การปรับปรุงความเร็วของวงจรการกำจัดสัญญาณเสียงสะท้อนด้วยการออกแบบไปป์ไลน์สำหรับวงจรกรองปรับตัวได้แบบ DLMS

A Speed Improvement of Echo-Noise Cancellation Circuit Using Pipeline Design for the DLMS Adaptive Filter

สายัณ ละอองโชค

Sayan La-ongchok

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า
มหาวิทยาลัยสงขลานครินทร์

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering
Prince of Songkla University
2555

ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์

ชื่อวิทยานิพนธ์ การปรับปรุงความเร็วของวงจรการกำจัดสัญญาณเสียงสะท้อนด้วยการออกแบบไปป์ไลน์สำหรับวงจรกรองปรับตัวได้แบบ DLMS

ผู้เขียน นายสายัณ ละอองโชค

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

…….....

(ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

…….....

(ผู้ช่วยศาสตราจารย์ ดร.พรชัย พฤกษ์ภัทรานนต์)

คณะกรรมการสอบ

…….....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต)

…….....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์)

…….....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร.พรชัย พฤกษ์ภัทรานนต์)

…….....กรรมการ

(ดร.ประกาศิต กายะสิทธิ์)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

…….

(ศาสตราจารย์ ดร.อมรรัตน์ พงศ์ดารา)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์ การปรับปรุงความเร็วของวงจรการกำจัดสัญญาณเสียงสะท้อนด้วยการออกแบบไปป์ไลน์สำหรับวงจรกรองปรับตัวได้แบบ DLMS

ผู้เขียน นายสายัณ ละอองโชค

สาขาวิชา วิศวกรรมไฟฟ้า

ปีการศึกษา 2554

บทคัดย่อ

วิทยานิพนธ์นี้นำเสนอการออกแบบวงจรกรองปรับตัวได้แบบ DLMS (Delayed Least Mean Square) ขนาด 32 แท็ปเพื่อเพิ่มความเร็วในการกำจัดสัญญาณเสียงสะท้อนในระบบเครื่องช่วยฟังด้วยเทคนิคการออกแบบไปป์ไลน์บนชิพ FPGA (Field Programmable Gate Array) รุ่น Xilinx Virtex-II Pro XC2VP30 โดยมีหน่วยประมวลผลขนาด 16 บิต แบบเลขทศนิยมคงตัว Q0.15 วงจรที่นำเสนอมี 2 แบบ คือ วงจรแบบไม่ใช้ทรัพยากรร่วมและวงจรที่ใช้ทรัพยากรร่วม เมื่อทำการออกแบบวงจรที่ไม่ใช้ทรัพยากรร่วมมีค่าดีเลย์สัญญาณ 10 หน่วย (D=10) ใช้พื้นที่ 63 วงจรบวก 65 วงจรคูณ 2424 ฟลิบฟลอบ มีความเร็วอยู่ที่ 94.266 ล้านตัวอย่างต่อวินาที ในขณะที่วงจรแบบใช้ทรัพยากรร่วมมีค่าดีเลย์สัญญาณ 2 หน่วย (D=2) มีการใช้เทคนิคการจัดรูปแบบเวลาการประมวลผล (Retiming) เพื่อนำไปประยุกต์ใช้กับอุปกรณ์ที่มีทรัพยากรจำกัด โดยการแบ่งการทำงานของวงจรกรองปรับตัวได้แบบ DLMS เป็นแบบไปป์ไลน์ 2 สเตจ ทำให้เกิดการดีเลย์ที่น้อยกว่าวงจรที่ไม่ใช้ทรัพยากรร่วม ใช้พื้นที่ 17 วงจรบวก 17 วงจรคูณ 2249 ฟลิบฟลอบ มีความเร็วในการประมวลผลอยู่ที่ 20.058 ล้านตัวอย่างต่อวินาที นอกจากนี้ได้ทำการเปรียบเทียบอัตราการลู่เข้าเทียบกับวงจรกรองปรับตัวได้แบบ LMS (Least Mean Square) เมื่อพิจารณาจำนวนตัวอย่างพบว่าวงจร LMS มีอัตราการลู่เข้าที่เร็วกว่าวงจร DLMS แต่เมื่อพิจารณาด้านเวลาในการประมวลผลพบว่า วงจรกรองปรับตัวได้แบบ DLMS มีอัตราการลู่เข้าที่เร็วกว่าเนื่องจากทำงานแบบไปป์ไลน์ โดยในกรณีวงจรแบบไม่ใช้ทรัพยากรร่วม วงจร LMS มีอัตราการลู่เข้าที่ 4.5 ไมโครวินาที ที่จำนวน 120 ตัวอย่าง ในขณะที่ DLMS (D=10) มีอัตราการลู่เข้าที่ 1 ไมโครวินาที ที่จำนวน 150 ตัวอย่าง สำหรับกรณีวงจรที่ใช้ทรัพยากรร่วมกัน วงจร LMS มีอัตราการลู่เข้าที่ 13 ไมโครวินาที วงจร DLMS (D=2) มีอัตราการลู่เข้าที่ 5 ไมโครวินาที

คำสำคัญ: วงจรกรองปรับตัวได้, ไปป์ไลน์, การใช้ทรัพยากรร่วมกัน

Thesis Title	A Speed Improvement of Echo-Noise Cancellation Circuit Using Pipeline Design for the DLMS Adaptive Filter
Author	Mr. Sayan La-ongchok
Major Program	Electrical Engineering
Academic Year	2011

ABSTRACT

This thesis presents a design of a 32-tap DLMS (Delayed Least Mean Square) adaptive filter to accelerate echo-noise cancellation in hearing aids by using a pipeline design technique on an FPGA (Field Programmable Gate Array) of Xilinx Virtex-II Pro XC2VP30. The processing unit is based on the Q0.15 16-bit fixed-point number. The proposed circuits have two styles: a DLMS adaptive filter with non-resource sharing and a DLMS adaptive filter with resource sharing. The DLMS adaptive filter (D=10) with non-resource sharing circuit was implemented resulting in the area of 63 adders, 65 multipliers, and 2424 flip-flops, and the speed of 94.266 MSPS (Million samples per second). In the latter DLMS circuit with resource sharing (D=2), the retiming technique was applied in order to use in the limited area devices. This DLMS circuit was implemented by a 2-stage pipeline resulting in the area of 17 adders, 17 multipliers, and 2249 flip-flops, and the speed of 20.058 MSPS. In addition, the proposed DLMS circuits were compared to the LMS (Least Mean Square) circuits in the convergence rate term. Certainly, the LMS circuits converged at the lower number of samples than that of the DLMS circuits. However, the DLMS circuits converged at the faster time than that of the LMS circuits. In the case of non-resource sharing, the LMS circuit converged at the 120^{th} sample with 4.5μs, whereas the DLMS (D=10) converged at the 150^{th} sample with only 1.0 μs. In the case of resource sharing, the LMS circuit converged at 13 μs, whereas the DLMS (D=2) converged at 5 μs.

KEYWORDS: Adaptive filter, Pipeline, Resource sharing

กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ณัฏฐา จินดาเพ็ชร์ ประธานกรรมการที่ปรึกษางานวิจัยที่ได้เสียสละเวลาในการให้คำปรึกษา แนวคิดในการทำวิจัย รวมถึงการช่วยเหลือแก้ไขปัญหาที่เกี่ยวกับการวิจัย ตลอดจนตรวจสอบและแก้ไขวิทยานิพนธ์ให้ดำเนินลุล่วงไปอย่างสมบูรณ์

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พรชัย พฤกษ์ภัทรานนต์ ที่ได้กรุณาให้คำปรึกษา คำแนะนำ และให้ความช่วยเหลือในงานวิจัย ตลอดจนช่วยตรวจทานแก้ไขวิทยานิพนธ์ให้ดำเนินไปด้วยดี

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.วรรณรัช สันติอมรทัต ที่กรุณาเสียสละเวลาเป็นประธานกรรมการสอบวิทยานิพนธ์ อีกทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ ดร.ประกาศิต กายะสิทธิ์ ที่กรุณาเสียสละเวลาเป็นกรรมการสอบวิทยานิพนธ์ อีกทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่ที่ให้การสนับสนุนทุนในการทำวิจัยและให้ความช่วยเหลือด้านการประสานงานต่างๆ

ขอขอบพระคุณ คณาจารย์ บุคลากร และเพื่อนๆน้องๆนักศึกษาปริญญาโท พี่ๆนักศึกษาปริญญาเอก ภาควิชาวิศวกรรมไฟฟ้าทุกคนที่ได้ให้คำปรึกษาและกำลังใจในการทำงานเป็นอย่างดีเสมอมา

และสุดท้าย ข้าพเจ้าขอน้อมรำลึกถึงพระคุณของ บิดามารดา และครอบครัว ที่ส่งเสริมและสนับสนุน ให้กำลังใจ ให้คำปรึกษาและคำชี้แนะ ให้เวลามาศึกษาต่อ ตลอดจนถึงทุนทรัพย์แก่ข้าพเจ้ามาโดยตลอดมาจนสำเร็จการศึกษา

สายัณ ละอองโชค

สารบัญ

	หน้า
สารบัญ	(6)
รายการตาราง	(9)
รายการภาพประกอบ	(10)
บทที่	
1. บทนำ	1
1.1 ความสำคัญและที่มาของหัวข้อวิจัย	1
1.2 การตรวจเอกสารบทความและงานวิจัยที่เกี่ยวข้อง	2
1.3 วัตถุประสงค์	7
1.4 ขอบเขตของการวิจัย	7
1.5 ขั้นตอนและวิธีการดำเนินงานวิจัย	7
1.6 ประโยชน์ที่คาดว่าจะได้รับ	8
2. ทฤษฎีที่เกี่ยวข้อง	9
2.1 วงจรกรองความถี่แบบเอฟไอาร์ (Finite impulse response filter)	9
2.1.1 คุณสมบัติของวงกรองแบบเอฟไอาร์	9
2.1.2 การสร้างวงจรกรองเอฟไอาร์	10
2.2 วงจรกรองปรับตัว (Adaptive filter)	11
2.2.1 วงจรกรองปรับตัวได้แบบ LMS	11
2.2.2 วงจรกรองปรับตัวได้แบบ DLMS (Delayed Least Mean Square Error)	13
2.3 ระบบเลขทศนิยมคงตัว (Fixed-point Number System)	16
2.3.1 การบวกเลขทศนิยม	18
2.3.2 การคูณเลขทศนิยม	20
2.4 ระบบไปป์ไลน์	21
2.4.1 การทำไปป์ไลน์ละเอียด (Fine-Grained pipeline)	22
2.4.2 วงจรบวก	23

สารบัญ (ต่อ)

	หน้า
2.4.3 วงจรคูณแบบไปป์ไลน์	25
2.5 รูปแบบวงจรและการวิเคราะห์	27
2.6 เครื่องช่วยฟัง (Hearing Aids)	29
2.6.1 ประเภทของเครื่องช่วยฟัง	30
2.6.2 ส่วนประกอบของเครื่องช่วยฟัง	30
2.6.3 บล็อกไดอะแกรมของเครื่องช่วยฟังแบบดิจิตอล	31
2.6.4 วงจรการกำจัดสัญญาณเสียงสะท้อนของเครื่องช่วยฟัง	31
3 .การออกแบบและวิธีทดสอบระบบ	34
3.1 การออกแบบด้วย AccelDSP Tool	34
3.1.1 การออกแบบหาค่าความเหมาะสมของขนาดของวงจรกรอง	34
3.1.2 การแบ่งจำนวนไปป์ไลน์	38
3.1.3 การสร้างวงจรกรองปรับตัวได้แบบ LMS	40
3.1.4 วงจรกรองปรับตัวได้แบบ DLMS	42
3.2 การออกแบบวงจรแบบการใช้ทรัพยากรร่วมกันด้วยภาษา Verilog HDL	46
3.2.1 การทรัพยากรร่วมกันของวงจรกรองเอฟไอาร์	47
3.2.2 การทรัพยากรร่วมกันของวงจรปรับปรุงค่าสัมประสิทธิ์	49
3.3 กระบวนการปัดเศษจากการคูณและการบวก (Round off)	49
3.4 วงจรควบคุมการทำงานโดยใช้ Finite State Machine	51
3.5 การออกแบบวงจรด้วยไปป์ไลน์ละเอียด	51
3.5.1 การออกแบบวงจรปรับปรุงค่าสัมประสิทธิ์	52
3.5.2 การออกแบบวงจรกรองเอฟไอาร์ส่วนที่ 1	52
3.3.3 การออกแบบวงจรกรองเอฟไอาร์ส่วนที่ 2	55
4. ผลการดำเนินงานและวิเคราะห์ผล	56
4.1 ผลการสังเคราะห์วงจร	56
4.2 การทดสอบวงจร	59
4.3 ผลจากการปัดเศษ (Round off)	63
4.4 การทดสอบอัตราการลู่เข้า	61

สารบัญ (ต่อ)

	หน้า
5. สรุปผลการวิจัยและข้อเสนอแนะ	69
5.1 สรุปผล	69
5.2 ปัญหาและอุปสรรคของการทำวิทยานิพนธ์	70
5.3 ข้อเสนอแนะ	70
บรรณานุกรม	72
ภาคผนวก	74
ภาคผนวก ผลงานตีพิมพ์เผยแพร่วิทยานิพนธ์	75
ประวัติผู้เขียน	81

รายการตาราง

ตาราง		หน้า
2-1	ค่า range และ precision ขนาด 16 บิต ตามรูปแบบ Q	15
2-2	แสดงช่วงค่าต่างๆของเลขฐานสิบของรูปแบบ16 บิตเมื่อถูกแทนด้วยเลข 16 บิต	16
2-3	สรุปการทำงานของวงจรบวกแบบต่างๆ ขนาด 16 บิต	21
3-1	ความเร็วในการทำงานของวงจรบวกและวงจรคูณ	39
4-1	การใช้ทรัพยากรในวงจรกรองปรับตัวได้แบบต่างๆ เป็นวงจรไม่ได้ใช้ทรัพยากรร่วม	56
4-2	การใช้ทรัพยากรในวงจรกรองปรับตัวได้แบบต่างๆ เป็นวงจรแบบใช้ทรัพยากรร่วมกัน	57
4-3	เปรียบเทียบจำนวนรอบการทำงาน(Clock cycle) ในวงจรกรองปรับตัวได้แบบต่างๆ	58

รายการภาพประกอบ

ภาพประกอบ		หน้า
1-1	โครงสร้างวงจรกรองปรับตัวได้แบบ DLMS	2
1-2	วงจรกรองปรับตัวได้แบบ DLMS ขนาด 1 แท็ป	3
1-3	การต่อโมดูลของวงจรกรองปรับตัว มีขนาดจำนวน N แท็ป รวมกัน	3
1-4	อัตราการลู่เข้าของวงจรกรองปรับตัวได้แบบ LMS กับ วงจรกรองปรับตัวได้แบบ DLMS มีจำนวน N = 16 แท็ป	4
1-5	การสร้างวงจรเพื่อปรับปรุงการลู่เข้าให้มีความรวดเร็ว	5
1-6	อัตราการลู่เข้าของวงจรกรองปรับตัวได้แบบ DLMS	5
1-7	การสร้างวงจรกรองปรับตัวได้แบบไปป์ไลน์ละเอียด	6
2-1	โครงสร้างวงจรกรองเอฟไออาร์แบบโดยตรง	10
2-2	โครงสร้างวงจรกรองเอฟไออาร์แบบทรานสโพส	10
2-3	โครงสร้างวงจรกรองแบบสมมาตร	11
2-4	โครงสร้างวงจรกรองปรับตัวได้แบบ LMS	11
2-5	โครงสร้างวงจรกรองปรับตัวได้แบบ DLMS	13
2-6	รูปแบบ Q*m.n* แบบต่างๆ	16
2-7	การบวกเลขทศนิยมที่มีจำนวนบิตเท่ากัน	19
2-8	การบวกเลขทศนิยมรูปแบบไม่เหมือนกัน	19
2-9	วงจรคูณเลขสองจำนวนตามรูปแบบเลขจำนวนเต็ม	20
2-10	วงจรไปป์ไลน์	22
2-11	วงจรไปป์ไลน์ละเอียด	22
2-12	วงจรบวกแบบ Full adder	23
2-13	วงจรบวกแบบ Full adder แบบตัวทดเลื่อน ขนาด 4 บิต	23
2-14	โครงสร้างของวงจรบวกแบบ Carry Select Adder ขนาด 4 บิต	24
2-15	วงจรคูณที่ถูกสร้างด้วย IP-core generator	25
2-16	การกำหนดค่าวงจรคูณในการเลือกความเร็วหรือพื้นที่	26
2-17	การกำหนดค่าวงจรคูณในการเลือกจำนวนไปป์ไลน์	26
2-18	รูปแบบของวงจรที่ใช้ออกแบบวงจรประมวลผลสัญญาณดิจิตอล	27
2-19	โครงสร้างวงจร Moore machine และ Mealy machine	28

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
2-20	บล็อกไดอะแกรมการทำงานของเครื่องช่วยฟัง	30
2-21	บล็อกไดอะแกรมการกำจัดสัญญาณเสียงสะท้อนของเครื่องช่วยฟัง	31
2-22	ผลตอบสนองอิมพัลส์การป้อนกลับของเครื่องช่วยฟังแบบในช่องหูขนาดเต็มใบหู	32
2-23	การเกิดเสียงสะท้อนในห้องและผลตอบสนองต่ออิมพัลส์ของระบบ	33
3-1	บล็อกไดอะแกรมของเครื่องช่วยฟัง	34
3-2	การทำงานของวงจรกรองปรับตัวได้ขนาด 8 แท็ป	35
3-3	การทำงานของวงจรกรองปรับตัวได้ขนาด 16 แท็ป	36
3-4	การทำงานของวงจรกรองปรับตัวได้ขนาด 32 แท็ป	37
3-5	วงจรกรองปรับตัวได้แบบ LMS ขนาด 64 แท็ป	38
3-6	การต่อวงจรบวกแบบไบนารีทรี	39
3-7	โครงสร้างของวงจรกรองปรับตัวได้แบบ LMS ขนาด 32 แท็ป	40
3-8	โครงสร้างภายในบล็อก MAC module ของวงจรกรองปรับตัวได้แบบ LMS	41
3-9	โครงสร้างวงจรภายใน Update Block ของวงจรกรองปรับตัวได้แบบ LMS	41
3-10	โครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS (D=5)	42
3-11	การแบ่งไปป์ไลน์วงจรกรองปรับตัวได้แบบ DLMS (D=5)	43
3-12	Update Block ของวงจรกรองปรับตัวได้ DLMS (D=5)	43
3-13	โครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS (D=7)	44
3-14	การแบ่งไปป์ไลน์วงจรกรองปรับตัวได้แบบ DLMS (D=7)	45
3-15	โครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS (D=10)	45
3-16	การแทรกไปป์ไลน์ในวงจรเอฟไออาร์ขนาด 8 แท็ป	46
3-17	การแบ่งรายละเอียดภายในวงจรกรองแบบ DLMS	47
3-18	การออกแบบวงจรกรองเอฟไออาร์ขนาด 32 แท็ป ส่วนที่ 1	48
3-19	การออกแบบวงจรกรองเอฟไออาร์ขนาด 32 แท็ป ส่วนที่ 2	48
3-20	การออกแบบวงจรปรับค่าสัมประสิทธิ์ขนาด 32 แท็ป แบบใช้ทรัพยากรร่วม	49
3-21	วิธีการปัดเศษหลังจากการประมวลผล	50
3-22	วิธีการปัดเศษในครั้งสุดท้ายเพียงครั้งเดียว	50

รายการภาพประกอบ (ต่อ)

ภาพประกอบ		หน้า
3-23	วงจรควบคุมการทำงานของ Finite State Machine	51
3-24	การออกแบบวงจรปรับปรุงค่าสัมประสิทธ์ 32 แท็ปโดยการใช้ทรัพยากรร่วม	52
3-25	การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 1 SEC1	53
3-26	การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 1 SEC2	53
3-27	การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 1 SEC3	54
3-28	การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 1 SEC4	54
3-29	การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 2	55
4-1	โมเดลการทดสอบการทำงานของวงจรกรองปรับตัวได้แบบไม่ใช้ทรัพยากรร่วม	59
4-2	การทดสอบการทำงานของวงจรกรองปรับตัวได้ LMS แบบไม่ใช้ทรัพยากรร่วม	60
4-3	เปรียบเทียบสัญญาณที่ได้จากวงจรกรองปรับตัวได้ LMS ใน MATLAB กับผลที่ได้จาก FPGA แบบไม่ใช้ทรัพยากรร่วม	60
4-4	เปรียบเทียบสัญญาณที่ได้จากวงจรกรองปรับตัวได้ LMS และ DLMS	61
4-5	การสร้างสัญญาณเสียงที่ใช้ในการทดสอบ	62
4-6	สัญญาณเสียงเอาต์พุตที่ผ่านวงจรกรองปรับตัวได้แบบ DLMS เปรียบเทียบกับสัญญาณที่ได้จาก MATLAB	63
4-7	ผลจากกระบวนการปัดเศษหลังจากประมวลผล	63
4-8	เปรียบเทียบกระบวนการปัดเศษหลังจากประมวลผลกับ MATLAB	64
4-9	ผลจากกระบวนการปัดเศษเพียงครั้งเดียว	64
4-10	เปรียบเทียบอัตราการอัตราลู่เข้าของวงจรกรองปรับตัวได้แบบต่างๆ	65
4-11	การเปรียบเทียบเวลาเมื่ออัตราการลู่เข้าของเวลาวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เป็นวงจรไม่ใช้ทรัพยากรร่วม	66
4-12	การเปรียบเทียบเวลาเมื่ออัตราการลู่เข้าของเวลาวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เป็นวงจรใช้ทรัพยากรร่วมกัน	66
4-13	การเปรียบเทียบอัตราการลู่เข้าวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เมื่อใช้อัตราการเรียนรู้ที่ค่าต่างๆ	67
4-14	การเปรียบเทียบอัตราการลู่เข้าวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เมื่อปรับอัตราการเรียนรู้ที่เท่ากัน	68

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มาของงานวิจัย

ปัจจุบันการพัฒนาทางด้านอิเล็กทรอนิกส์มีบทบาทที่สำคัญมากต่อวงการอุตสาหกรรม ซึ่งการประมวลผลสัญญาณดิจิตอลถูกนำมาใช้งานทางด้านต่างๆมากมาย เช่นการสื่อสาร ระบบคอมพิวเตอร์ งานควบคุมโดยวงจรดิจิตอลทั้งหลายและงานทางด้านการบันเทิง การประมวลผลสัญญาณเสียง ทำเป็นเสียงก้อง เสียงสะท้อนหรือเอฟเฟ็คต่างๆ โดยอาศัยหลักการประมวลผลสัญญาณทั้งนั้น สำหรับสัญญาณเสียงสะท้อนที่เราได้ยินเมื่อเกิดขึ้นในบางครั้งก็สามารถสร้างสุนทรีได้ เช่น การรับชมภาพยนตร์ในระบบเสียงดิจิตอลเซอร์ราวด์ ซึ่งเสียงมีคุณภาพที่สูงเนื่องจากอัตราการสุ่มตัวอย่าง (Sampling Rate) อยู่ที่ 96 kHz และสำหรับเครื่องเล่น MP3 DVD อยู่ที่ 48 kHz ทำให้สร้างบรรยากาศในการชมมากยิ่งขึ้นด้วย แต่ถ้าหากเสียงสะท้อนที่เกิดขึ้นทุกเมื่อในการรับชมก็จะทำให้เกิดการรบกวนจนผู้ชมอาจฟังไม่รู้เรื่องเลย และยังสร้างความรำคาญให้อีกด้วย เนื่องจากเสียงสะท้อนตัวที่เกิดขึ้นมีการทับหรือรวมกับสัญญาณเสียงที่เข้ามา เมื่อผ่านวงจรขยายเสียง สัญญาณที่ออกทางลำโพงจะไม่ชัดเจนเท่าที่ควรเมื่อเปรียบเทียบกับเสียงที่เราพูดออกไปอันเนื่องมาจากสัญญาณเสียงสะท้อนรวมอยู่ด้วย สาเหตุของปัญหาเสียงสะท้อนที่เกิดขึ้นเนื่องมาจากสัญญาณเสียงที่ออกมาจากลำโพงย้อนกลับมายังไมโครโฟนผู้พูด เมื่อสัญญาณเสียงมารวมกันก็ทำให้เกิดได้ยินเสียงที่ถูกพูดไปแล้วด้วยส่วนหนึ่ง เช่น ห้องประชุม ระบบสายส่งโทรศัพท์ และสาเหตุอีกประการหนึ่งมาจากการจัดวางลำโพงกับไมโครโฟนให้อยู่ใกล้ชิดกันมากไม่สามารถหลีกเหลี่ยงได้ เช่น ในเครื่องช่วยฟังดิจิตอล (Digital Hearing Aids) หรือโทรศัพท์มือถือ เพราะอุปกรณ์เหล่านี้มีขนาดที่เล็กมีพื้นที่จำกัดในการจัดวางอุปกรณ์ ดังนั้นปัญหาเสียงสะท้อนอาจจะหลีกเลี่ยงได้ยาก

การแก้ปัญหาเสียงสะท้อนที่เกิดขึ้นและสัญญาณรบกวนในระบบนั้นมีอยู่หลากหลายวิธีการที่นำมาประยุกต์ใช้ เช่น ใช้วงจรอนาล็อกในการกรองความถี่ เช่น วงจรกรองความถี่แบบน๊อต (Notch Filter) หรือวงกรองความถี่ดิจิตอลที่นิยมใช้กันอยู่อย่างแพร่หลาย ได้แก่ วงจรกรองเอฟไออาร์ (Finite Impulse Response Filter) วงจรกรองไอไออาร์ (Infinite Impulse Response Filter) วงจรกรองความถี่แบบปรับตัวได้ (Adaptive Filter) แบบต่างๆ เช่น LMS (Least Mean Square), NLMS (Normalised Least Mean Square) เป็นต้น ทำให้สามารถรับฟังเสียงนั้นได้ชัดเจนยิ่งขึ้น รู้และเข้าใจความหมายของคำ และการสื่อสารเป็นไปอย่างมีประสิทธิภาพมากยิ่งขึ้น อย่างไรก็ตามเมื่อพิจารณาวงจรกรองความถี่แบบปรับตัวได้แบบ LMS มาแก้ไขปัญหาเสียงสะท้อนพบว่าวงจรมีอัตราการลู่เข้าที่เร็ว แต่มีปัญหาเรื่องความเร็วในการทำงานช้าลง เมื่อจำนวนแท็ปของ

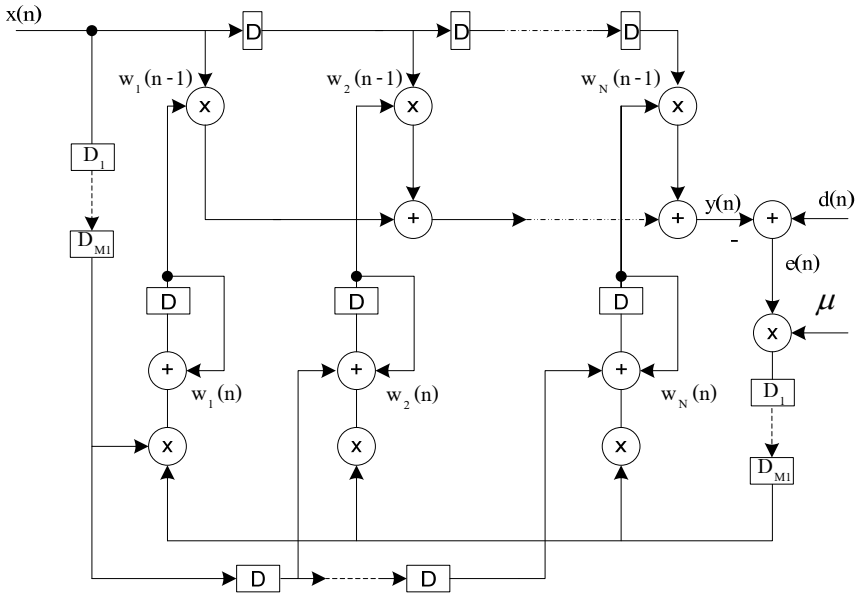
วงจรกรองมีจำนวนเพิ่มขึ้นและวงจรกรองปรับตัวได้แบบ LMS นี้มีการป้อนกลับด้วยจึงไม่สามารถทำการออกแบบวงจรไปป์ไลน์ (Pipeline) มาใช้ในการเพิ่มความเร็วในการทำงาน

โครงการวิจัยนี้เป็นการออกแบบสร้างวงจรประมวลผลสัญญาณดิจิตอล สำหรับวงจรกรองความถี่แบบปรับตัวได้ด้วย FPGA (Field Programmable Gate Array) โดยใช้วงจรกรองปรับตัวได้แบบ DLMS (Delayed Least Mean Square) มาทำการแก้ปัญหาของวงจรกรองปรับตัวได้แบบ LMS เพื่อให้ทำไปป์ไลน์ได้ เนื่องจากมีการหน่วงสัญญาณขาเข้าไว้เท่ากับจำนวนสเตจของไปป์ไลน์ โดยใช้เทคนิค Fine-Grained Pipeline มาทำการออกแบบซึ่งเป็นการทำไปป์ไลน์ละเอียดภายในวงจรคณิตศาสตร์ (เช่น วงจรคูณ วงจรบวก) สามารถเพิ่มความเร็วให้กับวงจรกำจัดสัญญาณเสียงสะท้อน

1.2 การตรวจเอกสารบทความและงานวิจัยที่เกี่ยวข้อง

1.2.1 The LMS algorithm with delayed coefficient adaptation [1]

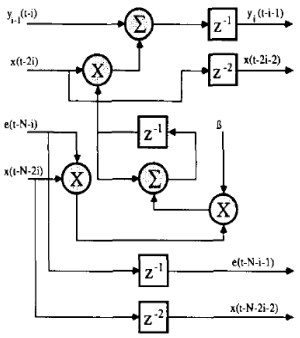
บทความนี้เป็นการนำเสนอการปรับปรุงวงจรกรองปรับตัวได้แบบ LMS โดยการเพิ่มดีเลย์ (Delay) เข้าไปแทรกในส่วนป้อนกลับเพื่อทำการปรับปรุงค่าสัมประสิทธิ์ใหม่ไปพร้อมๆกัน เรียกว่า วงจรกรองปรับตัวได้แบบ DLMS ซึ่งเป็นวงจรต้นแบบของวงจรกรองปรับตัวได้แบบ DLMS ทำให้สามารถนำไปสร้างวงจรในรูปแบบไปป์ไลน์ได้ ดังแสดงในภาพประกอบ 1-1 โดยการแทรก $D_1 - D_{M1}$ ทั้งสองส่วน ทำให้วงจรมีการทำงานที่เร็วขึ้น



ภาพประกอบ 1-1 โครงสร้างวงจรกรองปรับตัวได้แบบ DLMS [3]

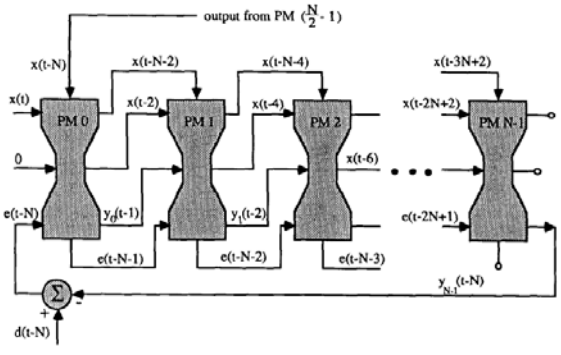
1.2.2 A high sampling rate delayed LMS filter architecture implement [2]

บทความนี้เป็นวงจรกรองปรับตัวได้แบบ DLMS อธิบายแยกออกเป็นโมดูล เพื่อที่สะดวกจะนำไปสร้างวงจร เป็นวงจรกรองที่มีขนาดต่างๆ ตามที่ออกแบบไว้ โดยการสร้างวงจรเป็นแบบทรานโพสฟอร์ม (Transpose Form) เมื่อขนาดของจำนวนวงจรกรอง (filter length) เพิ่มมากขึ้น ทำให้ขนาดของจำนวนทรัพยากรที่เพิ่มขึ้นตาม ทำให้เกิดจำนวนดีเลย์ที่เพิ่มขึ้นตามไปด้วย ส่งผลให้อัตราการลู่เข้าที่ช้า จำนวนดีเลย์ของวงจรแบบนี้มีค่าเท่ากับขนาดจำนวนแท็ปของวงจรกรอง



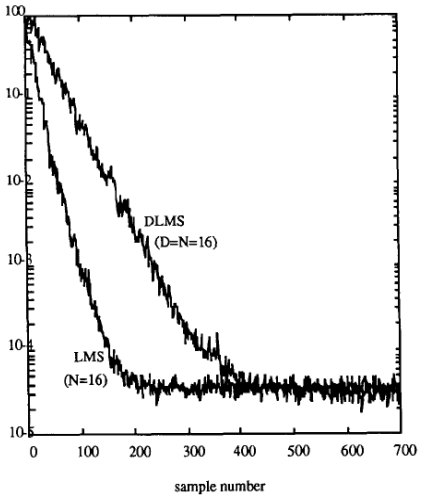
ภาพประกอบ 1-2 วงจรกรองปรับตัวได้แบบ DLMS ขนาด 1 แท็ป [2]

จากภาพประกอบ 1-2 การสร้างวงจรแบบนี้มีความสะดวกและง่าย เนื่องจากได้ทำโมดูลเสร็จเป็นที่เรียบร้อย เมื่อนำมาต่อรวมกันสามารถเชื่อมต่อตามจำนวนของวงจรกรองตามที่ออกแบบไว้ โดยที่ N คือ จำนวนแท็ปของวงจรกรอง



ภาพประกอบ 1-3 การต่อโมดูลของวงจรกรองปรับตัว มีขนาดจำนวน N แท็ป รวมกัน [2]

เมื่อนำจำนวน N แท็ปของวงจรกรองมาต่อรวมกันดังแสดงในภาพประกอบ 1-3 ขนาดของวงจรจะใหญ่ตามจำนวนแท็ปของวงจรกรองที่เพิ่มขึ้นและทำให้อัตราการลู่เข้าของวงจรช้าลงด้วย ดังในภาพประกอบ 1-4



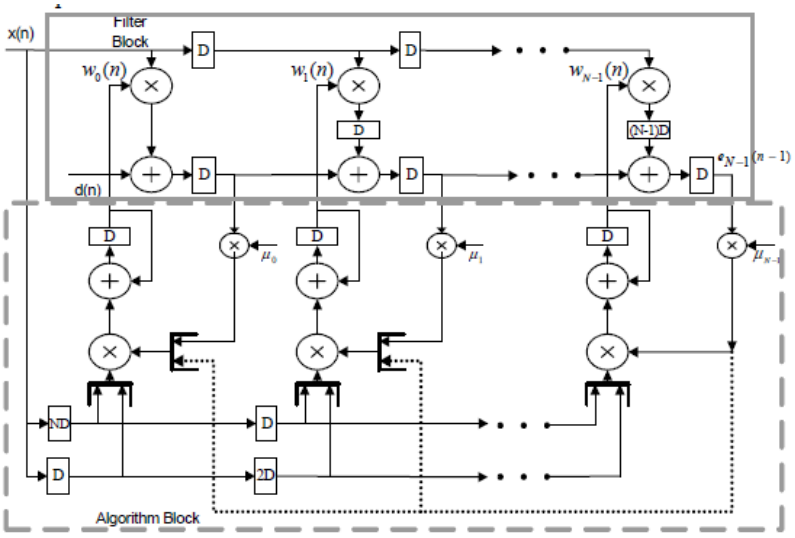
ภาพประกอบ 1-4 อัตราการลู่เข้าของวงจรกรองปรับตัวได้แบบ LMS กับ
วงจรกรองปรับตัวได้DLMS มีจำนวน N = 16 tap [2]

