สำนักหอสมุดกลาง พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การกำจัดสัญญาณเสียงสะท้อนด้วยตัวกรองแบบปรับตัว

ADAPTIVE FILTER FOR ECHO CANCELLATION

โดย

นางสาวจิตรลดา สุนทะศักดิ์

นายชวลิต อิทธิอาภา

นายชัชชล สมสุข

มท.
TK5101
.S45
เลขหมู่.....
เลขทะเบียน 62868
วัน,เดือน,ปี 23 ส.ค. 2549
b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำจัดสัญญาณเสียงสะท้อนด้วยตัวกรองแบบปรับตัว

Adaptive filter for echo cancellation

โดย

นางสาวจิตรลดา สุนทะศักดิ์ 45010113

นายชวลิต อิทธิอาภา 45010170

นายชัชชล สมสุข 45010171

อาจารย์ที่ปรึกษา

ผศ. อัครพล ตรีรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การกำจัดสัญญาณเสียงสะท้อนด้วยตัวกรองแบบปรับตัว**

ADAPTIVE FILTER FOR ECHO CANCELLATION

ผู้จัดทำ

1. นางสาวจิตรลดา	**สุนทะศักดิ์**	**45010113**
2. นายชวลิต	**อิทธิอาภา**	**45010170**
3. นายชัชชล	**สมสุข**	**45010171**

..... อาจารย์ที่ปรึกษา

(ผศ. อัครพล ตรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำจัดสัญญาณเสียงสะท้อนด้วยตัวกรองแบบปรับตัว

ADAPTIVE FILTER FOR ECHO CANCELLATION

โดย นางสาวจิตรลดา สุนทะศักดิ์ 45010113

นายชวลิต อิทธิอาภา 45010170

นายชัชชล สมสุข 45010171

อาจารย์ที่ปรึกษา ผศ. อัครพล ตรีรัตน์

บทคัดย่อ

การออกแบบและสร้างตัวกำจัดสัญญาณเสียงสะท้อน ไม่สามารถกำจัดได้ดีด้วยตัวกรองแบบที่มีค่าสัมประสิทธิ์เป็นค่าคงที่ ดังนั้นตัวกรองแบบปรับตัวเองจึงได้ถูกนำมาใช้ในงานนี้ โดยจะเลือกใช้อัลกอริธึมในการปรับสัมประสิทธิ์ของตัวกรองที่ใช้เงื่อนไขลีสท์มีนแสควร์เออเรอร์ (Least mean square error) โดยแบ่งการทดลองเป็นสองส่วนด้วยกันคือ การจำลองการทำงาน (Simulation) จะทำโดยใช้โปรแกรม matlab และ การสร้างตัวกรองสัญญาณแบบปรับตัวเองจะถูกสร้างโดยใช้ FPGA

ABSTRACT

This thesis presents a design and implementation of voice the echo canceller (VEC) can't cancel with fixed coefficient filter, so adaptive filter was developed for using in this case. Implement for apply to use. In this algorithm for adapted coefficient of filter which in condition of lest mean square error (LMS). So we split in to 2 case of the second case, real implemented with adaptive filter by FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 บทนำ	2
2.2 การกำจัดสัญญาณเสียงสะท้อน	2
2.2.1 สัญญาณเสียงสะท้อนในระบบโทรศัพท์	2
2.2.2 หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อน	3
2.3 ประเภทของสัญญาณเสียงสะท้อน	4
2.3.1 สัญญาณสะท้อนแบบ hybrid (Hybrid echo)	4
2.3.2 สัญญาณเสียงสะท้อนแบบ Acoustic (Acoustic echo)	4
2.4 Echo Cancellation	5
2.4.1 Stochastic process and models	6
2.4.2 ลักษณะของสัญญาณในรูปแบบต่างๆ	7
2.5 แนะนำเกี่ยวกับ Adaptive Filter	8
2.5.1 บทนำ	8
2.5.2 Adaptive Signal Processing	9
2.5.3 FIR filter	10
2.6 Wiener Filter	11
2.6.1 Method of Steepest decent	12
2.7 ทฤษฎีการใช้งาน adaptive filter	13
2.7.1 Least Mean Square Algorithm (LMS)	13
2.7.2 Normalized Least Mean Square algorithm (NLMS)	15
2.7.3 Quantized – error Algorithm	17
2.7.4 Sign – Error Algorithm	17
2.7.5 Sign Data Algorithm	18
บทที่ 3 การออกแบบ และการสร้างชุดการทำงานเพื่อกำจัดเสียงสะท้อน	19
3.1 บทนำ	19
3.2 การออกแบบโครงสร้างสำหรับการทดลอง	19
3.3 การออกแบบสำหรับ LMS Algorithm	20
3.4 การออกแบบสำหรับ Quantized – error algorithm	21
3.5 Normalized LMS algorithm (NLMS)	23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.6 แนวคิดในการออกแบบวงจรโดยใช้ FPGA	24
3.6.1 ส่วนของการควบคุม	25
3.6.2 หน่วยประมวลผลทางคณิตศาสตร์	26
3.6.3 ส่วนของระบบการจัดเก็บข้อมูล	27
3.6.4 การออกแบบและโครงสร้างการทำงานของชุดวงจร Analog to Digital Converter	28
3.6.5 การออกแบบและโครงสร้างการทำงานของชุดวงจร Digital to Analog Converter	29
3.6.6 แบบแผนวงจรรวมของวงจรกำจัดเสียงสะท้อน	29
3.6.7 การออกแบบการทดลองสำหรับการวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบ	30
บทที่ 4 การทดลองและผลการทดลอง	31
4.1 บทนำ	31
4.2 รูปแบบที่ใช้ในการทดลอง	31
4.3 ผลการทดลองเมื่อใช้ LMS Algorithm	31
4.4 ผลการทดลองเมื่อใช้ Quantized – error Algorithm	36
4.5 ผลการทดลองเมื่อใช้ Normalized LMS Algorithm	44
4.6 ผลการทดลองของวงจรที่ใช้ในการควบคุม	49
4.7 ผลการทดลองของวงจรหารความถี่	50
4.8 ผลการทดลองของวงจรบวก	51
4.9 ผลการทดลองของวงจรลบ	52
4.10 ผลการทดลองของวงจรบัฟเฟอร์	52
4.11 ผลการทดลองของวงจรขยาย	53
4.12 ผลการทดลองของวงจรคูณ	54
4.13 ผลการทดลองของวงจรเก็บค่าสัญญาณ	55
4.14 ผลการทดลองของวงจรเก็บค่าสัมประสิทธิ์	56
4.15 ผลการทดลองของวงจรรวม	57
4.16 ผลการทดลองของวงจร A/D และ D/A Converter	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 สรุปและวิเคราะห์ผลการทดลอง	60
5.1 สรุปผลการทดลองจากการชิมมูเลชั่นด้วยโปรแกรม MATLAB	60
5.2 สรุปผลการทดลองที่ได้จากบอร์ด FPGA	60
5.3 แนวทางแก้ไขปัญหาที่เกิดขึ้น	61
หนังสืออ้างอิง	
ภาคผนวก	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 สาเหตุการเกิดสัญญาณเสียงสะท้อนในระบบโทรศัพท์	2
รูปที่ 2.2 หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อนที่ใช้สำหรับการสื่อสาร	3
รูปที่ 2.3 หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อน	3
รูปที่ 2.4 แสดงสัญญาณสะท้อนแบบ Hybrid	4
รูปที่ 2.5 แสดงการเกิดสัญญาณสะท้อนแบบ Acoustic	5
รูปที่ 2.6 Adaptive Filter Configuration	5
รูปที่ 2.7 องค์ประกอบ adaptive filter	9
รูปที่ 2.8 FIR Filter structure	10
รูปที่ 2.9 Standard Adaptive Filter Problem	11
รูปที่ 2.10 แสดงวิธีการปรับเปลี่ยนค่า step size	12
รูปที่ 3.1 โครงสร้างของ adaptive filter	19
รูปที่ 3.2 แสดงโครงสร้างของ adaptive filter ชนิด LMS algorithm	20
รูปที่ 3.3 แสดงโครงสร้างของ adaptive filter ชนิด sign – error algorithm	22
รูปที่ 3.4 แสดงโครงสร้างของ adaptive filter ชนิด sign – data algorithm	22
รูปที่ 3.5 แสดงโครงสร้างของ adaptive filter ชนิด NLMS algorithm	23
รูปที่ 3.6 การเชื่อมต่อของวงจรกำจัดสัญญาณเสียงสะท้อน	24
รูปที่ 3.7 แสดงช่วงเวลาที่ใช้ในการประมวลผลในแต่ละ Sample	25
รูปที่ 3.8 แสดงบล็อกไดอะแกรมของวงจรในส่วนควบคุม (controller)	25
รูปที่ 3.9 แสดงบล็อกไดอะแกรมของวงจรบวก (adder)	26
รูปที่ 3.10 แสดงบล็อกไดอะแกรมของวงจรขยาย (gain)	26
รูปที่ 3.11 แสดงบล็อกไดอะแกรมของวงจรคูณ (multiplier)	26
รูปที่ 3.12 แสดงบล็อกไดอะแกรมของวงจรลบ (subtractor)	26
รูปที่ 3.13 แสดงบล็อกไดอะแกรมของวงจรบัฟเฟอร์ (buffer)	27
รูปที่ 3.14 แสดงบล็อกไดอะแกรมของวงจรที่ใช้ในการเก็บค่าสัญญาณ	27
รูปที่ 3.15 แสดงบล็อกไดอะแกรมของวงจรที่ใช้ในการเก็บค่าสัมประสิทธิ์ของฟิวเตอร์	28
รูปที่ 3.16 แสดงวงจรแปลงสัญญาณเชิงอนาลอกเป็นสัญญาณดิจิตอล	28
รูปที่ 3.17 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณเชิงอนาลอก	29
รูปที่ 3.18 แบบแผนวงจรรวมของระบบการทำงานที่จะถูกนำไปทำคอนฟิกูเรชันลงบนชิพ FPGA	29
รูปที่ 3.19 แสดงการเตรียมอุปกรณ์สำหรับการทดลองวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบขึ้น	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.1 แสดงรูปของวงจรที่ใช้สำหรับการจำลองผลการทำงาน 31

