเยื่อซึ่งมีลักษณะกว้าง และมีอัตราการเปลี่ยนแปลงของความชันต่ำ ดังนั้นการใช้ขนาดหน้าต่างวงจรกรองที่เล็กนั้นจะทำให้การลดทอนสัญญาณรบกวนไม่ดีนัก ในที่นี้จึงจะทดสอบประสิทธิภาพของแต่ละวงจรกรองที่ขนาดหน้าต่างขนาดกลางที่ 15x15 พิกเซล เรื่อยไปจนถึงขนาดหน้าต่างขนาดใหญ่ที่ 59x59 พิกเซล สำหรับค่าพารามิเตอร์วงจรกรอง SGMH จะใช้วงจรกรองซาวีตกี-โกเลย์สองมิติ มีค่าอันดับของฟังก์ชันพหุนาม S และ T เท่ากับ 2 สำหรับวงจรกรองลดทอนสัญญาณรบกวน Speckle แบบ Lee กำหนดให้ σ_n^2 เท่ากับ 0.2732 และสำหรับวงจรกรองลดทอนสัญญาณรบกวน Speckle แบบ Frost กำหนดให้ β เท่ากับ 0.3 ผลลัพธ์จากวงจรกรองทั้งสี่จะถูกป้อนให้กับวงจรตรวจจับเส้นขอบของ Canny

รูปที่ 10.7 แสดงภาพ SAR ในรูปที่ 10.6 (ก) หลังผ่านการกรองด้วยวงจกรองแต่ละชนิด ขนาดของหน้าต่างที่ใช้ได้ถูกเลือกให้ได้ผลลัพธ์ดีที่สุดสำหรับแต่ละวงจกรอง



(ก)



(ข)



(ค)



(ง)

รูปที่ 10.7 ผลลัพธ์ของวงจกรองทั้งสี่สำหรับภาพ SAR

(ก) วงจกรองแบบ SGMH ขนาดหน้าต่าง 37X37 พิกเซล

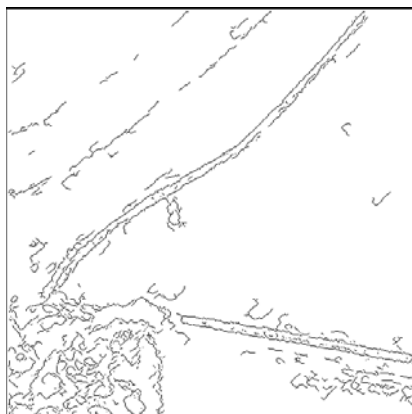
(ข) วงจกรองมัธยฐานขนาดหน้าต่าง 9x9 พิกเซล

(ค) วงจกรองแบบ Lee ขนาดหน้าต่าง 11X11 พิกเซล

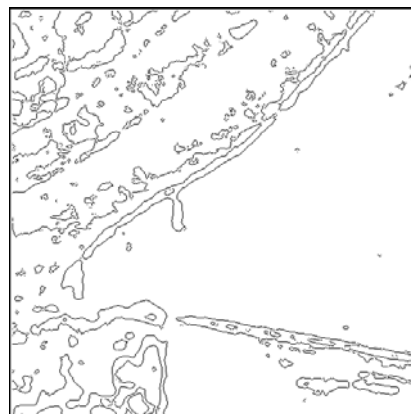
(ง) วงจกรองแบบ Frost ขนาดหน้าต่าง 9x9 พิกเซล

ผลลัพธ์ของวงจกรอง SGMH ดังแสดงในรูปที่ 10.7 (ก) นั้นยังดูไม่แตกต่างกับภาพต้นแบบเท่าไรนักแต่ทว่ามีจุด speckle น้อยลงมาก เส้นขอบของรูปดังกล่าวแสดงในรูปที่ 10.8(ก) สำหรับภาพที่ผ่านการกรองด้วยวงจกรองมัธยฐาน (รูปที่ 10.7 (ข)) มีลักษณะเป็นหมอกฝ้าแต่ภาพที่ได้ก็ดูราบเรียบขึ้น ซึ่งเป็นเทคนิคการกรองโดยทำให้ภาพพร่าลง (Diffusion filtering) เมื่อนำมา

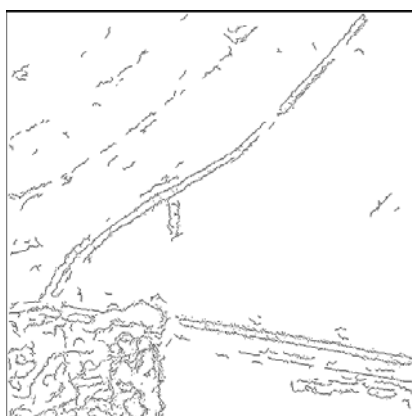
ตรวจจับเส้นขอบภาพได้ผลในรูปที่ 10.8 (ข) เห็นได้ว่ายังปรากฏเส้นขอบของสัญญาณรบกวนอยู่มากและยังไม่สามารถแสดงเส้นขอบที่สนใจได้ ในวงจรกรองมัธยฐาน สำหรับผลลัพธ์ของวงจรกรองแบบ Lee และ Frost จะให้ผลลัพธ์คล้ายกับผลลัพธ์ของวงจรกรอง SGMH เพียงแต่จะมีจุด speckle ชัดกว่า เส้นขอบของผลลัพธ์ของวงจรกรองทั้งสอง ดังแสดงในรูปที่ 10.8(ค) และ 10.8(ง) ก็ปรากฏว่ามีสัญญาณรบกวนมากกว่าและขอบต่อเนื่องน้อยกว่า