1.2.3 Low-power pipeline LMS adaptive filter architectures with minimal adaption delay [3]

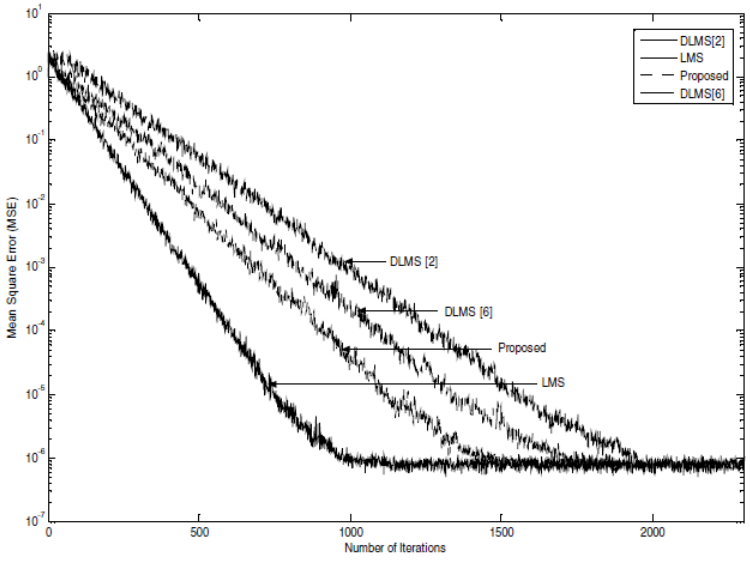
บทความนี้ได้อธิบายวิธีการลดจำนวนดีเลย์ที่เกิดขึ้นในวงจรกรองปรับตัวได้แบบ DLMS ขนาดของวงจรกรองที่มาก โดยอาศัยโมดูลย่อยมาทำการสร้างวงจรภายในโครงสร้างของโมดูลย่อยมีการใช้มัลติเพล็กเซอร์ (Multiplexer) มาออกแบบร่วมและวงจรบวกภายในเป็นแบบไบนารีทรีต่อร่วม ทำให้ลดค่าการดีเลย์ของวงจรกรองปรับตัวได้แบบ DLMS ส่งผลให้อัตราการลู่เข้าที่รวดเร็ว

1.2.4 A high-throughput DLMS adaptive algorithm [4]

บทความนี้เป็นวงจรกรองปรับตัวแบบ DLMS เพื่อทำการปรับปรุงอัตราการลู่เข้าของวงจรให้มีความรวดเร็วมากยิ่งขึ้น เมื่อนำไปสร้างวงจรทำให้การใช้จำนวนทรัพยากรเพิ่มขึ้นเป็นจำนวนมาก โดยที่ขนาดวงจรคูณเพิ่มเป็น 3 เท่า และมีรีจิสเตอร์มาแทรกกั้นอีก จำนวนดีเลย์ที่เพิ่มขึ้นเท่ากับจำนวนแท็ปของวงจรกรอง ดังภาพประกอบ 1-5 ทำให้อัตราการลู่เข้าของวงจรเร็วขึ้น โดยการเปรียบเทียบกับวงจรกรองปรับตัวได้แบบ LMS และ DLMS ดังแสดงในภาพประกอบ 1-6



ภาพประกอบ 1-5 การสร้างวงจรเพื่อปรับปรุงการลู่เข้าให้มีความรวดเร็ว [4]

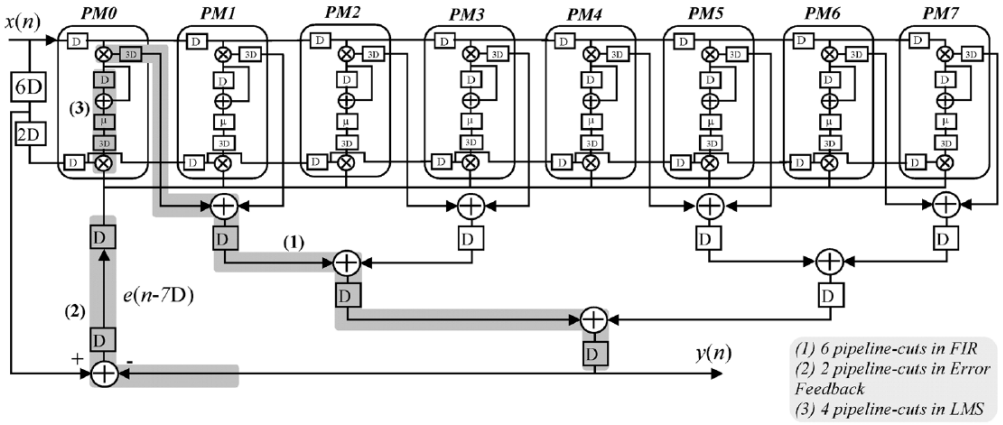


ภาพประกอบ 1-6 อัตราการลู่เข้าของวงจรกรองปรับตัวได้แบบ DLMS [4]

1.2.5 Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers [5]

บทความนี้โดยใช้การสร้างวงจรแบบโดยตรง (Direct form) การบวกเป็นแบบไบนารีทรี (Binary tree) เพื่อลดการดีเลย์ที่เกิดจากการเพิ่มขึ้นของจำนวนของวงจรกรองแบบทรานสโพสโดยที่จำนวนแท็ปของวงจรกรองแยกออกเป็นโมดูลย่อย วิธีการออกแบบเป็นแบบไปป์ไลน์ละเอียดเมื่อนำโมดูลแต่ละตัวมารวมกันแบบไบนารีทรี ความเร็วในการทำงาน แบบนี้ มีความเร็วที่สูงกว่า

แบบทรานสโพส [2] เนื่องจากวงจรที่ออกแบบเป็นไปป์ไลน์ละเอียด และทำให้ค่าการดีเลย์ของวงจรกรองปรับตัวได้ลดลงด้วย



ภาพประกอบ 1-7 การสร้างวงจรกรองปรับตัวได้แบบไปป์ไลน์ละเอียด [7]

1.2.6 Feedback cancellation in hearing aids: result from a computer simulation [6]

บทความนี้เป็นการอธิบายการแก้ปัญหาการป้อนกลับสัญญาณในเครื่องช่วยฟังด้วยวงจรกรองปรับตัวได้แบบ LMS และวงจรกรองปรับตัวได้แบบ Wiener ของเครื่องช่วยฟังในช่องหูขนาดเต็มใบหู (ITE: In the ear hearing aid) และแสดงการหาค่าสัมประสิทธิ์ของวงจรกรองปรับตัวได้แบบ LMS ขนาด 16, 32 และ 64 แท็ป มีประสิทธิภาพในการกำจัดสัญญาณการป้อนกลับของวงจรแต่ละแบบซึ่งวงจรขนาด 64 แท็ป สามารถให้ประสิทธิภาพที่ดีกว่าวงจรกรองขนาด 32 และ 16 แท็ป ด้วยการประมวลผลสัญญาณแบบ 32 บิต ส่วนที่ต่อมาได้อธิบายการหาค่าเฉลี่ยกำลังสองที่เหมาะสมในการออกแบบวงจรกรองปรับตัวได้

1.2.7 Reducing acoustic feedback in hearing aids [7]

บทความนี้อธิบายถึงการแก้ปัญหาที่เกิดจากสัญญาณป้อนกลับในเครื่องช่วยฟัง มีทั้งหมด 5 วิธีการ ประกอบไปด้วย วงจรกรองปรับตัวได้แบบน๊อต (Adaptive Notch Filter) แบบที่สองวงจร Adaptive Feedback Cancellation (AFC) ประกอบด้วย 3 วงจรกรองได้แก่ วงจร Continuous Adaption วงจร Noncontinuous Adaption และ วงจรกรองเอฟไออาร์กับวงจรกรองไอไออาร์ ส่วนวงจรกรองสุดท้าย คือ วงจรกรอง Novel AFC ซึ่งถูกพัฒนามาจากวงจรกรองแบบ Noncontinuous Adaption เมื่อนำไปสร้างวงจรในรูปแบบที่แตกต่างกันออกไป ผลตอบสนองของ

วงจรไม่เหมือนกัน และการใช้ทรัพยากรที่แตกต่างกัน เพื่อนำไปสร้างวงจรกำจัดสัญญาณการป้อนกลับภายในเครื่องช่วยฟัง

1.2.8 Modeling external feedback path of an ITE digital hearing instrument for acoustic feedback cancellation [8]

บทความนี้ได้อธิบายวิถีสัญญาณป้อนกลับ (Feedback path) ในเครื่องช่วยฟังแบบในช่องหูขนาดเต็มใบหู และวิธีการแก้ปัญหาสัญญาณการป้อนกลับด้วยการออกแบบวงจรกรองปรับตัวได้แบบ LMS และอธิบายในรูปแบบโดเมนของความถี่ของสัญญาณการป้อนกลับ

1.3 วัตถุประสงค์

1.3.1 เพื่อวิเคราะห์หาพารามิเตอร์ที่เหมาะสมในการออกแบบ DLMS ให้มีอัตราการลู่เข้าที่ดีในขณะที่วงจรสามารถทำงานได้เร็วและใช้ทรัพยากรน้อย

1.3.2 เพื่อประยุกต์เทคนิคการทำไปป์ไลน์ละเอียดกับวงจรกรองปรับตัว DLMS ให้มีประสิทธิภาพในการกำจัดเสียงสะท้อนในระบบเครื่องช่วยฟัง

1.4 ขอบเขตของการวิจัย

1.4.1 ออกแบบวงจรกำจัดสัญญาณเสียงสะท้อนและสัญญาณรบกวนด้วย FPGA โดยใช้เทคนิคไปป์ไลน์ละเอียดเพื่อเพิ่มความเร็ว

1.4.2 เป็นวงจรกรองความถี่แบบปรับตัวได้ใช้อัลกอริธึม DLMS มีสัญญาณอ้างอิง

1.4.3 วงจรประมวลผลสัญญาณดิจิตอลขนาด 16 บิต

1.4.4 การทดสอบเปรียบเทียบกับวงจรกรองปรับตัวได้แบบ LMS โดยใช้สัญญาณเสียงมาตรฐาน

1.4.5 วงจรสามารถทำงานได้เร็วกว่าวงจรกรองปรับตัวได้แบบ LMS มากกว่า 5 เท่าตัว ในขณะที่ขนาดวงจรเพิ่มขึ้นไม่เกิน 50 % บนเทคโนโลยี FPGA

1.5 ขั้นตอนและวิธีดำเนินงานวิจัย

1.5.1 ศึกษาปัญหา สาเหตุ เสียงสะท้อนที่เกิดขึ้นในระบบเครื่องช่วยฟัง

1.5.2 ศึกษาวิธีการแก้ปัญหาเสียงสะท้อนด้วยวิธีการออกแบบวงจรกรองปรับตัวได้แบบ LMS และวงจรกรองปรับตัวได้แบบ DLMS

1.5.3 หาค่าพารามิเตอร์ที่เหมาะสมของจรกรองปรับตัวได้ด้วยโปรแกรม MATLAB ในการกำจัดสัญญาณเสียงสะท้อน

1.5.4 ออกแบบวงจรกรองปรับตัวได้แบบไม่ใช้ทรัพยากรร่วม โดยใช้โปรแกรม Xilinx AccelDSP ที่รันใน MATLAB ในการสร้างวงจรการกำจัดสัญญาณเสียงสะท้อน

1.5.5 ทดสอบสัญญาณที่ได้จากวงจรที่ออกแบบด้วยโปรแกรม AccelDSP เปรียบเทียบกับสัญญาณที่ได้จาก MATLAB เพื่อต้องการปรับปรุงแก้ไขพารามิเตอร์ให้เหมาะสมของวงจรใน AccelDSP ให้ถูกต้อง

1.5.6 ออกแบบวงจรกรองปรับตัวได้ DLMS แบบใช้ทรัพยากรร่วมโดยใช้ภาษา Verilog HDL

1.5.7 ทดสอบสัญญาณที่ได้จากวงจรกรองปรับตัวได้แบบ DLMS แบบใช้ทรัพยากรร่วมและทำการปรับปรุงแก้ไขให้เหมาะสม

1.5.8 สรุปผลการวิจัย ข้อเสนอแนะและจัดทำรูปเล่มวิทยานิพนธ์

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1ได้รูปแบบวงจรประมวลผลสัญญาณดิจิตอลด้วยเอฟพีจีเอในการกำจัดสัญญาณเสียงสะท้อนและสัญญาณรบกวนให้มีประสิทธิภาพ

1.6.2 สามารถนำไปประยุกต์ใช้งานเพื่อลดปัญหาเสียงสะท้อนที่เกิดขึ้นในเครื่องช่วยฟังดิจิตอล ระบบสื่อสารเสียงทางอินเตอร์เน็ต เป็นต้น

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องในบทนี้เป็นการนำเสนอเพื่อให้เชื่อมโยงกับงานวิจัยที่ทำ และได้รวบรวมเนื้อหาข้อมูลที่เกี่ยวข้อง เพื่อนำมาเป็นแนวทางประกอบการวิเคราะห์และแก้ปัญหาโจทย์ที่วิจัย สามารถนำไปประกอบเป็นข้อมูลความรู้ในการศึกษาหรือพัฒนางานวิจัยต่อไป

2.1 วงจรกรองความถี่แบบเอฟไอาร์ (Finite impulse response filter)

2.1.1 คุณสมบัติของวงกรองแบบ เอฟไอาร์

วงจรกรองแบบเอฟไอาร์ เป็นวงจรกรองความถี่แบบเชิงเลข (Digital filter) ซึ่งมีหลายรูปแบบ เช่น วงจรกรองความถี่ต่ำ (Low pass filter) วงจรกรองความถี่สูง (High pass filter), วงจรกรองแถบความถี่ผ่าน (Band pass filter) และวงจรกรองตัดแถบความถี่ (Band stop filter) มีประโยชน์ในการสื่อสาร วีดิโอ เสียงดนตรี งานทางด้านการแพทย์ เป็นต้น มีคุณสมบัติดังนี้

- เฟสเป็นเชิงเส้น (linear phase) เฟสของสัญญาณขาออกมีการเปลี่ยนแปลงที่เป็นเชิงเส้นกับความถี่ของสัญญาณขาเข้า ทำให้เกิดการผิดเพี้ยนทางเฟส มีประโยชน์ในการสื่อสาร วีดิโอ เสียงดนตรี งานทางด้านการแพทย์
- มีความเสถียร (Stability) เนื่องจากเป็นวงจรที่ไม่มีการป้อนกลับเหมือนวงจรกรองไอไออาร์
- สามารถสร้างวงจรด้วยชิพ DSP ที่มีการรองรับคำสั่งที่เป็นวงจรกรองแบบเอฟไอาร์หรือมีหน่วยประมวลผลแบบ MAC (Multiply Accumulator) อยู่ภายในโครงสร้างแบบตัวประมวลผลเฉพาะทางมีฮาร์ดแวร์รองรับหน่วยความจำแบบ (Circular Addressing)
- มีความถูกต้องและแม่นยำ ผลมาจากการควอนไทซ์ (Quantization) การปัดเศษ (Truncation) มีขอบเขตเลขทศนิยมที่จำกัดได้ เมื่อนำไปสร้างวงจร

สมการเอฟไอาร์ สามารถอธิบายได้ดังต่อไปนี้

$$y(n) = \sum_{i=0}^{N-1} \mathrm{w}_{\mathrm{i}}\, x(n-i) \qquad (2.1)$$

หรือ

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n)$$

โดยที่ $\mathbf{w}^T(n) = [w_0, w_1, ..., w_{N-1}]$ เป็นค่าสัมประสิทธิ์ (Coefficient) ของวงจรกรอง

$\mathbf{x}(n) = [x(n), x(n-1), ..., x(n-N+1)]^T$ เป็นอินพุตของวงจรกรอง

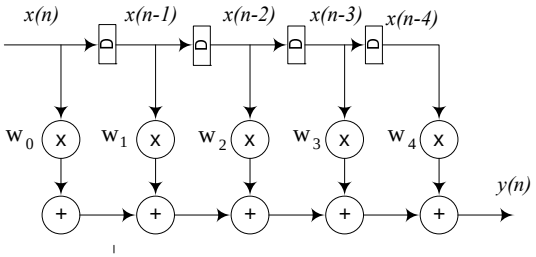
N = ความยาว (Length) ของวงจรกรองหรือเรียกว่าแท็ปดีเลย์ (Tapped delay)

2.1.2 การสร้างวงจรกรองเอฟไอาร์

วงจรกรองเอฟไอาร์สามารถทำการสร้างวงจร ได้ 3 แบบ คือ

2.1.2.1 การสร้างวงจรแบบโดยตรง (Direct form)

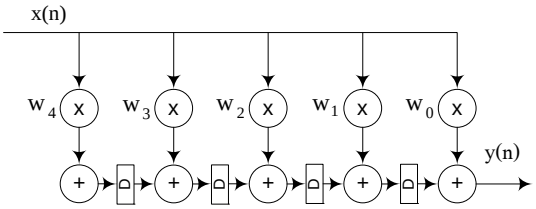
มีโครงสร้างดังภาพประกอบ 2-1 ซึ่งเป็นโครงสร้างวงจรกรองเอฟไอาร์ขนาด 5 แท็ป มีสัมประสิทธิ์ 5 ตัว $(w_0 - w_4)$



ภาพประกอบ 2-1 โครงสร้างวงจรกรองเอฟไอาร์แบบโดยตรง

2.1.2.2 การสร้างวงจรแบบทรานสโพส (Transpose Form)

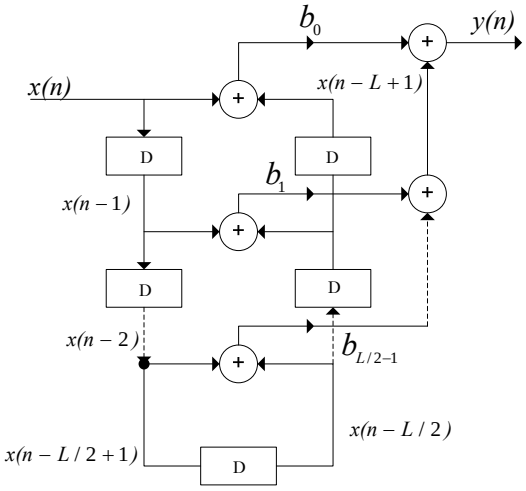
การสร้างวงจรแบบทรานสโพส มีการจัดวางดีเลย์ที่แตกต่างจากแบบโดยตรง คือ สัญญาณอินพุตที่ป้อนเข้ามาหรือ $x(n)$ มาคูณกับค่าสัมประสิทธิ์ทุกตัว และนำมาบวกกันของดีเลย์แต่ละตัว



ภาพประกอบ 2-2 โครงสร้างวงจรกรองเอฟไอาร์แบบทรานสโพส

2.1.2.3 การสร้างวงจรแบบสมมาตร (Symmetric Form)

การสร้างวงจรรูปแบบนี้ค่าสัมประสิทธิ์เป็นแบบสมมาตร สามารถใช้สัมประสิทธิ์ร่วมได้ เนื่องจากวงจรมีคู่สัมประสิทธิ์ที่เหมือนกัน ในการสร้างวงจรทำให้ลดการใช้ทรัพยากรของวงจร ดังภาพประกอบ 2-3

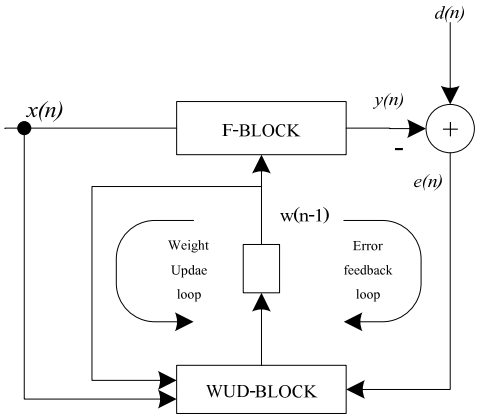


ภาพประกอบ 2-3 โครงสร้างวงจรกรองแบบสมมาตร

2.2.วงจรกรองปรับตัว (Adaptive filter)

2.2.1วงจรกรองปรับตัวได้แบบ LMS

ลักษณะวงจรกรองปรับตัวแบบ LMS มีโครงสร้างวงจรกรองเอฟไออาร์อยู่ภายในค่าสัมประสิทธิ์จะแปรเปลี่ยนตามเวลา เนื่องจากมีการป้อนกลับสัญญาณ สัญญาณป้อนกลับนี้จะนำไปคิดเพื่อปรับปรุงค่าสัมประสิทธิ์ใหม่ทุกรอบก่อนที่จะรับสัญญาณอินพุตถัดไป



ภาพประกอบ 2-4 โครงสร้างวงจรกรองปรับตัวได้แบบ LMS

สมการของวงจรกรองปรับตัวได้แบบ LMS สามารถอธิบายได้ดังต่อไปนี้

$y(n)$ คือ เอาต์พุทของวงจรกรองเอฟไอาร์

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \tag{2.2}$$

$e(n)$ คือ ค่าคลาดเคลื่อนของวงจรกรองปรับตัวได้

$$e(n) = d(n) - y(n) \tag{2.3}$$

$\mathbf{w}(n)$ คือ ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวได้

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n)\mathbf{x}(n) \tag{2.4}$$

วงจรกรองแบบปรับตัวได้ LMS [9] เป็นวงจรกรองซึ่งภายในโครงสร้างประกอบด้วย F-block (Forward path block)ดังภาพประกอบ 2-4 เป็นวงจรกรองเอฟไอาร์ จากสมการที่ (2.1)-(2.3) ค่าประมาณของสัญญาณที่ต้องการ $y(n)$ ได้ผลรวมจากการคูณค่าค่าสัมประสิทธิ์ กับสัญญาณอินพุต $\mathbf{x}(n)$ ที่ป้อนเข้ามาและ WUD-Block (Weight update block) เป็นวงจรปรับค่าสัมประสิทธิ์ โดยนำค่าประมาณความคลาดเคลื่อน $e(n)$ ได้จาก สัญญาณที่พึงประสงค์ (desired signal) $d(n)$ ลบด้วยค่าประมาณของสัญญาณที่ต้องการ $y(n)$ นำค่า $e(n)$ มาปรับค่าสัมประสิทธิ์ใหม่ก่อนที่จะรับสัญญาณอินพุตใหม่ ดังสมการที่ (2.4) ส่วน μ เป็นค่าอัตราการเรียนรู้ของระบบหรือ ค่าขนาดขั้น (Step Size) เนื่องจากมีผลต่อความเสถียร และความเร็วในการลู่เข้าซึ่งมีผลต่อความสามารถในการติดตามสัญญาณของตัวกรองด้วย เราสามารถหาค่า μ ได้ดังนี้

$$0 < \mu < \frac{1}{\lambda_{\max}} \tag{2.5}$$

โดยที่ค่า $\lambda_{\max}$ เป็นค่าไอเก็น (Eigen Value) ที่ใหญ่ที่สุดของเมตริกซ์ $\mathbf{R}$ และค่า $\lambda_{\max}$ สามารถหาได้จาก $\lambda_{\max} < tr[\mathbf{R}] = \sum\{Diagonal\ elements\ of\ \mathbf{R}\}$

ค่า $\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)]$ เป็นค่า Autocorrelation Matrix ขนาด $N \times N$ แสดงค่าความสัมพันธ์ของแต่ละค่าสัญญาณสุ่ม (sample) ของอินพุต $\mathbf{x}(n)$

2.2.2 วงจรกรองปรับตัวได้แบบ DLMS (Delayed Least Mean Square Error)

G. Long, F. Ling, and J. G. Proakis [1] ซึ่งเป็นผู้พัฒนาวงจรกรองแบบปรับตัวได้แบบ LMS แบบเดิมให้สามารถทำการสร้างวงจรแบบไปป์ไลน์ เนื่องจากโครงสร้างวงจรกรองปรับตัวได้แบบ LMS มีการป้อนกลับสัญญาณเพื่อนำมาปรับปรุงค่าสัมประสิทธิ์ ก่อนที่รับสัญญาณอินพุตในรอบถัดไป จึงไม่สามารถนำมาสร้างวงจรแบบไปป์ไลน์ได้ การออกแบบวงจรกรองปรับตัวได้แบบ DLMS นี้ โดยการแทรกการดีเลย์เข้าไปเป็นไปป์ไลน์เพื่อให้การทำงานของวงจรที่เร็วขึ้นแต่มีผลต่ออัตราการลู่เข้าของสัญญาณที่ช้ากว่าแบบวงจรกรองปรับตัวได้แบบ LMS

เอาต์พุตของวงจรกรองเอฟไออาร์

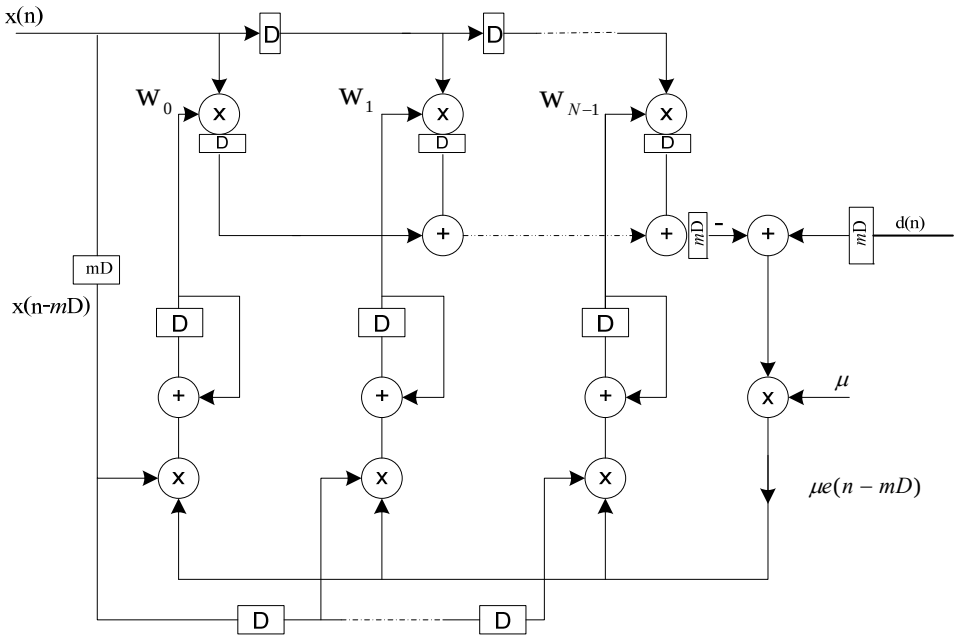
$$y(n-mD) = \mathbf{w}^T(n-mD)\mathbf{x}(n-mD) \tag{2.6}$$

$e(n)$ คือ ค่าคลาดเคลื่อนของวงจรกรองปรับตัวได้

$$e(n-mD) = d(n-mD) - y(n-mD) \tag{2.7}$$

$\mathbf{w}(n)$ คือ ค่าสัมประสิทธิ์ของวงจรกรองปรับตัวได้

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu e(n-mD)\mathbf{x}(n-mD) \tag{2.8}$$



ภาพประกอบ 2-5 โครงสร้างวงจรกรองปรับตัวได้แบบ DLMS

จากภาพประกอบ 2-5 จำนวนการดีเลย์ที่เกิดขึ้นในส่วนวงจรกรองเอฟไออาร์ มีค่าเท่ากับ mD นำมารวมกับค่า $e(n)$ คูณกับค่า μ เมื่อรวมจำนวนการดีเลย์ทั้งหมดมาถึงส่วนนี้มีค่าเท่ากับ mD คือจำนวนการดีเลย์ที่เกิดขึ้นทั้งหมดในวงจรกรองปรับตัวได้แบบ DLMS หรือการแทรกรีจิสเตอร์เข้าไปขั้นในหน่วยประมวลผลตัวถัดไปดังนั้นเมื่อนำไปคิดหาค่าสัมประสิทธิ์ของวงจรก็ต้องทำการดีเลย์ค่าสัญญาณอินพุตที่เท่ากับจำนวน mD เช่นกัน

วิธีการเลือกค่าอัตราการเรียนรู้ (μ) ที่เหมาะสมสำหรับวงจรกรองปรับตัวได้แบบ DLMS

เมื่อ

$$\mathbf{R} = E[\mathbf{x}(n)\mathbf{x}^T(n)] \tag{2.9}$$

$\mathbf{R}$ คือ Autocorrelation Matrix ขนาด $N \times N$

หรือ

$$\mathbf{R} = V\Lambda V^T \tag{2.10}$$

โดยที่

V คือ เมทริกซ์เชิงตั้งฉาก (orthogonal matrix)

Λ คือ ค่าไอเก็น(Eigen value) ของเมทริกซ์ $\mathbf{R}$

$$\Lambda = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N) \tag{2.11}$$

ค่าเฉลี่ยกำลังสองของ λ คือ

$$\lambda_{rms}^2 = \frac{1}{N}\sum_{i=0}^{N}\lambda_i^2 \tag{2.12}$$

ค่าเฉลี่ยของ λ คือ

$$\lambda_{av} = \frac{1}{N}\sum_{i=0}^{N}\lambda_i \quad = \quad \sigma^2 \tag{2.13}$$

ค่าของ α คือ อัตราของค่าเฉลี่ยกำลังสองต่อค่าเฉลี่ยของ λ

$$\alpha = \frac{\lambda^2_{rms}}{\lambda^2_{av}} \tag{2.14}$$

M คือค่าคงที่ได้จาก

$$M = N + 2 \tag{2.15}$$

ค่า μ ที่เหมาะสมสำหรับวงจรกรองปรับตัวได้ คือ

$$\mu_{opt} \approx \frac{1}{(\alpha M + 2D - 2)} \tag{2.16}$$

หรือ

$$\mu_{opt} \approx \frac{1}{(\alpha M + 2D - 2)\sigma^2} \tag{2.17}$$

ช่วงค่า μ ที่เหมาะสมสำหรับวงจรกรองปรับตัวได้ คือ

$$0 < \mu < \frac{2}{(\alpha M + 2D - 2)\sigma^2} \tag{2.18}$$

ในการกำหนดค่าอัตราการเรียนรู้ของระบบเมื่อทำการคำนวณเพื่อเลือกค่าที่เหมาะสม ในสมการที่ (2.16) และ (2.17) ทำให้อัตราการลู่เข้าของวงจรที่รวดเร็ว ซึ่งค่าที่คำนวณได้เมื่อนำไปใช้งานในบางครั้งอาจจะไม่ใช่ค่าที่เหมาะสมที่ดีที่สุด ดังนั้นควรใช้ค่าอัตราการเรียนรู้ที่อยู่ในช่วงขอบเขตในสมการที่ (2.18)

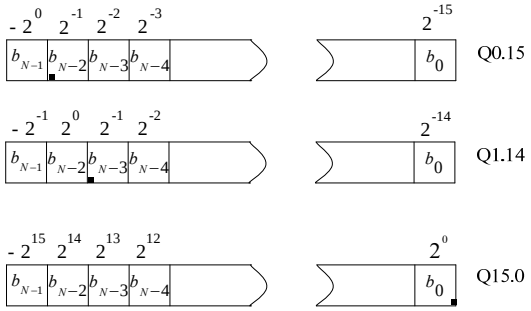
2.3 ระบบเลขทศนิยมคงตัว (Fixed-point Number System)

เลขทศนิยมประกอบด้วย 2 ส่วน คือ ส่วนที่เป็นเลขจำนวนเต็ม กับ จำนวนทศนิยม มีบิต MSB เป็นตัวกำหนดบิตเครื่องหมาย (Sign bit) สามารถเขียนอยู่ในรูป Q*m.n* โดยที่ m คือ จำนวนเลขจำนวนเต็มที่อยู่หน้าจุดทศนิยม และ n จำนวนเลขทศนิยม เช่น Q1.14 หรือ (Q14) หมายถึง เป็นเลขทศนิยมประกอบด้วย ทศนิยม 14 หลัก และเลขจำนวนเต็ม 1 หลัก ค่าของเลขทศนิยมนี้สามารถอธิบายในสมการที่ (2.19) $N = m + n + 1$, N คือ จำนวนบิตทั้งหมด

$$\begin{aligned} x &= (-b_{N-1}2^{N-1} + b_{N-2}2^{N-2} + b_{N-3}2^{N-3} + ... + b_0)2^{-n} \\ &= -b_{N-1}2^m + \sum_{k=0}^{N-2} b_k 2^{k-n} \end{aligned} \qquad (2.19)$$

$b_0 - b_{N-1}$ แทนค่าด้วยเลขฐานสอง

ซึ่งสามารถอธิบายวิธีการรูปแบบ Q*m.n* ในภาพประกอบ 2-6 สามารถนำมาคิดหาค่าตัวเลขเป็นคำตอบโดยใช้สมการที่ (2.19)



ภาพประกอบ 2-6 รูปแบบ Q*m.n* แบบต่างๆ

จากตารางที่ 2-1 แสดงค่าช่วงและความแม่นยำของเลขจำนวน 16 บิต ตามรูปแบบ Q จะเห็นได้ว่า Q0.15 มีความแม่นยำมากที่สุด ซึ่งมีช่วงตั้งแต่ -1 ถึง 0.999 ซึ่งเป็นช่วงที่แคบมาก แต่มีผลดี คือ การคูณเลขสองจำนวนเข้าด้วยกันจะไม่เกิดโอเวอร์โฟลว์ (Over flow) เพราะผลลัพธ์ที่ได้จะไม่มีทางเกินช่วง -1 ถึง 1 สำหรับ Q15.0 มีความแม่นยำน้อยที่สุด แต่มีช่วงค่าที่กว้างที่สุด คือ -32768 ถึง 32767

ตารางที่ 2-1 ค่า range และ precision ขนาด 16 บิต ตามรูปแบบ Q

Format	Largest positive value	Least negative value	Precision
Q0.15	0.999969482421875	-1	0.00003051757812
Q1.14	1.99993896484375	-2	0.00006103515625
Q2.13	3.9998779296875	-4	0.00012207031250
Q3.12	7.999755859375	-8	0.0002441406250
Q4.11	15.99951171875	-16	0.000488281250
Q5.10	31.9990234375	-32	0.00097656250
Q6.9	63.998046875	-64	0.0019531250
Q7.8	127.99609375	-128	0.003906250
Q8.7	255.9921875	-256	0.00781250
Q9.6	511.984375	-512	0.0156250
Q10.5	1023.96875	-1,024	0.031250
Q11.4	2047.9375	-2,048	0.06250
Q12.3	4095.875	-4,096	0.1250
Q13.2	8191.75	-8,192	0.250
Q14.1	16383.5	-16,384	0.50
Q15.0	32,767	-32,768	1.0

จากตารางที่ 2-2 แสดงช่วงค่าต่างๆของเลขจำนวน 16 บิต เมื่อแปลงเลขจำนวน 16 บิตมาเป็นเลขฐานสิบสามารถเขียนได้ดังนี้ เช่น รูปแบบ Q0.15 มีค่าช่วงอยู่ที่ -1 ถึง 0.999 ค่าที่ต่ำสุด (-1) ของแต่ละรูปแบบ Q0.15 ถูกแทนด้วยเลขฐาน 16 คือ (8000h) ในขณะที่ค่ามากที่สุด (1) ถูกแทนด้วยเลขฐาน 16 คือ (7FFFh) สำหรับรูปแบบอื่นๆ ค่าเลขฐาน 16 ก็อยู่ในช่วงนี้แต่จะใช้สเกลที่แตกต่างกันออกไปในการแทนค่า

ตารางที่ 2-2 แสดงช่วงค่าต่างๆของเลขฐานสิบของรูปแบบ16 บิตเมื่อถูกแทนด้วยเลข 16 บิต

Format	Scaling factor 2^n	Range in Hex (Decimal value)
Q0.15	$2^{15} = 32{,}768$	7FFFh(0.99) → 8000h(-1)
Q1.14	$2^{14} = 16{,}384$	7FFFh(1.99) → 8000h(-2)
Q2.13	$2^{13} = 8{,}192$	7FFFh(3.99) → 8000h(-4)
Q3.12	$2^{12} = 4{,}096$	7FFFh(7.99) → 8000h(-8)
Q4.11	$2^{11} = 2{,}048$	7FFFh(15.99) → 8000h(-16)
Q5.10	$2^{10} = 1{,}024$	7FFFh(31.99) → 8000h(-32)
Q6.9	$2^9 = 512$	7FFFh(63.99) → 8000h(-64)
Q7.8	$2^8 = 256$	7FFFh(127.99) → 8000h(-128)
Q8.7	$2^7 = 127$	7FFFh(255.99) → 8000h(-256)
Q9.6	$2^6 = 64$	7FFFh(511.99) → 8000h(-512)
Q10.5	$2^5 = 32$	7FFFh(1,023.99) → 8000h(-1,024)
Q11.4	$2^4 = 16$	7FFFh(2,047.99) → 8000h(-2,048)
Q12.3	$2^3 = 8$	7FFFh(4,095.99) → 8000h(-4,096)
Q13.2	$2^2 = 4$	7FFFh(8,191.99) → 8000h(-8,192)
Q14.1	$2^1 = 2$	7FFFh(16,383.99) → 8000h(-16,384)
Q15.0	$2^0 = 1$	7FFFh(32,767.99) → 8000h(-32,768)