รูปที่ 4.2 แสดงผลการทดลองของ Adaptive filter แบบ LMS Algorithm 33

(รูปบน) ค่า $\mu = 0.0007$ (รูปกลาง) ค่า $\mu = 0.0009$ (รูปล่าง) ค่า $\mu = 0.001$

รูปที่ 4.3 แสดงผลการทดลองของ Adaptive filter แบบ LMS Algorithm 34

(รูปบน) ค่า $\mu = 0.003$ (รูปกลาง) ค่า $\mu = 0.005$ (รูปล่าง) ค่า $\mu = 0.007$

รูปที่ 4.4 แสดงผลการทดลองของ Adaptive filter แบบ LMS Algorithm 35

(รูปบน) ค่า $\mu = 0.009$ (รูปกลาง) ค่า $\mu = 0.01$ (รูปล่าง) ค่า $\mu = 0.03$

รูปที่ 4.5 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Error Algorithm 38

(รูปบน) ค่า $\mu = 0.00007$ (รูปกลาง) ค่า $\mu = 0.00009$ (รูปล่าง) ค่า $\mu = 0.0001$

รูปที่ 4.6 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Error Algorithm 39

(รูปบน) ค่า $\mu = 0.00003$ (รูปกลาง) ค่า $\mu = 0.00005$ (รูปล่าง) ค่า $\mu = 0.0007$

รูปที่ 4.7 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Error Algorithm 40

(รูปบน) ค่า $\mu = 0.0009$ (รูปกลาง) ค่า $\mu = 0.001$ (รูปล่าง) ค่า $\mu = 0.003$

รูปที่ 4.8 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Data Algorithm 41

(รูปบน) ค่า $\mu = 0.0007$ (รูปกลาง) ค่า $\mu = 0.0009$ (รูปล่าง) ค่า $\mu = 0.001$

รูปที่ 4.9 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Data Algorithm 42

(รูปบน) ค่า $\mu = 0.003$ (รูปกลาง) ค่า $\mu = 0.005$ (รูปล่าง) ค่า $\mu = 0.007$

รูปที่ 4.10 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Data Algorithm 43

(รูปบน) ค่า $\mu = 0.009$ (รูปกลาง) ค่า $\mu = 0.01$ (รูปล่าง) ค่า $\mu = 0.03$

รูปที่ 4.11 แสดงผลการทดลองของ Adaptive filter แบบ Normalized LMS Algorithm เมื่อ $\gamma = 5$ 46

(รูปบน) ค่า $\mu = 0.03$ (รูปกลาง) ค่า $\mu = 0.08$ (รูปล่าง) ค่า $\mu = 0.2$

รูปที่ 4.12 แสดงผลการทดลองของ Adaptive filter แบบ Normalized LMS Algorithm เมื่อ $\gamma = 5$ 47

(รูปบน) ค่า $\mu = 0.5$ (รูปกลาง) ค่า $\mu = 0.8$ (รูปล่าง) ค่า $\mu = 1.3$

รูปที่ 4.13 แสดงผลการทดลองของ Adaptive filter แบบ Normalized LMS Algorithm เมื่อ $\gamma = 5$ 48

(รูปบน) ค่า $\mu = 1.8$ (รูปกลาง) ค่า $\mu = 2$ (รูปล่าง) ค่า $\mu = 2.2$

รูปที่ 4.14 แสดงบล็อกไดอะแกรมของวงจรในส่วนควบคุม (controller) 49

รูปที่ 4.15 แสดง timing diagram การทำงานของวงจรในส่วนควบคุม (controller) 50

รูปที่ 4.16 แสดงบล็อกไดอะแกรมของวงจรหาร 8 50

รูปที่ 4.17 แสดงบล็อกไดอะแกรมของวงจรหาร 1000 50

รูปที่ 4.18 แสดง timing diagram การทำงานของวงจรหาร 8 51

รูปที่ 4.19 แสดง timing diagram การทำงานของวงจรหาร 1000 51

รูปที่ 4.20 แสดงบล็อกไดอะแกรมของวงจรบวก (adder) 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.21 แสดง timing diagram การทำงานของวงจรบวก (adder) 51

รูปที่ 4.22 แสดงบล็อกไดอะแกรมของวงจรลบ (subtracter) 52

รูปที่ 4.23 แสดง timing diagram การทำงานของวงจรลบ (subtracter) 52

รูปที่ 4.24 แสดงบล็อกไดอะแกรมของวงจรบัฟเฟอร์ (buffer) 52

รูปที่ 4.25 แสดง timing diagram การทำงานของวงจรบัฟเฟอร์ (buffer) 53

รูปที่ 4.26 แสดงบล็อกไดอะแกรมของวงจรขยาย (gain) 53

รูปที่ 4.27 แสดง timing diagram การทำงานของวงจรขยาย (gain) 53

รูปที่ 4.28 แสดงบล็อกไดอะแกรมของวงจรคูณ (multiplier) 54

รูปที่ 4.29 แสดง timing diagram การทำงานของวงจรคูณ (multiplier) 54

รูปที่ 4.30 แสดงบล็อกไดอะแกรมของวงจรเก็บค่าสัญญาณ 55

รูปที่ 4.31 แสดง timing diagram การทำงานของวงจรเก็บค่าสัญญาณ 55

รูปที่ 4.32 แสดงบล็อกไดอะแกรมของวงจรเก็บค่าสัมประสิทธิ์ของฟิลเตอร์ 56

รูปที่ 4.33 แสดง timing diagram การทำงานของวงจรเก็บค่าสัมประสิทธิ์ของฟิลเตอร์ 56

รูปที่ 4.34 แสดง timing diagram การทำงานของวงจรรวม 57

รูปที่ 4.35 แสดงบล็อกไดอะแกรมของวงจรรวม 58

รูปที่ 4.36 (บน) รูปสัญญาณก่อนที่จะผ่านวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล 59
(ล่าง) รูปสัญญาณหลังจากผ่านวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 4.1 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ LMS Algorithm	32
ตารางที่ 4.2 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ Sign – Error Algorithm	37
ตารางที่ 4.3 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ Sign – Data Algorithm	37
ตารางที่ 4.4 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ Normalized LMS Algorithm	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของหัวข้อโครงงาน

ในโครงงานนี้จะเป็นการศึกษาเกี่ยวกับระบบ Digital Signal Processing ของอุปกรณ์ที่เรียกว่า Adaptive filter ซึ่งเป็น filter ที่สามารถปรับค่าสัมประสิทธิ์ได้ ซึ่งจะช่วยแก้ข้อบกพร่องของ filter ที่มีค่าสัมประสิทธิ์คายตัว (fixed-filter) ให้สัญญาณที่ออกมาจากระบบมีลักษณะใกล้เคียงสัญญาณที่ต้องการมากขึ้น Adaptive filter สามารถนำมาประยุกต์เป็นอุปกรณ์ได้หลายชนิด เช่น Echo Cancellation, Equalization of dispersive channel , System identification ,Signal enhancement, Adaptive beamforming และ Noise cancelling เป็นต้น โดยในโครงงานนี้เราจะนำ Adaptive filter นี้มาทำเป็นอุปกรณ์ที่เรียกว่า Echo cancellation ซึ่งใช้แก้ปัญหาของระบบโทรศัพท์ที่เกิดปัญหาแบบมีสัญญาณสะท้อน (Echo) ในระบบโทรศัพท์เป็นเหตุให้เกิดการรบกวนต่อการสื่อสารทำให้ประสิทธิภาพของการสื่อสารลดลง โดยจะทำการจำลอง (simulate) ด้วยโปรแกรม matlab และสร้างบนอุปกรณ์ FPGA

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาระบบการทำงานของ Adaptive filter และรูปแบบของ algorithm เพื่อใช้กำจัดสัญญาณสะท้อน

1.2.2 เพื่อศึกษาระบบของโทรศัพท์ซึ่งส่งผลให้เกิดสัญญาณสะท้อนขึ้น และรูปแบบของสัญญาณสะท้อนที่เกิดขึ้น

1.2.3 เพื่อศึกษาและประยุกต์ใช้งานโปรแกรมแมทแลบ (MATLAB) ในการประมวลผล

1.2.4 เพื่อศึกษาวิธีการเขียนโปรแกรมโดยใช้อุปกรณ์ FPGA

1.3 ขอบเขตของโครงงาน

ศึกษาการทำงานของอัลกอริทึม 3 อัลกอริทึม คือ Least Mean Square (LMS), Normalize Least Mean Square (NLMS) และ Quantized Error แล้วนำทั้ง 3 อัลกอริทึมมาเขียนโปรแกรมจำลองใน matlab เพื่อศึกษาข้อดีข้อเสียของอัลกอริทึมแต่ละแบบ แล้วนำแบบที่ได้ผลน่าพอใจที่สุดมาสร้างลงบนอุปกรณ์ FPGA

1.4 เนื้อหาของโครงงาน

บทที่ 2 ทฤษฎีและหลักการ

บทที่ 3 การคำนวณและการสร้าง

บทที่ 4 การทดลองและผลการทดลอง

บทที่ 5 บทวิจารณ์และบทสรุป

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

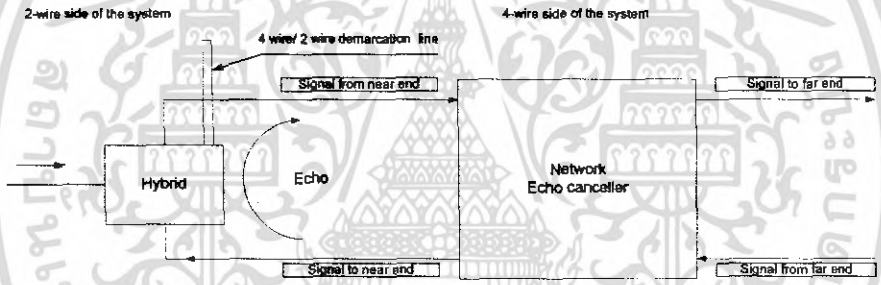
2.1 บทนำ

ในการสื่อสารด้วยโทรศัพท์ทั่วไปอาจจะมีการเกิดสัญญาณสะท้อนรบกวนในระบบทำให้เกิดความไม่สะดวกต่อทั้งผู้รับและผู้ส่งข้อมูลข่าวสาร และอาจมีบางครั้งที่รุนแรงจนกระทั่งการติดต่อสื่อสารไม่สามารถดำเนินต่อไปได้อย่างต่อเนื่อง ดังนั้น เพื่อให้การติดต่อสื่อสารด้วยโทรศัพท์มีประสิทธิภาพสูงขึ้นจึงมีความจำเป็นที่จะต้องกำจัดสัญญาณเสียงสะท้อนภายในระบบ ซึ่งในบทนี้จะได้ศึกษาถึงสาเหตุ แนวทางในการกำจัดสัญญาณเสียงสะท้อน และวิธีการที่จะนำมาใช้กำจัดสัญญาณเสียงสะท้อนตามลำดับ