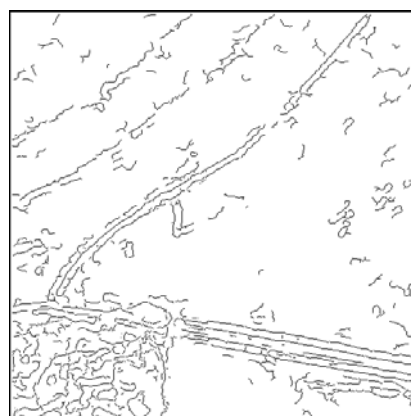
(ก)



(ข)



(ค)



(ง)

รูปที่ 10.8 ภาพเส้นขอบของผลลัพธ์ของวงจรกรองทั้งสี่สำหรับภาพ SAR

(ก) วงจรกรองแบบ SGMH ขนาดหน้าต่าง 37X37 พิกเซล

(ข) วงจรกรองมัธยฐานขนาดหน้าต่าง 9x9 พิกเซล

(ค) วงจรกรองแบบ Lee ขนาดหน้าต่าง 11X11 พิกเซล

(ง) วงจรกรองแบบ Frost ขนาดหน้าต่าง 9x9 พิกเซล

10.3.5 สรุปผลการทดลอง

การทดสอบเบื้องต้นในหัวข้อ 10.3.3 และ 10.3.4 ชี้ให้เห็นว่า วงจรกรอง SGMH มีประสิทธิภาพสูงกว่าวงจรกรองมัชฌิมาน วงจรกรองแบบ Lee และวงจรกรองแบบ Frost ในแง่ของการลดทอนสัญญาณรบกวน speckle และคงสภาพเส้นขอบในภาพถ่าย SAR ประสิทธิภาพของวงจรกรอง SGMH ที่สูงกว่านี้เกี่ยวเนื่องมาจากการใช้ค่าสถิติทาง rank และความเชื่อมโยงทาง spatial ระหว่างพิกเซลในพื้นที่เดียวกัน แต่อย่างไรก็ดี การคำนวณของวงจรกรอง SGMH จะซับซ้อนและกินเวลามากกว่ามาก

การปรับปรุงวงจรรองขาสติ-โกเล่ย์สำหรับลดทอนสัญญาณรบกวน Speckle บนภาพ อัลตราซาวนด์

ดังที่ได้นำเสนอวงจรรองขาสติ-โกเล่ย์สองมิติมาแล้วในบทที่ 10 จะเห็นได้ว่าวงจรรองขาสติ-โกเล่ย์สองมิติที่ได้มีผู้พัฒนาขึ้นในงานวิจัย เพื่อใช้ในการลดทอนสัญญาณรบกวนในภาพอัลตราซาวนด์นั้นใช้วิธีการกำลังสองน้อยสุด (Least Squares method) ประมาณกลุ่มข้อมูลด้วยฟังก์ชันพหุนาม ดังนั้นในบทนี้จึงมีแนวคิดที่จะพัฒนาวงจรรองขาสติ-โกเล่ย์สองมิติโดยการรวมรูปแบบของฟังก์ชันถ่วงน้ำหนักตามความโค้งของภาพ (Curvature) วงจรรองที่จะนำเสนอนี้คือวงจรรองขาสติ-โกเล่ย์แบบแอนไอโซทรอปิก (Anisotropic Savitzky-Golay : ASG filters) ซึ่งคือวงจรรองขาสติ-โกเล่ย์สองมิติที่ถูกปรับปรุงโดยการเพิ่มกลไกในการหาทิศทางในการถ่วงน้ำหนักโดยยังคงรักษาหลักการกำลังสองน้อยสุดเพื่อให้สัญญาณที่ได้ราบเรียบแบบมีทิศทางตามคุณสมบัติของแอนไอโซทรอปิกในย่านของชุดข้อมูลของภาพที่เรากำลังสนใจ

ดังที่ได้นำเสนอวงจรรองขาสติ-โกเล่ย์สองมิติมาแล้วในหัวข้อที่ 10.2.1 เราจะเห็นได้ว่าวงจรรองขาสติ-โกเล่ย์สองมิติใช้วิธีการกำลังสองน้อยสุด (Least square method) เพื่อประมาณกลุ่มข้อมูลด้วยฟังก์ชันพหุนาม ดังนั้นในงานวิจัยนี้จึงมีแนวคิดที่จะพัฒนาวงจรรองขาสติ-โกเล่ย์สองมิติโดยการรวมรูปแบบของฟังก์ชันถ่วงน้ำหนักตามความโค้งของภาพ (Curvature) วงจรรองที่จะนำเสนอนี้คือวงจรรองขาสติ-โกเล่ย์แบบแอนไอโซทรอปิก (Anisotropic Savitzky-Golay : ASG filters) ซึ่งคือวงจรรองขาสติ-โกเล่ย์สองมิติที่ถูกปรับปรุงโดยการเพิ่มกลไกในการหาทิศทางในการถ่วงน้ำหนักโดยยังคงรักษาหลักการกำลังสองน้อยสุดเพื่อให้สัญญาณที่ได้ราบเรียบแบบมีทิศทางตามคุณสมบัติของแอนไอโซทรอปิกในย่านของชุดข้อมูลของภาพที่เรากำลังสนใจ [22]