2.3.1 การบวกเลขทศนิยม

สำหรับการบวกเลขทศนิยมที่นำมาใช้ มี 2 แบบ คือ

2.3.1.1 การบวกด้วยจำนวนบิตที่มีรูปแบบเหมือนกัน เช่น Q0.15 บวกด้วย Q0.15 ผลลัพธ์ที่ได้เท่ากับ Q0.15 มีจำนวนบิตที่เท่ากัน จะมีโอเวอร์โฟลว์ เกิดขึ้นด้วย 1 บิต (c) ซึ่งหากจำนวนทั้งสองมาบวกกันมีค่ามากกว่ารูปแบบ Q0.15 แล้วเกิดโอเวอร์โฟลว์ก็ต้องทำการตรวจเช็คอีกครั้ง เพื่อให้ได้ค่าถูกต้องมากยิ่งขึ้น

$$Qm.n = Qm.n + Qm.n \qquad (2.20)$$

$$\begin{array}{r} |s.|x|x|x|x|x|x|x|x|x|x|x|x|x|x|x| \quad + \\ |s.|x|x|x|x|x|x|x|x|x|x|x|x|x|x|x| \\ \hline C|s.|x|x|x|x|x|x|x|x|x|x|x|x|x|x|x| \end{array}$$

ภาพประกอบ 2-7 การบวกเลขทศนิยมที่มีจำนวนบิตเท่ากัน

จากภาพประกอบ 2-7 เป็นการบวกเลขทศนิยมโดยใช้รูปแบบ Q0.15 สามารถอธิบายตัวอย่างได้ดังนี้

เลขฐานสอง 0.100 (0.5) + 0.011(0.375) = 0.111(0.875) ไม่เกิดโอเวอร์โฟลว์

เลข Q0.15 4000h + 3000h = 7000h

เลขฐานสอง 0.101 (0.625) + 0.011(0.375) = 1.000(-1) โอเวอร์โฟลว์

เลข Q0.15 5000h + 3000h = 8000h

2.3.1.2 การบวกเลขจำนวนทศนิยมรูปแบบไม่เหมือนกัน เมื่อนำเลขทศนิยมที่มีตำแหน่งเลขไม่เท่ากันมาบวกกัน ต้องทำให้ทศนิยมตัวที่น้อยกว่าให้ตรงกับตัวมาก โดยทำการใส่บิตเครื่องหมายเพิ่มในส่วนของตัวเลขทศนิยมตัวที่น้อยกว่า เช่น การบวกเลข Q0.15 กับ Q1.14 ผลลัพธ์ได้ Q1.14 ต้องใส่บิตเครื่องหมายในค่า Q0.15 อีก 1 ตำแหน่ง ซึ่งหากบิตเครื่องหมายเป็น 0 ก็ใส่ 0 เพิ่มเข้าไปที่บิต MSB เช่นเดียวกันหากบิตเครื่องหมายเป็น 1 หรือมีค่าเป็นลบก็ให้ใส่ 1 ที่ตำแหน่ง MSB บิต เมื่อทำการใส่บิตเครื่องหมายของเลข Q0.15 ลงไป ก่อนที่จะนำไปบวก

$$\begin{array}{rl} |s.|x|x|x|x|x|x|x|x|x|x|x|x|x|x|x| & \text{Q0.15} \\ |s|x.|x|x|x|x|x|x|x|x|x|x|x|x|x|x| + & \text{Q1.14} \\ \hline |s|x.|x|x|x|x|x|x|x|x|x|x|x|x|x|x| & \text{Q1.14} \end{array}$$

(a)

$$\begin{array}{r} |s|s.|x|x|x|x|x|x|x|x|x|x|x|x|x|x|x| \\ |s|x.|x|x|x|x|x|x|x|x|x|x|x|x|x|x| + \\ \hline |s|x.|x|x|x|x|x|x|x|x|x|x|x|x|x|x| \end{array}$$

(b)

ภาพประกอบ 2-8 การบวกเลขทศนิยมรูปแบบไม่เหมือนกัน

จากภาพประกอบ 2-8 (a) การบวกเลขจำนวนทศนิยมรูปแบบไม่เหมือนกัน ระหว่าง Q0.15 กับ Q1.14 นำมาบวกกันต้องปรับหลักทศนิยมตัวน้อยให้ตรงกับตัวมากโดยการใส่บิตเครื่องหมายเพิ่มเติมเข้าไปในส่วนของเลขทศนิยมตัวที่น้อย คือ Q0.15 อีก 1 ตัว ดังภาพประกอบ 2-8 (b) การบวกเลขในรูปแบบนี้สามารถอธิบายรายละเอียดดังตัวอย่างต่อไปนี้

เลขฐานสอง 0.100 (0.5) + 0.011(0.375) = 0.111(0.875)

เลขฐานสิบหก 4000h (Q0.15) + 1800h (Q1.14)

ใส่บิตเครื่องหมาย 4000h (Q0.15) $\rightarrow$ 2000h (Q1.14) ก่อนนำไปบวก

เลขฐานสิบหก 2000h (Q1.14) + 1800h (Q1.14) = 3800h(0.875)

เลขฐานสอง 0.101 (0.625) + 0.011(0.375) = 1.000 (1)

เลขฐานสิบหก 5000h(Q0.15) + 1800h(Q1.14)

ใส่บิตเครื่องหมาย จาก 5000h(Q0.15) $\rightarrow$ 2800h(Q1.14) + 1800h(Q1.14) = 4000h (1)

2.3.2 การคูณเลขทศนิยม

เมื่อนำเลขทศนิยมสองจำนวนมาทำการคูณกันผลลัพธ์ทำให้ได้เลขที่มีจำนวนบิตขนาดเท่ากับขนาดของเลขทั้งสองจำนวนรวมกัน คือ ขนาดของบิตทั้งหมด ตำแหน่งของทศนิยม

$$\mathbf{Q}m.n = \mathbf{Q}m1.n1 \times \mathbf{Q}m2.n2 \qquad (2.21)$$

$$\mathrm{Q}m.n = \mathrm{Q}(m1+m2).(n1+n2) \qquad (2.22)$$

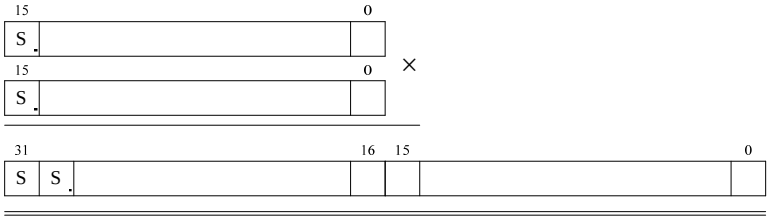
m คือ จำนวนบิตของเลขจำนวนเต็ม

n คือ จำนวนบิตของเลขทศนิยม

$m1, m2$ คือ จำนวนบิตของเลขจำนวนเต็ม

$n1, n2$ คือ จำนวนบิตของเลขทศนิยม

ตัวอย่าง เลข Q0.15 คูณ Q0.15 เท่ากับ Q0.30 มีขนาด 32 บิต จำนวนบิตเครื่องหมายเกิดขึ้น 2 บิต



ภาพประกอบ 2-9 การคูณเลขสองจำนวนตามรูปแบบเลขจำนวนเต็ม

จากภาพประกอบ 2-9 เป็นการคูณเลขระหว่าง Q0.15 กับ Q0.15 ผลลัพธ์ที่ได้จะเกิดบิตเครื่องหมายขึ้น 2 ตัว คือตำแหน่งบิตที่ 31 และบิตที่ 30 ส่วนข้อมูลบิตที่ 29 ถึงบิตที่ 0 เป็นผลลัพธ์ให้นำมาทำการปัดเศษ ตามตัวอย่างด้านล่างเป็นการปัดเศษให้เหลือเพียง 16 บิต จากผลลัพธ์ข้อมูลขนาด 32 บิต

เลขฐานสิบ (0.5) * (0.375) = 0.1875

เลข Q0.15 4000h * 3000h = 0C000000h

ปัดเศษ 0C000000h = 0000 1100 0000 0000 0000 0000 0000 0000

0.1875.1 = 1800h (Q0.15)

บิตที่ 31 เป็น บิตเครื่องหมาย นำมาใส่ที่ตำแหน่ง MSB ของรูปแบบ 16 บิต

บิตที่ 30 และ 29 รวมกันได้ 1 บิต (เอามาตรวจสอบเงื่อนไข)

ถ้า บิตเครื่องหมายเป็นจำนวนบวก (0) ให้นำ AND กัน

ถ้า บิตเครื่องหมายเป็นจำนวนลบ (1) ให้นำ OR กัน

บิตที่ 28 ดึงออกมา

ข้อมูลตั้งแต่บิต 31 ถึง บิตที่ 28 นี้ เมื่อทำการประมวลผลดังกล่าวที่อธิบายข้างต้นแล้ว ได้ข้อมูลออกมา 1 ชุด ซึ่งมีขนาด 4 บิต

ข้อมูลตั้งแต่บิต 27 ถึง บิตที่ 23 นี้ มารวมกันได้ข้อมูลออกมา 1 ชุด ซึ่งมีขนาด 4 บิต

ข้อมูลตั้งแต่บิต 22 ถึง บิตที่ 19 นี้ มารวมกัน ได้ข้อมูลออกมา 1 ชุด ซึ่งมีขนาด 4 บิต

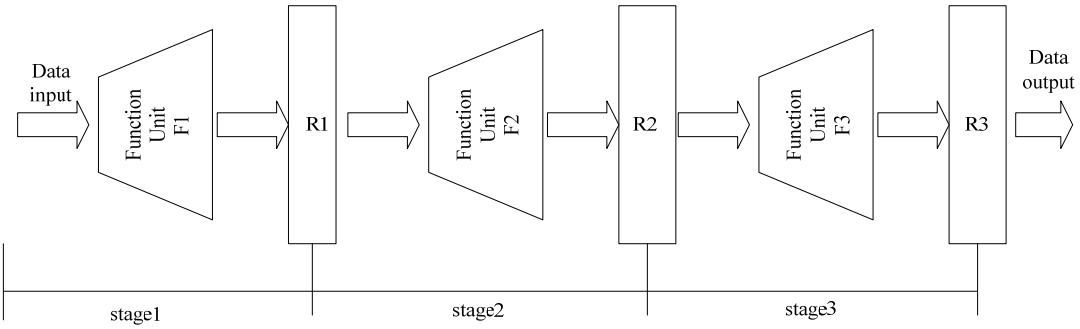
ข้อมูลตั้งแต่บิต 18 ถึง บิตที่ 15 นี้ มารวมกัน ได้ข้อมูลออกมา 1 ชุด ซึ่งมีขนาด 4 บิต

ส่วนบิตที่ 14 ถึง บิตที่ 0 ให้ทำการตัดทิ้งไปได้เลย

นำข้อมูลทั้ง 4 ชุดมาทำการรวมกัน ได้เป็นข้อมูลขนาด 16 บิต ซึ่งมีรูปแบบเป็นแบบ Q0.15

2.4 ระบบไปป์ไลน์

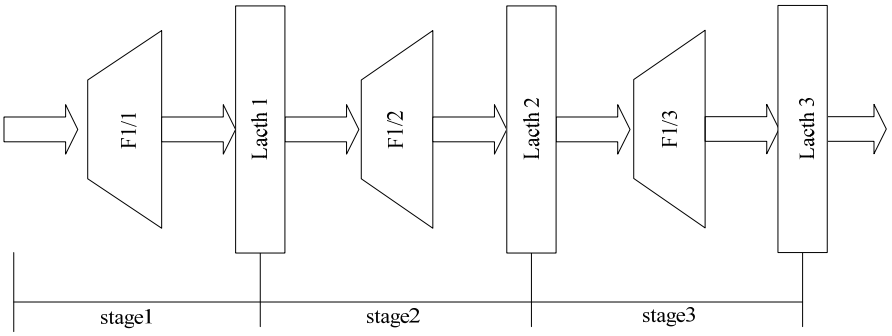
การทำไปป์ไลน์ (Pipelining) เป็นการแบ่งการทำงานของวงจรทั้งระบบออกเป็นส่วนๆ ทำงานต่อๆกันดังแสดงในภาพประกอบ 2-10 แต่ละส่วนเรียกว่า สเตจ (Stage) ประกอบด้วยฟังก์ชันประมวลผล (Function unit) และรีจิสเตอร์ (Register) วงจรที่ออกแบบเป็นไปป์ไลน์สามารถทำงานได้เร็ว เนื่องจากแต่ละสเตจประมวลผลไปพร้อมๆกัน อย่างต่อเนื่อง กล่าวคือ เมื่อ สเตจที่ 1 ประมวลผลเสร็จ และเก็บข้อมูลไว้ในรีจิสเตอร์ สเตจที่ 2 รับข้อมูลดังกล่าวไปประมวลผลต่อและพร้อมกันนี้ สเตจที่ 1 ก็จะรับข้อมูลใหม่มาประมวลผลไปพร้อมกัน



ภาพประกอบ 2-10 วงจรไปป์ไลน์ [10]

2.4.1 การทำไปป์ไลน์ละเอียด (Fine-Grained pipeline)

การทำไปป์ไลน์ละเอียดหรือการทำไมโครไปป์ไลน์ เป็นเทคนิคเดียวกันกับการทำไปป์ไลน์แต่เป็นการทำไปป์ไลน์ตรงที่ตัววงจรฟังก์ชันประมวลผลมักมีการใช้วงจรแลทช์ (Latches) คั่นระหว่างสเตจ เนื่องจากมีขนาดที่เล็กกว่ารีจิสเตอร์ดังแสดงในภาพประกอบ 2-11

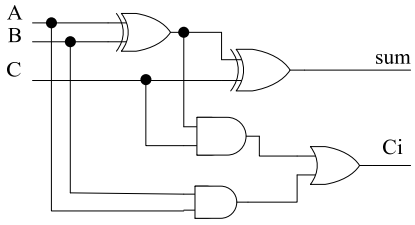


ภาพประกอบ 2-11 วงจรไปป์ไลน์ละเอียด [10]

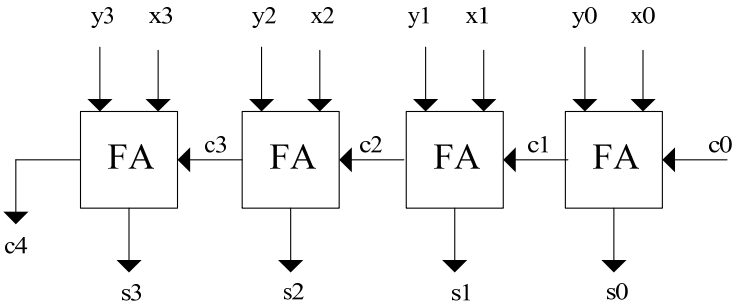
2.4.2 วงจรบวก

2.4.2.1 วงจรบวกแบบตัวทดเลื่อน (Ripple Carry Adder)

เป็นวงจรบวกแบบพื้นฐานเลขฐานสอง โดยอาศัยวงจร Full adder ต่อกันแบบบิตต่อบิต ดังภาพประกอบ 2-12 และเมื่อมีจำนวนบิตที่เพิ่มขึ้น การต่อวงจรดังภาพประกอบ 2-13 ซึ่งการบวกแบบนี้ทำให้เกิดช่วงเวลาที่ดีเลย์มากหากมีจำนวนบิตที่เพิ่มเข้ามา การบวกแบบนี้กระทำเสร็จภายในหนึ่งช่วงเวลา วงจรบวกแบบนี้การทำงานไม่เป็นแบบไปป์ไลน์



ภาพประกอบ 2- 12 วงจรบวกแบบ Full adder

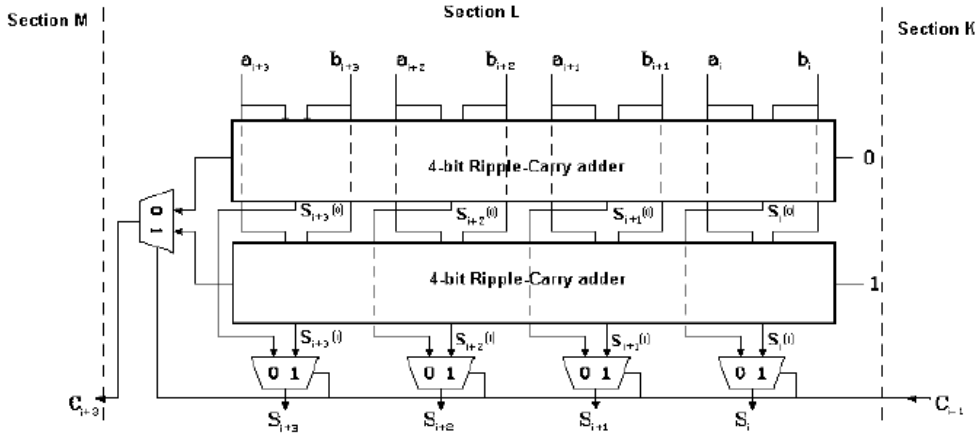


ภาพประกอบ 2-13 วงจรบวกแบบ Full adder แบบตัวทดเลื่อน ขนาด 4 บิต

2.4.2.2 วงจรบวกแบบเลือกตัวทด (Carry Select Adder)

เป็นวงจรบวกแบบไปป์ไลน์ โดยอาศัยวงจรบวกแบบ Full adder เป็นพื้นฐาน เพื่อลดค่าดีเลย์ในวงจรบวกแบบ Ripple Carry Adder ลงเนื่องจากเป็นวงจรที่ต้องรอค่าตัวทด (carry) ก่อนที่จะทำการบวกในบิตถัดไป วงจรบวกแบบ Carry Select Adder นี้ได้ทำการบวกไว้ก่อนล่วงหน้าทั้ง 2 แบบ คือ แบบไม่คิดตัวทด (c = 0) กับแบบคิดตัวทด (c = 1) และใช้วงจรมัลติเพล็กเซอร์ทำการเลือกบิตตัวทดอีกครั้ง ดังภาพประกอบ 2-14 ทำให้ขนาดของวงจรเพิ่มขึ้นมากกว่า 2 เท่าตัวเมื่อรวมกับค่าวงจรมัลติเพล็กเซอร์ด้วย การทำงานของวงจรเป็นแบบไปป์ไลน์ สามารถรับ

สัญญาณอินพุตได้ต่อเนื่องในขณะการบวกยังไม่สิ้นสุดกระบวนการ จำนวนขั้นตอนของไปป์ไลน์มีค่าเท่ากับ $\log_2 N$ โดยที่ N คือจำนวนบิต ที่นำมาบวกด้วยวงจรบวกแบบ Carry Select Adder



ภาพประกอบ 2-14 โครงสร้างของวงจรบวกแบบ Carry Select Adder ขนาด 4 บิต [11]

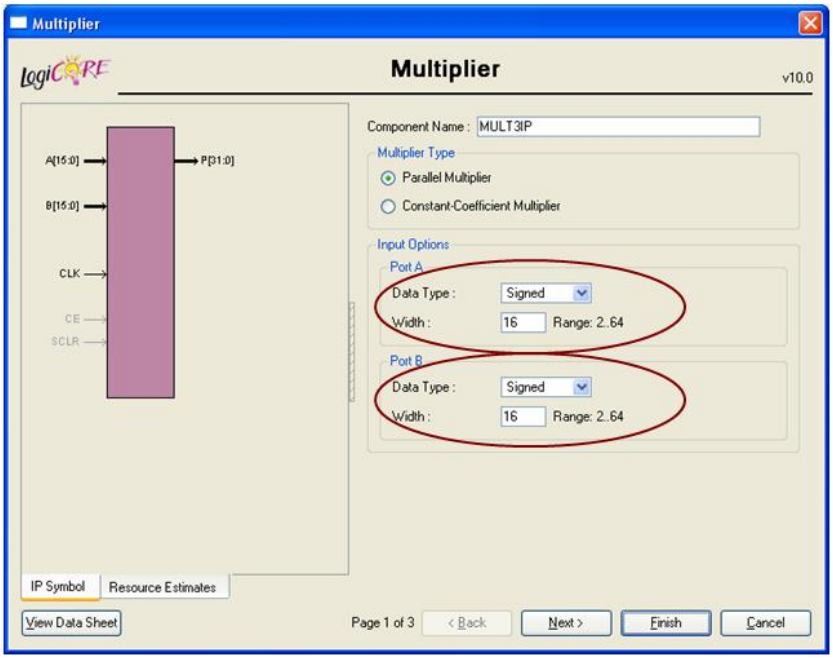
ตารางที่ 2-3 สรุปการทำงานของวงจรบวกแบบต่างๆ ขนาด 16 บิต [11]

Adder types	Delay(ns)	Area
Carry Ripple Adder	72.1	160
Carry Select Adder	26	356
Manchester Adder	27.58	256
Kogge-Stone prefix Adder	25.59	429

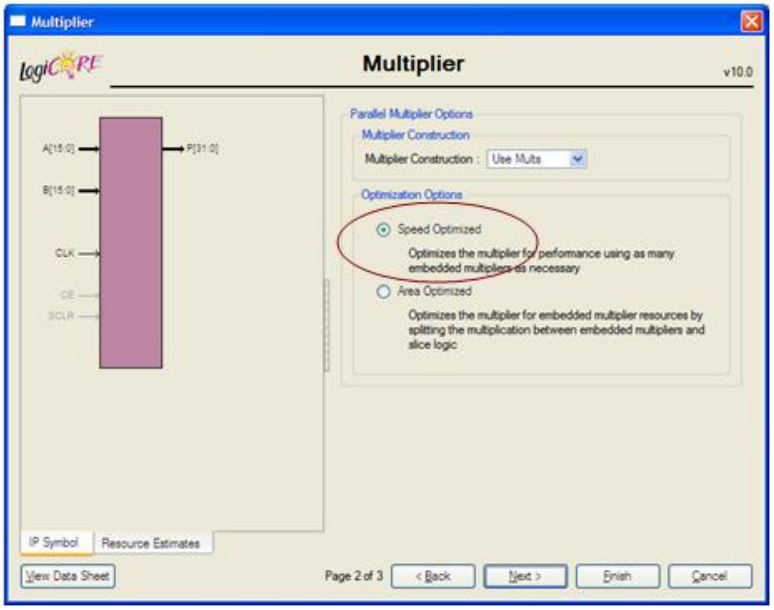
ความเร็วของการทำงานของวงบวกที่แตกต่างกัน ในตารางที่ 2-3 โดยที่วงจรบวกแบบ Ripple Carry Adder เป็นวงจรแบบจัดเชิงหมู่ (Combination) การทำงานเพียงแค่ 1 รอบสัญญาณนาฬิกาเท่านั้น ส่วนวงจรบวกแบบ Carry Select Adder, Manchester และ Kogge-stone Prefix เป็นวงจรบวกแบบไปป์ไลน์

2.4.3 วงจรคูณแบบไปป์ไลน์

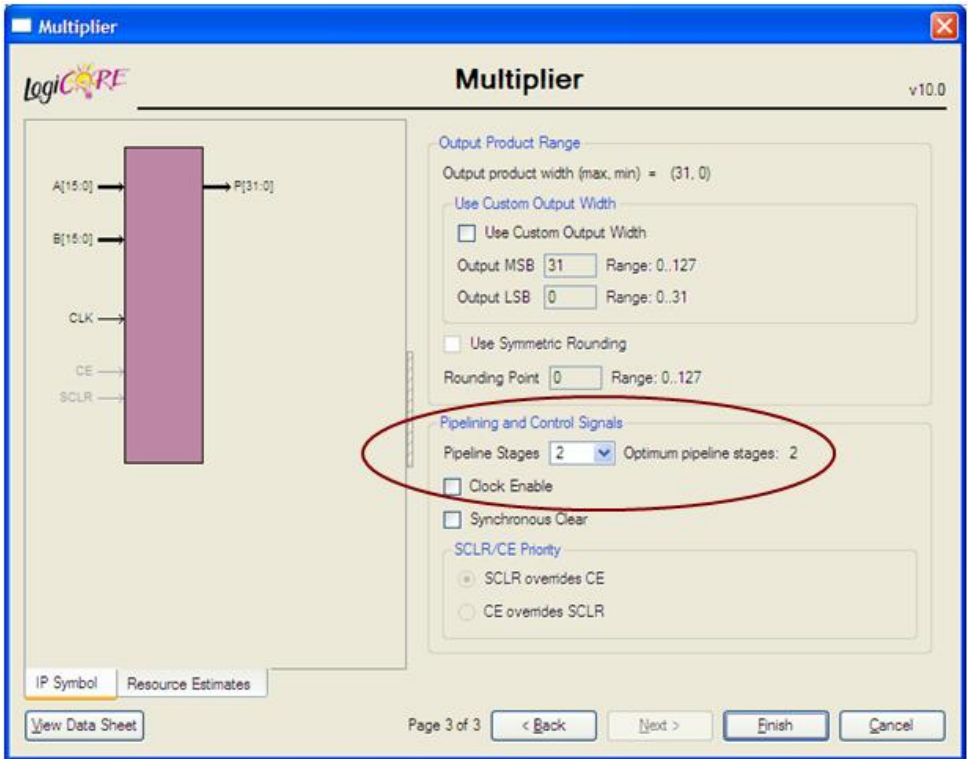
สถาปัตยกรรมของ Xilinx ตั้งแต่ Spartan-3 – Virtex-5 มีวงจรคูณที่เป็นแบบไปป์ไลน์โดยการออกแบบโดยใช้ IP-core generator สามารถสร้างขึ้นจากเครื่องมือที่ให้มา และกำหนดพารามิเตอร์แบบต่างๆ ในเครื่องมือ(จำนวนบิตข้อมูล จำนวนไปป์ไลน์กี่ขั้นที่เหมาะสม) ดังแสดงในภาพประกอบ 2-15 เป็นการตั้งค่าวงจรคูณมีอินพุตจำนวน 16 บิตเป็นแบบคิดเครื่องหมาย ส่วนในภาพประกอบ 2-16 เป็นการกำหนดวงจรคูณในการเลือกความเร็วหรือการจัดการพื้นที่ และภาพประกอบ 2-17 เป็นการกำหนดจำนวนไปป์ไลน์ของวงจรที่เหมาะสมว่าจำนวนกี่สเตจ



ภาพประกอบ 2-15 วงจรคูณที่ถูกสร้างด้วย IP-core generator



ภาพประกอบ 2-16 การกำหนดค่าวงจรคูณในการเลือกความเร็วหรือพื้นที่

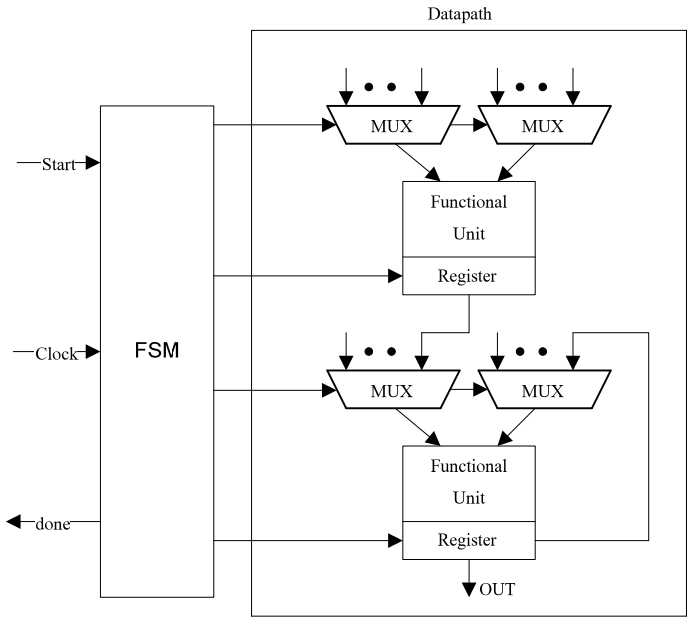


ภาพประกอบ 2-17 การกำหนดค่าวงจรคูณในการเลือกจำนวนไปป์ไลน์

2.5 รูปแบบวงจรและการวิเคราะห์

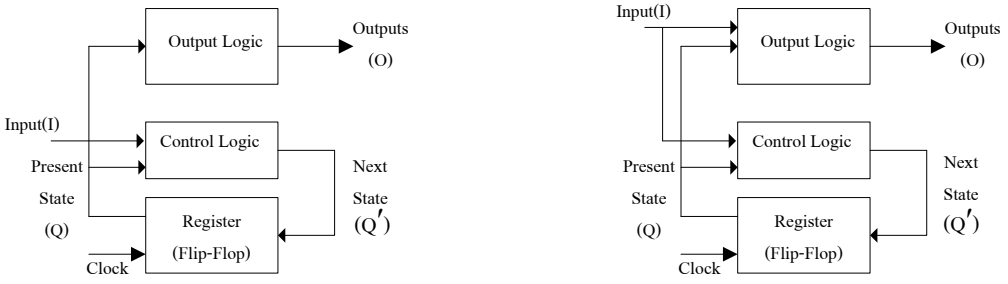
การออกแบบวงจรประมวลผลสัญญาณดิจิตอลแบ่งออกเป็น 2 ส่วนหลัก คือ การออกแบบวงจรดาต้าพาทและการออกแบบวงจรควบคุม ดังภาพประกอบ 2-18 วงจรดาต้าพาททำหน้าที่ประมวลสัญญาณที่ข้อมูลที่เข้ามาและส่งออกไปส่วนที่เกี่ยวข้อง โมดูลต่างๆ ในวงจรดาต้าพาทประกอบด้วย วงจรฟังก์ชัน (คณิตศาสตร์ บวก/ลบ คูณ และลอจิก เป็นต้น) สำหรับการคำนวณ รีจิสเตอร์สำหรับเก็บข้อมูล มัลติเพล็กเซอร์สำหรับในกรณีที่มีการใช้วงจรฟังก์ชันร่วมกันของหลายโอเปอร์เรเตอร์ เป็นต้น

ส่วนวงจรควบคุมทำหน้าที่ควบคุมการทำงานของวงจรดาต้าพาทให้เป็นไปตามลำดับที่ถูกต้องโดยอาจจะมีสัญญาณ start และ done สำหรับเชื่อมต่อวงจรอื่นเพื่อควบคุมให้วงจรเริ่มทำงานและเพื่อแจ้งการจบการทำงานตามลำดับ และวงจรถูกกำหนดจังหวะการทำงานด้วยสัญญาณนาฬิกา (Clock) เดียวทั้งระบบ



ภาพประกอบ 2-18 รูปแบบของวงจรที่ใช้ออกแบบวงจรประมวลผลสัญญาณดิจิตอล [13]

Finite State Machine (FSM) ถูกใช้เป็นวงจรควบคุม เนื่องจากสังเคราะห์ได้ง่าย รูปแบบไม่ซับซ้อนและเป็นที่นิยมใช้เป็นวงจรควบคุมในการออกแบบวงจรรวมดิจิตอลโดยทั่วไป โดย FSM ที่นิยมใช้มี 2 รูปแบบด้วยกันคือ Moore machine และ Mealy machine ดังโครงสร้างของวงจรในภาพประกอบ 2-19



(a) Moore machine (b) Mealy machine

ภาพประกอบ 2-19 โครงสร้างวงจร Moore machine และ Mealy machine [13]

โครงสร้างวงจรของ FSM ทั้งสองแบบประกอยด้วยวงจร 3 ส่วนดังนี้

- วงจรลอจิกขาออก (Output logic) ทำหน้าที่สร้างสัญญาณเอาต์พุต ซึ่งเป็นสัญญาณที่ใช้ในการควบคุมและติดต่อกับวงจรส่วนอื่น

- วงจรลอจิกควบคุม (Control logic) ทำหน้าที่ควบคุมวงจรโดยการรับสัญญาณอินพุตและสเตทปัจจุบันมาประมวลผลเพื่อตัดสินใจว่าสเตทถัดไปควรเป็นอะไร

- รีจิสเตอร์ (Register) ทำหน้าที่จำสเตทปัจจุบันไว้ รีจิสเตอร์นี้สามารถถูกสร้างจากฟลิบ-ฟลอบ จำนวนของฟลิบ-ฟลอบขึ้นกับจำนวนสเตททั้งหมดของวงจร

สัญญาณนาฬิกา (Clock) ทำหน้าที่สำคัญคือ กำหนดจังหวะการทำงานของวงจร FSM ค่าของสเตทที่ได้จากวงจรลอจิกจะถูกบันทึกในรีจิสเตอร์ทุกครั้งที่ขอบสัญญาณนาฬิกา ถ้าฟลิบ-ฟลอบที่ใช้เป็นชนิดที่ทำงานที่ขอบขาขึ้น (Positive edge-triggered) ค่าของสเตทดังกล่าวจะถูกบันทึกไว้ในรีจิสเตอร์ภายในเวลาไม่กี่นาโนวินาทีหลังจากที่ขอบขาขึ้นของสัญญาณนาฬิกา วงจรนี้ถูกเรียกว่า วงจรซิงโคนัส (Synchronous circuit) การทำงานของวงจรต้องซิงโครไนซ์กับสัญญาณนาฬิกาเดียวทั้งระบบ

ความแตกต่างของ Moore machine และ Mealy machine อยู่ที่วงจรลอจิกขาออก โดยในกรณีของ Moore machine ดังแสดงในภาพประกอบ 2-19(a) และสมการที่ (2.23) สัญญาณเอาต์พุตเป็นฟังก์ชันของสเตทปัจจุบันเพียงอย่างเดียว การเปลี่ยนแปลงของสัญญาณอินพุตไม่มีผลต่อสัญญาณเอาต์พุต ข้อดีคือเอาต์พุตจะมีค่าคงที่หรือเสถียรตลอดสเตท เอาต์พุตจะเปลี่ยนแปลงก็ต่อเมื่อมีการเปลี่ยนสเตทใหม่

Moore machine

$$O = f(Q)$$
$$Q' = f(Q,I) \quad (2.23)$$

โดยที่ O คือ สัญญาณเอาต์พุต

I คือ สัญญาณอินพุต

Q คือ สัญญาณสเตทปัจจุบัน และ

Q' คือ สัญญาณสเตทถัดไป

ในทางตรงกันข้ามกรณีของ Mealy machine ดังแสดงในภาพประกอบ 2-19(b) และสมการที่ (2.24) สัญญาณเอาต์พุตเป็นฟังก์ชันของสเตทปัจจุบันและอินพุต เอาต์พุตสามารถเปลี่ยนแปลงค่าได้หากมีการเปลี่ยนแปลงของอินพุต ไม่จำเป็นต้องรอเปลี่ยนในสเตทถัดไป ข้อดีคือในงานบางลักษณะจำนวนสเตทของ Mealy machine จะน้อยกว่าจำนวนสเตทของ Moore machine

Mealy machine

$$O = f(Q,I)$$
$$Q' = f(Q,I) \quad (2.24)$$

ในการออกแบบวงจรประมวลผลสัญญาณดิจิตอลควรใช้ Moore machine เนื่องจากนำไปประยุกต์ใช้ในการควบคุมการทำงานของวงจรดาต้าพาท (Datapath) สัญญาณเอาต์พุตที่ไปควบคุมการทำงานของวงจรดาต้าพาทต้องคงที่ตลอดจนกว่าวงจรดาต้าพาทส่วนที่เกี่ยวข้องทำงานเสร็จ

2.6 เครื่องช่วยฟัง (Hearing Aids)

เครื่องช่วยฟังเป็นอุปกรณ์ช่วยการได้ยินสำหรับบุคคลที่มีความบกพร่องทางการได้ยิน เพื่อการสื่อสารที่มีประสิทธิภาพมากยิ่งขึ้น เป็นประโยชน์ทั้งสองฝ่าย

2.6.1 ประเภทของเครื่องช่วยฟัง

เครื่องช่วยฟังสามารถแบ่งออกเป็น 3 ประเภท ด้วยกัน ซึ่งประกอบด้วย เครื่องช่วยฟังแบบอนาลอก เครื่องช่วยฟังแบบกึ่งดิจิตอล และเครื่องช่วยฟังแบบดิจิตอล การพัฒนาเครื่องช่วยฟังเริ่มจากเครื่องช่วยฟังแบบอนาลอก ลักษณะของวงจรเป็นวงจรขยายสัญญาณเสียงทั่วไปเพื่อให้สามารถรับฟังได้ ไม่มีระบบการจัดการทางด้านเสียงเข้ามาช่วยหรือสัญญาณรบกวนต่อมามีการพัฒนาทางด้านดิจิตอลเข้ามาช่วยทำให้เครื่องช่วยฟังแบบดิจิตอลเข้ามาแทนที่ เนื่องจากมีข้อดีกว่าเครื่องช่วยฟังแบบอนาลอก มีการปรับปรุงทางสัญญาณเสียง เช่น มีวงจรชดเชยความถี่ของ