2.2 การกำจัดสัญญาณเสียงสะท้อน

2.2.1 สัญญาณเสียงสะท้อนในระบบโทรศัพท์

ต้นกำเนิดของเสียงสะท้อนสามารถที่จะพิจารณาได้จากการเชื่อมต่อกันระหว่างตัว Subscriber S1 และ S2 ที่แสดงไว้ในรูป 2.1



รูปที่ 2.1 สาเหตุการเกิดสัญญาณเสียงสะท้อนในระบบโทรศัพท์

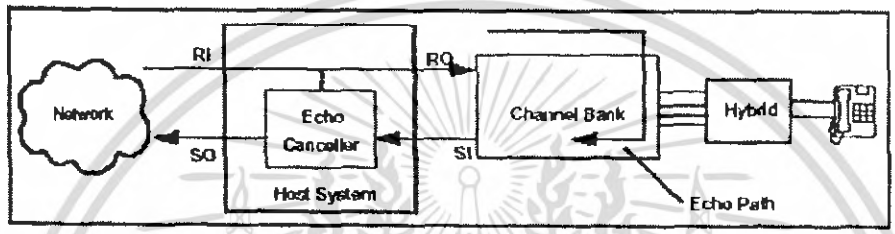
จากรูปที่ 2.1 การเชื่อมต่อจะประกอบไปด้วยส่วนของโครงข่ายแบบ 2 สาย (2-wire network) ที่อยู่ด้านปลายทั้งสองและมีการเชื่อมต่อแบบ 4 สาย (4-wire network) อยู่ตรงกลาง ส่วนอุปกรณ์ไฮบริด (Hybrid) ที่อยู่ปลายของระบบคอยทำการปรับเปลี่ยนการสื่อสารแบบ 2 สาย ไปเป็นการสื่อสารแบบ 4 สาย ในส่วนของการสื่อสารแบบ 2 สาย ก็จะประกอบไปด้วย Subscriber loops และก็อาจจะมีบางส่วนที่เป็นโครงข่ายท้องถิ่น (local network) ยิ่งกว่านั้นในส่วนนี้จะมีทิศทางของการส่งผ่าน (Transmission) แบบสองทางโดยจะถูกทำการส่งผ่านด้วยสายคู่เดียว สัญญาณจากผู้พูด S1 และ S2 จะถูกป้อนใส่เข้าไปในส่วนนี้ แต่วงจรส่วนที่เป็นการสื่อสารแบบ 4 สายนั้นการสื่อสารในแต่ละทางจะถูกแยกออกจากกัน สัญญาณเสียงจากผู้พูด S1 ก็จะวิ่งไปตามเส้นทางด้านบนของรูปและมีทิศทางตามลูกศร ในขณะที่เสียงจากจุดกำเนิด S2 ก็จะไปตามเส้นทางด้านล่างของรูปการแบ่งแยกสัญญาณทั้งสองออกจากกันนั้นเป็นสิ่งที่จำเป็นเนื่องจากต้องมีการนำไปรวมกับสัญญาณพาหะ (Carrier) ระบบขยายสัญญาณ (Amplifier) และระบบสลับสายสัญญาณดิจิตอล (Digital Switch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

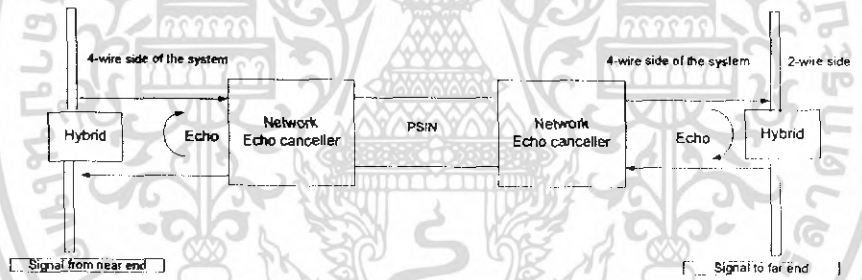
อุปกรณ์ไฮบริดจะทำหน้าที่เปลี่ยนการส่งผ่านจากระบบการสื่อสารแบบ 2 สาย ให้เป็นการสื่อสารแบบ 4 สาย หน้าที่ของไฮบริดที่อยู่ทางด้านขวามือก็จะทำการควบคุมการให้พลังงานของสัญญาณจาก S1 ไปยังส่วนที่เป็นการสื่อสารแบบ 2 สายของ S2 โดยที่จะไม่ยินยอมให้มันย้อนกลับมาที่ S1 อีกด้วยเส้นทางที่อยู่ด้านล่าง แต่เนื่องจากเกิดการไม่ match ในอุปกรณ์ ไฮบริดจึงทำให้พลังงานบางส่วนเกิดการย้อนกลับไปยัง S1 ทำให้ได้ยินเสียงที่มีการหน่วงเวลาไปจากปกติซึ่งการเกิดเหตุการณ์เช่นนี้จะเรียกว่า "เสียงสะท้อนถึงผู้พูด" (Talker echo)

2.2.2 หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อน

หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อนที่ใช้สำหรับการสื่อสารในระบบทิศทางเดียวได้แสดงไว้ในรูปที่ 2.2



รูปที่ 2.2 หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อนที่ใช้สำหรับการสื่อสาร



รูปที่ 2.3 หลักการทำงานของตัวกำจัดสัญญาณเสียงสะท้อน

รูปที่ 2.2 แสดงถึงบางส่วนของการเชื่อมต่อสื่อสารแบบ 4 สายที่อยู่ใกล้กับส่วนของการสื่อสารแบบ 2 สาย ด้วยการส่งผ่านสัญญาณเสียงไปในทิศทางเดียวระหว่างพอร์ท A ไปยังพอร์ท C และระหว่างพอร์ท D ไปยังพอร์ท B สัญญาณที่แสดง far – end – talker แสดงด้วยสัญลักษณ์ y(i) สัญญาณสะท้อนแสดงด้วย x(i) และสัญญาณ near – end – talker แสดงด้วยสัญญาณสะท้อนที่ไม่ต้องการจะรวมอยู่กับสัญญาณ near – end – talker บน part D สัญญาณที่ถูกรับได้จาก far – end – talker y(i) ก็จะถูกใช้ให้เป็นเหมือนกับสัญญาณอ้างอิงของตัวกำจัดสัญญาณสะท้อน โดย y(i) จะถูกใช้เพื่อทำการสร้างสัญญาณลอกเลียนแบบซึ่งเรียก r(i) จากนั้นสัญญาณลอกเลียนแบบที่ได้นี้จะถูกนำไปหักลบออกจาก near – end talker ที่ถูกรวมอยู่กับสัญญาณสะท้อนก็จะทำให้ได้เป็นการส่งผ่านเฉพาะ near – end signal u(i) กลับไปด้าน far – end – listener เท่านั้น เมื่อ u(i) = x(i) + r(i) - r(i)ในทางอุดมคติแล้วส่วนที่เหลือของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

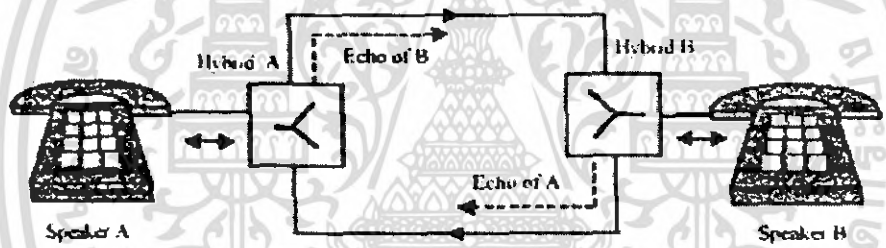
สะท้อน (echo error) จะมีค่าเป็น e(i) = r(i) – r(i) ซึ่งจะมีค่าน้อยมากหลังจากได้ทำการกำจัดสัญญาณสะท้อนแล้ว

2.3 ประเภทของสัญญาณเสียงสะท้อน

สัญญาณสะท้อนเกิดขึ้นจากสถานการณ์ต่างๆในระบบเครือข่ายการสื่อสารข้อมูล และมันมักจะเป็นตัวการที่ทำให้คุณภาพระบบการสื่อสารลดลง ซึ่งสามารถแบ่งประเภทของสัญญาณสะท้อนได้เป็น 2 ประเภท คือ Hybrid echo และ acoustic echo

2.3.1 สัญญาณสะท้อนแบบ Hybrid (Hybrid echo)

สัญญาณสะท้อนประเภทนี้มีต้นกำเนิดจาก hybrid transformer ซึ่งใช้เป็นที่เชื่อมต่อระหว่าง 2-wire ไปยัง 4-wire ในการเชื่อมต่อแบบ end-to-end ในระบบโทรศัพท์ที่มีใช้อยู่ในปัจจุบันนี้ประกอบด้วยส่วนที่ใช้เป็น link 2 ส่วนคือ 2-wire กับ 4-wire เป็นส่วนที่จะต่อไปยังตัวเครื่องโทรศัพท์ (long-haul repeater link) ซึ่ง hybrid transformer ถูกใช้เป็นจุดเชื่อมต่อของทั้งสองส่วนนี้ และระหว่างที่เกิดปรากฏการณ์ที่เรียกว่า impedance matching สัญญาณสะท้อนจะถูกกำเนิดขึ้น เราจะใช้ตัว echo canceller ติดตั้งเพื่อทำการกำจัดสัญญาณเหล่านี้ในด้านของ 4-wire รูปที่ 2.4