11.1 วงจรรองขาสติ-โกเล่ย์แบบแอนไอโซทรอปิก

ระเบียบวิธีของวงจรรองขาสติ-โกเล่ย์แบบแอนไอโซทรอปิกสามารถอธิบายด้วยสองขั้น ตอนหลักดังนี้ ในขั้นตอนแรกเราจะได้นำเสนอการคำนวณหาคุณลักษณะแบบคอนทัวร์ของภาพในย่านที่เราสนใจซึ่งสัมพันธ์กับทิศทางของความโค้งหลัก (Principal curvature direction) ที่ตำแหน่ง (i, j) ด้วย local curvatures โดยหาได้จากค่าเจาะจง (Eigenvalues) ของ Hessian Matrix ซึ่งได้มาจากอนุกรมเทย์เลอร์ (Taylor series) ในการประมาณข้อมูลตำแหน่ง (m, n) ใด ๆ รอบจุด

กึ่งกลางหน้าต่างของวงจรกรองที่ตำแหน่ง (0,0) โดยฟังก์ชัน p ที่จุดดังกล่าวสามารถประมาณได้ดังสมการที่ (11-1)

$$\begin{aligned} \tilde{p}(m,n) = & p(0,0) + [\Delta m \quad \Delta n] \begin{bmatrix} \frac{\partial}{\partial m} p \\ \frac{\partial}{\partial n} p \end{bmatrix} \\ & + \frac{1}{2!} [\Delta m \quad \Delta n] \begin{bmatrix} \frac{\partial^2}{\partial m^2} p & \frac{\partial^2}{\partial m \partial n} p \\ \frac{\partial^2}{\partial n \partial m} p & \frac{\partial^2}{\partial n^2} p \end{bmatrix} \begin{bmatrix} \Delta m \\ \Delta n \end{bmatrix} + O(\Delta m^3, \Delta n^3) \end{aligned} \quad (11-1)$$

โดย $O(\Delta m^3, \Delta n^3)$ เป็นพจน์ที่เหลือซึ่งมีการเข้าใกล้ Δm^3 หรือ Δn^3 ซึ่งสำหรับ Δm หรือ Δn น้อย ๆ เราอาจตัดพจน์เหล่านี้ทิ้งไปโดยค่าประมาณที่ได้ไม่ผิดพลาดมากนัก ส่วนเมตริกซ์ของอนุพันธ์ย่อยอันดับสองจะเป็นตัวกำหนดความโค้งของโครงสร้างภาพซึ่งเป็นเทอมที่เราสนใจ ดังนั้นเราจึงลดรูปให้อยู่ในรูปของ Hessian Matrix ดังสมการ

$$H(i, j) = \begin{bmatrix} f_{uu}(i, j) & f_{uv}(i, j) \\ f_{uv}(i, j) & f_{vv}(i, j) \end{bmatrix} \quad (11-2)$$

โดยที่ f_{uu} เป็นอนุพันธ์ย่อยอันดับที่สองของ f เทียบกับแกน u , f_{uv} เป็นอนุพันธ์ย่อยอันดับที่สองเทียบกับแกน u และ v และ f_{vv} เป็นอนุพันธ์ย่อยอันดับที่สองเทียบกับแกน v ในการคำนวณค่าอนุพันธ์เหล่านี้เราจะประยุกต์หลักการพิทเพนผิวด้วยฟังก์ชันพหุนามกำลังสองน้อยสุดดังที่เราเคยได้กล่าวมาแล้วในหัวข้อวงจรกรองซาวีตกี-โกเลย์สองมิติ ต่างกันที่ในหัวข้อวงจรกรองซาวีตกี-โกเลย์สองมิตินั้นเราจะคำนวณหาค่าสัมประสิทธิ์ $a_{i,j}(0,0)$ แต่ในที่นี้เราจะคำนวณหาค่าสัมประสิทธิ์ $a_{i,j}(2,0)$, $a_{i,j}(1,1)$, $a_{i,j}(0,2)$ มาใช้เป็นค่าในการประมาณ $f_{uu}(i, j)$, $f_{uv}(i, j)$, $f_{vv}(i, j)$ ตามลำดับดังสมการ

$$\begin{aligned} f_{uu}(i, j) &= \frac{\partial^2}{\partial u^2} f(i, j) \approx \frac{\partial^2}{\partial m^2} p_{i,j}(m, n) \Big|_{m=0, n=0} = 2a_{2,0} \\ f_{uv}(i, j) &= \frac{\partial^2}{\partial uv} f(i, j) \approx \frac{\partial^2}{\partial mn} p_{i,j}(m, n) \Big|_{m=0, n=0} = a_{1,1} \\ f_{vv}(i, j) &= \frac{\partial^2}{\partial v^2} f(i, j) \approx \frac{\partial^2}{\partial n^2} p_{i,j}(m, n) \Big|_{m=0, n=0} = 2a_{0,2} \end{aligned} \quad (11-3)$$