สัญญาณทำให้ผู้สูญเสียการได้ยินสามารถรับฟังได้ วงจรป้องกันเสียงสะท้อน และวงจรกำจัดสัญญาณรบกวนของระบบ สัญญาณที่ได้จากเครื่องช่วยฟังแบบดิจิตอลจึงมีประสิทธิภาพมาก

2.6.2 ส่วนประกอบของเครื่องช่วยฟัง

เครื่องช่วยฟังประกอบด้วยส่วนสำคัญต่างๆ ดังต่อไปนี้ [15] ไมโครโฟน วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล วงจรขยายเสียง วงจรประมวลผลสัญญาณดิจิตอล วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกและลำโพง

- ไมโครโฟน (Microphone) มีหน้าที่รับเสียงและเปลี่ยนพลังงานเสียง ให้เป็นพลังงานไฟฟ้า (Electrical energy) จากนั้นจะส่งต่อไปยังตัวขยายเสียง เพื่อขยายพลังงานไฟฟ้าให้มากขึ้น

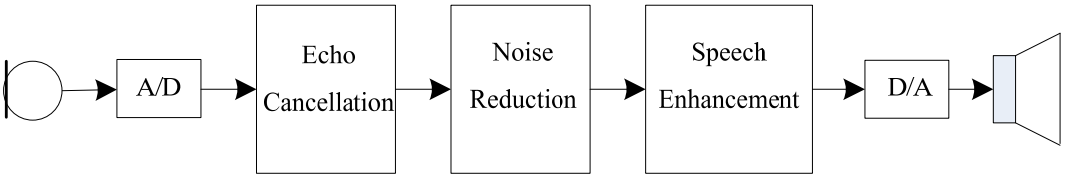
- วงจรแปลงสัญญาณเสียงเป็นสัญญาณดิจิตอล (Analog to Digital Conversion) เมื่อสัญญาณเสียงผ่านไมโครโฟนมาแล้วนำมาแปลงเป็นสัญญาณดิจิตอล เพื่อนำมาประมวลสัญญาณ ให้มีความถูกต้องชัดเจนมากยิ่งขึ้น ซึ่งการประมวลผลสัญญาณของเครื่องช่วยฟังแต่ละแบบมีความสลับซับซ้อนไม่เหมือนกัน

-วงจรขยายเสียง (Amplifier) เป็นส่วนที่ขยายสัญญาณหรือพลังงานไฟฟ้าที่ส่งมาจากไมโครโฟนให้ดังขึ้น ซึ่งการขยายจะมีมากน้อยเพียงใดขึ้นอยู่กับอัตราการขยายของเครื่องช่วยฟัง

-ลำโพง (Speaker) สัญญาณที่ได้จากวงจรประมวลผลสัญญาณเรียบร้อยแล้วผ่านวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกไปยังลำโพง ทำหน้าที่เปลี่ยนพลังงานไฟฟ้าที่ได้เป็นพลังงานเสียง

- แบตเตอรี่ ทำหน้าที่ เป็นแหล่งจ่ายพลังงานไฟฟ้าให้กับเครื่องช่วยฟัง โดยปกติการใช้งานของแบตเตอรี่ขึ้นอยู่กับการใช้งานมากน้อยแค่ไหนและการปรับอัตราการขยายของวงจรเครื่องช่วยฟัง

2.6.3. บล็อกไดอะแกรมของเครื่องช่วยฟังแบบดิจิตอล

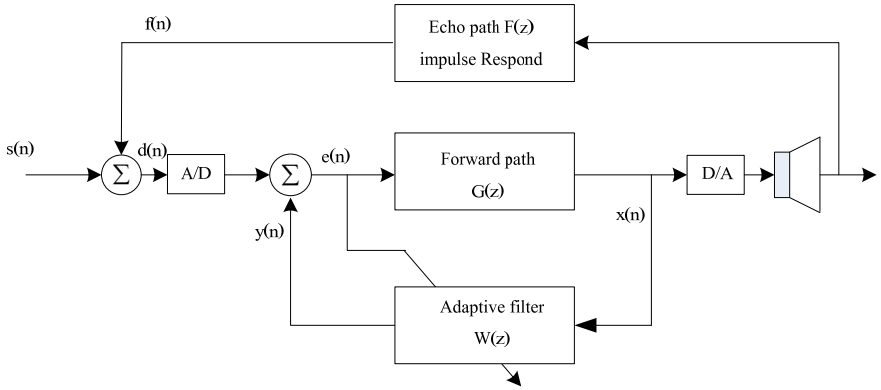


ภาพประกอบ 2-20 บล็อกไดอะแกรมของการทำงานของเครื่องช่วยฟัง

จากภาพประกอบ 2-20 การทำงานของเครื่องช่วยฟังแบบดิจิตอลประกอบไปด้วยวงจรกำจัดสัญญาณเสียงสะท้อน (Echo Cancellation) ทำหน้าที่ กำจัดสัญญาณเสียงสะท้อนจากลำโพงมายังไมโครโฟน เช่น เสียงหวีด เสียงหอน วงจรกำจัดสัญญาณรบกวน (Noise Reduction) ทำหน้าที่กำจัดสัญญาณรบกวนออกจากระบบ และวงจรปรับแต่งเสียง (Speech Enhancement) ทำหน้าที่ปรับแต่งเสียงให้มีความเหมาะสมกับคนแต่ละคน ซึ่งผู้พิการทางการได้ยินมีประสาทการรับรู้ที่แตกต่างกันไปในแต่ละย่านความถี่หรือชดเชยในความถี่ที่ขาดหายไป ภายในวงจรนี้ เช่น วงจรฟิลเตอร์แบงค์ (Filter bank) และมีวงจรร่วมอีกสองชุดคือ วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลและแปลงสัญญาณดิจิตอลไปเป็นอนาลอก

2.6.4. วงจรการกำจัดสัญญาณเสียงสะท้อนของเครื่องช่วยฟัง

ลักษณะของวงจรการกำจัดสัญญาณสะท้อนของเครื่องช่วยฟัง ประกอบไปด้วยส่วนต่างๆ ดังภาพประกอบ 2-21 โดยมีสัญญาณต่างๆดังนี้



ภาพประกอบ 2-21 บล็อกไดอะแกรมการกำจัดสัญญาณเสียงสะท้อนของเครื่องช่วยฟัง [8]

x(n) = สัญญาณอ้างอิงของวงกรองปรับตัวได้

f(n) = สัญญาณการป้อนกลับจากลำโพง

s(n) = สัญญาณผ่านไมโครโฟน

d(n) = สัญญาณผ่านไมโครโฟนรวมกับสัญญาณการป้อนกลับ

$y(n)$ = สัญญาณการประมาณค่าจากวงจรกรองปรับตัวได้

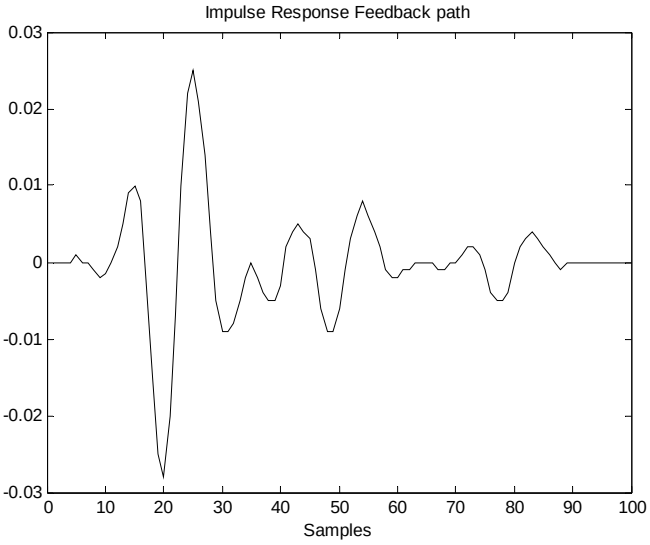
e(n) = สัญญาณคลาดเคลื่อน

F(Z) = ฟังก์ชันของการป้อนกลับ

W(Z) = ฟังก์ชันของวงจรกรองปรับตัวได้

G(Z) = ฟังก์ชันอัตราการขยายของเครื่องช่วยฟัง

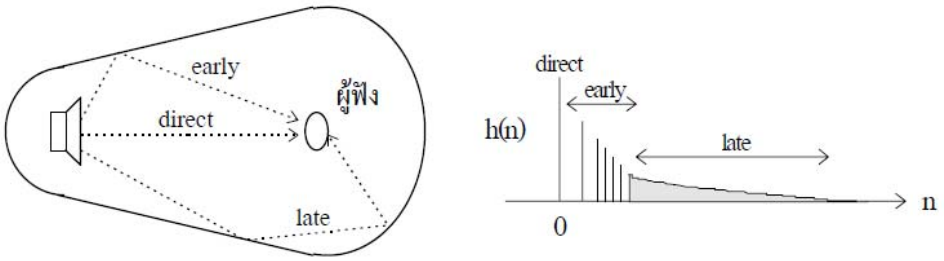
จากภาพประกอบ 2-21 สัญญาณอินพุต s(n) เข้ามาทางไมโครโฟนเป็นสัญญาณเสียงพูดมีส่วนผสมของสัญญาณรบกวนอยู่ด้วยเข้ามารวมกับสัญญาณ f(n) เป็นสัญญาณป้อนกลับที่มาจากลำโพงสะท้อนกลับเข้ามารวมกับสัญญาณมาจากไมโครโฟน ในการกำจัดสัญญาณเสียงสะท้อนนี้จำเป็นต้องมีฟังก์ชัน W(Z) ซึ่งเป็นวงจรกรองปรับตัวได้ ทำหน้าที่กำจัดสัญญาณเสียงสะท้อน f(n) ออกจากระบบ ทำให้ได้ยินสัญญาณพูด s(n) ผ่านเข้ามาทางไมโครโฟนเพียงอย่างเดียวได้ชัดเจนมากยิ่งขึ้น ส่วน G(Z) ซึ่งเป็นอัตราการขยายของเครื่องช่วยฟัง มีค่าเท่าประมาณ 20 dB



ภาพประกอบ 2-22 ผลตอบสนองอิมพัลส์การป้อนกลับของเครื่องช่วยฟังแบบในช่องหูขนาดเต็มใบหู [8]

ภาพประกอบ 2-22 แสดงผลตอบสนองต่ออิมพัลส์การป้อนกลับ (Impulse Response Feedback path) ของเครื่องช่วยฟังแบบในช่องหูขนาดเต็มใบหู [8] เป็นผลตอบสนองระบบในเครื่องช่วยฟัง สัญญาณเอาต์พุตที่ออกจากลำโพงผ่านผลตอบสนองต่ออิมพัลส์ส่วนนี้ไปรวมกับสัญญาณที่เข้ามาทางลำโพง ทำให้เกิดการสะท้อนของสัญญาณ

หลักการเกิดเสียงสะท้อนภายในห้อง [17] เมื่อเสียงเดินทางไปกระทบกับวัตถุหรือผนังของห้องทำให้เสียงสะท้อน เสียงที่ได้ยินประกอบด้วย เสียงที่มาตรงจากแหล่งกำเนิด จากนั้นเป็นเสียงสะท้อนในระยะแรก (early reflection) มีความหนาแน่นน้อย แต่มีขนาดใหญ่ ซึ่งเกิดจากการสะท้อนของผนังที่อยู่ใกล้แหล่งกำเนิด และต่อมาได้ยินการสะท้อนในระยะท้าย (late reflection) ซึ่งมาจากการสะท้อนของผนังที่อยู่ไกลออกไป มีค่าความหนาแน่นมาก แต่มีขนาดเล็ก ดังแสดงในภาพประกอบ 2-23



ภาพประกอบ 2-23 การเกิดเสียงสะท้อนในห้องและผลตอบสนองต่ออิมพัลส์ของระบบ [17]

การเกิดเสียงสะท้อนในระบบของเครื่องช่วยฟัง จะเกิดการสะท้อนในระยะแรกซึ่งมีความหนาแน่นของเสียงน้อย แต่มีขนาดใหญ่ เนื่องจากระยะของลำโพงกับไมโครโฟนของเครื่องช่วยฟังมีระยะที่ใกล้กันมาก

บทที่ 3
การออกแบบและวิธีทดสอบระบบ

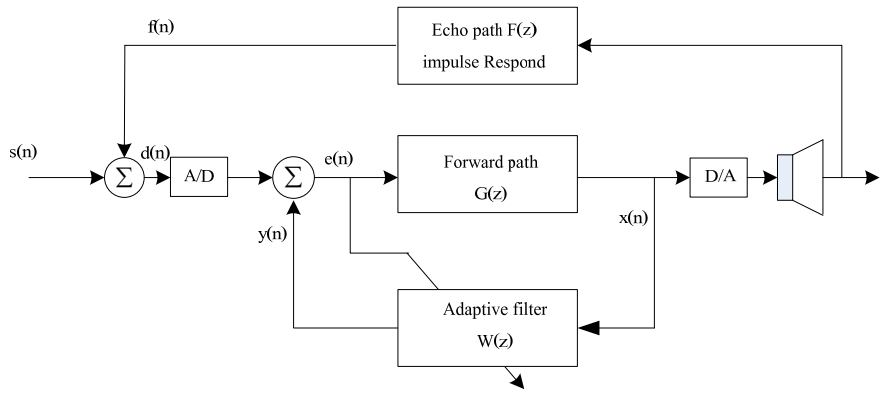
เนื้อหาในบทนี้จะกล่าวถึงการออกแบบวงจรกรองปรับตัวได้นั้นมีด้วยกัน 2 ส่วน คือ การออกแบบด้วย AccelDSP Tool ใน MATLAB Simulink และการออกแบบด้วยภาษาบรรยาย Verilog HDL สำหรับการใช้ทรัพยากรร่วมเพื่อลดขนาดของวงจร

3.1 การออกแบด้วย AccelDSP Tool

ในการออกแบบวงจรเพื่อกำจัดสัญญาณเสียงสะท้อนในเครื่องช่วยฟัง ที่มีผลตอบสนองต่ออิมพัลส์ในภาพประกอบ 2-22 เป็นสัญญาณรบกวนป้อนเข้ามาไมโครโฟน สามารถทำการทดสอบด้วยวงจรกรองปรับตัวได้ด้วยโปรแกรม MATLAB เพื่อเลือกขนาดวงจรกรองที่เหมาะสมก่อนมาทำการออกแบบวงจรกรองปรับตัวได้แบบ LMS และ DLMS ด้วย AccelDSP Tool

3.1.1 การออกแบบหาค่าความเหมาะสมของขนาดของวงจรกรอง

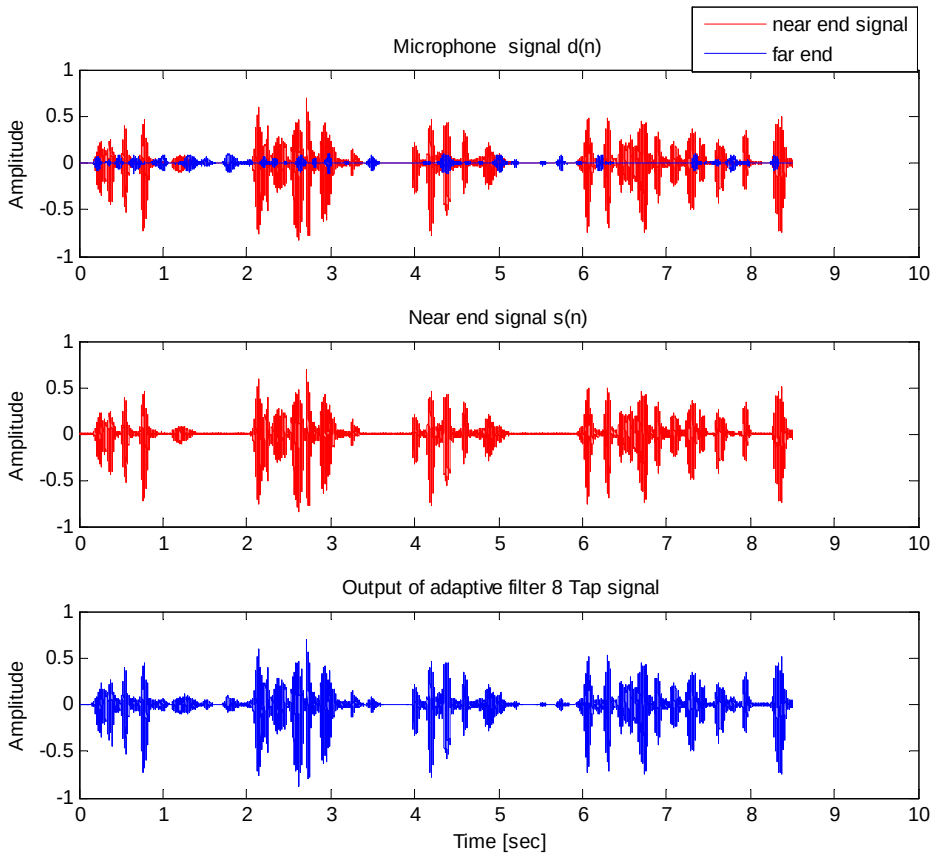
จากผลตอบสนองสัญญาณต่ออิมพัลส์ในภาพประกอบที่ 2-22 สามารถนำมาทำการทดสอบเพื่อเลือกขนาดของวงจรกรองที่เหมาะสม ในที่นี้พิจารณาขนาด 8,16,32 และ 64 แท็ป



ภาพประกอบ 3-1 บล็อกไดอะแกรมของเครื่องช่วยฟัง

ภาพประกอบ 3-1 เป็นบล็อกไดอะแกรมของเครื่องช่วยฟัง เพื่ออธิบายสัญญาณต่างๆ ที่ป้อนเข้ามาก่อนที่จะนำสัญญาณเหล่านี้ไปออกแบบด้วยโปรแกรม MATLAB จากภาพประกอบนี้สัญญาณต่างๆประกอบด้วย s(n) เป็นสัญญาณเสียงพูดที่ไม่มีสัญญาณรบกวนเข้ามา

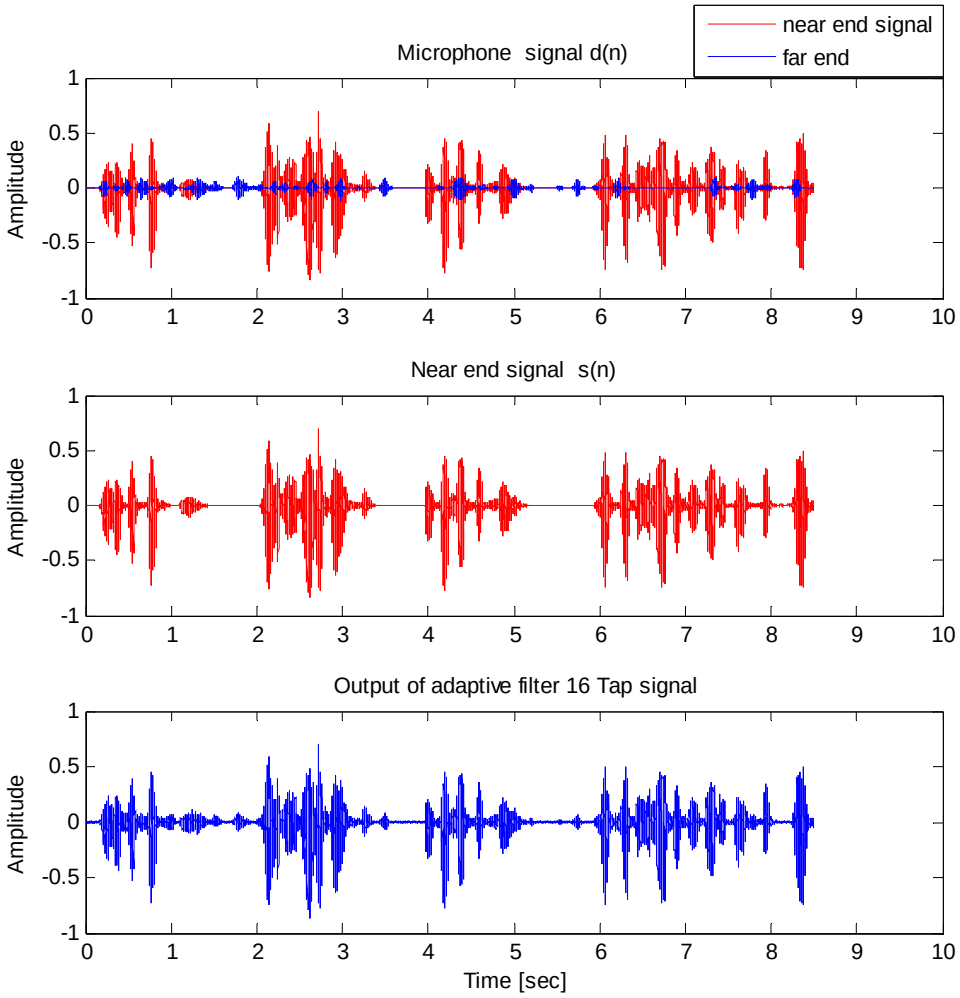
แทรก สัญญาณ f(n) เป็นสัญญาณเสียงสะท้อนที่ได้จากผลตอบสนองอิมพัลส์รวมกับสัญญาณรบกวนจึงได้สัญญาณ d(n) ที่มีส่วนผสมสัญญาณทั้งสอง คือ s(n) และ f(n) ต่อมา สัญญาณ x(n) เป็นสัญญาณอ้างอิงที่ป้อนให้กับวงจรกรองปรับตัวได้ ซึ่งโครงสร้างภายในเป็นวงจรกรองแบบเอฟไออาร์สัญญาณเอาต์พุตที่ได้ คือ สัญญาณ y(n) ถูกนำมาลบกับสัญญาณ d(n) ซึ่งเป็นสัญญาณอินพุต ทำให้ได้สัญญาณ e(n) เป็นเอาต์พุตของวงจรกรองแบบปรับตัวได้ สัญญาณนี้ถูกนำไปปรับค่าสัมประสิทธิ์ใหม่ในวงจรกรองแบบเอฟไออาร์ สัญญาณต่างๆเหล่านี้ถูกนำมาพิจารณาเพื่อเลือกขนาดของวงจรกรองที่เหมาะสม



ภาพประกอบ 3-2 การทำงานของวงจรกรองปรับตัวได้ขนาด 8 แท็ป

ภาพประกอบ 3-2 แสดงผลของการออกแบบวงจรกรองสัญญาณเสียงสะท้อนด้วยวงจรกรองปรับตัวได้ขนาด 8 แท็ป สัญญาณ ในภาพประกอบ (บน) คือ สัญญาณเสียงพูด(แดง) ที่ไม่มีสัญญาณเสียงสะท้อน ส่วนสัญญาณเสียงสะท้อน (น้ำเงิน) ทั้งสองสัญญาณนี้รวมกันเป็นสัญญาณ d(n) ป้อนให้กลับวงจรกรองปรับตัวได้ ภาพประกอบ (กลาง) คือ สัญญาณเสียงพูด (แดง)

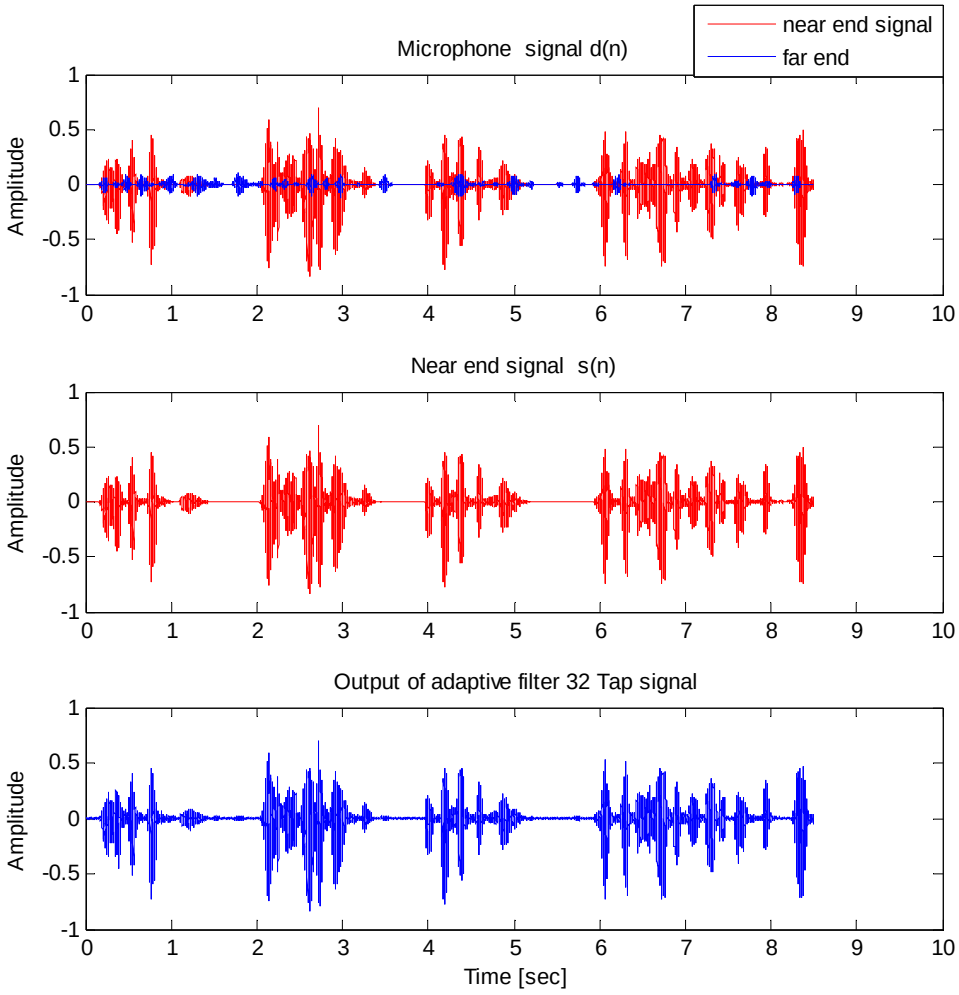
ที่ไม่มีสัญญาณเสียงสะท้อน หรือสัญญาณ s(n) เพื่อนำมาเปรียบเทียบกับภาพ(ล่าง) ซึ่งเป็นเอาต์พุตของวงจรกรองปรับตัวได้ขนาด 8 แท็ป จะเห็นได้ว่าวงจรไม่สามารถกำจัดสัญญาณเสียงสะท้อนให้หมดไปได้เมื่อเปรียบเทียบระหว่างภาพ (กลาง) กับ (ล่าง)



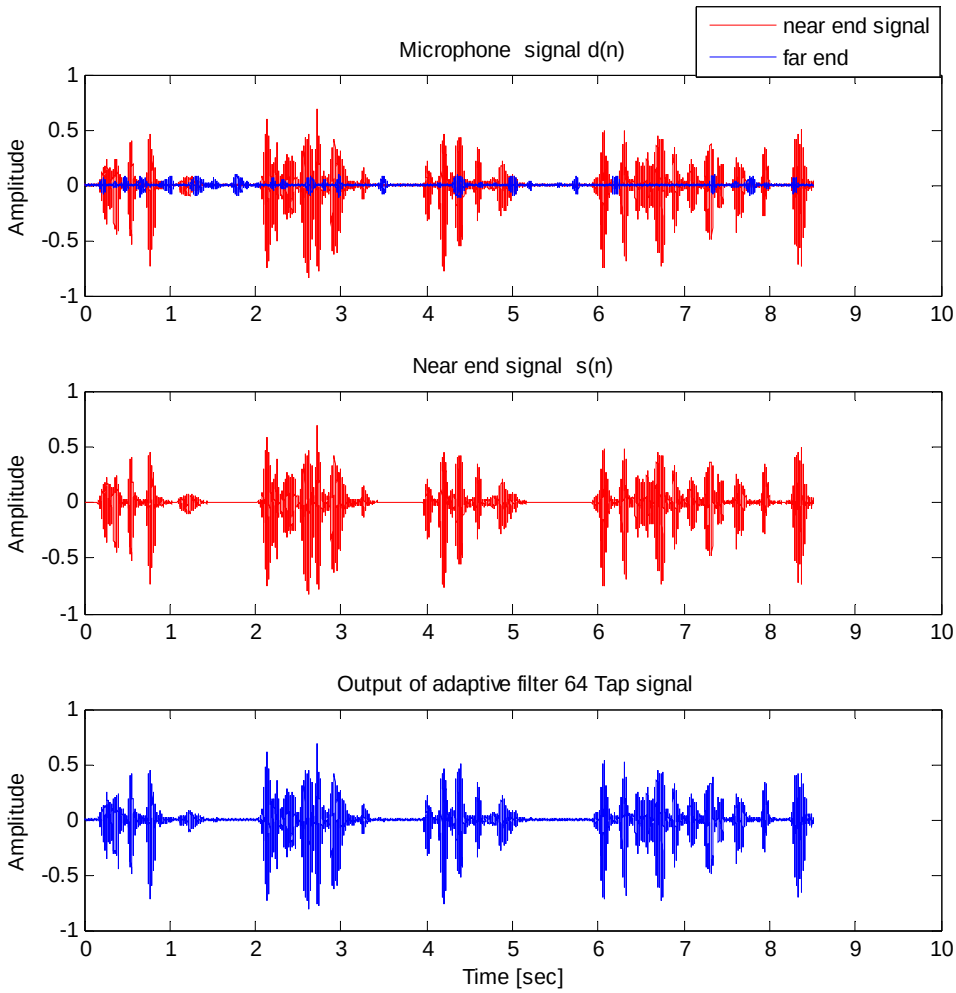
ภาพประกอบ 3-3 การทำงานของวงจรกรองปรับตัวได้ขนาด 16 แท็ป

ภาพประกอบ 3-3 แสดงผลการออกแบบวงจรกรองสัญญาณเสียงสะท้อนด้วยวงจรกรองปรับตัวได้ขนาด 16 แท็ป สัญญาณทีออกจากเอาต์พุต (ล่าง) มีสัญญาณรบกวนปะปนมาอยู่อีกเมื่อทำการเปรียบเทียบกับสัญญาณเสียง (กลาง) ไม่มีสัญญาณรบกวน วงจรกรองปรับตัวได้ขนาด 16 แท็ป จึงไม่สามารถกรองสัญญาณเสียงสะท้อนออกไปได้

ในทางตรงข้ามภาพประกอบ 3-4 และภาพประกอบ 3-5 ใช้วงจรกรองปรับตัวได้ขนาด 32 แท็ป และ 64 แท็ป เมื่อมองภาพสัญญาณเอาต์พุต สังเกตเห็นว่าสามารถกรองสัญญาณเสียงสะท้อนออกไปได้ ดังนั้นการออกแบบที่จะนำไปใช้ คือวงจรกรองปรับตัวได้ขนาด 32 แท็ป ถ้าเป็นขนาด 64 แท็ป มีการใช้ทรัพยากรที่มากเกินไปไม่จึงเหมาะสมกับการนำไปสร้างวงจร



ภาพประกอบ 3-4 การทำงานของวงจรกรองปรับตัวได้ขนาด 32 แท็ป



ภาพประกอบ 3-5 การทำงานของวงจรกรองปรับตัวได้ขนาด 64 แท็ป

3.1.2 การแบ่งจำนวนไปป์ไลน์

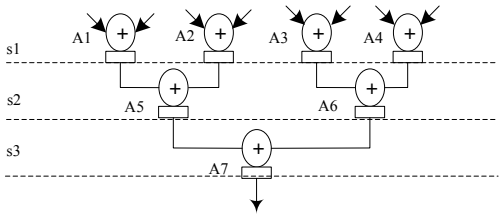
การแบ่งจำนวนไปป์ไลน์ที่เหมาะสมในการออกแบบทำให้วงจรสามารถทำงานได้รวดเร็ว โดยการออกแบบวงจรกรองปรับตัวได้ ซึ่งภายในประกอบด้วยวงจรคูณและวงจรบวก เมื่อทำการสร้างวงจรคูณและวงจรบวกใน Xilinx Virtex-II Pro XC2VP30 ความเร็วการทำงานของวงจรมีดังนี้

ตารางที่ 3-1 ความเร็วในการทำงานของวงจรบวกและวงจรคูณ

Circuit	Speed (ns)
MULTIPLY	10.752
ADDER	4.202

จากตาราง 3-1 เมื่อทำการพิจารณาการทำงานของวงจรคูณ คิดเป็น 2.5 เท่าของวงจรบวก ดังนั้นการแบ่งการทำงานแบบไปป์ไลน์ของวงจรคูณสามารถทำการแบ่งไปป์ไลน์ได้สูงสุด 3 สเตจ เพื่อนำไปใช้ในการออกแบบวงจร

เมื่อนำวงจรบวกมาต่อแบบไบนารีทรี ในภาพประกอบ 3-6 ประกอบด้วยวงจรบวกทั้งหมด 7 ตัว สามารถแบ่งไปป์ไลน์ของวงจรบวกได้สูงสุดเท่ากับ 3 สเตจ โดยสามารถหาค่าจำนวนไปป์ไลน์ในการต่อของวงจรบวกได้จาก $\log_2 N$ เมื่อ N เป็นค่าจำนวนวงจรบวกที่นำมาต่อ เมื่อพิจารณาวงจรกรองปรับตัวได้ขนาด 32 แท็ป ใช้วงจรบวกในส่วนของวงจรกรองเอฟไออาร์เท่ากับ 31 ตัว โดยสามารถแบ่งไปป์ไลน์ได้สูงสุดเท่ากับ 5 สเตจ เพื่อนำไปใช้ในการออกแบบวงจร



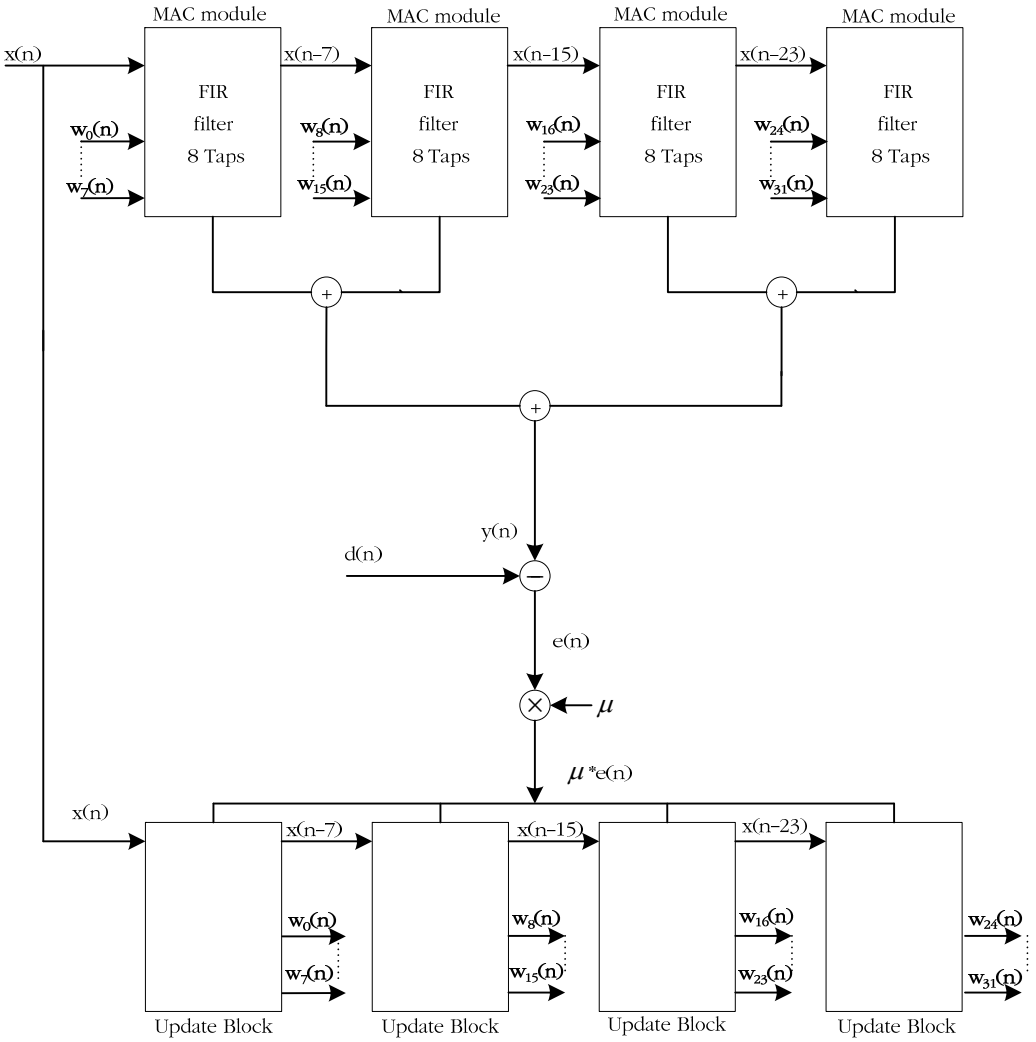
ภาพประกอบ 3-6 การต่อวงจรบวกวงจรแบบไบนารีทรี