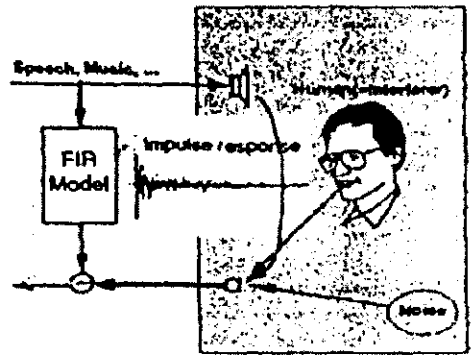


รูปที่ 2.4 แสดงสัญญาณสะท้อนแบบ Hybrid

2.3.2 สัญญาณเสียงสะท้อนแบบ Acoustic (Acoustic echo)

สัญญาณสะท้อนประเภทนี้มีผลมาจากการสะท้อนของคลื่นเสียงและการเหนี่ยวนำกันระหว่าง microphone และ ลำโพง สัญญาณสะท้อนจะขัดจังหวะในการสนทนาของผู้พูดและเป็นสาเหตุของการเกิดเสียงหอน (voice switches) และทิศทางของ microphone เป็นวิธีการที่ใช้มาแต่ดั้งเดิมเพื่อบรรเทาปัญหาเหล่านี้แต่ว่ายังติดปัญหาในเรื่องของสถานที่ที่จำกัดของการติดตั้งลำโพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



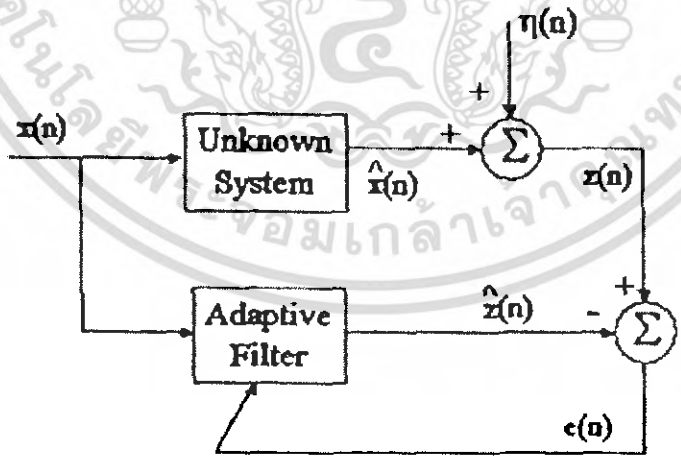
รูปที่ 2.5 แสดงการเกิดสัญญาณสะท้อนแบบ Acoustic

2.4 Echo Cancellation

ถึงแม้ว่าความสามารถทางเทคโนโลยีในปัจจุบันของวงการโทรศัพท์จะพัฒนาไปอย่างรวดเร็วก็ตาม หากแต่ว่าพื้นฐานของวิธีการกำจัดสัญญาณสะท้อนในระบบโทรศัพท์ก็ยังเหมือนเดิมไม่ได้มีการเปลี่ยนแปลงไปและหลักการนั้นก็ยังมีความถูกต้องทั้งการกำจัดสัญญาณสะท้อนที่เป็นเสียงพูด (speed echo cancellers).

ในบทนี้จะได้กล่าวถึงวิธีการพื้นฐานที่ใช้ในการกำจัดสัญญาณสะท้อนโดยจะเริ่มที่การศึกษาโครงสร้างของตัวกรองสัญญาณ filter โดยเน้นศึกษาตัวกรองสัญญาณที่ชื่อว่า FIR filter (finite Impulse Response filter) หลังจากนั้นจะเป็นการศึกษาตัวกรองสัญญาณ LMS (Least Mean Square) algorithm

พื้นฐานของการกำจัดสัญญาณสะท้อนนั้นส่วนประกอบหลักที่สำคัญที่จะเรียกว่าเป็นหัวใจหลักของวงจรการกำจัดสัญญาณสะท้อนคือ Adaptive filter สามารถแสดงได้ดังรูปที่ 2.6



รูปที่ 2.6 Adaptive Filter Configuration

จากรูปสามารถอธิบาย parameter ต่างๆได้ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\hat{z}(n)$ = สัญญาณที่สังเคราะห์ขึ้น (replica signal)

z(n) = สัญญาณจากเครือข่าย (desired signal)

$\eta(n)$ = สัญญาณรบกวน (noise)

x(n) = สัญญาณเข้า (input signal)

e(n) = สัญญาณความผิดพลาด (error signal)

จากรูปที่ 2.6 แสดงโครงสร้างของวงจรกรอง (filter) เมื่อมีสัญญาณเข้ามายัง block ที่ชื่อว่า Unknown system ซึ่งเป็น block ที่ใช้แทนวงจร Adaptive filter อีกทางหนึ่ง ส่วนสัญญาณเอาท์พุท ที่ออกมาจาก Unknown system เรียกว่า desired signal และ output จาก Adaptive filter เรียกว่า synthesized signal ซึ่งก็คือสัญญาณที่จำลองขึ้นมาจากสัญญาณ input ต้นแบบ โดยพื้นฐานของวงจรกรอง (filter) ก็คือพยายามลดผลต่างหรือเรียกว่า สัญญาณ error ระหว่างสัญญาณจากเครือข่าย และสัญญาณที่มีการสังเคราะห์ขึ้นไปให้น้อยที่สุดจนอยู่ในค่าที่ยอมรับได้โดยที่สัญญาณความผิดพลาดที่ได้นั้นจะถูกส่งย้อนกลับเข้าที่ไป Adaptive filter อีกครั้งหนึ่ง ถ้าหากค่าความผิดพลาดนั้นยอมรับไม่ได้เพื่อทำการปรับเปลี่ยนสัมประสิทธิ์ filter อีกครั้งหนึ่ง

2.4.1 Stochastic process and models

Stochastic process (random process) คือกระบวนการเปลี่ยนแปลงของสัญญาณเชิงเวลาทางสถิติโดยอาศัยกฎความน่าจะเป็น (probabilistic laws) เนื่องจาก Stochastic process เป็นฟังก์ชั่นของเวลา เราสามารถเขียนแทนได้ด้วย time series u(n) โดยที่ n แทน time samples 0,1,2,...

2.4.1.1 ค่าทางสถิติของ Stochastic process มีดังนี้

1. ค่าเฉลี่ยของ process (expected value) : $E[u(n)] = \int u(n) fu_{(n)}(u(n))du(n)$

โดยที่ fu(n)(u(n)) คือ probability density function ของ u(n)

2. ค่าความแปรปรวน (variance)

$$V[u(n)] = \iint [u(n)\cdot u^*(n)] fu_{(n)}, U(n)(u(n), u(n))du(n) = \iint |u(n)|^2 fu_{(n)}(u(n))du(n) \quad (2.1)$$

โดยที่ $fu_{(n)}, U(n)(u(n), u(n))du(n)$ คือ joint probability density function ของ u(n)

3. ค่า Covariance หรือ correlation : $c(n, n-k) = E[u(n)u^*(n-k)]$

$$= \iint [u(n)\cdot u^*(n-k)] fu_{(n)}, U(n-k)(u(n), u(n-k))du(n)du(n-k) \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยคุณสมบัติที่สำคัญอีกอย่างหนึ่งของ Stochastic process คือ ความเป็น stationary ซึ่ง Stationary process มีคุณลักษณะดังต่อไปนี้

1. $E[u(n)] = \mu$ for all n :ค่า expected value มีค่าคงที่ไม่ขึ้นกันเวลา
2. $V[u(n)] = \sigma^2$ for all n: ค่าความแปรปรวนมีค่าคงที่ไม่ขึ้นกับเวลา
3. $c(n, n-k) = c(k)$ for all n: ค่า covariance หรือ correlation มีค่าคงที่ไม่ขึ้นกับเวลา n แต่ขึ้นกับหรือเป็นฟังก์ชั่นของ delay เท่านั้น

เราสามารถนำเอาค่า variance และค่า covariance มาสร้างเป็น correlation matrix ได้โดยคุณสมบัติของ correlation matrix ของ stationary process มีดังต่อไปนี้

1. Correlation matrix ของ stationary stochastic process เป็น Hermittian

$$R = \begin{pmatrix} r(0) & r(1) & \cdots & r(M-1) \\ r^*(1) & r(0) & \cdots & r^*(M-2) \\ \cdots & & r^*(M-2) & \cdots \\ r^*(M-1) & & & \end{pmatrix} \qquad (2.3)$$

2. Correlation matrix ของ stationary stochastic process เป็น Toeplitz หมายความว่า ค่าในแนวทแยงมุมมีค่าคงที่

3. Correlation matrix of identically independent stochastic process (white noise) $r(0) = 1$; all other's are 0

4. Correlation matrix of correlated stationary stochastic process (e.g. colored noise) $r(0) = 1; r(1), r(2), r(3), \ldots$ are not 0 ขึ้นกับว่าความสัมพันธ์ระหว่าง samples มีมากแค่ไหน

5. Correlation matrix of uncorrelated stochastic process (e.g. speech signal) ค่า Correlation ในแนวทแยงไม่เท่ากัน Correlation matrix ไม่อยู่ในรูปของ Toeplitz

2.4.2 ลักษณะของสัญญาณในรูปแบบต่างๆ

- **White noise** พลังงานสัญญาณจะคงที่อยู่ตลอดเวลา ทำให้สัญญาณแต่ละ sample ไม่มีความสัมพันธ์ต่อกัน โดยที่ไม่จำเป็นต้องรู้ค่าในอดีตก็หาค่าในอนาคตได้
- **Color noise** พลังงานของ noise จะมีการเปลี่ยนแปลงไปตลอดเวลาจะเกิดความสัมพันธ์กันขึ้นระหว่าง sample ปัจจุบันและอดีตเพื่อที่จะทำนายหรือประมาณการค่าในอนาคตได้ว่าจะมีความน่าจะเป็นที่จะเกิดขึ้นต่อไปอย่างไรบ้าง
- **Speed signal** เป็นสัญญาณที่อยู่ในรูปแบบของ non-stationary