โดยฟังก์ชันถ่วงน้ำหนักของทุกตัวมีค่าเท่ากับหนึ่ง

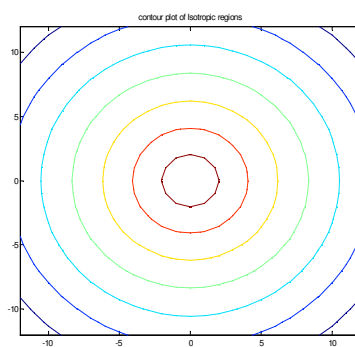
ในขั้นตอนถัดไป เราจะทำการเพิ่มประสิทธิภาพของวงจรกรองซาวีตกี-โกเลย์สองมิติให้สามารถลดทอนสัญญาณรบกวนในแนวทิศทางและโครงสร้างของภาพโดยการสร้างฟังก์ชันถ่วงน้ำหนัก $w_{i,j}$ ซึ่งได้นิยามไว้ในสมการที่ (10-3) ด้วยค่าเจาะจงและเวกเตอร์เจาะจงซึ่งเป็นค่าที่บอกถึง

ลักษณะความโค้งและทิศทางของโครงสร้างภาพจาก Hessian Matrix ในสมการที่ (11-2) และ (11-3) โดยกำหนดให้ λ_1 เป็น maximum curvature, λ_2 เป็น minimum curvature และพิจารณาค่าสัมบูรณ์ของผลต่างระหว่างขนาดของค่าเจาะจงทั้งสอง $\|\lambda_1 - \lambda_2\|$ เราจะแบ่งโครงสร้างของภาพดิจิทัลในย่านที่เราสนใจอยู่ออกเป็น 3 กรณี

- **กรณีที่ 1 :** โครงสร้างของภาพแบบไอโซทรอปิก โดยเราจะตัดสินใจว่าโครงสร้างในย่านที่เราสนใจเป็นแบบไอโซทรอปิกเมื่อ $\|\lambda_1 - \lambda_2\| \leq \varepsilon$ เมื่อ ε เป็นค่าตั้งแต่ศูนย์จนถึงค่าบวกค่าหนึ่ง นั่นคือถ้าขนาดของค่าเจาะจงทั้งสองมีค่าไม่ต่างกันมากหรือค่าสัมบูรณ์ของผลต่างระหว่างขนาดของค่าเจาะจงทั้งสองมีค่าน้อยกว่า ε จะได้ว่าโครงสร้างของภาพแบบคอนทัวร์ในย่านที่เรากำลังสนใจมีลักษณะเป็นแบบวงกลมดังแสดงในรูปที่ 11.1 ดังนั้นเราจะทำการสร้างฟังก์ชันถ่วงน้ำหนักดังสมการ

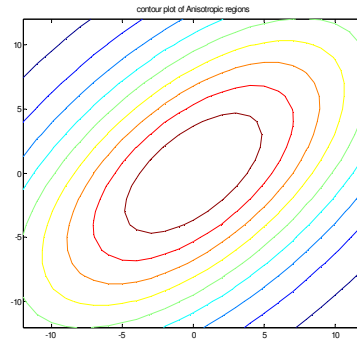
$$w_{i,j}(m,n) = \sigma^{m^2+n^2} \quad (11-4)$$

โดย σ เป็นค่าบวกและมีค่าน้อยกว่าหนึ่ง



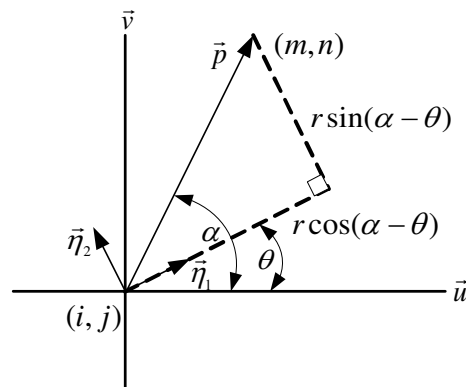
รูปที่ 11.1 แผนภาพคอนทัวร์ของฟังก์ชันถ่วงน้ำหนักแบบไอโซทรอปิก

- **กรณีที่ 2 :** โครงสร้างของภาพแบบแอนไอโซทรอปิก โดยเราจะตัดสินใจว่าโครงสร้างในย่านที่เราสนใจเป็นแบบแอนไอโซทรอปิกเมื่อ $\varepsilon < \|\lambda_1 - \lambda_2\| \leq \delta$ เมื่อ δ เป็นจุดเริ่มเปลี่ยนซึ่งมีค่ามากกว่า ε นั่นคือถ้าขนาดค่าเจาะจงทั้งสองมีค่าค่อนข้างที่จะแตกต่างกันหรือค่าสัมบูรณ์ของผลต่างระหว่างขนาดของค่าเจาะจงทั้งสองมีค่ามากกว่า ε และน้อยกว่า δ จะได้ว่าโครงสร้างของภาพแบบคอนทัวร์ในย่านที่เรากำลังสนใจมีลักษณะเป็นแบบวงรีดังแสดงในรูปที่ 11.2