จากจำนวนไปป์ไลน์สำหรับการออกแบบการออกแบบวงจรกรองปรับตัวได้ขนาด 32 แท็ป สามารถแบ่งไปป์ไลน์การทำงานของวงจรคูณได้เท่ากับ 3 สเตจ และวงจรบวกเมื่อนำมาต่อแบบไบนารีทรีได้เท่ากับ 5 สเตจ นำไปใช้ในการออกแบบ

จำนวนการแบ่งไปป์ไลน์ในวงจรกรองปรับตัวได้สามารถแบ่งได้หลายสเตจ ก็สามารถทำได้ ขึ้นอยู่กับการออกแบบ

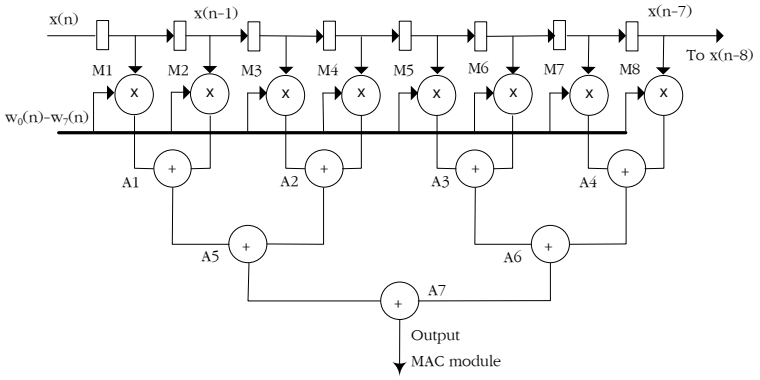
3.1.3 การสร้างวงจรกรองปรับตัวได้แบบ LMS

ผลที่ได้จากการวิเคราะห์เพื่อหาขนาดของวงจรกรองพบว่า วงจรกรองปรับตัวได้ที่เหมาะสม คือ ขนาด 32 แท็ป ถูกนำการสร้างวงจรด้วยการออกแบบวงจรด้วย AccelDSP Tool ใน MATLAB simulink เพื่อทำการวิเคราะห์หาจำนวนทรัพยากรที่ใช้บน Xilinx Virtex-II Pro XC2VP30



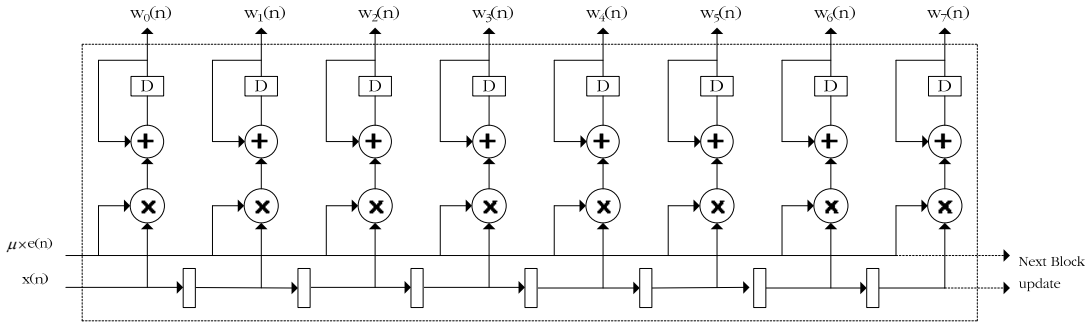
ภาพประกอบ 3-7 โครงสร้างของวงจรกรองปรับตัวได้แบบ LMS ขนาด 32 แท็ป

ภาพประกอบ 3-7 แสดงวงจรกรองปรับตัวได้แบบ LMS ขนาด 32 แท็ป เมื่อนำทำการสร้างวงจร บน AccelDSP Tool โดยใช้รูปแบบการคำนวณแบบเลขทศนิยม (Fixed-point) Q0.15 สัญญาณอินพุตและเอาต์พุตมีขนาด 16 บิต มาทำการประมวลผล ส่วนประกอบของวงจรมีดังนี้ คือ วงจรกรองเอฟไอาร์ขนาด 8 แท็ปต่อรวมกันโดยใช้บล็อก MAC module ต่อกันเป็น 32 แท็ป ให้สัญญาณเอาต์พุต y(n) ส่วนวงจรปรับปรุงค่าสัมประสิทธิ์ใช้ Update Block ขนาด 8 แท็ป ต่ออยู่ ดังนั้นการใช้ทรัพยากรประกอบด้วย วงจรคูณทั้งหมด 65 ตัววงจรบวกอีก 63 ตัว วงจรลบ 1 ตัว



ภาพประกอบ 3-8 โครงสร้างภายในบล็อก MAC module ของวงจรกรองปรับตัวได้แบบ LMS

ภาพประกอบ 3-8 แสดงภายในบล็อก MAC module เป็นส่วนหนึ่งของวงจรกรองแบบเอฟไอาร์ขนาด 8 แท็ป ประกอบด้วย วงจรคูณ 8 ตัว และวงจรบวก 7 ตัว มีทั้งหมด 4 บล็อก ถูกนำมาต่อรวมกันซึ่งมีขนาด 32 แท็ป ส่วนค่าสัมประสิทธิ์ที่นำมาคำนวณในวงจรกรองปรับตัวได้มาจากวงจร Update Block หรือ วงจรปรับปรุงค่าสัมประสิทธิ์



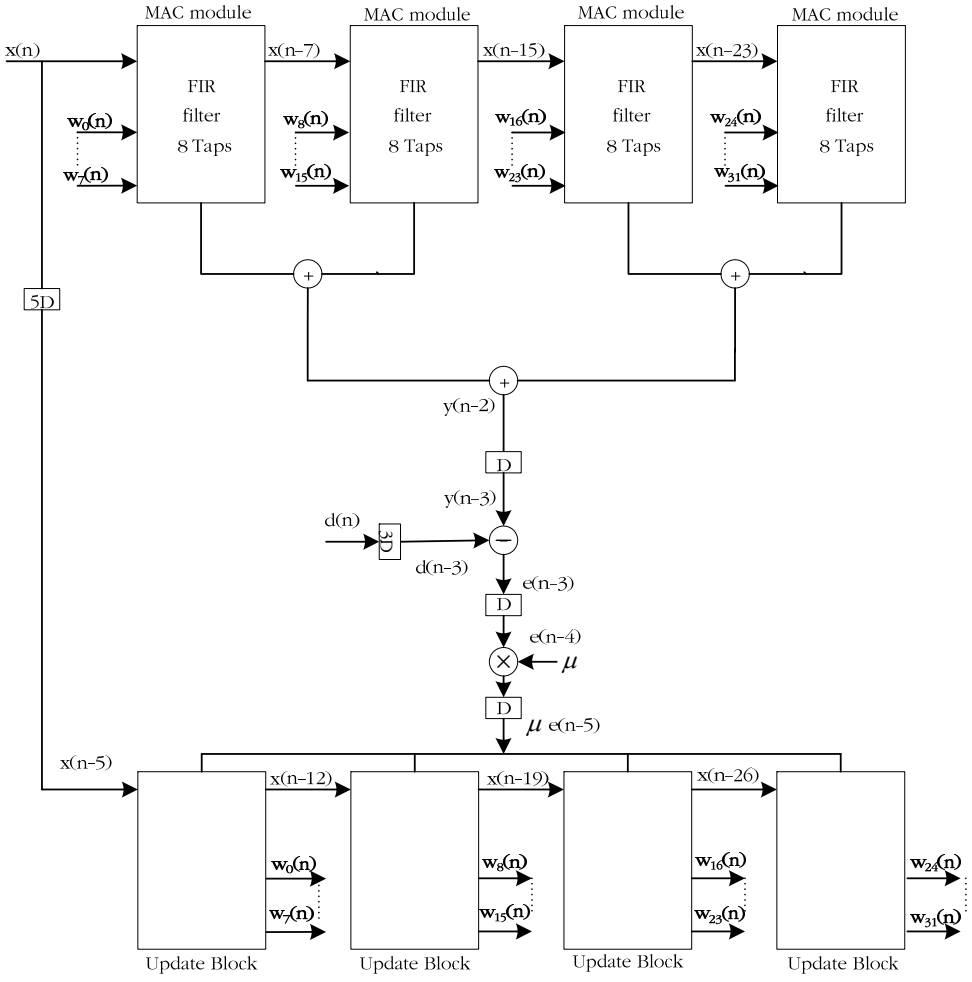
ภาพประกอบ 3-9 โครงสร้างวงจรภายใน Update Block ของวงจรกรองปรับตัวได้แบบ LMS

วงจรปรับปรุงค่าสัมประสิทธิ์ ทำหน้าที่ เพื่อปรับปรุงค่าสัมประสิทธิ์ใหม่ ก่อนที่จะรับค่าสัญญาณอินพุตในรอบถัดไป โดยนำสัญญาณ e(n) หรือค่าผิดพลาดมาคำนวณกับค่าอัตราการเรียนรู้ของระบบและสัญญาณอินพุต x(n) วงจรนี้ประกอบด้วยวงจรบวกทั้งหมด 32 ตัวและวงจรคูณอีก 32 ตัว ซึ่งแสดงในภาพประกอบ 3-9 เป็นเพียงแค่สัมประสิทธิ์เพียง 8 ตัวของวงจรกรองปรับตัวได้แบบ LMS ต้องใช้ Update Block ทั้งหมด 4 ตัว มาต่อรวมกันให้ได้ 32 แท็ป

3.1.4 วงจรกรองปรับตัวได้แบบ DLMS

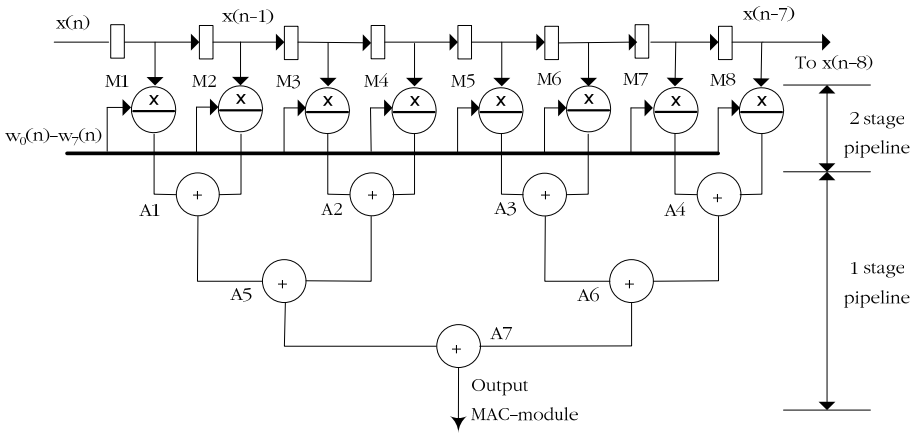
การสร้างวงจรวงจรกรองปรับตัวได้แบบ DLMS ทดลองโดยการเพิ่มดีเลย์เข้าไปในส่วนที่เป็นวงจรกรองเอฟไอาร์ขนาด 32 แท็ป ซึ่งมี 3 รูปแบบในวงจรกรองปรับตัวได้แบบ DLMS คือ วงจรกรองปรับตัวได้แบบ DLMS (D=5), DLMS (D=7) และวงจรกรองปรับตัวได้ DLMS (D=10) สามารถอธิบายการแทรกดีเลย์ได้ดังต่อไปนี้

(1) วงจรกรองปรับตัวได้ DLMS (D=5)



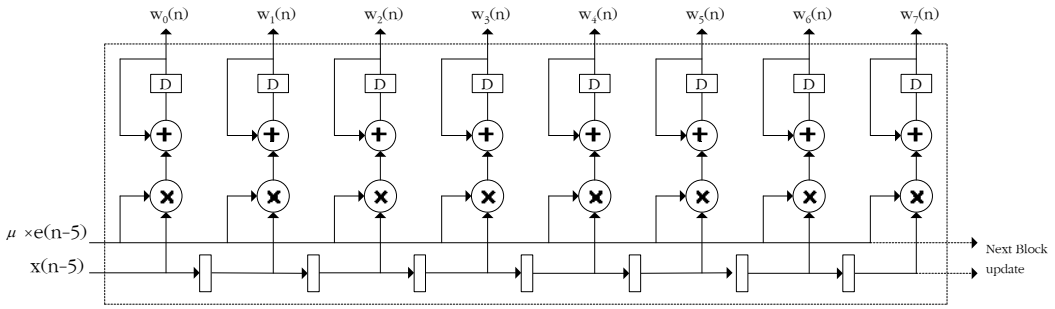
ภาพประกอบ 3-10 โครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS (D =5)

โครงสร้างภาพประกอบที่ 3-10 แสดงวิธีการแทรกดีเลย์เข้าไปในวงจรกรองปรับตัวได้แบบ DLMS มีค่าดีเลย์เท่ากับ 5 มีรายละเอียดดังนี้ คือ ในส่วนของวงจรคูณ 2 สเตจ ในภาพประกอบ 3-11 และวงจรบวกในสเตจสุดท้ายของวงจรกรองเอฟไออาร์ 32 แท็ป ที่ y(n-3) 1 สเตจ และเมื่อรวมกับค่าเอาต์พุตของวงจรลบระหว่าง d(n-3) กับ y(n-3) เป็น 1 สเตจ และที่ $\mu e(n-5)$ ทำให้ได้ดีเลย์ทั้งหมดของวงจรรวมเป็น 5 สเตจ ดังนั้นต้องมีการดีเลย์สัญญาณอินพุต x(n) เท่ากับ 5 สเตจ หรือ x(n-5) ด้วยก่อนนำมาปรับปรุงค่าสัมประสิทธิ์ใหม่



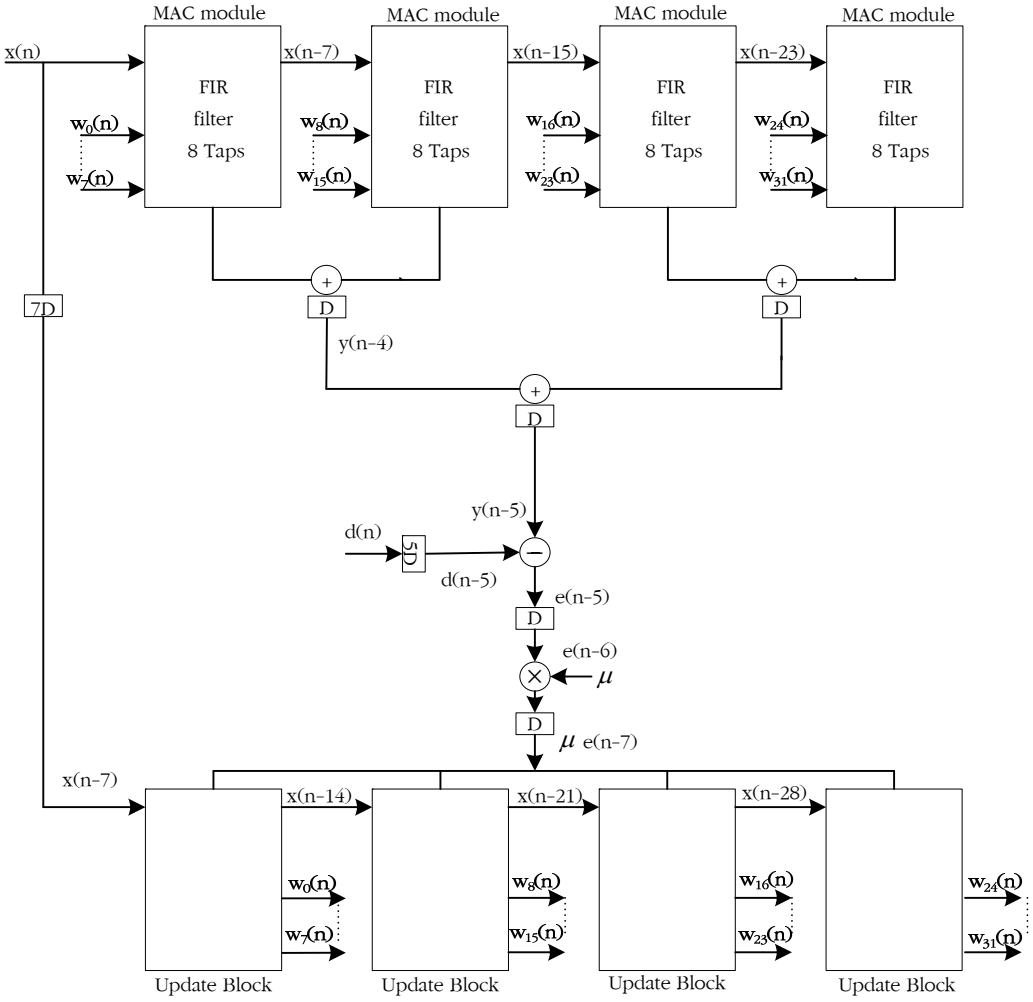
ภาพประกอบ 3-11 การแบ่งไปป์ไลน์วงจรกรองปรับตัวได้แบบ DLMS (D=5)

วงจรปรับปรุงค่าสัมประสิทธิ์ของวงจรกรองปรับตัวได้แบบ DLMS จะใช้วงจรเหมือนกับวงจรกรองปรับตัวได้แบบ LMS แต่จะแตกต่างกันอยู่ที่การดีเลย์สัญญาณอินพุต x(n) ของวงจรกรองปรับตัวได้แบบ DLMS มีค่าเท่ากับจำนวนดีเลย์หรือไปป์ไลน์ที่แบ่งไว้ในวงจรกรองปรับตัวได้ เช่น DLMS (D=5) ต้องมีการดีเลย์สัญญาณอินพุต เท่ากับ 5 หน่วยหรือ x(n-5) แสดงในภาพประกอบ 3-12 เป็นบลอกของวงจรปรับปรุงค่าสัมประสิทธิ์



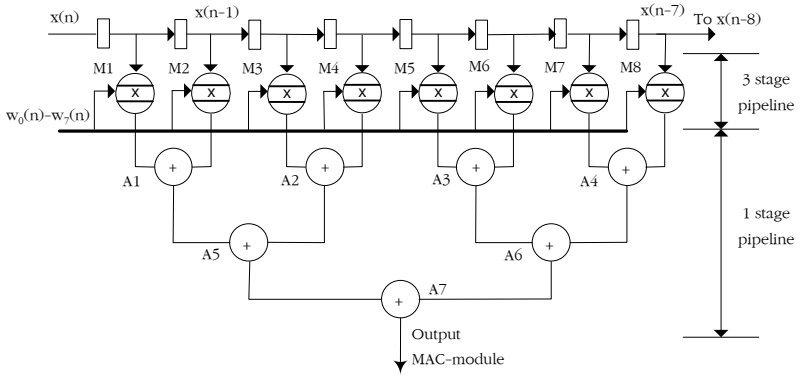
ภาพประกอบ 3-12 Update Block ของวงจรกรองปรับตัวได้ DLMS (D=5)

(2) วงจรกรองปรับตัวได้ DLMS (D=7)



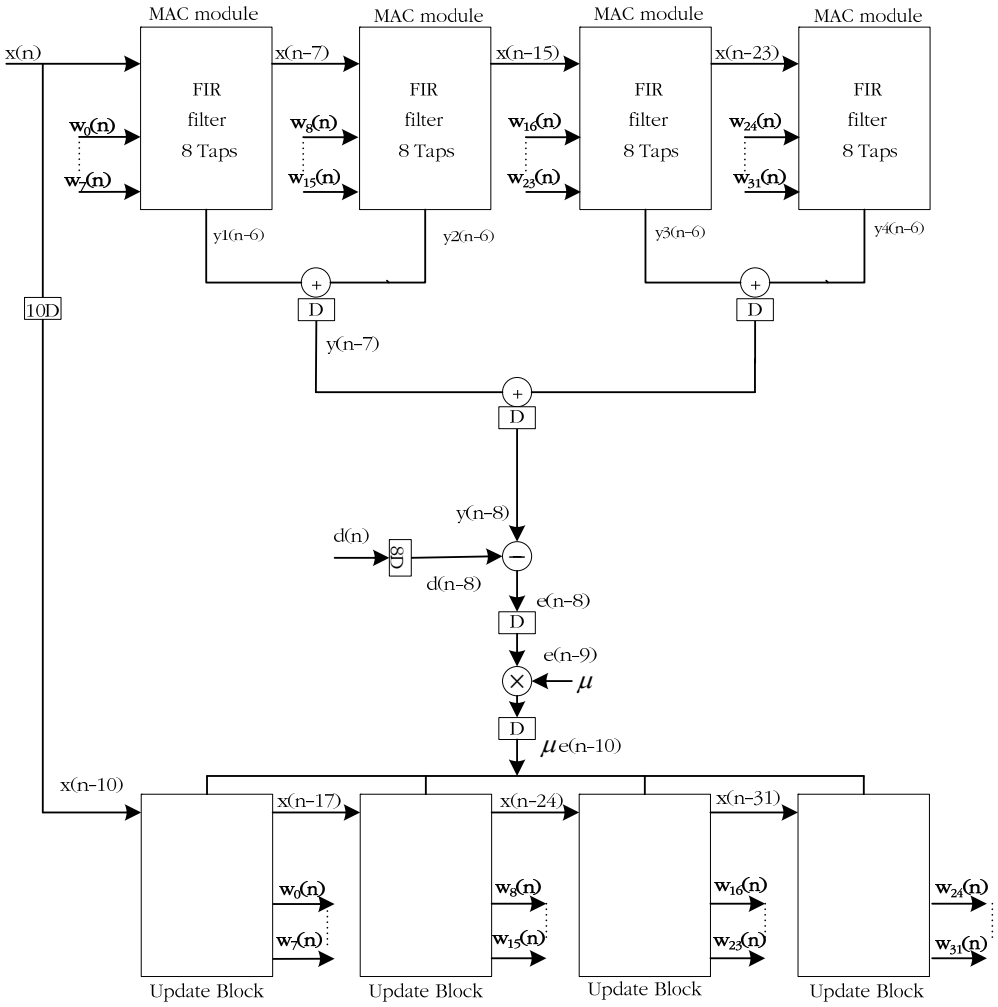
ภาพประกอบ 3-13 โครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS (D =7)

วงจรกรองปรับตัวได้แบบ DLMS (D=7) แสดงในภาพประกอบ 3-13 มีการดีเลย์สัญญาณอินพุตไป 7 หน่วย มีรายละเอียดดังนี้ คือ ใช้วงจรคูณแบบไปป์ไลน์ 3 สเตจ ในวงจรกรองเอฟไออาร์ 8 แท็ป หรือ MAC module ในภาพประกอบ 3-14 ต่อมามีการแทรกดีเลย์ที่ 16 แท็ป อีก 1 สเตจ เป็น y(n-4) และที่ 32 แท็ป 1 สเตจ เป็น y(n-5) ดังนั้นไปป์ไลน์เอาต์พุตของวงจรกรองเอฟไออาร์เท่ากับ 5 สตจหรือ y(n-5) และเมื่อรวมไปป์ไลน์ของ e(n-6) และผลคูณกับค่าอัตราการเรียนรู้มีค่าเท่ากับ 7 สเตจ ส่วนวงจรปรับปรุงค่าสัมประสิทธิ์เหมือนกับวงจรกรองปรับตัวได้แบบ DLMS (D=5) ในภาพประกอบ 3-12 เพียงแต่มีการดีเลย์สัญญาณอินพุตออกไป 7 หน่วย หรือ x(n-7)

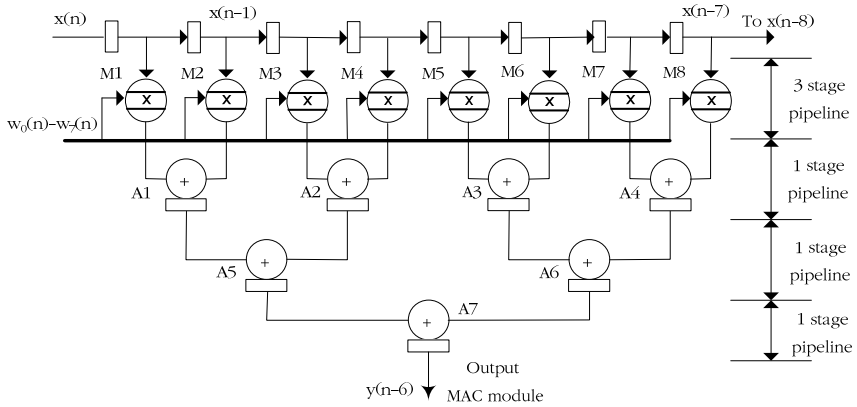


ภาพประกอบ 3-14 การแบ่งไปป์ไลน์วงจรกรองปรับตัวได้แบบ DLMS (D=7)

(3) วงจรกรองปรับตัวได้ DLMS (D=10)



ภาพประกอบ 3-15 โครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS (D =10)



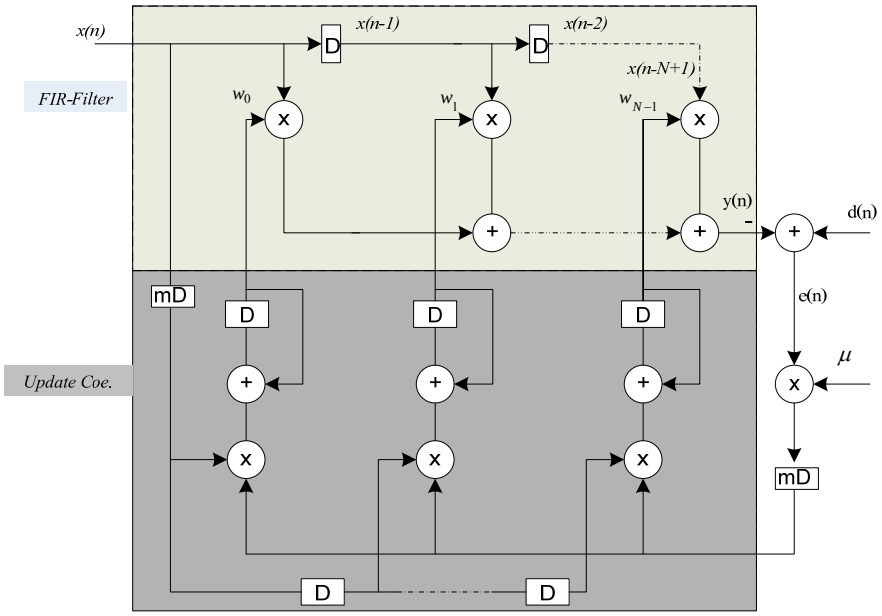
ภาพประกอบ 3-16 การแทรกไปป์ไลน์ในวงจรเอฟไอาร์ขนาด 8 แท็ป

การสร้างวงจรกรองปรับตัวได้ DLMS (D=10) แสดงในภาพประกอบ 3-16 โดยวงจรคูณมีการทำไปป์ไลน์ 3 สเตจ และทำไปป์ไลน์ของวงจรบวกของภายในวงจรกรองเอฟไอาร์ขนาด 8 แท็ปอีก 3 สเตจ รวมเป็น 6 สเตจ y(n-6) แสดงในภาพประกอบ 3-16 จากนั้นนำไปต่อรวมกันกับวงจร 16 แท็ป 1 สเตจ และ 32 แท็ป 1 สเตจ รวมเป็น 2 สเตจ เอาต์พุตของ y(n) เมื่อรวมดีเลย์ที่เกิดขึ้นแล้วทั้งหมด 8 สเตจ y(n-8) เมื่อรวมกับค่าเอาต์พุตของ e(n-9) และ $\mu e(n-10)$ ทำให้ได้ดีเลย์ทั้งหมดของวงจรรวมเป็น 10 สเตจ ส่วนวงจรปรับปรุงค่าสัมประสิทธิ์ใช้แบบเดียวกันเพียงแต่มีการดีเลย์สัญญาณอินพุตที่แตกต่างกันเท่านั้น

จากการออกแบบวงจรกรองปรับตัวได้แบบ DLMS ขนาด 32 แท็ปนี้เป็นแบบไม่ใช้ทรัพยากรร่วม สามารถทำการแทรกไปป์ไลน์ได้สูงสุดเพียง 10 สเตจ ซึ่งหากต้องการจะแทรกไปป์ไลน์เพิ่มก็สามารถทำได้แต่จะสิ้นเปลื้องทรัพยากร

3.2 การออกแบบวงจรแบบการใช้ทรัพยากรร่วมด้วยภาษา Verilog HDL

จากภาพประกอบ 3-17 แสดงโครงสร้างของวงจรกรองปรับตัวได้แบบ DLMS สามารถอธิบายการใช้ ทรัพยากรได้ดังต่อไปนี้ ซึ่งประกอบไปด้วย 2 ส่วน คือวงจรกรองเอฟไอาร์ขนาด 32 แท็ป และวงจรปรับค่าสัมประสิทธิ์ (Update coefficient) วงจรกรองเอฟไอาร์นี้ภายในประกอบด้วยวงจรคูณ 32 ตัว และวงจรบวก 31 ตัว และส่วนวงจรปรับค่าสัมประสิทธิ์ ก็ประกอบด้วยวงจรคูณ 32 ตัว และวงจรบวกอีก 32 ตัว เมื่อรวมวงจรคูณอีก 1 ตัว ของการปรับค่าอัตราการเรียนรู้ สามารถทำการสรุปวงจรต่างๆ ประกอบด้วย วงจรคูณ 65 ตัว วงจรบวก 63 ตัว วงจรลบ 1 ตัว

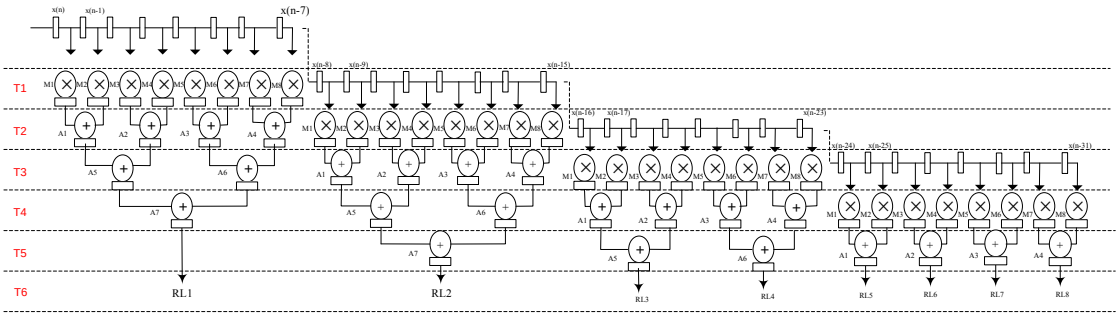


ภาพประกอบ 3-17 แบ่งรายละเอียดภายในวงจรกรองแบบ DLMS

3.2.1 การใช้ทรัพยากรร่วมกันของวงจรกรองเอฟไอาร์

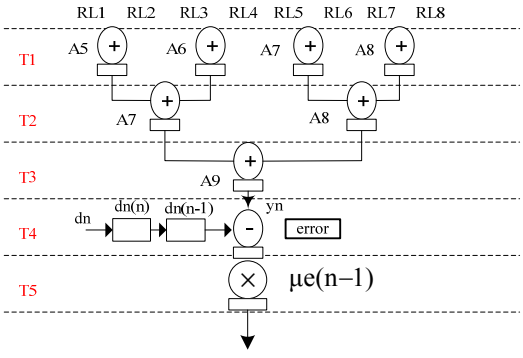
จากภาพประกอบ 3-17 เป็นวงจรแบบไม่ใช้ทรัพยากรร่วมขนาด 32 แท็ป เมื่อนำวงจรกรองเอฟไอาร์มาออกแบบโดยใช้ทรัพยากรร่วม (Resource sharing) วงจรถูกแบ่งวงจรออกเป็น 2 ส่วน ทำงานแบบไปป์ไลน์ ประกอบด้วยวงจรกรองเอฟไอาร์ส่วนที่ 1 และส่วนที่ 2

วงจรกรองเอฟไอาร์ในส่วนที่ 1 จะใช้วงจรคูณเพียง 8 ตัว และวงจรบวก 7 ตัว แสดงในภาพประกอบ 3-18 ซึ่งแสดงขั้นตอนการทำงานของแต่ละรอบคำสั่ง ออกแบบใช้ทรัพยากรร่วม สัญญาณอินพุต x(n-k) คูณกับค่าสัมประสิทธิ์ w_k โดยที่ k คือ จำนวนดีเลย์ในค่าของสัญญาณอินพุต x(n) และอีกความหมายหนึ่ง ใช้แทนค่าสัมประสิทธิ์แต่ละตัว ซึ่งมีทั้งหมด 32 ตัว โดยที่ค่า k มีค่าเท่ากับ (k = 0,1,2,..31) เมื่อได้นำมารวมกันมีเอาต์พุตทั้งหมด 8 ตัว ประกอบด้วย RL1-RL8 ทั้งหมดนี้จะถูกส่งต่อไปยังไปป์ไลน์ส่วนที่ 2 อีกต่อไป การทำงานแบบไปป์ไลน์จะทำไปพร้อมๆกัน ใช้การประมวลผลสัญญาณ 6 รอบสัญญาณนาฬิกา (Clock cycle) ต่อสัญญาณอินพุต 1 สัญญาณ



ภาพประกอบ 3-18 การออกแบบวงจรกรองเอฟไออาร์ขนาด 32 แท็ป ส่วนที่ 1

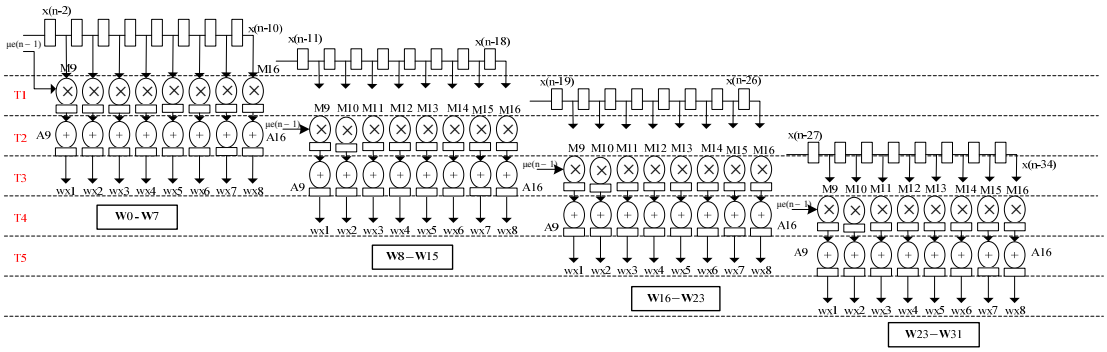
ภาพประกอบ 3-19 แสดงวงจรกรองเอฟไออาร์ส่วนที่ 2 ซึ่งรับสัญญาณอินพุตมาจาก RL1-RL8 ของ วงจรกรองเอฟไออาร์ ส่วนที่ 1 มาบวกกันอีกครั้งโดยใช้วงจรบวก (A5-A9) เอาต์พุตของ วงจรวงจรกรองเอฟไออาร์ อยู่ที่ T3 ถูกนำมาลบกับสัญญาณ d(n) ซึ่งมีการดีเลย์สัญญาณไป 2 หน่วย เพื่อให้เท่ากับการประมวลผลของ วงจรกรองเอฟไออาร์ โดยที่สัญญาณ d(n) จะดีเลย์ 1 ครั้งในทุกๆ 6 รอบ เนื่องจากวงจรกรองเอฟไออาร์ที่ออกแบบถูกแบ่งออกเป็น 2 ส่วนในการคำนวณ ดังนั้นจำเป็นต้องมีการดีเลย์สัญญาณ d(n) ออกเป็น 2 หน่วย ด้วยเช่นกันให้เกิดความสมดุล