Process คือ เป็นแบบ time variance ค่าทางสถิติจะมีการเปลี่ยนแปลงในแต่ละ sample เพราะธรรมชาติในการออกเสียง จะมีการเปล่งเสียงในระดับที่สูง-ต่ำ แตกต่างกัน ดังนั้นค่าในแนวทแยงมุมของ Correlation matrix จะไม่เป็น 0 หมด ขึ้นอยู่กับเสียงในตอนที่เปล่งออกมาว่ามีลักษณะเป็นอย่างไร ส่วนค่าในแนวทแยงมุมก็จะไม่เท่ากันซึ่งไม่อยู่ในรูปแบบของ Toeplitz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 แนะนำเกี่ยวกับ Adaptive Filter

2.5.1 บทนำ

ในเรื่องนี้เราจะอธิบายเกี่ยวกับชนิดของระบบ Signal Processing ซึ่งเราจะนำมาใช้ในการกำจัดสัญญาณสะท้อน ในระยะเวลา 30 กว่าปีที่ผ่านมานี้ ความก้าวหน้าทางด้านดิจิตอลเป็นเทคโนโลยีที่สำคัญที่คนหันมาสนใจกันอย่างแพร่หลาย จนหันมาสนใจ Digital Signal Processing เพิ่มขึ้นด้วย การที่ระบบ Digital Signal Processing นี้ถูกสนใจก็เพราะผลของระบบมันน่าสนใจ ทั้งด้าน ความน่าเชื่อถือ, ความถูกต้องแม่นยำ, มีขนาดทางกายภาพเล็ก และมีลักษณะเปลี่ยนแปลงได้

ตัวอย่างหนึ่งของระบบ DSP นั่นคือ Filtering ซึ่งเป็นกระบวนการทาง Signal Processing ซึ่งสามารถเปลี่ยนแปลงลักษณะข้อมูลของสัญญาณให้เหมาะสม กล่าวคือ Filter เป็นอุปกรณ์ที่แปลงสัญญาณอินพุตให้ออกมาเป็นสัญญาณเอาท์พุตที่เปลี่ยนแปลงจากสัญญาณอินพุตให้ดีตามที่เราต้องการได้ ถ้าเป็น time-invariant filter นั้น เราจะต้องกำหนดลักษณะโครงสร้างกับเงื่อนไขภายในตัวมันด้วยและถ้าเป็น linear filter สัญญาณเอาท์พุตจะเป็นฟังก์ชั่นเชิงเส้นของสัญญาณอินพุต ถ้ามีการกำหนดเงื่อนไขในการออกแบบขึ้น การออกแบบ time-invariant linear filter จะต้องคำนึงถึงเงื่อนไข 3 ข้อ คือ การประมาณรายละเอียดด้วยเศษส่วน Transfer function , การเลือกโครงสร้างที่เหมาะสม, การเลือก algorithm เพื่อนำไปใช้งาน

Adaptive Filter จะถูกนำไปใช้เมื่อเราไม่ทราบรายละเอียดที่แน่นอนของสัญญาณหรือเป็นรายละเอียดที่ไม่สามารถจัดการได้ด้วย time-invariant filter จริงๆแล้วก็คือ Adaptive filter เป็น filter แบบไม่เชิงเส้น เมื่อลักษณะสำคัญของมันนั้นจะขึ้นกับสัญญาณอินพุต แต่อย่างไรก็ตามถ้าเราหยุดทุกๆตัวแปรของ filter ไว้ที่ชั่วขณะหนึ่ง adaptive filter ที่เห็นก็จะแสดงเป็นลักษณะเชิงเส้นโดยสัญญาณเอาท์พุตจะเป็นฟังก์ชั่นเชิงเส้นของสัญญาณอินพุต

Adaptive filter นั้นยังช่วยแก้ปัญหาของ filter แบบ fixed ค่า ในด้านของการปฏิบัติซึ่งโดยทั่วไปแล้ว การออกแบบ fixed-filter นั้น ต้องการทั้งลักษณะของสัญญาณอินพุตและสัญญาณอ้างอิงเพื่อออกแบบ filter ได้เหมาะสม แต่ในบางกรณีเราไม่สามารถทราบว่าอุปกรณ์รอบข้างนั้นถูกกำหนดค่ามาดีแค่ไหน ทำให้สัญญาณผิดเพี้ยนไปแค่ไหน และการออกแบบ filter ที่ช่วยในกรณีเช่นนี้นั้นจะมีราคาแพงและยากในการนำมาใช้งานจริง เราสามารถแก้ปัญหานี้ได้โดยใช้ Adaptive filter ซึ่งมีการ update ค่าอยู่ตลอดเวลาโดยใช้ข้อมูลที่ปรากฏอยู่รอบข้าง อาจเรียกได้ว่า Adaptive filter นั้นมีลักษณะการประมาณค่าข้อมูลที่เคลื่อนที่ ซึ่งในหัวข้อต่อๆไปเราจะมาศึกษาเกี่ยวกับการเลือกโครงสร้าง และอัลกอริทึมของ Adaptive filter เพื่อนำมาประยุกต์ใช้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

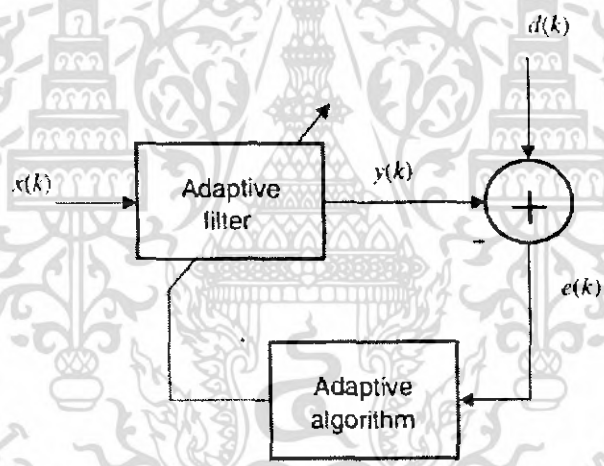
2.5.2 Adaptive Signal Processing

การออกแบบ Digital filter แบบที่เรากำหนดสัมประสิทธิ์คงที่นั้นต้องการขอบเขตของรายละเอียดที่ดี แต่ถึงอย่างไรก็ยังมีกรณีที่ไม่สามารถแสดงรายละเอียดได้ หรือเกิดการเปลี่ยนแปลงทางเวลา เราสามารถแก้ปัญหานี้ได้โดยใช้ digital filter ที่สามารถปรับค่าสัมประสิทธิ์ได้ ซึ่งเรียกว่า adaptive filter

ใน Adaptive algorithm ซึ่งเป็นตัวกำหนดขอบเขตของการ update ค่าสัมประสิทธิ์ของ filter นั้นต้องการข้อมูลพิเศษ ซึ่งปกติจะมาในรูปแบบของสัญญาณ เราเรียกสัญญาณนี้ว่า สัญญาณที่ต้องการหรือสัญญาณอ้างอิง

การติดตั้งขององค์ประกอบ Adaptive filter แสดงดังรูปที่ 2.7 ซึ่งค่า k คือ ค่าที่วนซ้ำ, x(k) แทนสัญญาณอินพุต, y(k) แทนสัญญาณเอาท์พุตและ d(k) แทนสัญญาณที่ต้องการ ค่าความผิดพลาด e(k) นั้นจะหาได้จาก d(k) – y(k) สัญญาณผิดพลาดนี้ใช้ในฟังก์ชั่นเพื่อใช้สร้างสัญญาณ และจะเป็นสิ่งจำเป็นใน algorithm เพื่อที่จะกำหนดค่า update ของสัมประสิทธิ์ให้เหมาะกับการที่จะลดค่าของฟังก์ชั่นเป้าหมายเพื่อให้เอาท์พุตนั้นตรงกับสัญญาณที่ต้องการ

รายละเอียดที่สมบูรณ์ของระบบ Adaptive แสดงในรูปที่ 2.7 โดยประกอบไปด้วยอุปกรณ์ 3 ชนิด



รูปที่ 2.7 องค์ประกอบ adaptive filter

การประยุกต์ : ชนิดของการประยุกต์นั้นจะกำหนดได้ด้วยตัวเลือกของสัญญาณที่ได้รับมาจากอุปกรณ์ ข้างเคียง ในช่วง 10 ปีที่ผ่านมามีการประยุกต์ที่ใช้เทคนิค adaptive มากมาย เช่น echo - cancellation, equalization of dispersive channel, system identification, signal enhancement, adaptive – beam forming and noise canceling.