รูปที่ 11.2 แผนภาพคอนทัวร์ของฟังก์ชันถ่วงน้ำหนักแบบแอนไอโซทรอปิก

โดยการสร้างฟังก์ชันถ่วงน้ำหนักลักษณะนี้จะต้องทำการเปลี่ยนระบบพิกัดปกติไปยังระบบพิกัดใหม่ในทิศทางของความโค้งหลักเพื่อให้ฟังก์ชันถ่วงน้ำหนักของเราสามารถปรับทิศทางตามทิศทางของ *principal maximum curvature*, $\vec{\eta}_1$ [22-23]



รูปที่ 11.3 การกระจายตัวของทิศทาง Curvature ในระบบพิกัดหน้าต่าง $D_{i,j}$

รูปที่ 11.3 แสดงการกระจายตัวของ *principal curvature axes* ที่ตำแหน่งหน้าต่างข้อมูลที่จะทำการกรอง $D_{i,j}$ โดยให้ $\vec{\eta}_1$ เป็นเวกเตอร์หนึ่งหน่วยในทิศทางของ *principal maximum curvature*, $\vec{\eta}_2$ เป็นเวกเตอร์หนึ่งหน่วยในทิศทางของ *principal minimum curvature* และ θ เป็นมุมระหว่าง $\vec{\eta}_1$ และแกน $\vec{u}$ สำหรับที่ตำแหน่งพิกเซล (m,n) เราจะนิยามตำแหน่งของพิกเซลเหล่านี้ด้วยเวกเตอร์ $\vec{p}$ ซึ่งเริ่มต้นจากตำแหน่งศูนย์กลางของหน้าต่าง (i,j) ซึ่งไปยังตำแหน่งของพิกเซล (m,n) โดยเวกเตอร์ $\vec{p}$ สามารถแตกได้เป็น 2 องค์ประกอบในระบบพิกัดใหม่ในทิศทางของ η_1 และ η_2 ดังสมการ

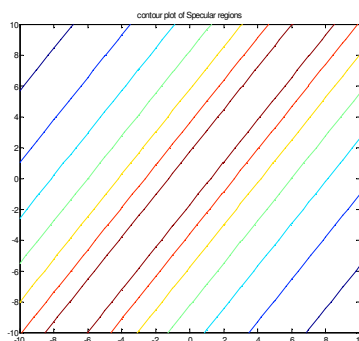
$$\vec{p} = r \cos(\alpha - \theta) \vec{\eta}_1 + r \sin(\alpha - \theta) \vec{\eta}_2 \quad (11-5)$$

โดย $r = (\sqrt{m^2 + n^2})$ และ $\alpha = \arctan(n/m)$ ในการควบคุมทิศทางในการกรองของวงจกรองชาวีสกี-โกเลย์แบบแอนไอโซทรอปิก เราจะให้ระดับทิศทางในการกรองในแนว $\vec{u}_2$ มีค่ามากกว่าในแนว $\vec{u}_1$ โดยการกำหนดฟังก์ชันถ่วงน้ำหนักให้เป็นไปตามสมการ

$$w_{i,j}(m,n) = \sigma_1^{r^2 \cos^2(\alpha-\theta)} \sigma_2^{r^2 \sin^2(\alpha-\theta)} \quad (11-6)$$

โดยที่พารามิเตอร์ σ_1 และ σ_2 เป็นค่าคงที่อยู่ในช่วง $0 < \sigma_1 < \sigma_2 < 1$

- **กรณีที่ 3 : โครงสร้างของภาพแบบเส้นตรง** โดยเราจะตัดสินใจว่าโครงสร้างในย่านที่เราสนใจเป็นแบบเส้นตรงเมื่อ $\|\lambda_1 - \lambda_2\| > \delta$ นั่นคือถ้าขนาดของค่าเจาะจงทั้งสองมีค่าแตกต่างกันมากหรือค่าสัมบูรณ์ของผลต่างระหว่างขนาดของค่าเจาะจงทั้งสองมีค่ามากกว่า δ จะได้ว่าโครงสร้างของภาพแบบคอนทัวร์ในย่านที่เรากำลังสนใจมีลักษณะเป็นแบบเส้นตรงในทิศทางของ maximum curvature ดังแสดงในรูปที่ 11.4



รูปที่ 11.4 แผนภาพคอนทัวร์ของฟังก์ชันถ่วงน้ำหนักแบบเส้นตรง

โดยฟังก์ชันถ่วงน้ำหนักจะสอดคล้องกับสมการที่ (11-5) และ (11-6) ต่างกันที่จะกำหนดให้ค่า σ_1 ในสมการที่ (11-6) มีค่าเท่ากับหนึ่งเพื่เน้นการกรองในทิศทางของ maximum curvature ดังสมการ