ภาพประกอบ 3-19 การออกแบบวงจรวงจรกรองเอฟไออาร์ขนาด 32 แท็ป ในส่วนที่ 2

3.2.2 การใช้ทรัพยากรร่วมกันของวงจรปรับปรุงค่าสัมประสิทธิ์

วงจรในภาพประกอบ 3-17 ซึ่งวงจรเดิมที่ไม่มีการใช้ทรัพยากรร่วมใช้วงจรคูณทั้งหมด 32 ตัว และวงจรบวก 32 ตัว จำนวนของแท็ปของวงจรกรองทั้งหมด 32 ตัว เมื่อนำมาออกแบบด้วยวงจรใช้ทรัพยากรร่วม กำหนดให้ใช้วงจรคูณ 8 ตัว และวงจรบวก 8 ตัว ใช้เวลาในการประมวลผลสัญญาณ 6 รอบสัญญาณนาฬิกา ดังภาพประกอบ 3-20



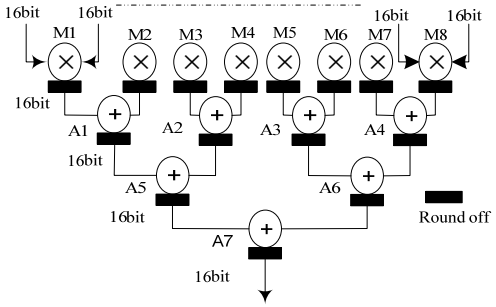
ภาพประกอบ 3-20 การออกแบบวงจรปรับค่าสัมประสิทธิ์ขนาด 32 แท็ป แบบใช้ทรัพยากรร่วม

การประมวลของวงจรปรับค่าสัมประสิทธิ์ใช้เวลาการประมวลผล 6 รอบสัญญาณ ดังแสดงในภาพประกอบ 3-20 ก่อนที่รับสัญญาณอินพุตในรอบถัดไป ซึ่งเวลาในส่วนนี้เป็นตัวกำหนดในการออกแบบวงจรกรองเอฟไอาร์ โดยที่แต่ละไปป์ไลน์ต้องทำงานให้เสร็จภายในเวลา 6 รอบสัญญาณหรือ 1 สเตจไปป์ไลน์และส่งต่อไปยังไปป์ไลน์ในขั้นต่อไป ซึ่งจะเห็นได้ว่าไปป์ไลน์ของวงจรกรองเอฟไอาร์ นี้ใช้ไป 2 สเตจ

3.3 กระบวนการปัดเศษจากการคูณและการบวก (Round off)

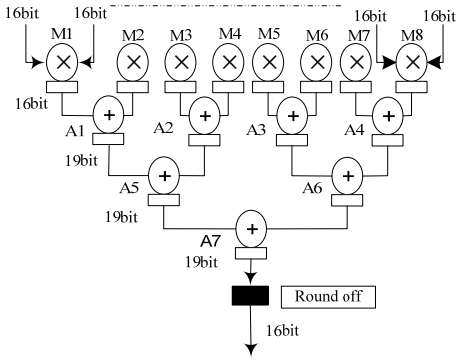
วงจรประมวลผลสัญญาณเลขทศนิยม แบบ Q0.15 มีขนาด 16 บิต วิธีคิดการปัดเศษที่นำมาใช้ มี 2 แบบ คือ

1) วิธีการปัดเศษหลังจากการประมวลผลเสร็จแล้ว คือ การปัดเศษหลังจากการบวกหรือการคูณ วิธีการนี้จะทำการปัดเศษทุกครั้งที่มีการประมวลผล เพื่อให้ได้ผลลัพธ์ขนาด 16 บิต นำไปกับไว้เพื่อประมวลในรอบถัดไป ดังภาพประกอบ 3-21



ภาพประกอบ 3-21 วิธีการปัดเศษหลังจากการประมวลผล

2) วิธีการปัดเศษหลังจากการบวกค่าในครั้งสุดท้ายของการบวก วิธีการแบบนี้การปัดเศษของการบวกกระทำเพียงครั้งเดียวเท่านั้นที่เอาต์พุตของวงจรกรองเอฟไออาร์ คือ ให้ทำการบวกรวมกันให้หมดก่อน เมื่อถึงค่า y(n) ก็ทำการปัดเศษ วงจรบวก (A1-A4) มีขนาด 16 บิต และ (A5-A9) มีขนาด 19 บิต ส่วนการคูณเกิดขึ้นเพียงแค่ครั้งเดียว คือ สัญญาณอินพุต x(n) กับค่าสัมประสิทธิ์ w_k ก็ให้ทำการปัดเศษเหลือเพียง 16 บิตได้เลยแล้วนำไปบวกต่อ ดังภาพประกอบ 3-22

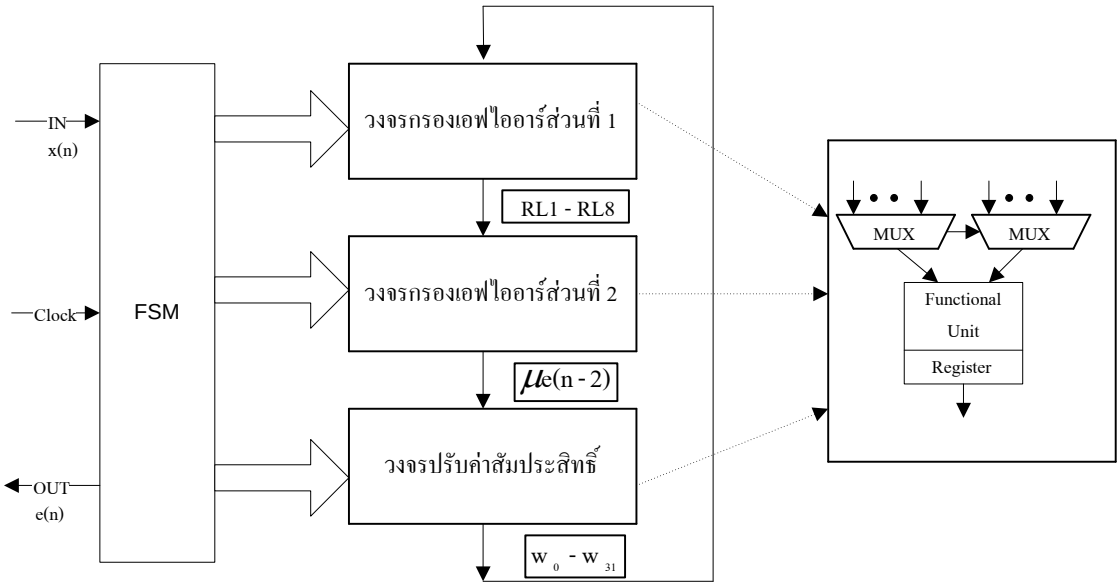


ภาพประกอบ 3-22 วิธีการปัดเศษในครั้งสุดท้ายเพียงครั้งเดียว

จากวิธีการปัดเศษทั้ง 2 แบบนำไปประยุกต์ใช้ในวงจรกรองปรับตัวได้แบบ DLMS ขนาด 32 แท็ป ซึ่งให้ค่าผลลัพธ์ที่แตกต่างกันและความถูกต้องแม่นยำการทำงานของวงจรกรองปรับตัวได้ไม่เหมือนกัน เมื่อนำไปใช้งานจริงกับวงจรกรองปรับตัวได้แบบ DLMS จะใช้วิธีการปัดเศษในครั้งสุดท้ายเพียงครั้งเดียว ให้ค่าที่ถูกต้องแม่นยำกว่าวิธีการปัดเศษในแบบแรก ดังผลการวิเคราะห์ในบทที่ 4

3.4 วงจรควบคุมการทำงานโดยใช้ Finite State Machine

การควบคุมการทำงานของวงจรกรองปรับตัวได้แบบ DLMS อาศัยการควบคุมแบบ Moore Machine มีทั้งหมด 6 รอบสัญญาณก่อนจะรับสัญญาณอินพุตในรอบถัดไป แสดงในภาพประกอบ 3-23 การควบคุมการทำงานของวงจรประกอบด้วย 3 ส่วน คือ วงจรกรองเอฟไอาร์ส่วนที่ 1 (ภาพประกอบ 3-18) กับ ส่วนที่ 2 (ภาพประกอบ 3-19) ทำงานแบบไปป์ไลน์และวงจรปรับค่าสัมประสิทธิ์ ทั้ง 3 ส่วนนี้จะทำงานไปพร้อมกัน โดยโครงสร้างภายในแต่ละส่วนประกอบด้วยมัลติเพล็กเซอร์ รีจิสเตอร์ ฟังก์ชันการประมวลผล โดยแสดงการใช้ทรัพยากรร่วมที่ได้กล่าวไว้ในข้างต้นแล้ว



ภาพประกอบ 3-23 วงจรควบคุมการทำงานของ Finite State Machine

3.5 การออกแบบวงจรด้วยไปป์ไลน์ละเอียด

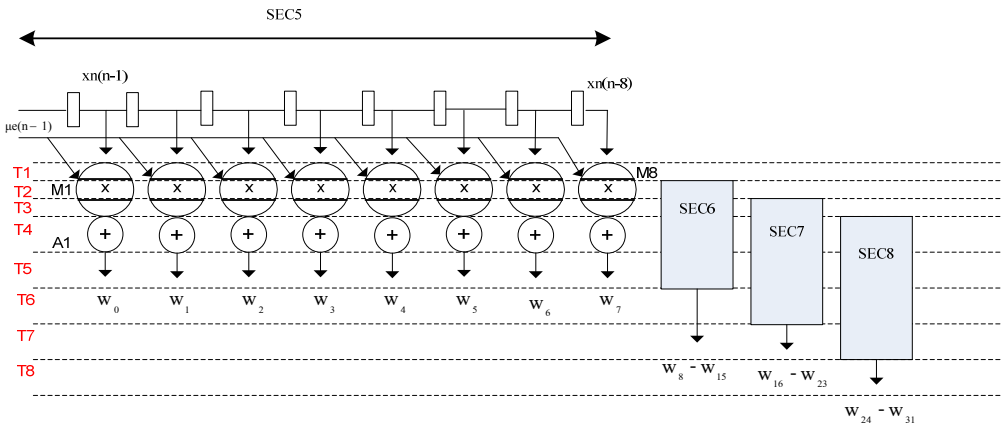
การทำไปป์ไลน์ละเอียดจากวงจรคูณ โดยทำการแบ่งวงจรคูณออกเป็น 3 ขั้น ส่วนวงจรบวกและลบเป็นแบบ 1 ขั้น ทำให้เพิ่มจำนวนรอบการทำงานของวงจรที่เพิ่มมาเพียง 2 รอบการทำงาน การออกแบบวงจรด้วยไปป์ไลน์ละเอียดนี้ใช้หลักการเช่นเดียวกับการออกแบบวงจรใช้ทรัพยากรร่วม โดยทำการออกแบบในส่วนวงจรปรับปรุงค่าสัมประสิทธิ์ เพื่อพิจารณาจำนวนรอบว่าวงจรทำงานไปกี่รอบ จากนั้นมาทำการแบ่งขั้นไม่เกินจำนวนการทำงานของวงจรป้อนกลับ

3.5.1 การออกแบบวงจรปรับปรุงค่าสัมประสิทธิ์

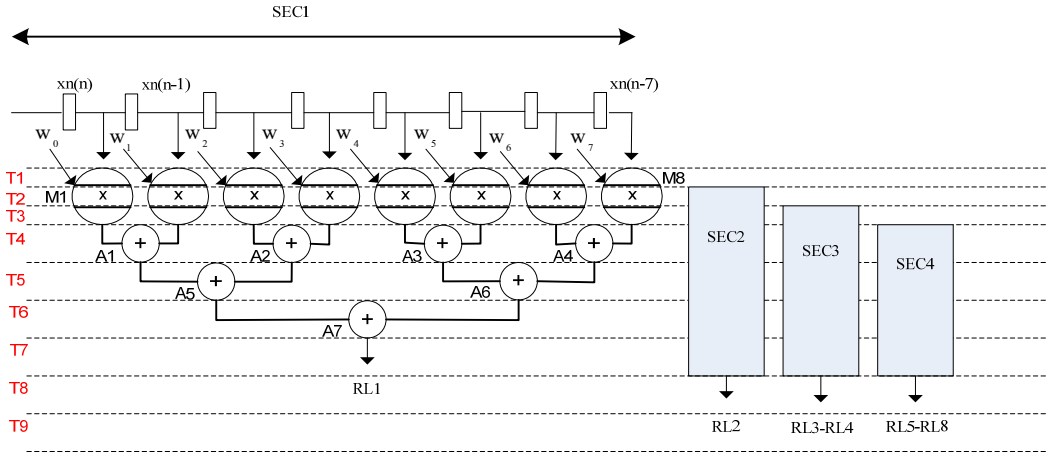
การออกแบบวงจรนี้ด้วยวิธีการใช้ทรัพยากรร่วม ประกอบด้วยค่าสัมประสิทธิ์ทั้งหมด 32 ตัว มาคำนวณก่อนที่รับสัญญาณรอบถัดไปเข้ามา โดยใช้วงจรคูณทั้งหมด 8 ตัว และวงจรบวก 8 ตัว ทำงานร่วมกัน มีการทำงานทั้งหมด 8 รอบการทำงาน(รวมกับค่าโหลดสัญญาณใหม่เข้ามาอีก 1 รอบ) ก่อนที่รับสัญญาณในรอบถัดไป ดังภาพประกอบ 3-24 โดยนำค่าของ 8 รอบนี้มากำหนดการทำงานในส่วนของการวงจรกรองเอฟไออาร์ เพื่อลดจำนวนดีเลย์ที่เกิดขึ้น ในส่วนของวงจรกรองปรับตัวได้แบบ DLMS

3.5.2 การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 1

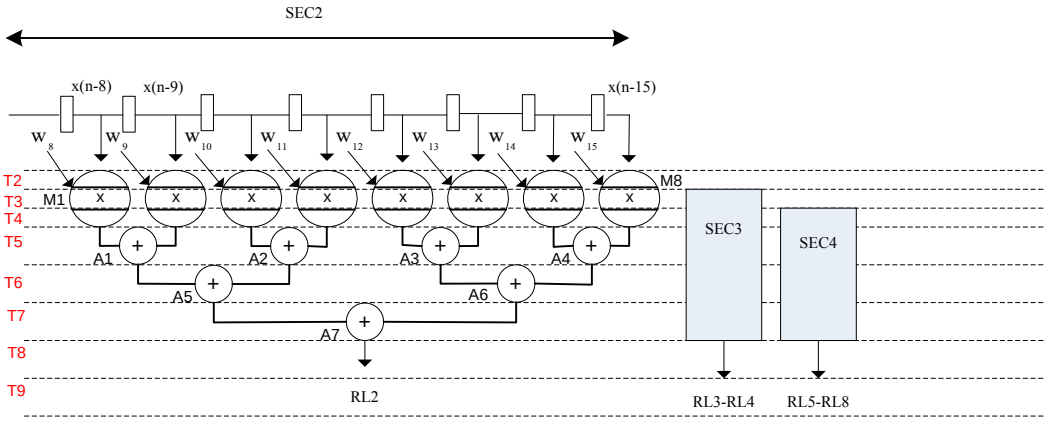
วงจรกรองเอฟไออาร์ส่วนที่ 1 การออกแบบใช้ทรัพยากรร่วมกันประกอบด้วยวงจรคูณ 8 ตัว และวงจรบวกอีก 7 ตัว มีการทำงานทั้งหมด 8 รอบการทำงานเมื่อรวมกับการโหลดสัญญาณอินพุตเข้ามา วงจรนี้แบ่งออกเป็น 4 SEC ภายในบล็อกของ SEC1 กับ SEC2 เหมือนกัน ให้เอาต์พุตออกมา 2 ตัว คือ RL1 และ RL2 ส่วนใน SEC3 ให้ค่าเอาต์พุตออกมา 2 ตัว RL3 และ RL4 ส่วน SEC4 ให้เอาต์พุตออกมา 4 ตัว คือ RL5– RL8 ดังแสดงภาพประกอบ 3-25 ถึง 3-28 แบ่งออกเป็น 4 ส่วน ตามรายละเอียด



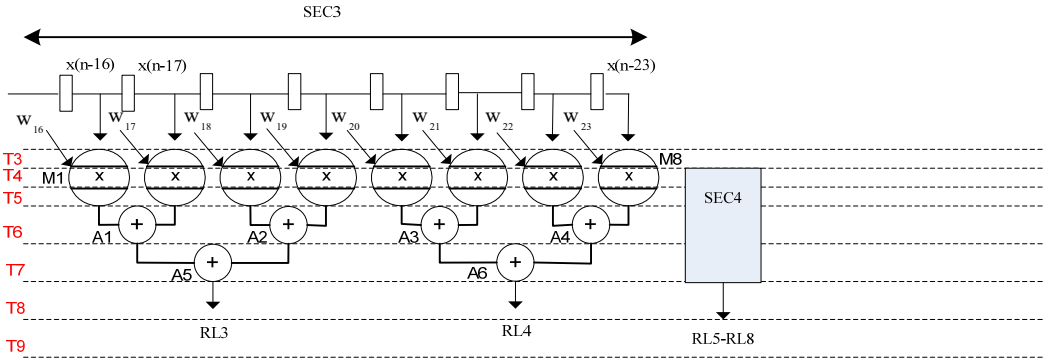
ภาพประกอบ 3-24 การออกแบบวงจรปรับปรุงค่าสัมประสิทธิ์ 32 แท็ปโดยการใช้ทรัพยากรร่วม



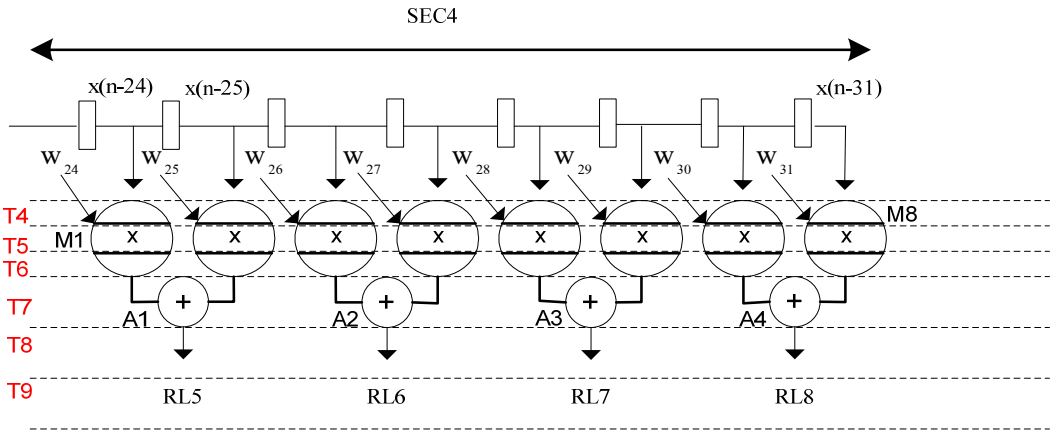
ภาพประกอบ 3-25 การออกแบบวงจรกรองเอฟไอาร์ส่วนที่ 1 SEC 1



ภาพประกอบ 3-26 การออกแบบวงจรกรองเอฟไอาร์ส่วนที่ 1 SEC2



ภาพประกอบ 3-27 การออกแบบวงจรกรองเอฟไอาร์ส่วนที่ 1 SEC3

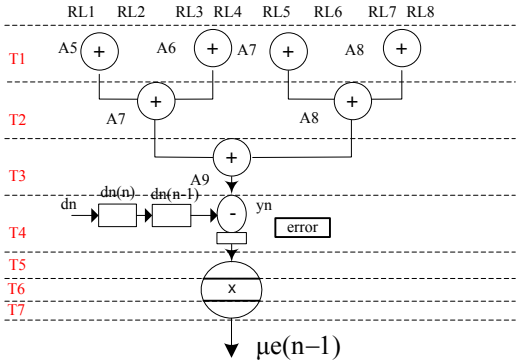


4

ภาพประกอบ 3-28 การออกแบบวงจรกรองเอฟไอาร์ส่วนที่ 1 SEC4

สัญญาณเอาต์พุตจากภาพประกอบ 3-25 ถึง 3-28 คือ RL1-RL8 มีทั้งหมด 8 ตัว ถูกส่งไปคำนวณในวงจรกรองเอฟไอาร์ส่วนที่ 2 ต่อไป โดยข้อมูลจะถูกส่งต่อเป็นแบบไปป์ไลน์ ทำให้จำนวนรอบในการทำงานของวงจรแบบใช้ทรัพยากรร่วมกันลดน้อยลง อาศัยการจัดรูปแบบเวลาการประมวลผลใหม่ (Retime) มาใช้ในการออกแบบ

3.5.3 การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 2



ภาพประกอบ 3-29 การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 2

การออกแบบวงจรกรองเอฟไออาร์ส่วนที่ 2 วงจรในภาพประกอบ 3-29 มีการทำงานคล้ายกับภาพประกอบ 3-19 เพียงแต่วงจรคูณเป็นแบบไปป์ไลน์ทำให้การทำงานจากเดิม 6 รอบการทำงานเพิ่มมาเป็น 8 รอบ ส่วนในวงจรบวกเป็นแบบเดิม การทำงานของวงจรกรองเอฟไออาร์ทำงานแบ่งเป็น 2 ส่วน ทำให้ต้องมีการดีเลย์สัญญาณออกเป็น 2 หน่วยก่อนที่ทำการปรับปรุงค่าสัมประสิทธิ์ใหม่

บทที่ 4
ผลการดำเนินงานและวิเคราะห์ผล

จากการออกแบบวงจรรวมทั้งสองแบบ คือ วงจรแบบไม่ใช้ทรัพยากรร่วม กับวงจรแบบใช้ทรัพยากรร่วม เมื่อนำมาทำการวิเคราะห์การใช้ทรัพยากร ความเร็วที่ทำงาน ในแต่ละแบบการทำงานที่แตกต่างกันออกไป สามารถนำไปใช้งานได้ตามความเหมาะสมของวงจรแต่ละแบบดังต่อไปนี้ วงจรที่ไม่ใช้ทรัพยากรร่วมประกอบด้วย วงจรกรองปรับตัวได้แบบ LMS และวงจรกรองปรับตัวได้แบบ DLMS (D=5, D=7, D=10) ใช้ AccelDSP Tool ใน MATLAB Simmulink เพื่อวิเคราะห์การใช้ทรัพยากรบน Xilinx Virtex-II Pro XC2VP30 ส่วนวงจรที่ใช้ทรัพยากรร่วมซึ่งเป็นวงจรกรองปรับตัวได้แบบ DLMS ค่าดีเลย์เท่ากับ 2 มีขนาด 32 แท็ป โดยได้จากการวิเคราะห์ออกแบบในบทที่ 3 เพื่อใช้ในการกำจัดสัญญาณเสียงสะท้อน

4.1 ผลการสังเคราะห์วงจร

การออกแบบวงจรไม่ใช้ทรัพยากรร่วม ประกอบไปด้วย วงจรกรองปรับตัวได้แบบ LMS จากภาพประกอบ 3-7 และวงจรกรองปรับตัวได้แบบ DLMS (D=5) จากภาพประกอบ 3-10 วงจรกรองปรับตัวได้แบบ DLMS (D=7) จากภาพประกอบ 3-13 และวงจรกรองปรับตัวได้แบบ DLMS (D=10) จากภาพประกอบ 3-15 สามารถสรุปการใช้ทรัพยากรบน Xilinx Virtex-II Pro XC2VP30 ได้จากตารางที่ 4-1

ตารางที่ 4-1 การใช้ทรัพยากรในวงจรกรองปรับตัวได้แบบต่างๆ เป็นวงจรไม่ได้ใช้ทรัพยากรร่วม

Algorithm	LUTs	FFs	Slices	MULT 18X18s	Throughput (MSPS)	Clock period (ns)
LMS	4317	1009	2524	65	21.337	46.866
DLMS [D = 5]	4826	2102	3025	65	76.703	13.037
DLMS [D= 7]	5084	2137	3152	65	82.267	12.156
DLMS [D = 10]	5523	2424	3368	65	94.266	10.608

จากตารางที่ 4-1 วงจรกรองปรับตัวได้แบบต่างๆ โดยไม่ใช้ทรัพยากรร่วม ทำการเปรียบเทียบวงจรกรองปรับตัวได้แบบ LMS และ DLMS พบว่าวงจรกรองปรับตัวได้แบบ LMS ทำงานประมวลผลในเวลา 1 วินาที สามารถทำการคำนวณให้ปริมาณงาน (throughput) เท่ากับ

21.337 MSPS (Mega Sample Per Second) ในขณะที่วงจรกรองปรับตัวได้แบบ DLMS (D=5) ให้ปริมาณงานเท่ากับ 76.703 MSPS สูงกว่าวงจรกรองปรับตัวได้แบบ LMS ถึง 3 เท่า และเมื่อทำการเปรียบเทียบการใช้ทรัพยากร (Slices,LUTs,FFs) วงจรกรองปรับตัวได้แบบ DLMS สูงอยู่เพียงเล็กน้อย ขณะที่วงจรคูณเท่ากันทั้งสองแบบ

การหาปริมาณงานเพื่อวัดการทำงานของหน่วยประมวลผลโดยใช้อัลกอริธึมแบบเดียวกัน ทำได้ตามสมการที่ (4.1) และ (4.2)

การคำนวณหาปริมาณงานสำหรับวงจรไม่ใช้ทรัพยากรร่วมได้จาก

$$\text{throughput} = \frac{1}{T_c} \tag{4.1}$$

T_c คือ เวลาในการประมวลผลสัญญาณ (Clock period)

การคำนวณหาปริมาณงานในกรณีสำหรับวงจรใช้ทรัพยากรร่วม ได้จาก

$$\text{throughput} = \frac{1}{T_C + T_n} \tag{4.2}$$

T_n คือ จำนวนรอบที่ใช้ในการประมวลผล (Clock/sample)

ตารางที่ 4-2 การใช้ทรัพยากรในวงจรกรองปรับตัวได้แบบต่างๆ เป็นวงจรแบบใช้ทรัพยากรร่วม

Algorithm	Slices	FFs	LUTs	MULs	Subs	Clock/sample	Clock period (ns)	Throughput (MSPS)
LMS	1604	2064	1758	17	1	15	8.350	7.984
DLMS (D=2)	2023	2249	2415	17	1	6	8.309	20.058
DLMS (D=5)	2087	2364	2839	17	1	6	9.306	17.909
DLMS (D=2) Fine Grained Pipeline	2352	2831	2687	17	1	8	7.510	16.644

จากตารางที่ 4-2 เมื่อทำการออกแบบวงจรที่ใช้ทรัพยากรร่วมกัน ทำการเปรียบเทียบวงจรกรองปรับตัวได้แบบ LMS วงจรกรองปรับตัวได้แบบ DLMS (D=2) จากภาพประกอบ (3-18 ถึง 3-20) และด้วยวิธีการออกแบบไปป์ไลน์ละเอียด ในภาพประกอบ (3-24 ถึง 3-28) พบว่า การทำงานของวงจรกรองปรับตัวได้แบบ DLMS (D=2) และแบบไปป์ไลน์ละเอียดมีการทำงานเพียง 6 และ 8 รอบสัญญาณ (Clock/sample) ต่อการประมวลผล 1 อินพุตที่ป้อนเข้ามา ในขณะที่วงจรกรองปรับตัวได้แบบ LMSใช้การประมวลผล 15 รอบ ก่อนที่จะรับสัญญาณอินพุตในรอบถัดไป ในขณะที่ Clock Period ทั้งสองวงจรมีความใกล้เคียงกัน โดยที่การออกแบบไปป์ไลน์ละเอียดมีค่าน้อยสุด คือ 7.510 ns และเมื่อทำการเปรียบเทียบปริมาณงานในการทำงานในเวลา 1 วินาที วงจรกรองปรับตัวได้แบบ DLMS ให้ปริมาณงานที่สูงกว่าวงจรกรองปรับตัวได้แบบ LMS

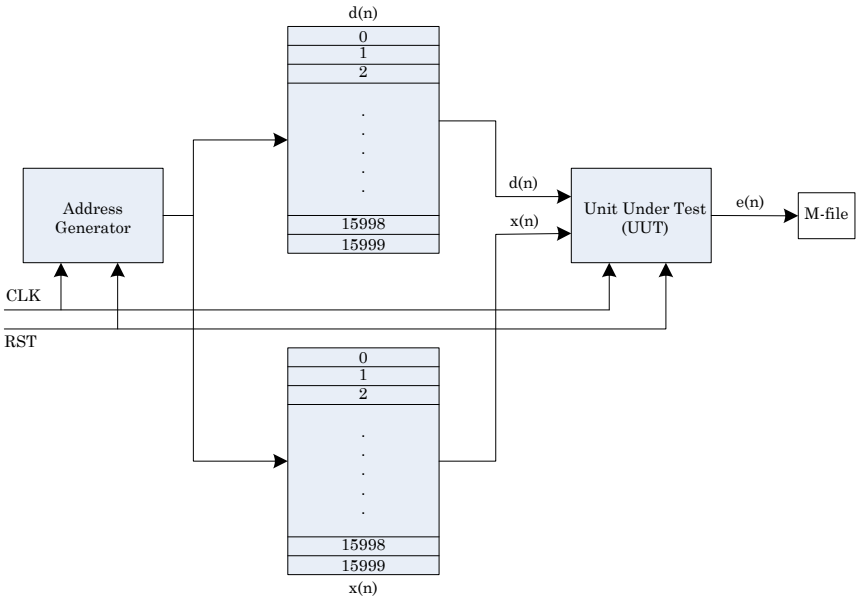
ตารางที่ 4-3 เปรียบเทียบจำนวนรอบการการทำงาน (Clock cycle)ในวงจรกรองปรับตัวได้แบบต่างๆ

Architecture	Adder	Multiplier	Clock Cycle	
			LMS	DLMS
Non-resource sharing	63	65	1	1
Resource sharing	17	17	15	11
Resource sharing [proposed]	17	17	15	6
Resource sharing Fine-Grained Pipeline [proposed]	17	17	22	8

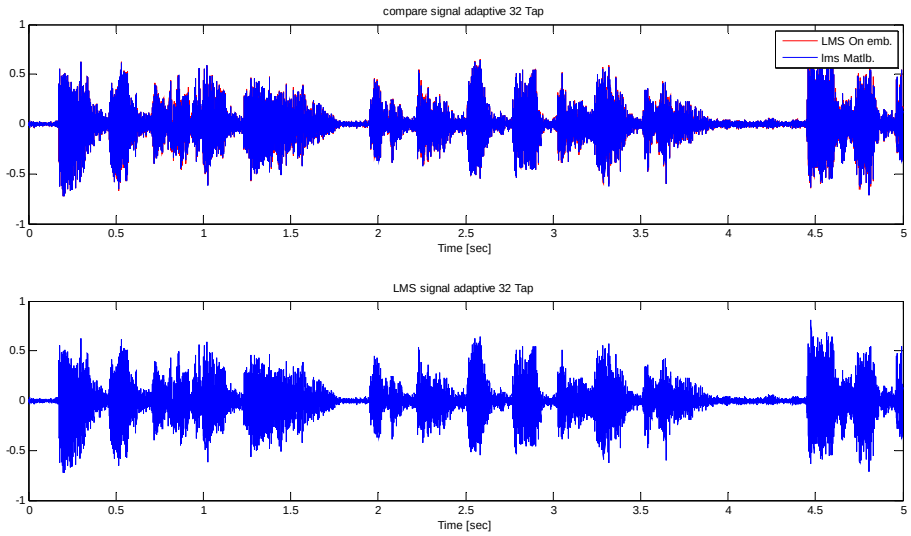
เมื่อเปรียบเทียบจำนวนรอบในการทำงานของวงจกรกรองปรับตัวได้แบบ LMS และ DLMS จากตารางที่ 4.3 วงจรที่ใช้ทรัพยากรร่วมกันทั้ง 2 แบบนี้ วงจรกรองปรับตัวได้แบบ LMS มีการทำงานที่ 15 รอบ ส่วน วงจรกรองปรับตัวได้แบบ DLMS มีการทำงาน 11 รอบ และที่นำเสนอในงานวิจัยนี้มีการทำงานที่ 6 รอบ และ 8 รอบ สำหรับการออกแบบวงจรกรองปรับตัวได้แบบ DLMS ด้วยไปป์ไลน์ละเอียด

4.2 การทดสอบวงจร

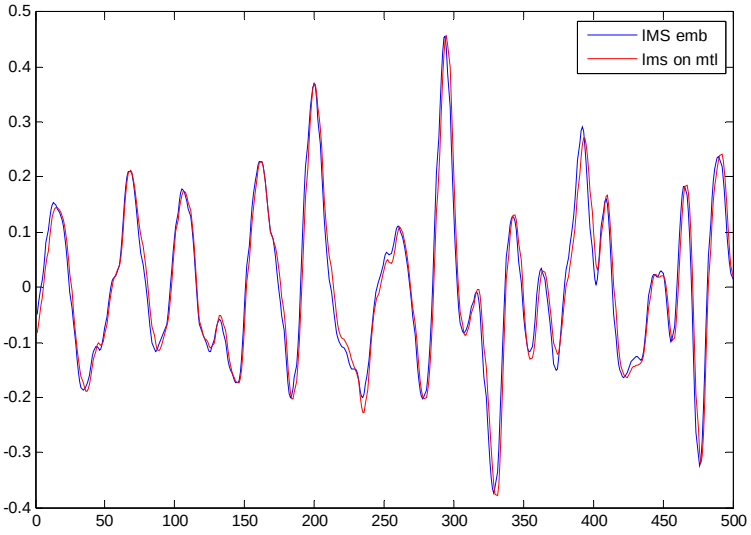
ในการทดสอบวงจรที่ออกแบบด้วย AccelDSP Tool โดยสร้างสัญญาณอินพุตจากวงจรนำมาเก็บในหน่วยความจำที่เป็นไฟล์ข้อมูล d(n) และ x(n) มีข้อมูลทั้งหมด 16000 ตัวอย่าง (Samples) มาเป็นข้อมูล โดยมี Address Generator ทำหน้าที่ ควบคุมการอ่านจากไฟล์ เพื่อส่งต่อไปยัง Unit Under Test ซึ่งเป็นวงจรที่แปลงมาจาก AccelDSP Tool เป็นภาษา Verilog HDL ดังแสดงในภาพประกอบ 4-1



ภาพประกอบ 4-1 โมเดลการทดสอบการทำงานของวงจรกรองปรับตัวได้แบบไม่ใช้ทรัพยากรร่วม

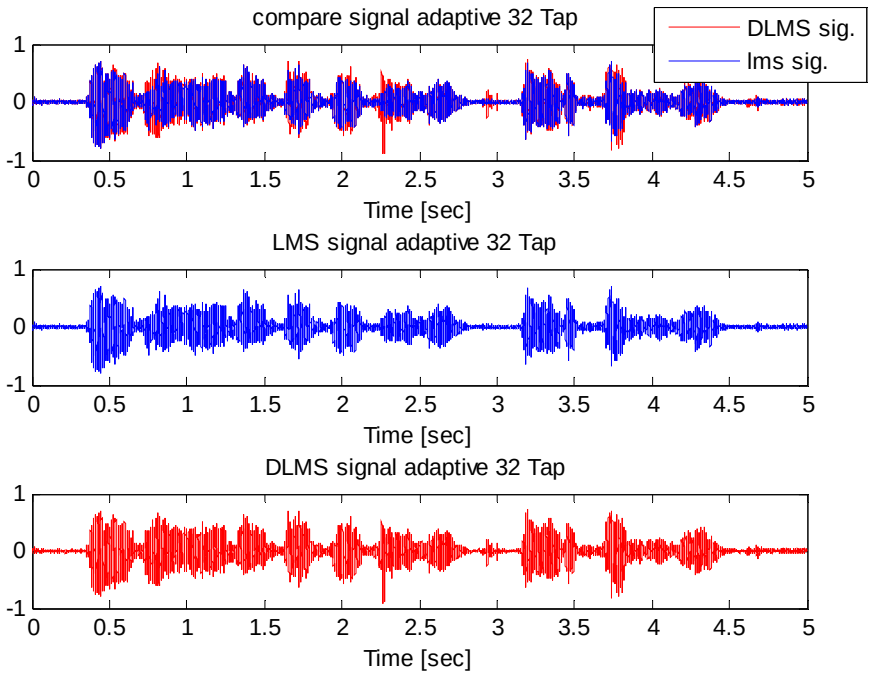


ภาพประกอบ 4-2 การทดสอบการทำงานของวงจรกรองปรับตัวได้ LMS แบบไม่ใช้ทรัพยากรร่วม



ภาพประกอบ 4-3 เปรียบเทียบสัญญาณที่ได้จากวงจรกรองปรับตัวได้ LMS ใน MATLAB กับผลที่ได้จาก FPGA แบบไม่ใช้ทรัพยากรร่วม