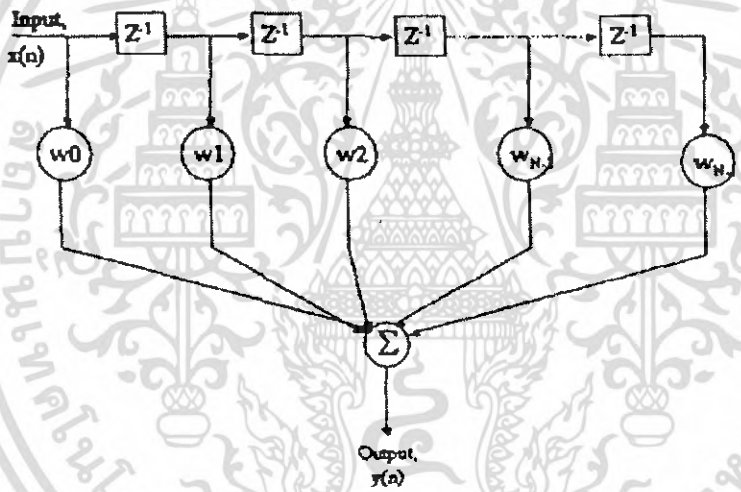
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้าง Adaptive filter : adaptive filter สามารถนำมาใช้ในโครงสร้างที่ต่างกันได้ การเลือกโครงสร้างนี้จะเป็นตัวบ่งบอกความยุ่งยากซับซ้อนในการคำนวณ และจำนวนของการทำซ้ำเพื่อที่จะแสดงข้อมูลในระดับที่ต้องการ โดยทั่วไปแล้ว adaptive filter ที่ใช้กันหลักๆ มีอยู่ 2 ชนิด ซึ่งแยกตามลักษณะของ Impulse-response เรียกว่า Finite-duration Impulse Response (FIR filter) และ Infinite-duration Impulse Response (IIR filter)

Algorithm : คือ กระบวนการที่ใช้ปรับค่าสัมประสิทธิ์ของadaptive filter เพื่อที่จะลดข้อกำหนดต่างๆ ตัวเลือกของ algorithm นั้นจะเป็นตัวกำหนดลักษณะ ทั้งหมดของกระบวนการ adaptive

2.5.3 FIR filter

Echo canceller เป็นวงแบบ linear adaptive filter ชนิดที่เรียกว่าวงปิด (close loop) โดยที่สามารถใช้วงจรกรอง (filter) ที่มีความแตกต่างในด้านองค์ประกอบและ algorithm การใช้งานได้หลากหลายขึ้นอยู่กับ applications ที่ใช้งาน โดยเราจะศึกษาเน้นไปที่โครงสร้างของ FIR filter เพื่อเป็นพื้นฐานในการนำไปสร้าง LMS algorithm FIR filter ซึ่งแสดงดังรูปที่ 2.8



รูปที่ 2.8 FIR Filter structure

FIR filter มีการใช้งานอย่างแพร่หลาย เนื่องจากการใช้งานที่ง่ายและภาวะความเสถียรคงที่ แต่ว่ามีปัญหาสำคัญคือ เมื่อเกิดช่วงสัญญาณความผิดพลาด (error signal) ยาวนานจำนวนของ tap จะเพิ่มขึ้นเป็นสัดส่วนกับช่วงการเกิด echo และอัตราของ convergence จะลดลง โดยที่ FIR filter จะประกอบด้วยลำดับของ delay, ส่วนวงจรการคูณและวงจรการบวก มี 1 input และ 1 output โดยที่จะเขียนเป็นสมการได้ดังนี้

$$y(n) = \sum_{i=0}^{N-1} w_i(n) x_i(n-1) \qquad (2.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ $w_i(n)$ คือสัมประสิทธิ์ของ filter

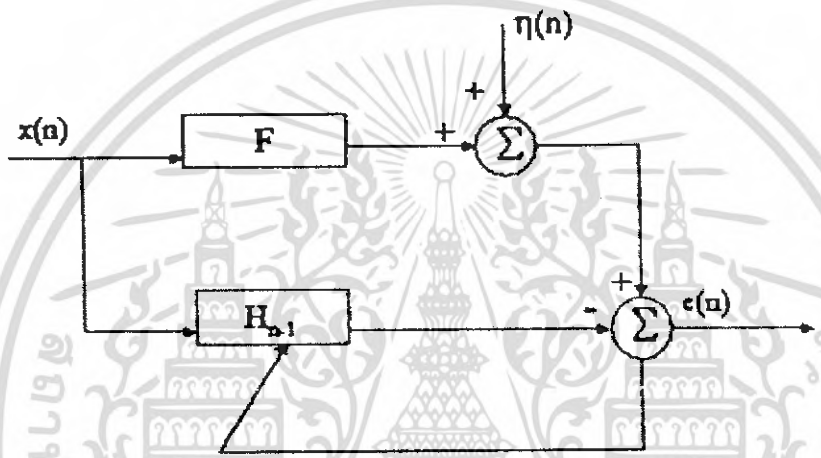
และ N คือ ความยาวของ filter

y(n) คือผลคูณระหว่าง vector 2 vector คือ w(n) และ x(n)

ข้อดีของ FIR filter ที่เหนือกว่า filter ชนิดอื่นคือ transfer function จะประกอบด้วย zeros เท่านั้น

2.6 Wiener Filter

สำหรับปัญหาของ linear filter ในส่วนของ inputs adaptive algorithms จะมุ่งเน้นที่รวมการแก้ปัญหาต่างๆจนเป็น wiener filter ซึ่งใช้กันอย่างกว้างขวางในการแก้ปัญหาของสัญญาณสะท้อนในปัจจุบันนี้



รูปที่ 2.9 Standard Adaptive Filter Problem

จากรูปแสดงปัญหาในการทำ Filter จากการประมาณค่า หรือกระบวนการ interference ซึ่งก็คือกระบวนการกระตุ้นหรือขัดจังหวะในสัญญาณ input สามารถแสดงเป็นสมการได้ดังต่อไปนี้

$$w_0 = R^{-1}p \qquad (2.5)$$

โดยที่ w_0 คือ tab-weight vector

R คือ ($N \times N$) auto-correlation matrix ของ tab inputs

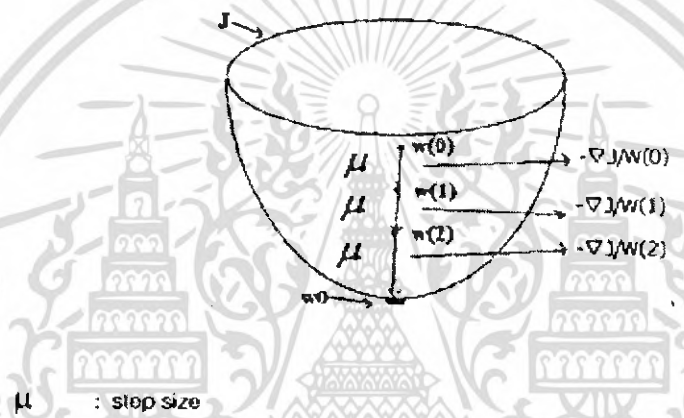
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 Method of Steepest decent

จากหลักการของ wiener filter จะต้องมีการหาค่า R และ R^{-1} ไม่ต้องหา ค่า R^{-1} จะใช้วิธีการ interactive แทน ดังนั้นวิธีการนี้จึงเป็นวิธีการหาจุดต่ำสุดด้วยความเร็วสูงสุด ต่างรู้ค่า R กับค่า p เพื่อที่จะหาค่า W ให้เร็วที่สุด โดยต้องปรับค่า m ให้เหมาะสมดังรูป

$$\begin{aligned} w(n+1) &= w(n) + \frac{1}{2}\mu[-\nabla J(n)] \\ &= w(n) + \mu[p - Rw(n)] \end{aligned} \tag{2.6}$$

w(n) : weight vector ที่ต้องการปรับเปลี่ยน



รูปที่ 2.10 แสดงวิธีการปรับเปลี่ยนค่า step size

จากรูปจะเห็นได้ว่า J คือ ความผิดพลาด (error) จะมีการหาจุดต่ำสุดโดยการ differential เทียบกับค่า w(n) โดยทำการปรับเปลี่ยนค่า μ ให้เหมาะสมแล้วหาค่า w_0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ทฤษฎีการใช้งาน adaptive filter

ในหัวข้อนี้เราจะกล่าวถึงการใช้งานAdaptive filter ซึ่งนำมาใช้งานอย่างแพร่หลายโดยวงจรกรองที่จะกล่าวถึงคือ Least Mean Square (LMS), Normalize Least Square (NLMS)

2.7.1 Least Mean Square Algorithm (LMS)

LMS เป็นการหา weight vector จากการ minimize ค่า $E[e(n)^2]$ โดยค่าของ weight vector จะปรับให้ใกล้เคียงกับค่า ที่อยู่ในระบบ Network ข้อดีคือสามารถ apply กับ non-stationary signal ได้ เป็นการ implement steepest descent โดยไม่ต้องรู้ค่า p และ R จากสมการของ steepest descent เราจะแปลงเป็นสมการของ LMS ดังนี้

เมื่อเราให้

$$\hat{R}(n) = u(n)u^{H}(n) \qquad \hat{p}(n) = u(n)d^{*}(n) \tag{2.7}$$

เราจะได้สมการมาใหม่ดังนี้

$$\begin{aligned} w(n+1) &= w(n) + \mu[p - Rw(n)] \\ &= w(n) + \mu[u(n)d^{*}(n) - u(n)u^{H}(n)w(n)] \\ &= w(n) + \mu u(n)[d^{*}(n) - y^{*}d(n)] \\ &= w(n) + \mu u(n)e^{*}(n) \end{aligned} \tag{2.8}$$

จากสมการจะเห็นได้ว่าเราจะใช้สมการของ Method of Steepest decent มาใช้อยู่โดยเมื่อแทนค่า R และ p ใหม่เข้าไปก็จะได้สมการใหม่ดังกล่าว ดังนั้นสมการนี้คือ สมการของการหาค่า Weight vector ของ LMS Algorithm

- e คือ ค่าความผิดพลาด (Error) ระหว่างสัญญาณออกของ acoustic echo path กับสัญญาณที่ได้จาก adaptive filter
- w(n+1) คือ สมการปรับค่าสัมประสิทธิ์ของ adaptive filter เพื่อนำไปคำนวณค่าผิดพลาดในรอบใหม่ต่อไป
- u(n) คือ สัญญาณ(input signal)
- d(n) คือ สัญญาณอ้างอิง (desired signal)
- e(n) คือ สัญญาณความผิดพลาดซึ่งจะมีค่าน้อยที่สุดเมื่อปราศจาก echo
- * คือ complex conjugate
- H คือ Hermitian transposition