$$w_{i,j}(m,n) = \sigma^{r^2 \sin^2(\alpha-\theta)} \quad (11-7)$$

ค่าพารามิเตอร์ σ เป็นค่าบวกและมีค่าน้อยกว่าหนึ่ง

จะเห็นว่าฟังก์ชันถ่วงน้ำหนัก $w_{i,j}$ ที่ได้นิยามไว้ในสมการที่ (11-6) และ (11-7) นั้นขึ้นอยู่กับค่า θ โดยมีค่าต่อเนื่องตั้งแต่ 0 จนถึง 2π ซึ่งจะทำให้ฟังก์ชันถ่วงน้ำหนักของเราให้ผลลัพธ์ชุดของค่าสัมประสิทธิ์ของวงจกรองที่แตกต่างกันอยู่มากมาย เพื่อหลีกเลี่ยงการเตรียมการคำนวณค่าสัมประสิทธิ์อันมหาศาลนี้ เราจะทำการแบ่งระดับช่วงของมุม θ ออกเป็น K ระดับที่แตกต่างกัน $\{\hat{\theta}_0, \dots, \hat{\theta}_k, \dots, \hat{\theta}_{K-1}\}$ โดย K เป็นตัวเลขจำนวนจำกัดค่าหนึ่ง โดยค่า θ ที่เราได้มาจะถูกควอนไทซ์ไปยังมุมอ้างอิง $\hat{\theta}$ ดังนั้นเราจึงเตรียมค่าสัมประสิทธิ์ของวงจกรองไว้เพียงแค่ K ชุดสำหรับทิศทางอ้างอิง K ระดับซึ่งจะทำให้การคำนวณของเรามีความซับซ้อนน้อยลง และหลังการสร้าง

ฟังก์ชันถ่วงน้ำหนัก $w_{i,j}(m,n)$ เสร็จแล้วต้องทำ Normalized Sum ด้วยทุกครั้งเพื่อให้ฟังก์ชันถ่วงน้ำหนักของเรามีผลรวมของค่าถ่วงน้ำหนักมีค่าเท่ากับหนึ่ง

11.2 ขั้นตอนการประเมินประสิทธิภาพของวงจรกรอง

เพื่อทดสอบประสิทธิภาพของวงจรกรองซาวีตกี-โกเลย์แบบแอนไอโซทรอปิก ในงานวิจัยนี้ได้ใช้ภาพทดสอบซึ่งถูกรบกวนด้วยสัญญาณรบกวนที่มีแบบจำลองเช่นเดียวกับสัญญาณรบกวน Speckle อันเป็นสัญญาณรบกวนหลักในภาพอัลตราซาวนด์ และเปรียบเทียบผลที่ได้กับวงจรกรองที่เป็นที่นิยมใช้กันซึ่งประกอบไปด้วยวงจรกรองลดทอนสัญญาณรบกวน Speckle แบบปรับตัวได้ วงจรกรองมัชฐานถ่วงน้ำหนักแบบปรับตัวได้ และวงจรกรองซาวีตกี-โกเลย์สองมิติ ในด้านการลดทอนสัญญาณรบกวน การคงสภาพเส้นขอบเมื่อนำไปตรวจจับเส้นขอบภาพ จากนั้นจึงนำไปทดสอบกับภาพอัลตราซาวนด์ทางการแพทย์ เพื่อวิเคราะห์ประสิทธิภาพของวงจรกรองในการนำไปใช้งานจริง

วงจรกรองลดทอนสัญญาณ Speckle แบบปรับตัวได้

วงจรกรองลดทอนสัญญาณ รบกวน Speckle แบบปรับตัวได้ อาศัยหลักการประมาณค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNR) จากค่าเชิงสถิติในย่านที่เราสนใจด้วยอัตราส่วนของค่าความแปรปรวนและค่าเฉลี่ยเพื่อแยกความแตกต่างระหว่างหมวดของโครงสร้างภาพ FFS (Fully Formed Speckle) และ NRLR (Non-Randomly distributed with Long-Range order) ซึ่งเป็นหมวดของสัญญาณที่สะท้อนกลับมา [24] โดยมีรูปแบบสมการของวงจรกรองดังนี้

$$\hat{I} = \bar{I} + k(I - \bar{I}) \quad (11-8)$$

เมื่อ $\hat{I}$ เป็นพิกเซลเอาต์พุตใหม่ที่ได้จากการประมาณค่ากลุ่มข้อมูล, I เป็นค่าเดิมหรือเป็นค่า ณ ตำแหน่งศูนย์กลาง และ $\bar{I}$ เป็นค่าเฉลี่ยของพิกเซลข้อมูลในย่านที่เรากำลังสนใจ โดยค่าสัมประสิทธิ์ในการปรับตัว k สามารถคำนวณได้จาก

$$k = \frac{p - \bar{p}_s}{p} \quad (11-9)$$

และ

$$p = \frac{\text{var}(I)}{\bar{I}} \quad (11-10)$$

$\text{var}(I)$ เป็นค่าความแปรปรวนของข้อมูลในหน้าต่างวงจรกรอง และ $\bar{p}_s$ เป็นค่าเฉลี่ยของ p ในย่านที่เราสนใจเช่นกัน โดยค่าสัมประสิทธิ์ในการปรับตัว k จะสนใจในช่วงค่าเท่ากับศูนย์จนถึงหนึ่ง $0 \leq k \leq 1$ เพื่อให้ค่าเอาต์พุตที่ได้มีค่าอยู่ในช่วงเท่ากับค่าเดิมกับค่าเฉลี่ย ดังนั้นในบางครั้งจึงมีผู้เรียกชื่อวงจรกรองนี้ว่า Adaptive mean-based filter [24-25]