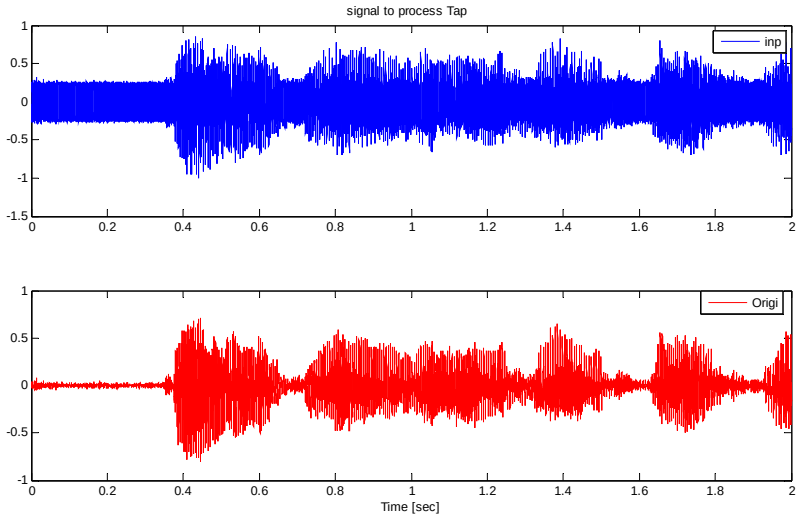
เมื่อทำการทดสอบสัญญาณของวงจรที่สร้างขึ้นใน FPGA เพื่อทำการเปรียบเทียบกับผลของ MATLAB สัญญาณที่ได้มีความใกล้เคียงกันในภาพประกอบ 4-2 และ 4-3 ซึ่งเป็นวงจรกรองปรับตัวได้แบบ LMS



ภาพประกอบ 4-4 เปรียบเทียบสัญญาณที่ได้จากวงจรกรองปรับตัวได้ LMS และ DLMS

จากภาพประกอบ 4-4 เมื่อทำการเปรียบเทียบสัญญาณของวงจรกรองปรับตัวได้แบบ LMS และ DLMS สัญญาณที่ได้จากวงจรกรองปรับตัวได้แบบ DLMS มีค่าความผิดพลาด เมื่อทำการเปรียบเทียบกับสัญญาณต้นแบบ โดยที่ค่าผิดพลาด MSE (Mean Square Error) ของวงจรกรองปรับตัวได้แบบ LMS มีค่า MSE = 0.0571 และค่าผิดพลาดของวงจรกรองปรับตัวได้แบบ DLMS มีค่า MSE = 0.0644 ซึ่งค่าความแตกต่างระหว่างวงจรกรองปรับตัวได้ทั้งสองแบบ มีค่าเท่ากับ 0.0073 โดยใช้การสุ่มสัญญาณมาเพียง 2000 สัญญาณ

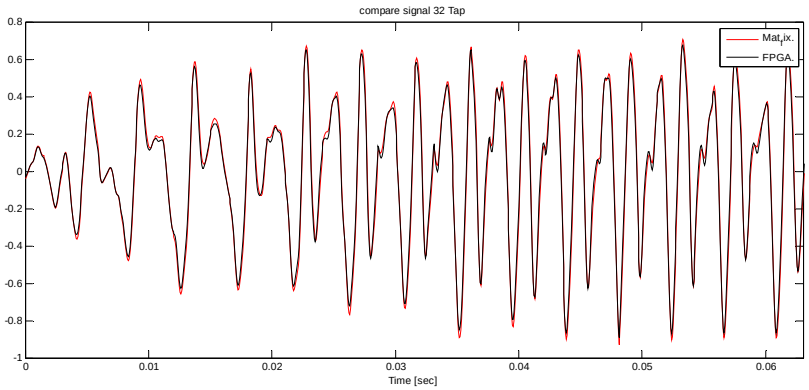
การทดสอบวงจรกรองปรับตัวได้แบบ DLMS ด้วยการออกแบบการใช้ทรัพยากรร่วม ทำการทดสอบเช่นเดียวกับแบบแบบภาพประกอบ 4-1 แต่เอาต์พุตจะมีการเปลี่ยนแปลงทุก 6 รอบสัญญาณนาฬิกา



ภาพประกอบ 4-5 การสร้างสัญญาณเสียงที่ใช้ในการทดสอบ

สัญญาณในภาพประกอบ 4-5 ซึ่งประกอบด้วยสัญญาณอินพุตถูกสร้างขึ้นให้มีสัญญาณรบกวน ความถี่ 1 kHz (บน) โดยผ่านวงจรกรองที่มีผลตอบสนองอิมพัลส์ของเครื่องช่วยฟัง รวมมากับสัญญาณเสียงพูด ส่วนในภาพด้านล่างเป็นสัญญาณเสียงที่ไม่มีสัญญาณรบกวน

จากภาพประกอบ 4-6 สัญญาณที่ได้จากวงจรกรองปรับตัวได้แบบ DLMS ที่มีการใช้ทรัพยากรร่วมใน FPGA มีการดีเลย์สัญญาณเท่ากับ 2 มีความใกล้เคียงกับสัญญาณที่ได้จากโปรแกรม MATLAB

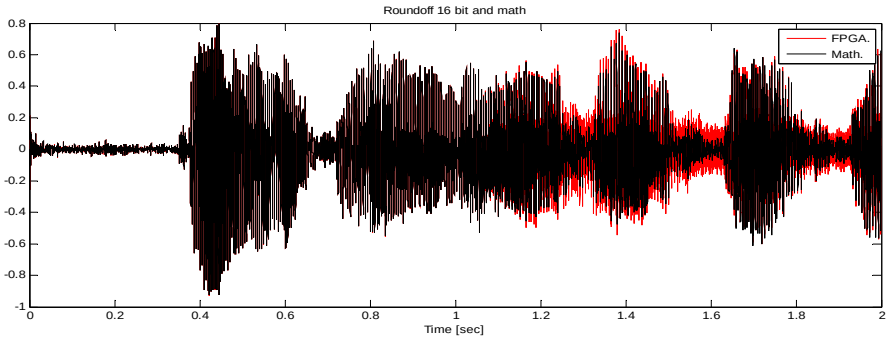


ภาพประกอบ 4-6 สัญญาณเสียงเอาต์พุตที่ผ่านวงจรกรองปรับตัวได้แบบ DLMS เปรียบเทียบกับสัญญาณที่ได้จาก MATLAB

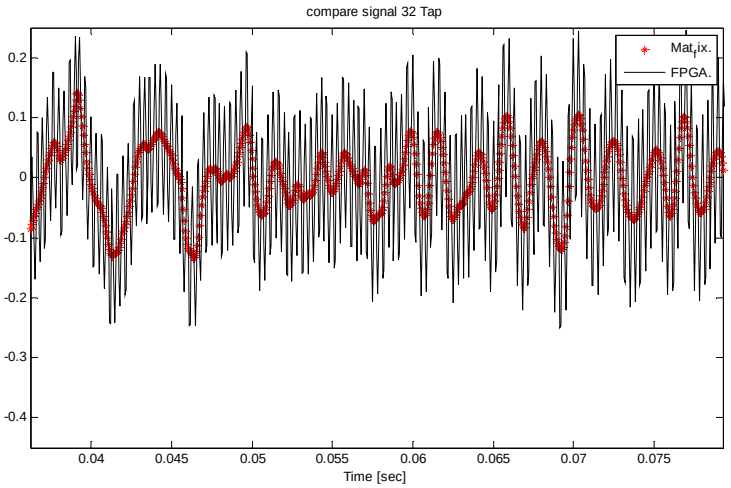
การประมวลผลด้วยไปป์ไลน์ละเอียดมีผลเหมือนกันใน ภาพประกอบ 4-6 เนื่องจากมีการดีเลย์สัญญาณออกไป 2 หน่วยเท่ากัน แต่การประมวลด้วยไปป์ไลน์ละเอียดมีรอบการทำงาน 8 รอบสัญญาณนาฬิกา

4.3 ผลจากการปัดเศษ (Round off)

ผลจากการปัดเศษหลังจากประมวลผลเสร็จ คือ เมื่อทำการคูณค่าของสัญญาณกับค่าสัมประสิทธิ์ได้เอาต์พุตขนาด 32 บิต แล้วทำการสเกลให้เหลือ 16 บิต เพื่อนำไปบวกต่อ เมื่อนำมาบวกก็ต้องสเกลค่าให้เหลือเพียง 16 บิต ก่อนที่จะนำไปบวกตัวถัดไป ซึ่งวิธีการปัดเศษแบบนี้ทำให้สัญญาณเอาต์พุตมีค่าผิดเพี้ยนไปจากโปรแกรม MATLAB ในช่วงปลาย (สัญญาณสีแดง) อยู่เพียงเล็กน้อย ซึ่งได้ค่าคำตอบจากการประมวลผลด้วย FPGA แสดงในภาพประกอบ 4-7



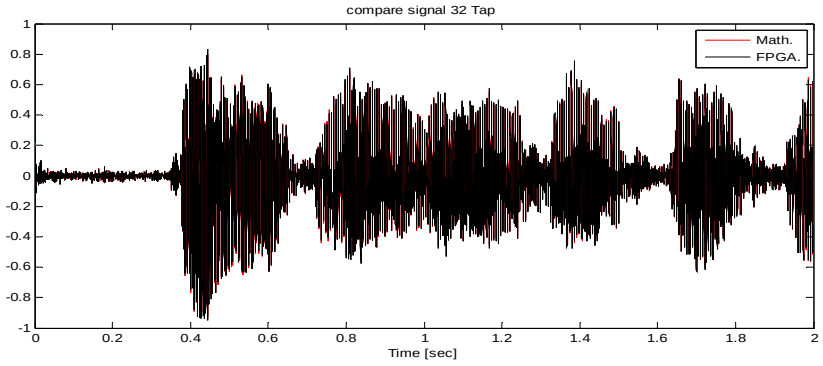
ภาพประกอบ 4-7 ผลจากกระบวนการปัดเศษหลังจากการประมวลผล



ภาพประกอบ 4-8 เปรียบเทียบกระบวนการปัดเศษหลังจากประมวลผลกับ MATLAB

จากภาพประกอบ 4-8 แสดงค่าของความแตกต่างระหว่างการประมวลด้วย MATLAB กับการประมวลสัญญาณด้วย FPGA สัญญาณที่จาก FPGA นั้นมีการเปลี่ยนแปลงไม่คงที่ไปกับการประมวลผลด้วย MATLAB เนื่องจากกระบวนการปัดเศษที่ได้หลังจากการประมวลผลเสร็จที่ละขั้น แล้วนำค่าที่ได้ไปทำการประมวลผลสัญญาณต่อจนเสร็จกระบวนการ

ผลจากการปัดเศษในครั้งสุดท้ายเพียงครั้งเดียว คือ เมื่อทำการคูณค่าของสัญญาณกับค่าสัมประสิทธิ์ได้เอาต์พุตขนาด 32 บิต แล้วทำการสเกลให้เหลือ 16 บิต เพื่อนำไปบวกต่อ ผลจากการบวกค่าเลข 16 บิต กับ 16 บิต ได้ผลลัพธ์เท่ากับ 19 บิต เป็นการจองข้อมูลบิตเพิ่มมา เมื่อนำมาบวกตัวถัดไปก็ได้ผลลัพธ์ขนาด 19 บิต จนถึงตัวสุดท้ายแล้วจึงทำการปัดเศษให้เหลือ 16 บิต ซึ่งเป็นค่าของเอาต์พุตที่ได้จากวงจรกรองเอฟไออาร์ ก่อนที่นำไปถูกลบด้วยสัญญาณ d(n) วิธีการปัดเศษแบบนี้ทำให้สัญญาณเอาต์พุตมีค่าใกล้เคียงกับโปรแกรม MATLAB มาก ซึ่งแสดงในภาพประกอบ 4-9 ซึ่งถูกนำไปใช้ในการออกแบบวงจรด้วยวิธีนี้ให้ความถูกต้องมากกว่าแบบแรก



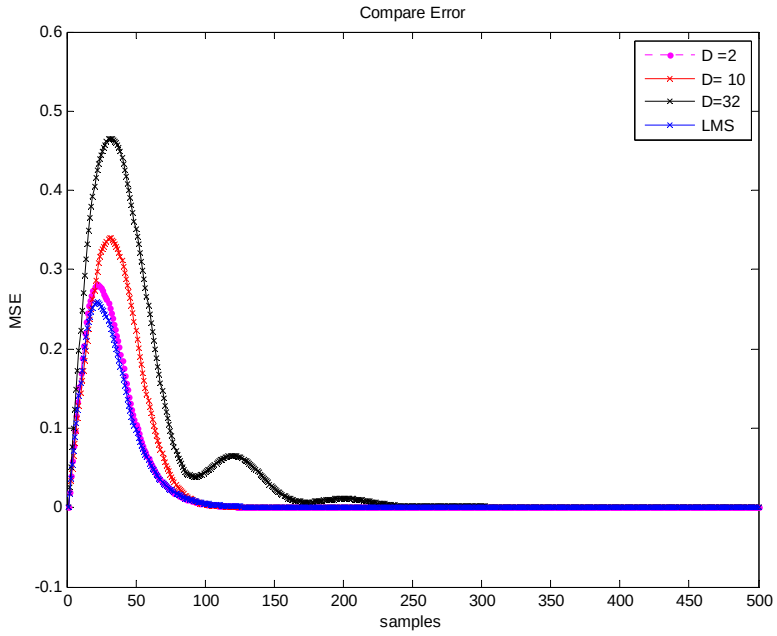
ภาพประกอบ 4-9 ผลจากกระบวนการปัดเศษในครั้งสุดท้ายเพียงครั้งเดียว

4.4 การทดสอบอัตราการลู่เข้า

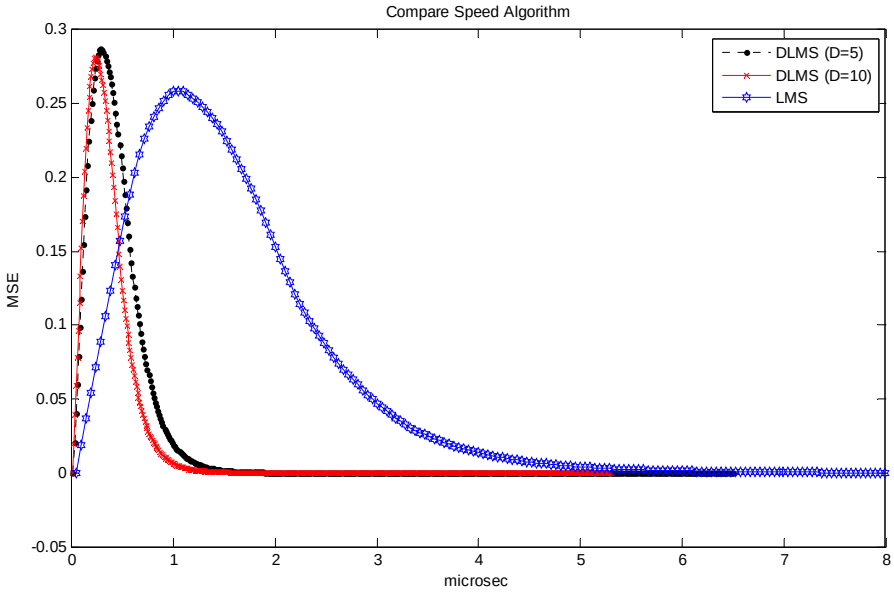
การเปรียบเทียบสัญญาณเอาต์พุตของวงจรกรองปรับตัวได้ว่าใกล้เคียงกับสัญญาณอินพุตที่ป้อนเข้ามาในวงจรกรองปรับตัวได้ มีการเปรียบเทียบอัตราการลู่เข้าของวงจรว่ามีความเร็วเท่าไร ที่สามารถกำจัดสัญญาณรบกวนออกไปจากระบบได้

ภาพประกอบ 4-10 เมื่อเปรียบเทียบวงจรกรองปรับตัวได้แบบ DLMS ที่มีค่าดีเลย์เท่ากับ 2, 10 และ 32 เปรียบเทียบกับวงจรกรองปรับตัวได้แบบ LMS ซึ่งมีอัตราการลู่เข้าที่ประมาณ 110 ตัวอย่าง เร็วกว่าวงจรกรองปรับตัวได้แบบ DLMS ที่มีค่าดีเลย์เท่ากับ 2 และ 10 เพียงเล็กน้อย ส่วน DLMS ที่มีค่าดีเลย์เท่ากับ 32 มีการลู่เข้าอยู่ที่ประมาณ 250 ตัวอย่าง ซึ่งจะช้ากว่าเมื่อจำนวนดีเลย์ที่เพิ่มขึ้น

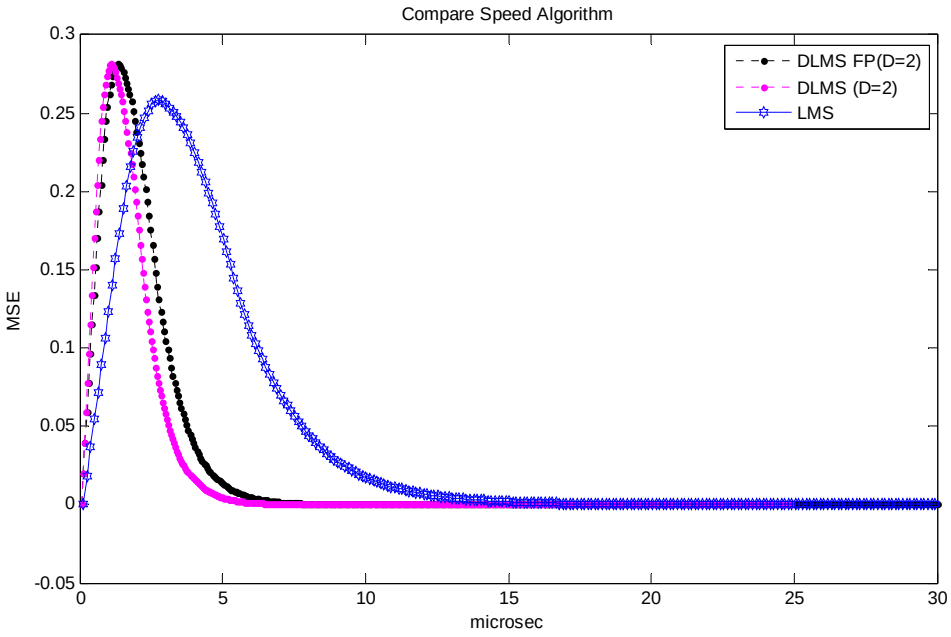
ภาพประกอบ 4-11 เมื่อทำการเปรียบเทียบเวลาของวงจรกรองปรับตัวได้แบบ LMS และ DLMS ของวงจรที่ไม่ใช้ทรัพยากรร่วม โดยที่วงจรกรองปรับตัวได้แบบ LMS มีอัตราการสุ่มตัวอย่างสูงสุดอยู่ที่ 21.33 MHz ส่วน DLMS มีอัตราการสุ่มตัวอย่างที่ 94.26 MHz วงจรกรองปรับตัวได้แบบ LMS มีอัตราการลู่เข้าอยู่ที่ 4.5 ไมโครวินาที มีจำนวน 120 ตัวอย่างในขณะที่วงจรกรองปรับตัวได้แบบ DLMS มีอัตราการลู่เข้าอยู่ที่ 1 ไมโครวินาที ที่มีจำนวนตัวอย่าง 150 ตัวอย่าง



ภาพประกอบ 4-10 เปรียบเทียบอัตราการอัตราลู่เข้าของวงจรกรองปรับตัวได้แบบต่างๆ

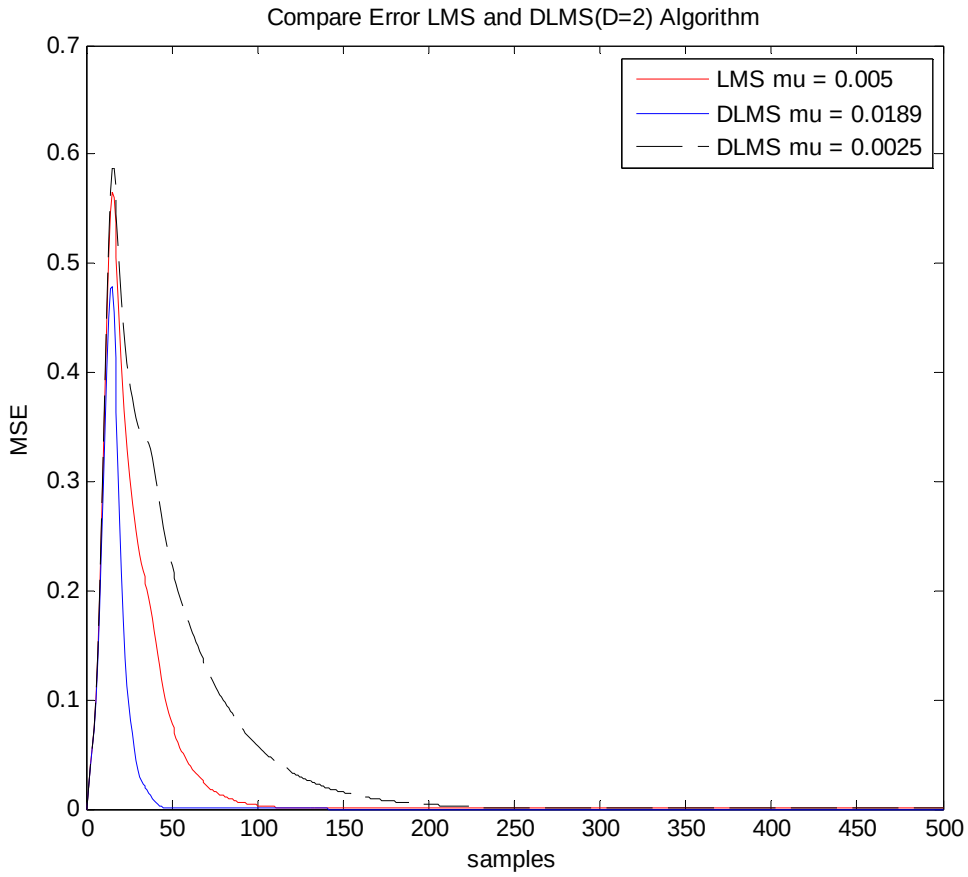


ภาพประกอบ 4-11 การเปรียบเทียบเวลาเมื่ออัตราการลู่เข้าของเวลาวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เป็นวงจรไม่ใช้ทรัพยากรร่วม



ภาพประกอบ 4-12 การเปรียบเทียบเวลาเมื่ออัตราการลู่เข้าของเวลาวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เป็นวงจรใช้ทรัพยากรร่วมกัน

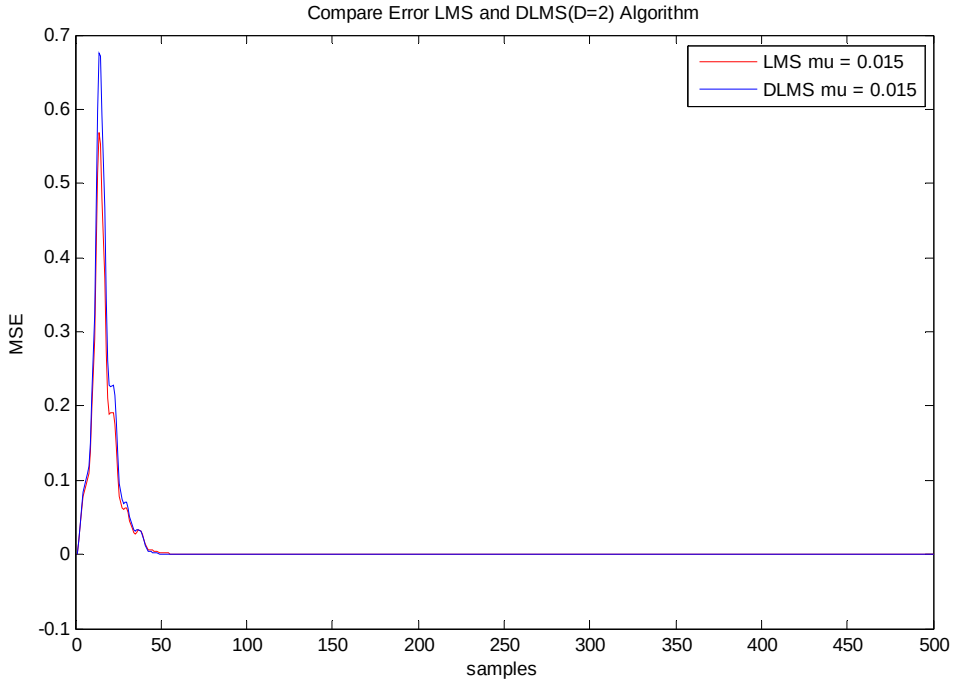
อัตราการลู่เข้าของวงจรที่ใช้ทรัพยากรร่วมกัน เมื่อเปรียบเทียบกันของวงจรกรองทั้งสองแบบ แสดงในภาพประกอบ 4-12 วงจรกรองปรับตัวได้แบบ DLMS (D=2) และแบบไปป์-ไลน์ละเอียด DLMS FP(D=2) มีความเร็วที่ประมาณ 5 และ 7 ไมโครวินาที ตามลำดับ ในขณะที่วงจรกรองปรับตัวได้แบบ LMS มีอัตราการลู่เข้าที่ 13 ไมโครวินาที ซึ่งช้ากว่าวงจรกรองปรับตัวได้แบบ DLMS 1.8-2.6 เท่า เมื่อเปรียบเทียบเวลาในการประมวลผล สำหรับการประมวลแบบใช้ทรัพยากรร่วมทำให้ใช้เวลาในการประมวลผลที่นานขึ้น แต่คาบเวลาในประมวลผลแต่ละรอบ (Clock period) น้อยกว่าแบบวงจรไม่ใช้ทรัพยากรร่วม



ภาพประกอบ 4-13 การเปรียบเทียบอัตราการลู่เข้าของวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เมื่อใช้อัตราการเรียนรู้ที่ค่าต่างๆ

เมื่อทดสอบใช้อัตราการเรียนรู้ที่ค่าต่างๆ ในภาพประกอบ 4-13 เมื่อใช้ค่าอัตราการเรียนรู้น้อย (μ = 0.0025) ทำให้อัตราการลู่เข้าของวงจรที่ช้า และเมื่อใช้อัตราการเรียนรู้ที่มาก (μ = 0.0189) ทำให้อัตราการลู่เข้าของวงจรเร็วขึ้น ดังนั้นอัตราการเรียนรู้ที่นำมาใช้ได้จากการ

คำนวณจากสัญญาณอินพุต เพื่อให้ได้ค่าเหมาะสม ทำให้วงจรทำงานได้ถูกต้อง สามารถกำจัดสัญญาณรบกวนออกไปได้



ภาพประกอบ 4-14 การเปรียบเทียบอัตราการลู่เข้าของเวลาวงจรกรองปรับตัวได้แบบ LMS กับวงจรกรองปรับตัวได้แบบ DLMS เมื่อปรับอัตราการเรียนรู้ที่เท่ากัน

เมื่อทำการปรับอัตราการเรียนรู้ของวงจรให้เท่ากันซึ่งเป็นค่าที่เหมาะสม แสดงในภาพประกอบ 4-14 ระหว่างวงจรกรองปรับตัวได้แบบ LMS และวงจรกรองปรับตัวได้แบบ DLMS มีค่าดีเลย์เท่ากับ 2 ทำให้อัตราการลู่เข้าใกล้เคียงกัน แต่วงจรกรองปรับตัวได้แบบ DLMS มีค่าผิดพลาดมากกว่า ประมาณ 0.1 เมื่อเปรียบเทียบกับวงจรกรองปรับตัวได้แบบ LMS

บทที่ 5
สรุปผลการวิจัยและข้อเสนอแนะ

ในบทนี้กล่าวถึงการสรุปผลและข้อเสนอแนะที่ได้จากการดำเนินการทำงานวิจัย ตลอดจนปัญหาและอุปสรรคที่เกิดขึ้นขณะทำงานวิจัย และให้ข้อเสนอแนะแก่ผู้สนใจจะนำงานวิจัยชุดนี้ไปพัฒนาต่อ

5.1 สรุปผล

การออกแบบวงจรกรองปรับตัวได้แบบ DLMS ในงานวิจัยนี้แบ่งออกเป็น 2 ส่วน คือ วงจรแบบไม่ใช้ทรัพยากรร่วมกัน และ วงจรที่ใช้ทรัพยากรร่วมกัน ทำการเปรียบเทียบผลกับวงจรกรองปรับตัวได้แบบ LMS บนชิพ FPGA รุ่น Xilinx Virtex-II Pro XC2VP30 ซึ่งผลที่ได้จากวงจรที่ไม่ใช้ทรัพยากรร่วมกันคือ ความเร็วในการทำงานของวงจรกรองปรับตัวได้แบบ DLMS (D=10) มากกว่าวงจรกรองปรับตัวได้แบบ LMS ที่ 4.41 เท่า ในขณะที่มีจำนวนพื้นที่ (Slices) เพิ่มขึ้นเพียง 33.43 % เท่านั้น กล่าวคือ DLMS มีความเร็วในการประมวลผลเท่ากับ 94.266 MSPS ในขณะที่ LMS มีความเร็วเท่ากับ 21.377 MSPS มีการใช้วงจรบวก 63 ตัว วงจรคูณ 65 ตัว ซึ่งทั้งสองวงจรใช้จำนวนวงจรบวกและวงจรคูณเท่ากัน แต่วงจร DLMS มีการใช้รีจิสเตอร์มากกว่า สำหรับการออกแบบวงจรที่ใช้ทรัพยากรร่วมกัน ของวงจรกรองปรับตัวได้แบบ DLMS มีความเร็วในการประมวลผลเท่ากับ 20.058 MSPS ส่วนความเร็วของวงจรกรองปรับตัวได้แบบ LMS อยู่ที่ 7.984 MSPS โดยใช้วงจรบวก 17 ตัว วงจรคูณ 17 ตัว

เมื่อเปรียบเทียบอัตราการลู่เข้าในเชิงจำนวนตัวอย่างที่ลู่เข้าของวงจรกรองปรับตัวได้แบบ DLMS พบว่าช้ากว่าวงจรกรองปรับตัวได้แบบ LMS แต่เมื่อเปรียบเทียบในเชิงเวลาในการประมวลผลสัญญาณ พบว่าวงจรกรองปรับตัวได้แบบ DLMS มีความเร็วกว่าวงจรกรองปรับตัวได้แบบ LMS ทำให้อัตราการลู่เข้าของสัญญาณเร็วกว่า โดยวงจรที่ไม่ใช้ทรัพยากรรวมกันมีอัตราการลู่เข้าของสัญญาณอยู่ที่ 1 μs ในขณะที่วงจรกรองปรับตัวได้แบบ LMS อยู่ที่ 4.5 μs สำหรับวงจรที่ใช้ทรัพยากรร่วมกันมีอัตราการลู่เข้าของสัญญาณวงจรกรองปรับตัวได้แบบ DLMS อยู่ที่ 5 μs ส่วนวงจรกรองปรับตัวได้แบบ LMS อยู่ที่ 13 μs

5.2 ปัญหาและอุปสรรคของการทำวิทยานิพนธ์

จากการออกแบบวงจรกรองปรับตัวได้แบบ DLMS ด้วยวิธีการใช้ทรัพยากรร่วม ปัญหาดังต่อไปนี้

ปัญหาของการจองบิตในวงจรกรองเอฟไออาร์ MAC (Multiply Accumulator) คือ วิธีการบวกในเลขทศนิยมแบบ 16 บิต เมื่อทำการบวกกันเพียง 1 ครั้งแล้วทำการสเกล (Scaling) ค่าให้เหลือเพียง 16 บิต แล้วนำผลลัพธ์ที่ได้ไปบวกกันต่อเนื่องกันในลักษณะแบบนี้ ผลลัพธ์ในขั้นสุดท้าย จะไม่ตรงกับตัวเลขที่เราทำการคำนวณในระบบเลขจำนวนเต็ม ทำให้ได้ค่าที่ผิดพลาดในผลลัพธ์ขั้นสุดท้ายออกมา เมื่อนำไปคำนวณต่อเนื่องกันก็จะให้ผลลัพธ์ที่ออกมากมาผิด โดยเมื่อนำไปปรับปรุงค่านำหนักของวงจรกรองปรับตัวได้ ทำให้ค่าไม่ถูกต้อง ส่งผลให้วงจรไม่สามารถกำจัดสัญญาณรบกวนออกจากระบบไปได้ ดังนั้นวิธีการแก้ปัญหานี้โดย การจองบิตข้อมูลของ MAC ไว้ที่ 20 บิต เมื่อทำการบวกข้อมูล 16 บิต จะให้ผลลัพธ์ของวงจรได้ 20 บิต และนำข้อมูลขนาด 20 บิตตัวนี้ไปบวกต่อเนื่องกัน แล้วให้ทำการสเกลค่าในขั้นสุดท้ายที่เดียวให้เหลือเพียง 16 บิต ซึ่งเป็นค่าของ y(n) แล้วนำไปลบกับข้อมูล d(n) แล้วนำค่าผิดพลาดไปทำการปรับปรุงค่าสัมประสิทธิ์ใหม่จะให้ค่าที่ถูกต้องมากกว่าแบบข้างต้นที่ได้กล่าวมานี้

การใช้ทรัพยากรร่วมกันสิ่งที่ต้องระมัดระวัง คือ ใช้รีจิสเตอร์ในการอ่านเขียนข้อมูลที่เวลาเดียวกัน ส่งผลให้เกิดความผิดพลาดจากการคำนวณได้มาก เนื่องจากระบบไม่รู้ชัดแน่นอนของข้อมูลที่ได้จากการอ่านหรือเขียน ดังนั้น การแก้ปัญหานี้ จะไม่ใช้รีจิสเตอร์การอ่านหรือเขียนไปพร้อมๆกัน ที่เวลาเดียวกัน หรือที่รอบการทำงานเดียวกัน

5.3 ข้อเสนอแนะ

ในการออกแบบวงจรกรองปรับตัวได้แบบ DLMS ที่มีค่าจำนวนแท็ปของวงจรกรองที่สูงๆสามารถทำได้ด้วยวิธีการออกแบบโดยใช้เทคนิคการจัดรูปแบบเวลาการประมวลผล (Retiming) ลักษณะนี้ได้เพื่อลดค่าดีเลย์ของวงจรกรองปรับตัวได้แบบ DLMS ในการสร้างวงจรที่ไม่ใช้ทรัพยากรร่วมจะเกิดดีเลย์ที่มาก หากทำการแก้ไข้ปัญหาด้วยวิธีแบบนี้จะช่วยลดปัญหาในกรณีนี้ได้ และวงจรสามารถประหยัดพลังงานในการประมวลผลสัญญาณได้ด้วย

วงจรกรองปรับตัวได้แบบ DLMS เมื่อใช้ทรัพยากรที่น้อยในการประมวลผลแบบใช้ทรัพยากรร่วม จะทำให้เกิดการดีเลย์เพิ่มขึ้นเพียงเล็กน้อยแต่จะใช้รอบการคำนวณที่นานขึ้น เมื่อใช้ทรัพยากรที่มากขึ้นทำให้เกิดดีเลย์เพิ่มขึ้นมาก เพราะสามารถทำงานได้รวดเร็ว จำนวนดีเลย์ที่เกิดขึ้นนี้อยู่ในวงจรวงจรกรองเอฟไออาร์

สัญญาณที่ทำการทดลองอยู่ที่ช่วง ± 1 หากนำไปประยุกต์ใช้งานที่มีช่วงของสัญญาณสูงหรือต่ำกว่านี้การทำสเกลค่า ควรปรับใหม่อีกรอบ เพื่อให้สามารถคิดหรือคำนวณได้ถูกต้อง เช่น ในวงจรกรองเอฟไออาร์ การคูณเลขทศนิยม การบวกเลข ต้องปรับให้เหมาะสมกับวิธีการนำไปใช้งาน

บรรณานุกรม

[1] C. Long, E. Ling, and J. G. Proakis, "The LMS algorithm with delayed coefficient adaptation," *IEEE Trans. on Acoust., Speech, Signal processing*, vol. 37, no.9, pp. 1397-1405, Sept.1989.