LMS algorithm คือ ความสามารถของการมาถึงที่การประมาณค่าใกล้เคียง ของวิธีการแก้ปัญหาเดียวกัน เช่นเดียวกับ Weiner-Hop equation แต่ปราศจากการแก้ปัญหาสมการโดยตรง LMS algorithm ใช้วิธี stochastic ในการค้นหา optimum Weiner solution คือ w_0 โดยลดทอนค่าของ mean-square error ซึ่งค่า error เป็นผลต่างระหว่าง output จาก filter และ desired response พื้นฐานของมันเป็นวิธีการแก้ปัญหาที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดีที่สุด ซึ่งในกรณีนี้คือ mean square error สามารถแสดงได้ดังสมการนี้

$$e(n) = d(n) - \sum_{k=0}^{M-1} w(n)x(n-k) \tag{2.9}$$

$$J = \sum_{e=1}^{N} e^2(n) \tag{2.10}$$

LMS มีจำนวนคุณสมบัติที่เหมาะสมที่ทำให้เป็น algorithm ซึ่งเป็นที่นิยมมากในการใช้และข้อดีที่สุดของ LMS คือ ง่ายในการสร้าง มันไม่ต้องการการคำนวณโดยตรงของ function ที่มีความสัมพันธ์ ซึ่งคำนวณได้ยาก และในการดำเนินการมีเพียงการทำซ้ำของ (N+1) การคูณและ N การบวกเท่านั้น โดยที่ N ก็คือ tab-weight ใน filter ส่วนคุณสมบัติเด่นๆอีกอย่างหนึ่งก็คือ มันมีความเสถียรสูง ความคลาดเคลื่อนน้อย เงื่อนไขซึ่งพอดีกับสภาวะเสถียรของ LMS คือ step size อยู่ในช่วง 0 < u < umax โดยที่ umax คือ eigenvalue ที่ใหญ่ที่สุดของ input (auto correlation matrix) อัตรา convergence ของ LMS ขึ้นอยู่กับค่าทางสถิติของสัญญาณ input

การปรับเปลี่ยนค่า Weight - vector (LMS)

Filter output

$$y(n) = \hat{w}^H U(n) \tag{2.11}$$

Estimation error

$$e(n) = d(n) - y(n) \tag{2.12}$$

tap – weight adaptation

$$w(n+1) = w(n) + \mu u(n)e^*(n) \tag{2.13}$$

วิธีการหาค่า step size (μ)

$$\mu <= 2/\text{max} \tag{2.14}$$

ค่า max คือ ค่า eigenvalue ที่มากที่สุดของ Correlation Matrix ของ tap input

$$\mu <= 2/\text{MSmax} \tag{2.15}$$

ค่า M คือ ความกว้างของ filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า Smax คือ Power spectral density of the tap input ต้องใช้ Furrier transform เข้ามาช่วย

$$\mu <= \min\ 1<=n<=N\ (1/\|u(n)\|^2) \tag{2.16}$$

N คือ ค่าจำนวนรอบที่วนรอบ

u(n) คือ tap weight vector

เราเลือกใช้วิธีที่ 3 เพราะว่าง่ายต่อการคำนวณและนำมาใช้งาน

เมื่อเรามี w(n) ปัจจุบันแล้วการหา Y เราต้องคูณ w(n) เข้ากับ $u^H(n)$ จะได้ Y และเมื่อนำ d มาลบออกจะได้ค่า e (error) เมื่อค่า error ยังมากจนยังได้ยินเสียง echo เราต้องทำการปรับเปลี่ยน weight คือ ค่า $w(n+1) = w(n) + \mu u(n)e^*(n)$ จะสังเกตว่า $Z^{-1}I$ คือ delay จะมีค่าเท่ากับ w(n) หรือ weight ครั้งที่แล้วนั่นเอง

2.7.2 Normalized Least Mean Square algorithm (NLMS)

จาก LMS algorithm ที่เราได้สมการเปลี่ยนค่าของ weight vector มาว่า

$$w(n+1) = w(n) + \mu u(n)e^*(n) \tag{2.17}$$

ซึ่ง ค่า μ จะขึ้นอยู่กับ eigen value ถ้ามีมากอัตรา convergence ก็จะมา ในวิธีนี้เราต้องทำการ Normalized ค่า eigen value ก่อนทำได้โดยการ minimum disturbance คือ weight vector เปลี่ยนแปลงน้อยที่สุด โดยมีเงื่อนไขที่ filter output เราพยายามที่จะเปลี่ยนแปลง w(n+1) ให้น้อยที่สุด และพยายามให้ y(n+1) = d(n) จะทำให้ e(n) มีค่าน้อย โดยปรับเปลี่ยน w(n) จากค่าปัจจุบันไปยังค่าในอนาคตให้น้อยที่สุด

$$\delta w(n+1) = w(n+1) - w(n) \tag{2.18}$$

$$w^H(n+1)u(n) = d(n) \tag{2.19}$$

เราใช้ Method of Lagrange multiplier เข้ามาช่วยในการปรับแต่งจะได้

$$J(n) = \|\delta w(n+1)\|^2 + \mathrm{Re}\left[\lambda(d(n) - w^H(n+1)u(n))\right] \tag{2.20}$$

จะได้ $\frac{\partial J(n)}{\partial W_k} = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w(n+1) = w(n) + \frac{\tilde{\mu}}{\|u(n)\|^2} u(n)e^*(n) \tag{2.21}$$

แต่เนื่องจากบางทีค่า $\|u(n)\|^2$ อาจเป็น 0 ได้ เราจึงต้องเพิ่มค่าเข้าไป

การหาค่า $\tilde{\mu}$

$$0 < \tilde{\mu} < \frac{2E[|u(n)|^2]d(n)}{E[|e(n)|^2]} \tag{2.22}$$

โดย $E[|u(n)|^2]$, $E[|e(n)|^2]$ สามารถหาได้จากสูตร

$$x^2(n+1) = (1-\gamma)x^2(n+1) + \gamma x^2(n) \tag{2.23}$$

ค่า γ = [0.9, 0.999]

d(n) = mean square deviation

d(n)= $E[\|\varepsilon(n)\|^2$

$\varepsilon(n) = w_0 - w(n)$

w_0 คือ ค่า tap weight สุดท้ายหรือค่า tap weight ของ network

ค่าของ d(n) จะมีการเปลี่ยนแปลงตลอดเวลา เรารู้แค่ค่าของ w(n) แต่เราไม่มีทางรู้ค่าของ w_0 ได้เลย เราเลยต้องกำหนดค่า delay จำนวน M_D ประมาณ 10% เพิ่มก่อนที่ เสียงจะผ่านเข้าไปใน Network ของเรา แล้วทำการเทียบบัญญัติไตรยางศ์หาค่าผลต่างที่เหลือตามสูตร

$$d(n) = \frac{M}{M_D} \sum_{k=0}^{M-1} w_k^2(n) \tag{2.24}$$

การเพิ่ม M_D จะทำให้เรารู้ weight vector ของ network จำนวน M_D ตัวแรก ว่าเป็นศูนย์ ดังนั้นผลต่างก็คือ weight vector เริ่มต้นนั่นเอง แต่การตั้งค่าเริ่มต้นของ weight vector จะตั้งให้เป็นศูนย์หมดเลยไม่ได้ เพราะจะทำให้ค่า d(n) เป็นศูนย์หมด ค่า μ ก็จะเป็นศูนย์ด้วยทำให้ไม่มีการเปลี่ยนแปลง weight vector เลย จึงต้องตั้งค่า weight vector ใหม่ให้ M_D ค่าแรกๆเป็นค่าน้อยๆ จึงจะมีการปรับเปลี่ยน weight vector

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 Quantized – error Algorithm

ความยุ่งยากซับซ้อนในการคำนวณนั้นโดยส่วนใหญ่แล้วจะเกิดจากรูปแบบของการคูณในการ update ค่าสัมประสิทธิ์ และการคำนวณเอาท์พุตของ adaptive filter ใน Application ที่ adaptive filter ต้องการการประมวลผลที่เร็ว เช่น echo cancellation และ channel equalizer มันจำเป็นที่จะต้องลดความยุ่งยากพวกนี้ออกไป

วิธีการหนึ่งที่จะทำให้การคำนวณใน LMS algorithm นั้นง่ายขึ้น คือ ปรับปรุงการ quantization ของสัญญาณที่ผิดเพี้ยน วิธีการสร้าง quantized-error algorithm จะเป็นดังสมการ

ซึ่ง Q[.] แสดงการ quantization ฟังก์ชั่น ของการ quantization นั้นเป็นค่า discrete ที่มีขอบเขต และไม่มีการลดลง ชนิดของการ quantization จะเป็นตัวที่แสดง quantized-error algorithm

ถ้า μ คือ ค่าเลข power - of - two ของการ update ค่าสัมประสิทธิ์ สามารถใช้การคูณอย่างง่ายๆ ที่ประกอบไปด้วย bit shift และ ตัวบวก ในการไปประยุกต์เป็น echo cancellation ในการกระจายแบบ full duplex และ channel equalizer ที่ข้อมูลเป็น binary สัญญาณอินพุต x(k) จะเป็น สัญญาณ binary แทนด้วย +1 และ -1 ซึ่งถ้าเป็นในกรณีนี้ adaptive filter จะสามารถดำเนินการโดยไม่มีความยุ่งยากเลย

การ quantization error เป็นตัวที่บ่งบอกว่าการปรับตัวใน function นั้นจะถูกลดค่าลง แทนด้วย F[e(k)] ในการ update ทั่วไปจะแสดงด้วย

$$\begin{aligned} w(k+1) &= w(k) - u\frac{\partial F[e(k)]}{\partial w(k)} \\ &= w(k) - u\frac{\partial F[e(k)]}{\partial e(k)}\frac{\partial e(k)}{\partial w(k)} \end{aligned} \tag{2.25}$$

ซึ่ง F[e(k)] หาได้จากการ integrate Q[e(k)] ซึ่งจะสัมพันธ์กับ e(k) ผลของ quantized-error และ LMS algorithm จะต่างกันอย่างเห็นได้ชัด

2.7.4 Sign – Error Algorithm

รูปแบบที่ง่ายที่สุดของ quantization function คือ sign (sgn) function ซึ่งแสดงโดย

$$\text{sgn}[b] = \begin{cases} 1, & b > 0 \\ 0, & b = 0 \\ -1, & b < 0 \end{cases} \tag{2.26}$$

Sign error algorithm ใช้ sign function เหมือน error quantization ซึ่ง การ update สัมประสิทธิ์ เป็นดังสมการ

$$w(k+1) = w(k) + 2\mu\,\text{sgn}[e(k)]x(k) \tag{2.27}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62868

โครงสร้างของ sign – error algorithm สำหรับเส้นทางของอินพุต x(k) ที่ delay การทำซ้ำของ sign – error algorithm ต้องการการคูณ N+1 ครั้งเพื่อสร้าง error และจำนวนการบวกทั้งหมดคือ 2N+2 จะเห็นว่า x(0) และ w(0) จะถูกแสดงในทางตรงกันข้ามซึ่ง อธิบายใน algorithm