วงจรกรองมัธยฐานถ่วงน้ำหนักแบบปรับตัวได้

วงจรกรองมัธยฐานถ่วงน้ำหนักแบบปรับตัวได้พัฒนามาจากวงจรกรองมัธยฐานโดยการประมาณค่าอัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNR) ในย่านของข้อมูลที่กำลังสนใจเช่นเดียวกับในวงจรกรองลดทอนสัญญาณรบกวน Speckle แบบปรับตัวได้มาใช้ในการคำนวณฟังก์ชันถ่วงน้ำหนักดังสมการ

$$w(i, j) = \lfloor w(K+1, K+1) - cd\sigma^2/m \rfloor \quad (11-11)$$

โดย c เป็นสัมประสิทธิ์ค่าคงที่ d เป็นระยะทางจากศูนย์กลางหน้าต่างวงจรกรองไปยังตำแหน่งข้อมูลใด ๆ ในหน้าต่างวงจรกรอง $w(K+1, K+1)$ เป็นค่าถ่วงน้ำหนัก ณ ตำแหน่งศูนย์กลางของหน้าต่างวงจรกรอง m, σ^2 เป็นค่าเฉลี่ยและค่าความแปรปรวนของข้อมูลในย่านที่กำลังสนใจตามลำดับ และ $\lfloor \cdot \rfloor$ เป็นฟังก์ชัน floor ในการปัดค่าให้เป็นจำนวนเต็มที่ใกล้เคียงที่สุด โดยหากค่าอัตราส่วนความแปรปรวนต่อค่าเฉลี่ย σ^2/m มีค่าต่ำ เอาต์พุตของวงจรกรองมัธยฐานถ่วงน้ำหนักแบบปรับตัวได้จะคำนวณจากค่ามัธยฐานของข้อมูลซึ่งสอดคล้องกับฟังก์ชันถ่วงน้ำหนักในสมการข้างต้นในย่านของข้อมูลที่กำลังสนใจ หาก σ^2/m มีค่าสูง เอาต์พุตที่ได้จะถูกไบแอสให้มีค่าเท่ากับค่าความเข้มเดิม [26]

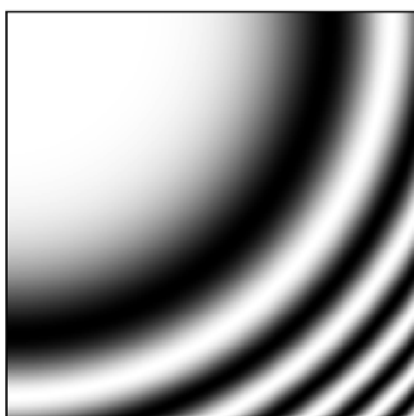
ตัวอย่างสมมุติคำนวณฟังก์ชันถ่วงน้ำหนักของลำดับข้อมูล $\{X_1, X_2, X_3\}$ ได้ค่า $w_1 = 2$, $w_2 = 3$ และ $w_3 = 2$ ดังนั้นเอาต์พุต $y_{i,j}$ หาได้จาก

$$y_{i,j} = \text{median}\{X_1, X_1, X_2, X_2, X_2, X_3, X_3\} \quad (11-12)$$

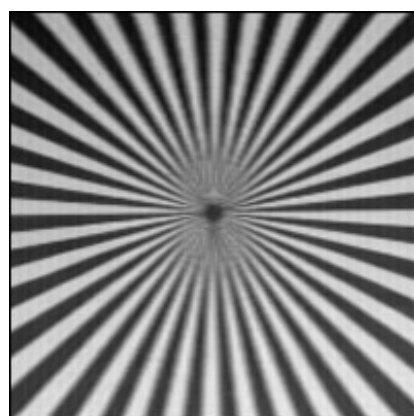
หากค่าสัมประสิทธิ์ของฟังก์ชันถ่วงน้ำหนัก $w(i, j)$ ในหน้าต่างมีค่าน้อยกว่าเป็นลบ จะปัดค่าดังกล่าวให้เท่ากับศูนย์หรือไม่มีการถ่วงน้ำหนักค่าข้อมูล ณ ตำแหน่งนั้น ด้วยความง่ายในตัวละครเบียบวิธีของวงจรกรองนี้ จึงทำให้เป็นวงจรกรองที่นิยมนำมาประยุกต์ใช้ในงานด้านต่าง ๆ

11.3 การทดสอบประสิทธิภาพวงจรกรองกับภาพสังเคราะห์ทางคณิตศาสตร์

ภาพที่ใช้ในการทดสอบเป็นภาพที่สร้างขึ้นจากสมการคณิตศาสตร์ดังรูปที่ 11.5 (ก) และ (ข) เพื่อทดสอบประสิทธิภาพในการลดทอนสัญญาณรบกวนเมื่อมีทิศทางของเส้นขอบและความชันของเส้นขอบต่าง ๆ กัน ในการทดสอบจะนำภาพทดสอบทั้งหมดมาใส่สัญญาณรบกวนแบบคูณ เพื่อพิจารณาการทำงานของวงจรกรองในสภาพที่ใกล้เคียงกัน จากนั้นจึงนำวงจรกรองไปทดสอบกับภาพอัลตราซาวนด์ทางการแพทย์จริง



(ก)



(ข)