[2] M.D. Meyer and D.P. Agrawal, "A high sampling rate delayed LMS filter architecture**,"** *IEEE Trans. on Circuits and Systems I,* vol.40, no.11, pp. 727-729, Nov.1993.

[3] S. Ramanathan and V. Visvanathan, "Low-power pipeline LMS adaptive filter architectures with minimal adaption delay," *the VLSI Journal Integration,* vol.27, pp. 1-32, Jan.1999.

[4] E. Mahfuz, C. Wang and M.O. Ahmad, "A high-throughput DLMS adaptive algorithm," *IEEE International Symposium on Circuits and Systems,* vol.4, pp. 3753-3756, May 2005.

[5] L. Ting, R. Woods, and C. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems,* vol.13, pp. 86-95, Jan.2005.

[6] J.M. Kates, "Feedback cancellation in hearing aids : result from a computer simulation," *IEEE Ttrans. on Signal Processing*, vol.39, no.3, pp. 553-562, Mar.1991.

[7] J. Maxwell and P. Zurek, "Reducing acoustic feedback in hearing aids," *IEEE Trans. on Speech Audio Processing*, vol.3, pp. 304-313, Jul.1995.

[8] J. Yang, M.T. Tan and J.S. Chang, "Modeling external feedback path of an ITE digital hearing instrument for acoustic feedback cancellation," *IEEE International Symposium on Circuits and Systems,* vol.2, pp. 1326–1329, May 2005.

[9] K. Parhi, *VLSI digital signal processing systems: design and implementation*, A wiley-interscience publication, USA. 1999.

[10] ชำนาญ ปัญญาใส และ วัชรากร หนูทอง, *ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล*, กรุงเทพฯ:ซีเอ็ดยูเคชั่น,2547.

[11] A.J. Al-Khalili, "Parallel Adder," online http://users.encs.concordia.ca/~asim/COEN_6501/Lecture_Notes/L2_Notes.pdf (สืบค้นเมื่อ 4 กรกฎาคม 2554).

[12] E. L.Oberstar, " Fixed-point representation & fraction math," Online http://www.superkits.net (สืบค้นเมื่อ 15 พฤษภาคม 2554).

[13] ณัฏฐา จินดาเพ็ชร์, *การออกแบบและสร้างแบบจำลอง วงจรรวมดิจิตอลด้วยภาษา Verilog HDL (Design and Modeling Digital VLSI with Verilog HDL)*, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ 2554.

[14] วุฒิ วิริยะสม, *การออกแบบหน่วยประมวลผลคณิตศาตร์ความเร็วสูงสำหรับวงจรกรองปรับตัวได้บน FPGAs*, วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต, สาขาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยสงขลานครินทร์, 2550.

[15] เกรียงไกร ทัศนวิภาส, *การพัฒนาต้นแบบเครื่องช่วยฟังดิจิตอลพลังงานต่ำและขนาดเล็กโดยใช้ดิจิตอลฟิลเตอร์แบงก์*, วิทยานิพนธ์วิศวกรรมศาสตร์มหาบัณฑิต, สาขาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยสงขลานครินทร์, 2552.

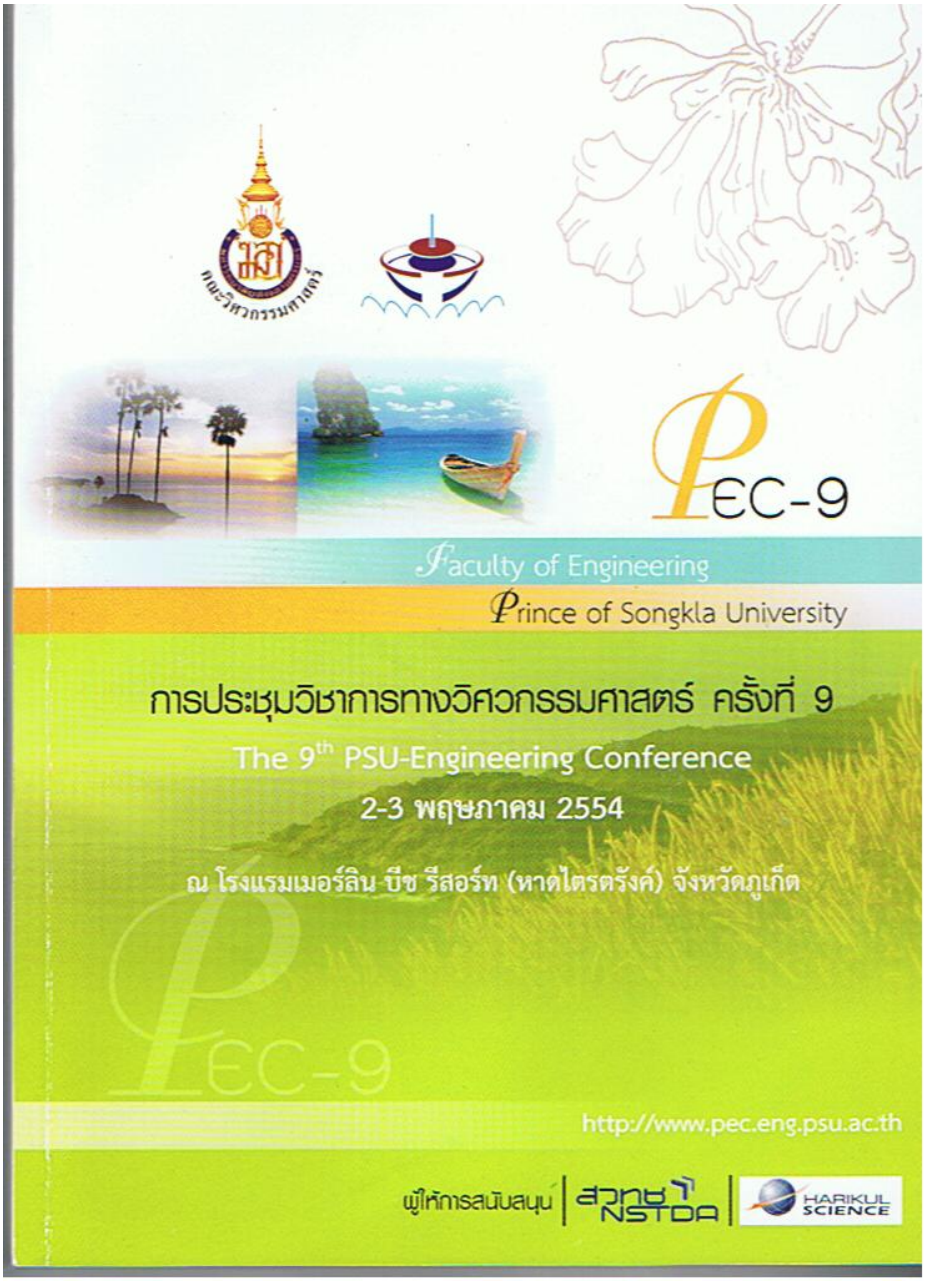
[16] Xilinx Corp, http://www.xilinx.com. (สืบค้นเมื่อ 10 ตุลาคม 2553).

[17] พรชัย ภววงษ์ศักดิ์, *การประมวลผลสัญญาณดิจิตอลเบื้องต้น Introduction to Digital Signal Processing Revision 1.0,* Online http://www.ee.mut.ac.th/home/suriya/dsp-pornchai.pdf (สืบค้นเมื่อ 9 ธันวาคม2551).

ภาคผนวก

ภาคผนวก

ผลงานตีพิมพ์เผยแพร่วิทยานิพนธ์



การประชุมวิชาการทางวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์ ครั้งที่ 9
2-3 พฤษภาคม 2554

การออกแบบวงจรกรองปรับตัวได้แบบ DLMS สำหรับการกำจัดสัญญาณเสียงสะท้อนในระบบเครื่องช่วยฟัง

Design DLMS Adaptive filter for Echo cancellation in Hearing Aids

สายัณ ละอองโชค ณัฏฐา จินดาเพ็ชร์ พรชัย พฤกษ์ภัทรานนต์ สมพัฒน์ รุ่งตะวันเรืองศรี
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ อ.หาดใหญ่ จ.สงขลา 90112
Sayan Laongchok Nattha Jindapetch Pornchai Phukpattaranont Sompat Roongtawanreongsri
Department of Electrical Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla 90112
E-mail:sayan_klh@hotmail.com , nattha.s@psu.ac.th, pornchai.p@psu.ac.th, sompat.r@psu.ac.th

บทคัดย่อ

บทความนี้นำเสนอการออกแบบวงจรกรองแบบปรับตัวได้แบบ DLMS ขนาด 32 Taps แบบ fixed point 16 บิต โดยการออกแบบใช้ไปป์ไลน์ เพื่อเพิ่มความเร็วในการทำงานของวงจรให้สูงขึ้น ในขณะที่การใช้ทรัพยากรเพิ่มขึ้นเพียงเล็กน้อย เมื่อทำการเปรียบเทียบกับวงจรกรองแบบปรับตัวได้แบบ LMS เพื่อนำวงจรที่ได้จากการออกแบบไปกำจัดสัญญาณสัญญาณเสียงสะท้อนในระบบเครื่องช่วยฟัง จากการทดสอบวงจรกรองปรับตัวได้แบบ DLMS (ดีเลย์=10) สามารถเพิ่มความเร็วในการทำงานเป็น 4.41 เท่า ขนาดพื้นที่เพิ่มขึ้น 1.39 เท่า บน Xilinx Virtex-II Pro XC2VP30

คำหลัก วงจรกรองปรับตัวได้แบบ LMS, วงจรกรองปรับตัวได้แบบ DLMS, ไปป์ไลน์

Abstract

This paper presents a design of a delayed least mean square (DLMS) adaptive filtering algorithm for echo cancellation in hearing aids. The circuit was implemented in 32 taps length and 16 bit fixed-point arithmetic representation. The pipelining technique was applied to increase the speed while slightly increase the resources. The experimental results shows that the DLMS (delay=10) circuit can run at speed 4.41 times faster than the LMS adaptive filter while the area (slices) was only 1.39 times on Xilinx Virtex-II Pro XC2VP30.

Keywords: LMS Adaptive filter, DLMS Adaptive filter,pipelined

1.บทนำ

ปัญหาการเกิดสัญญาณเสียงสะท้อนในระบบเครื่องช่วยฟังซึ่งเป็นอุปกรณ์ขนาดเล็กสำหรับผู้มีปัญหาทางการได้ยิน ทำให้การสื่อสารมีปัญหาไม่เป็นที่น่าพอใจต่อผู้ใช้งาน สาเหตุมาจากการจัดวางไมโครโฟนและลำโพงที่มีความใกล้ชิดและผลของอัตราการขยายในเครื่องช่วยฟัง จึงเกิดสัญญาณเสียงสะท้อนป้อนกลับเข้าสู่ไมโครโฟนอีกครั้ง จากการวิเคราะห์แนวทางการแก้ปัญหาที่เกิดขึ้นนี้โดยอาศัยอัลกอรึทึมทางคณิตศาสตร์ มีอยู่หลายแบบ เช่น วงจรกรองปรับตัวได้แบบ LMS [1] ซึ่งผลของวงจรมีอัตราการลู่เข้าของวงจรที่รวดเร็วกว่าวงจรกรองปรับตัวได้แบบ DLMS [2] เมื่อพิจารณาจำนวนตัวอย่าง (samples) ของสัญญาณเอาท์พุทที่ได้รับและวงจรกรองปรับตัวได้แบบ NLMS [3] ถูกพัฒนาให้สามารถเปลี่ยนอัตราการเรียนรู้เพื่อให้ลู่เข้าเร็วขึ้น แต่เมื่อเปรียบเทียบการทำงานของฮาร์ดแวร์ทางด้านความเร็ว วงจรกรองปรับตัวได้แบบ DLMS มีความเหมาะสม สามารถทำงานที่ความเร็วสูงกว่าวงจรกรองปรับตัวได้แบบ LMS และวงจรกรองปรับตัวได้แบบ NLMS ซึ่งมีโครงสร้างที่ซับซ้อน

ในงานวิจัยนี้ได้ทำการศึกษาออกแบบวงจรกรองปรับตัวได้แบบ DLMS เปรียบเทียบกับวงจรกรองปรับตัวได้แบบ LMS ทางด้านความเร็วการทำงานของวงจรและการใช้ทรัพยากรต่างๆ บน Xilinx Virtex-II Pro XC2VP30 โดยใช้ MATLAB/Simulink Xilinx Accel DSP Toolbox ในการออกแบบวงจร

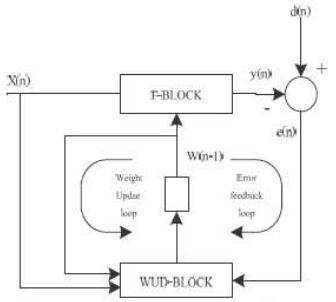
2. วงจรกรองปรับตัวได้ (Adaptive filter)

วงจรกรองดิจิตอลแบบ FIR และ IIR เป็นวงจรกรองสัญญาณที่มีค่าน้ำหนักคงที่ เมื่อนำมาใช้ในระบบที่มีการเปลี่ยนแปลงตามเวลา ก็ไม่สามารถลดสัญญาณรบกวนลงได้ แต่สำหรับวงจรกรองปรับตัวได้สามารถปรับค่าน้ำหนักของวงจรให้

สามารถเปลี่ยนแปลงตามเวลา ทำให้สามารถลดสัญญาณรบกวนลงได้ ซึ่งวงจรกรองปรับตัวได้มีอยู่หลายแบบ ในหัวข้อนี้จะกล่าวถึงวงจรกรองปรับตัวได้ LMS และ DLMS

2.1 วงจรกรองปรับตัวได้แบบ LMS (least Mean Square adaptive filter)

วงจรกรองแบบปรับตัวได้ LMS (Least Mean Square) [1] เป็นวงจรกรองซึ่งภายในโครงสร้างประกอบด้วย F-block ในรูปที่ 1 เป็นวงจรกรองแบบ FIR จากสมการที่ (1)- (3) ค่าประมาณที่ต้องการ $y(n)$ ได้ผลรวมจากการคูณค่าน้ำหนักกับสัญญาณอินพุท $\mathbf{x(n)}$ ที่ป้อนเข้ามาและ WUD-Block เป็นวงจรปรับค่าน้ำหนักโดยนำค่าประมาณความผิดพลาด $e(n)$ ได้จาก สัญญาณที่ต้องการ $d(n)$ ลบด้วยค่าประมาณที่ต้องการ $y(n)$ นำค่า $e(n)$ มาปรับค่าสัมประสิทธ์ $\mathbf{w(n)}$ ใหม่ทุกๆ รอบ ส่วน N เป็นจำนวน Tap ฟิลเตอร์ μ ซึ่งเป็นค่าอัตราเรียนรู้ของระบบ



รูปที่ 1 บล็อกไดอะแกรมวงจรกรองแบบปรับค่าได้แบบ LMS

$$\mathbf{y(n) = W^T}(n)\mathbf{X}(n) \quad (1)$$

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

$$\mathbf{w(n+1) = w(n)} + \mu e(n)\mathbf{x(n)} \quad (3)$$

โดยที่ $\mathbf{W(n)} = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^T$

$\mathbf{x(n)} = [x(n), x(n-1), \ldots, x(n-N+1)]^T$

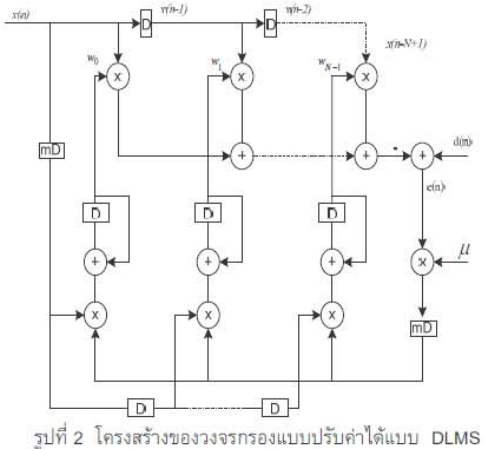
2.2 วงจรกรองปรับตัวได้แบบ DLMS (Delayed LMS adaptive filter)

วงจรกรองปรับตัวได้แบบ DLMS [2] แสดงดังสมการที่ (4)-(6) และรูปที่ 2 ได้ถูกพัฒนาจาก LMS เพื่อเป็นประโยชน์ในการทำไปป์ไลน์ เนื่องจากอัลกอริธึมที่เป็น LMS มีการป้อนกลับไม่สามารถทำไปป์ไลน์ในวงจรได้ สัญญาณค่าประมาณที่ต้องการ $y_D(n)$ ได้จากผลรวมจากการคูณค่าสัมประสิทธ์ $\mathbf{w(n)}$ กับสัญญาณอินพุท $\mathbf{x(n)}$ ใช้วงจรคูณและวงจรบวกเป็นแบบไปป์ไลน์ โดยที่ mD คือ จำนวนดีเลย์ในวงจรกรองปรับตัวได้แบบ DLMS ค่าประมาณความผิดพลาด $e_D(n)$ นำมาปรับปรุงค่าสัมประสิทธ์ใหม่อีกครั้ง และสัญญาณอินพุท $\mathbf{x(n)}$ ที่นำมาปรับค่าสัมประสิทธ์ต้องมีการดีเลย์เท่ากับจำนวนดีเลย์ที่เกิดขึ้นในส่วนของ F-BLOCK รวมกับจำนวนดีเลย์ผลคูณของอัตราการเรียนรู้

$$\mathbf{y}_D(n) = \mathbf{W}^T(n - mD)\mathbf{X}(n - mD) \quad (4)$$

$$e_D(n) = d(n - mD) - y(n - mD) \quad (5)$$

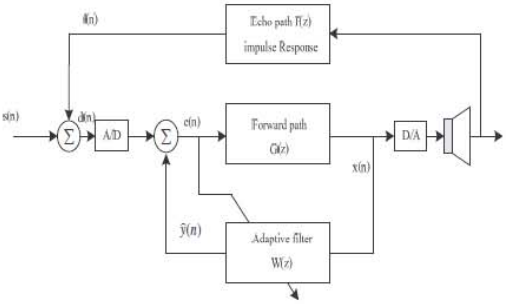
$$\mathbf{w(n+1) = w}(n) + \mu e(n)\mathbf{x}(n - mD) \quad (6)$$



รูปที่ 2 โครงสร้างของวงจรกรองแบบปรับค่าได้แบบ DLMS

2.3 วงจรกรองปรับตัวได้ในเครื่องช่วยฟัง

ปัญหาที่เกิดขึ้นในระบบเครื่องช่วยฟัง คือ สัญญาณเสียงสะท้อนที่มาจากลำโพงป้อนกลับมาสู่ไมโครโฟน เนื่องจากเป็นอุปกรณ์ขนาดเล็กมีการวางใกล้ชิดกัน



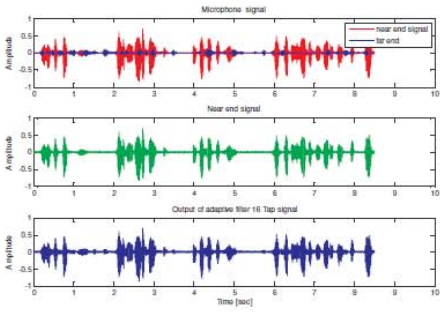
รูปที่ 3 บล็อกไดอะแกรมของเครื่องช่วยฟัง

จากบล็อกไดอะแกรมของเครื่องช่วยฟังในรูปที่ 3 ประกอบด้วย Forward path G(z) ทำหน้าที่ขยายสัญญาณ Echo path impulse Response F(z) คือ ผลตอบสนองของอิมพัลส์ [3] สัญญาณเสียงสะท้อนในระบบเครื่องช่วยฟังแบบ ITE และวงจรกรองปรับตัวได้ Adaptive filter W(z) ซึ่งทำหน้าที่ กำจัดสัญญาณเสียงสะท้อนที่เกิดขึ้นในระบบเครื่องช่วยฟัง ที่เกิดขึ้นจากผลตอบสนองของอิมพัลส์

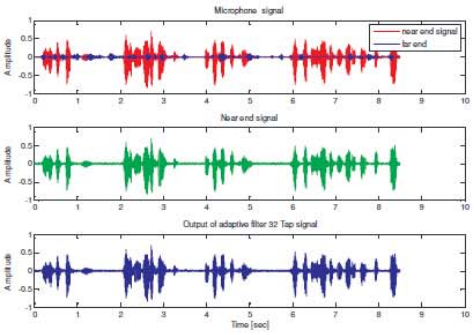
3. การวิเคราะห์และออกแบบวงจรกรองปรับตัวได้

3.1 ทำการทดสอบสัญญาณของวงจรกรองปรับตัวได้แบบ LMS ด้วยโปรแกรม MATLAB

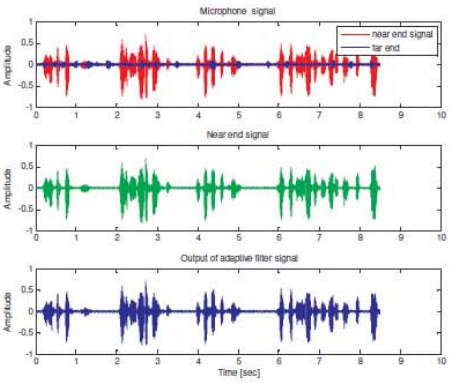
สัญญาณเสียงที่นำมาทดลองมีอัตรา sampling Rate 16 KHz ประกอบด้วย สัญญาณเสียงพูด (nearend signal) ผสมกับสัญญาณเสียงสะท้อน (far end) ที่เกิดขึ้นจากการสร้างด้วยผลตอบสนองอิมพัลส์ของเครื่องช่วยฟัง และสัญญาณอ้างอิง ผ่านวงจรกรองปรับตัวได้ขนาด 16, 32, 64 Taps ผลการทดลองด้วยโปรแกรม MATLAB สามารถกรองสัญญาณเสียงสะท้อนที่เกิดขึ้นได้ขึ้นกับขนาดความยาวของวงจรกรอง



รูปที่ 4 สัญญาณเสียงที่ผ่านวงจรกรองปรับตัวได้ LMS 16 Taps



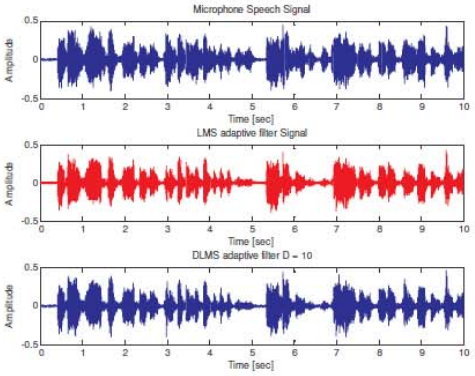
รูปที่ 5 สัญญาณเสียงที่ผ่านวงจรกรองปรับตัวได้ LMS 32 Taps



รูปที่ 6 สัญญาณเสียงที่ผ่านวงจรกรองปรับตัวได้ LMS 64 Taps

จากรูปที่ 4,5,6 (บน) สัญญาณเสียงที่ผ่านไมโครโฟนประกอบด้วย สัญญาณเสียงพูดรวมกับสัญญาณเสียงสะท้อนที่ผ่านผลตอบสนองอิมพัลส์ (กลาง) สัญญาณเสียงพูด และ(ล่าง) สัญญาณที่ผ่านวงจรกรองแบบปรับตัวได้ 16, 32, 64 Taps ซึ่งผลของวงจรกรองสัญญาณเสียงสะท้อนที่ 32 และ 64 Taps สามารถกำจัดสัญญาณเสียงสะท้อนออกไปได้ขณะที่วงจรกรองขนาด 16 Taps ยังมีเหลือสัญญาณสะท้อนอยู่อีก ดังนั้นการ implement วงจรเลือกที่ 32 Taps เนื่องจากใช้ทรัพยากรน้อยกว่า 64 Taps แต่ให้คุณภาพสัญญาณเสียงที่ใกล้เคียงกัน

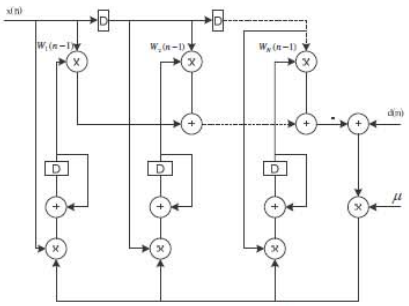
ผลจากวงจรกรองปรับตัวได้แบบ LMS เมื่อทำการเปรียบเทียบกับวงจรกรองปรับตัวได้แบบ DLMS ในรูปที่ 7 (บน) สัญญาณเสียงอินพุทซึ่งรวมกับสัญญาณเสียงสะท้อน (กลาง) สัญญาณเสียงที่ผ่านวงจรแบบ LMS (ล่าง) สัญญาณเสียงที่ผ่านวงจรกรองแบบ DLMS ซึ่งมีความใกล้เคียงกันกับวงจรกรองแบบ LMS ค่าของวงจรกรองปรับตัวได้แบบ DLMS มีค่าดีเลย์ (D = 10) มีค่าผิดพลาดเพียงเล็กน้อย แต่จะมากขึ้นหากค่าดีเลย์มากกว่านี้



รูปที่ 7 เปรียบเทียบสัญญาณเสียงที่ได้จากวงจรกรองปรับตัวได้แบบ LMS กับ DLMS

3.2 การออกแบบวงจรกรองปรับตัวได้แบบ LMS

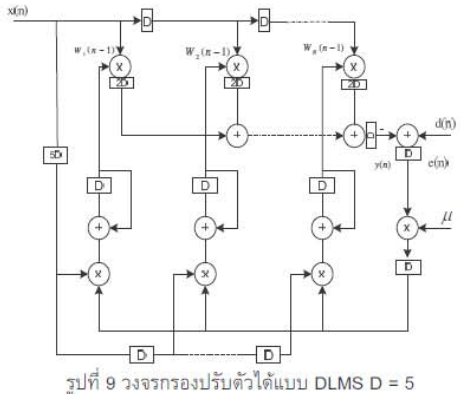
วงจรกรองปรับตัวได้แบบ LMS ที่นำมาออกแบบใช้จำนวน 32 Taps ใช้ MATLAB/Simulink Xilinx Accel DSP Toolbox ในการออกแบบวงจรดังรูปที่ 8 เป็นวงจรคูณและวงจรบวกไม่เป็นแบบไปป์ไลน์ โดยไม่มีการใช้ทรัพยากรร่วมกัน



รูปที่ 8 วงจรกรองปรับตัวได้แบบ LMS

3.3 การออกแบบวงจรกรองปรับตัวได้แบบ DLMS D = 5

วงจรกรองปรับตัวได้แบบ DLMS ขนาด 32 Taps มีหน่วยประวิงเวลา D = 5 ประกอบด้วย วงจรคูณที่เป็นแบบไปป์ไลน์ 2 ขั้น ทำการคูณกับค่าน้ำหนักกับสัญญาณอินพุท ส่วนในวงจรบวกได้ใส่ไปป์ไลน์ในขั้นสุดท้ายที่ 32 Taps 1 ขั้น และทำไปป์ไลน์ในวงจรลบกับวงจรคูณรวมเป็น 2 ขั้น ทั้งหมดเมื่อรวมกันเป็น 5 ดังนั้นการดีเลย์สัญญาณอินพุทมาทำการปรับค่าน้ำหนักของวงจรกรองปรับตัวได้แบบ DLMS มีค่าเท่ากับ 5 เช่นกัน ส่วนวงจรการป้อนกลับมาปรับค่าน้ำหนักใหม่ เพื่อนำมาคูณกับสัญญาณอินพุทในรอบใหม่ การออกแบบไม่ได้ทำไปป์ไลน์ของการปรับค่าน้ำหนัก



รูปที่ 9 วงจรกรองปรับตัวได้แบบ DLMS D = 5

3.4 การออกแบบวงจรกรองปรับตัวได้แบบ DLMS D = 7

วงจรกรองปรับตัวได้แบบ DLMS ขนาด 32 Taps มีหน่วยประวิงเวลา D = 7 ประกอบด้วย วงจรคูณที่เป็นแบบไปป์ไลน์ 3 ขั้น ทำการคูณกับค่าน้ำหนักกับสัญญาณอินพุท ทำไปป์ไลน์ 2 ขั้น ในวงจรบวกที่เป็นวงจรกรองแบบ FIR และเมื่อรวมไปป์ไลน์ในวงจรลบกับวงจรคูณระหว่างค่าความผิดพลาดกับค่าอัตราเรียนรู้ของระบบร่วมเป็น 2 ขั้น รวมทั้งหมดเป็น 7 ขั้น

3.5 การออกแบบวงจรกรองปรับตัวได้แบบ DLMS D = 10

โดยการทำไปป์ไลน์ในวงจรคูณ 3 ขั้น และทำไปป์ไลน์ในวงจรบวกได้สูงสุดเท่ากับ 5 ขั้น ($\log_2 N$) โดยที่ N คือ ลำดับของวงจรกรองแบบปรับตัวได้ และเมื่อรวมไปป์ไลน์ในวงจรลบกับวงจรคูณค่าอัตราการเรียนรู้อีก 2 ขั้น เท่ากับ 10

4 ผลการทดลอง

จากการออกแบบวงจรทั้ง 4 แบบที่กล่าวมาจากหัวข้อ 3.2-3.5 เป็นวงจรประมวลผลขนาด 16 บิต โดยใช้ MATLAB/Simulink Xilinx Accel DSP Toolbox ได้ผลดังหัวข้อย่อยต่อไปนี้

4.1 สรุปการใช้ทรัพยากรของ Xilinx Virtex-II Pro XC2VP30 ของวงจรแบบต่าง ๆ

ตารางที่ 1 แสดงผลการสังเคราะห์วงจรด้วยโปรแกรม Xilinx ISE 10.1 และตารางที่ 2 แสดงอัตราส่วน ของวงจร DLMS ต่างๆเมื่อเปรียบเทียบกับวงจรกรองแบบปรับตัวได้แบบ LMS พบว่าการทำไปป์ไลน์ให้ละเอียดขึ้นหรือมีจำนวนขั้นมากขึ้นทำให้วงจรทำงานเร็วขึ้นตามลำดับเป็นเท่าตัว ในขณะที่ขนาดของวงจรเมื่อพิจารณาที่จำนวน Silces นั้นเพิ่มขึ้นเพียงเล็กน้อย

ตารางที่ 1 การใช้ ทรัพยากรของวงจรต่างๆ

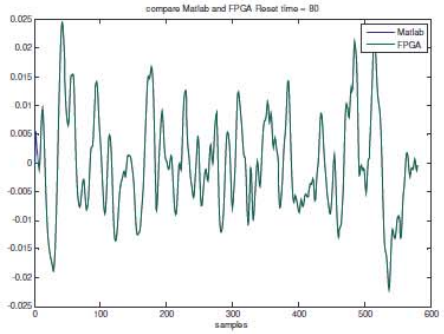
Algorithm	Slices	FFs	LUTs	MULs	Speed (MHz)
LMS	2524	1009	4317	65	21.33
DLMS (D=5)	3025	2102	4826	65	76.70
DLMS (D=7)	3153	2138	5084	65	82.26
DLMS (D=10)	3518	2936	5789	65	94.26

ตารางที่ 2 อัตราส่วน ของวงจรต่างๆเมื่อเปรียบเทียบกับวงจรกรองแบบปรับตัวได้แบบ LMS

Algorithm	Slices	FFs	LUTs	MULs	Speed
LMS	1	1	1	1	1
DLMS (D=5)	1.19	2.08	1.11	1	3.60
DLMS (D=7)	1.24	2.11	1.17	1	3.85
DLMS (D=10)	1.39	2.90	1.34	1	4.41

4.2 ทำการทดสอบสัญญาณของวงจรกรองปรับตัวได้แบบ DLMS เทียบกับสัญญาณในโปรแกรม Matlab

การออกแบบวงจรกรองปรับตัวแบบ DLMS D = 7 โดยใช้ MATLAB/Simulink Xilinx Accel DSP Toolbox ในการออกแบบได้อธิบายไปก่อนหน้านี้ วงจรที่ได้จากโปรแกรม MATLAB สามารถทำการแปลงเป็น Verilog code เพื่อทำการรันบน FPGA และทำการทดสอบโดยการป้อนสัญญาณอินพุท $d(n)$, $x(n)$ สัญญาณอ้างอิงให้กับวงจร เพื่อนำผลที่ได้จากการสังเคราะห์มาเปรียบเทียบกับผลที่ได้กับ MATLAB



รูปที่ 10 เปรียบเทียบสัญญาณระหว่าง Matlab กับ FPGA ของวงจรกรองปรับตัวได้ DLMS D = 7

จากรูปที่ 10 เมื่อทำการเปรียบเทียบสัญญาณที่ได้จากวงจรกรองปรับตัวได้จากโปรแกรม MATLAB กับสัญญาณที่ได้จากการ

สังเคราะห์ของ FPGA มีความผิดพลาดเพียงเล็กน้อยในช่วงแรก เนื่องจากการรีเซตของวงจร เมื่อนำมาเปรียบเทียบกับ โปรแกรม MATLAB มีความใกล้เคียงกัน เช่นเดียวกับวงจรกรองปรับตัวได้ที่มีค่า D = 5 และ D = 10 แตกต่างกันเพียงช่วงเริ่มต้นของสัญญาณเท่านั้น

5. สรุป

บทความนี้ได้นำเสนอการออกแบบวงจรการกำจัดสัญญาณเสียงสะท้อนโดยใช้วงจรกรองแบบปรับตัวได้แบบ LMS 32 Taps เปรียบเทียบกับวงจรกรองปรับตัวได้แบบ DLMS 32 Taps ความเร็วในการทำงานของวงจรกรองแบบปรับตัวได้แบบ DLMS มีความเร็วที่สูงกว่าวงจรแบบ LMS ประมาณ 3-4 เท่าตัว บน Xilinx Virtex-II Pro XC2VP30 ความเร็วสูงสุดที่ได้คือ 94.26 MHz ในขณะที่ขนาดวงจรใหญ่กว่าเพียง 1.39 เท่า

กิตติกรรมประกาศ

ขอขอบคุณทีมวิจัย High-Performance Embedded system ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยสงขลานครินทร์

เอกสารอ้างอิง

[1] Parhi, k. 1959. VLSI digital signal processing systems: design and implementation, A wiley-interscience publication, USA.

[2] Long, G., Ling, F. and Proakis, J.G. 1989. The LMS Algorithm with Delayed Coefficient Adaptation. IEEE Transactions on Acoustics Speech and Signal Processing, 37 : 1397-1405.

[3] Siqueira, M.G. and Alwan, A. 2000. Steady-State Analysis of Continuous Adaptation in Acoustic Feedback Reduction Systems for Hearing-Aids. IEEE Transactions on Speech and Audio Processing, 8: 443-453.

[4] Kates, J.M. 1991. Feed back cancellation in Hearing aids: Result from a computer Simulation. IEEE Transactions on Signal Processing, 39: 553-562.

[5] Maxwell, J.A. and Zurek, P.M. 2000. Reducing acoustic Feedback in Hearing aids. IEEE Transactions on Speech and Audio Processing, 3: 304-313.

ประวัติผู้เขียน

ชื่อ สกุล นายสายัณ ละอองโชค

รหัสประจำตัวนักศึกษา 5110120095

วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
อุตสาหกรรมศาสตรบัณฑิต (เทคโนโลยีอิเล็กทรอนิกส์โทรคมนาคม)	สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ	2545

ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนอุดหนุนงานวิจัยมหาวิทยาลัยสงขลานครินทร์ประจำปีการศึกษา 2552

การตีพิมพ์เผยแพร่ผลงาน

สายัณ ละอองโชค, ณัฏฐา จินดาเพ็ชร, พรชัย พฤกษ์ภัทรานนต์ และ สมพัฒน์ รุ่งตะวันเรืองศรี, การออกแบบวงจรกรองปรับตัวได้แบบ DLMS สำหรับการกำจัดสัญญาณเสียงสะท้อนในระบบเครื่องช่วยฟัง ,การประชุมวิชาการทางวิศวกรรมศาสตร์ ครั้งที่ 9 ประจำปี 2554.