รูปที่ 11.5 ภาพที่ใช้ในการทดสอบ

11.3.1 ขั้นตอนการวิเคราะห์สมรรถนะในการลดทอนสัญญาณรบกวนของวงจรกรอง

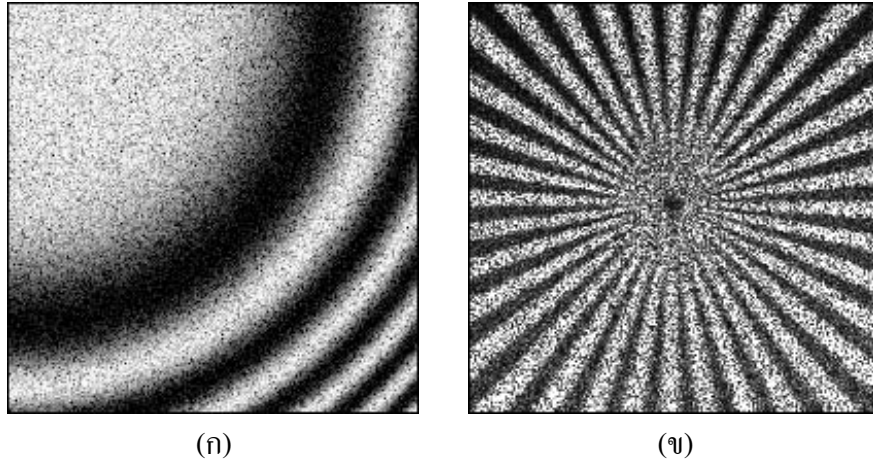
การวิเคราะห์สมรรถนะในการลดทอนสัญญาณรบกวนของวงจรกรองแบ่งออกเป็น 2 ขั้นตอน คือ

ขั้นตอนแรกของการวิเคราะห์สมรรถนะในการลดทอนสัญญาณรบกวนในภาพสังเคราะห์ของวงจรกรองได้ใช้ภาพทดสอบในรูปที่ 11.5 (ก) คูณกับสัญญาณรบกวนเฉลี่ยซึ่งเป็นแบบจำลองของสัญญาณรบกวน Speckle ดังสมการ

$$E[f_{image}] = E[f] \cdot E[N] \quad (11-13)$$

เมื่อ $E[\cdot]$ เป็นค่าคาดหวังหรือค่าเฉลี่ยเชิงสถิติ โดยสัญญาณรบกวนเฉลี่ย N ที่เรากำหนดให้กับสัญญาณจริง f เป็นอิสระต่อกัน ดังนั้นเราจึงให้ค่าเฉลี่ยของสัญญาณรบกวนมีค่าเท่ากับหนึ่ง เพื่อให้ค่าเฉลี่ยของภาพที่จะนำมาทดสอบมีค่าคงเดิม ส่งผลให้ค่าความแปรปรวนซึ่งมีความ มีค่าเท่า

กับ $\sigma^2 = 0.2732$ รูปที่ 11.6 แสดงภาพที่ใช้ในการทดสอบในรูปที่ 11.5 ซึ่งเป็นภาพสะอาด (Clean images) และนำมาใส่สัญญาณรบกวนเรย์ลีแบบคูณในสมการที่ (11-13)



รูปที่ 11.6 ภาพทดสอบที่ถูกกรบกวนด้วยสัญญาณรบกวนเรย์ลีแบบคูณ

เพื่อให้แบบจำลองของสัญญาณที่จะนำมาทดสอบมีรูปแบบการบีบอัดแบบลอการิทึมเช่นเดียวกับที่ใช้ในเครื่องตรวจจับสัญญาณภาพ อัลตราซาวนด์จริง ดังนั้นเราจึงจำลองสัญญาณภาพที่จะนำมาทดสอบเป็น

$$f_{image}(i, j) = \log\{\exp\{f(i, j)\} \cdot N(i, j)\} \quad (11-14)$$

โดยภาพที่จะนำมาทดสอบในรูปที่ 11.5 (ก) นี้เป็นภาพซึ่งสังเคราะห์จากสมการ

$$f(i, j) = \cos(0.5 \times 10^{-8}(i^4 + 2i^2j^2 + j^4)) \quad (11-15)$$

ดังแสดงในรูปที่ 11.6 (ก) แล้วให้ทดสอบโดยใช้ค่าความผิดพลาดกำลังสองเฉลี่ยแบบนอร์มัลไลซ์ (Normalized Mean Square Error: NMSE) เทียบกับขนาดหน้าต่างเป็นตัวแทนในการตัดสินใจหาค่าได้จากสมการ

$$NMSE = \frac{\sum_{\Omega} (f_{filtered} - f)^2}{\sum_{\Omega} (f_{image} - f)^2} \quad (11-16)$$

เมื่อ Ω เป็นกลุ่มของพิกเซลข้อมูลในภาพ ส่วน f_{image} เป็นค่าความเข้มของพิกเซลในภาพทดสอบที่ถูกกรบกวนด้วยสัญญาณรบกวน Speckle $f_{filtered}$ เป็นค่าความเข้มของพิกเซลในภาพที่ผ่านการกรองมาแล้ว และ f เป็นค่าความเข้มของพิกเซลข้อมูลในภาพทดสอบโดยไม่ถูกรบกวนด้วยสัญญาณรบกวนใดๆ