Function จะถูกลดรูปโดย sign – error algorithm คือ สัมประสิทธิ์ของ error จะถูกคูณด้วย 2 ดังนี้

$$F[e(k)] = 2|e(k)| \tag{2.28}$$

สังเกตว่าตัวประกอบ 2 จะถูกรวมเพื่อแสดง sign – error และ LMS algorithm ในรูปแบบเดียวกัน เห็นได้ว่าในการปฏิบัติจริง ตัวประกอบนี้ถูกรวมไว้กับค่า μ

ลักษณะต่างๆที่สัมพันธ์กับการทำงานของ sign – error algorithm ในสภาพแวดล้อมที่เหมือนเดิม ไม่เปลี่ยนแปลงจะอธิบายได้ตามกระบวนการเช่นเดียวกับ LMS algorithm

2.7.5 Sign Data Algorithm

ปกติ algorithm ในหัวข้อนี้จะไม่พิจารณาได้ว่าเหมือนกันกับ quantized – error algorithm แต่ที่มันถูกเสนอด้วยเหตุจูงใจหลายๆอย่างที่เหมือนกันนี่เป็นทางเลือกที่จะทำให้ LMS algorithm ง่ายขึ้น โดยประยุกต์ quantization เป็น vector ข้อมูล x(k) โดยแบบที่เป็นไปได้คือ การเปลี่ยน sign function เป็นสัญญาณอินพุต ซึ่งรูปแบบ update สัมประสิทธิ์ คือ

$$w(k+1) = w(k) + 2\mu e(k)\,\mathrm{sgn}[x(k)] \tag{2.29}$$

ซึ่งขั้นตอนนี้จะถูกประยุกต์เป็นส่วนประกอบของ input vector การ quantization ของ vector ข้อมูลจะนำไปสู่การลดลงของความเร็วในการ convergence หรือบางทีอาจนำไปสู่การ divergence ใน LMS algorithm ข้อจำกัดของ sign data algorithm อาจอยู่ที่การ update ค่าเนื่องจากการเพิ่มขึ้นเรื่อยๆ ของ square error ซึ่งอาจนำไปสู่ความไม่เสถียร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

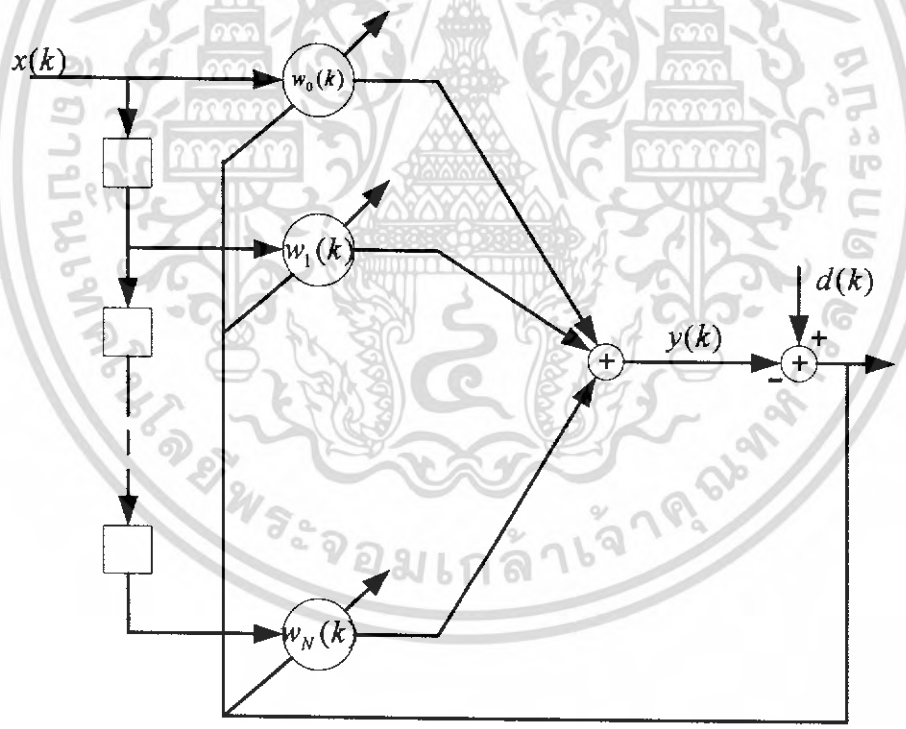
บทที่ 3
การออกแบบและการสร้างชุดการทำงานเพื่อกำจัดเสียงสะท้อน

3.1 บทนำ

ในบทนี้จะเป็นการกล่าวถึงแนวทาง และวิธีการที่นำมาใช้ในการสร้างตัวกำจัดเสียงสะท้อน โดยจะใช้โปรแกรม MATLAB มาช่วยในการจำลองการทำงานของตัวกำจัดเสียงสะท้อน เพื่อศึกษาถึงคุณสมบัติต่างๆ ของพารามิเตอร์ (parameter) ที่เกี่ยวข้องกับตัวกำจัดเสียงสะท้อน และนำผลที่ได้มาประยุกต์ใช้งานผ่าน FPGA โดยใช้ภาษา VHDL

3.2 การออกแบบโครงสร้างสำหรับการทดลอง

โครงสร้างของอะแดปทีปฟิลเตอร์ (adaptive filter) ที่นำมาเป็นต้นแบบในการใช้สร้างตัวกำจัดเสียงสะท้อนแสดงได้ดังรูปที่ 3.1 โดยโครงสร้างของอะแดปทีปฟิลเตอร์นี้จะใช้อัลกอริธึม(algorithm) ชนิด LMS (Least – mean – square) มาใช้ โดยอัลกอริธึมนี้เป็นวิธีที่สามารถนำไปประยุกต์ใช้ได้มากมาย



รูปที่ 3.1 โครงสร้างของ adaptive filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบสำหรับ LMS Algorithm

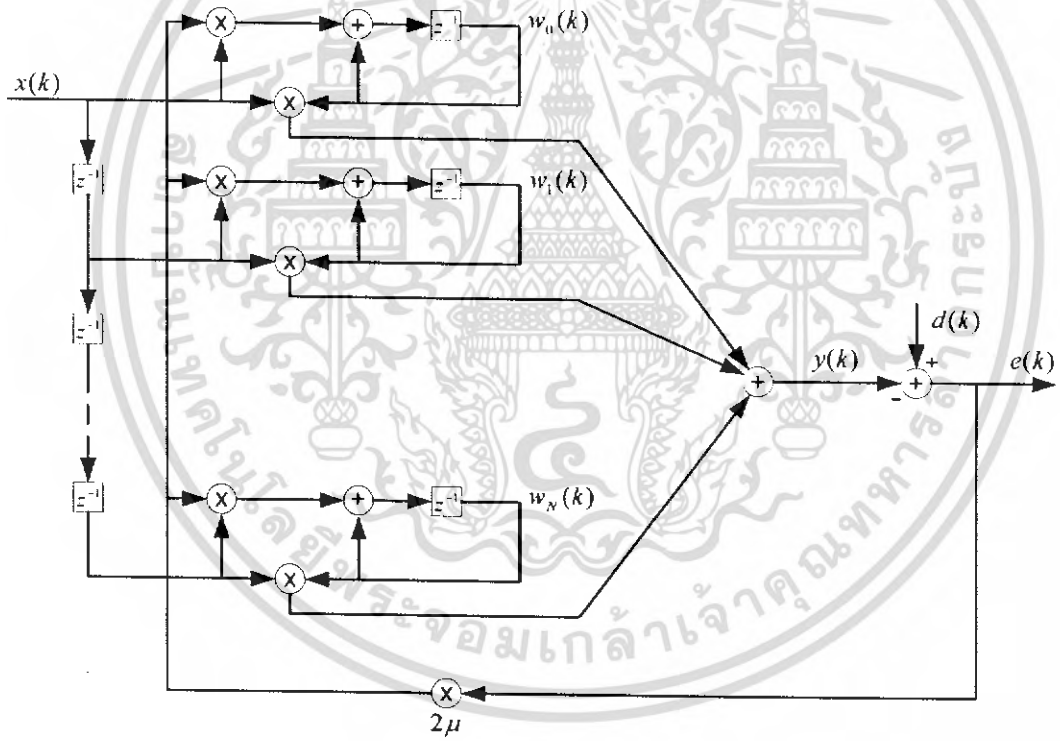
ในการทดลองสำหรับวิธี LMS algorithm จะสามารถหาค่าเอาท์พุท (output) จากตัวกำจัดเสียงสะท้อนจาก

$$y(k) = \sum_{i=0}^{N} w_i(k)x_i(k) \qquad (3.1)$$

โดยจะสามารถปรับค่าสัมประสิทธิ์ของฟิลเตอร์ได้จากสมการคือ

$$w(k+1) = w(k) + 2\mu e(k)x(k) \qquad (3.2)$$

โดยสามารถที่จะแสดงโครงสร้างของ adaptive filter ชนิด LMS algorithm ได้คือ



รูปที่ 3.2 แสดงโครงสร้างของ adaptive filter ชนิด LMS algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบสำหรับ Quantized – error algorithm

ในการทดลองสำหรับวิธี Quantized – error algorithm จะสามารถหาค่าเอาท์พุท (output) จากตัวกำจัดเสียงสะท้อนจากสมการที่ (3.1) โดยในการปรับค่าสัมประสิทธิ์ของฟิลเตอร์สามารถที่จะทำได้ 2 วิธี คือ sign – error algorithm และ sign – data algorithm โดยสามารถปรับค่าสัมประสิทธิ์ของฟิลเตอร์ได้ดังสมการ

$$w(k+1) = w(k) + 2\mu \operatorname{sgn}(e(k))x(k) \qquad (3.3)$$

$$\operatorname{sgn}(e(k)) = \begin{cases} 1 & , e(k) > 0 \\ 0 & , e(k) = 0 \\ -1 & , e(k) < 0 \end{cases} \qquad (3.4)$$

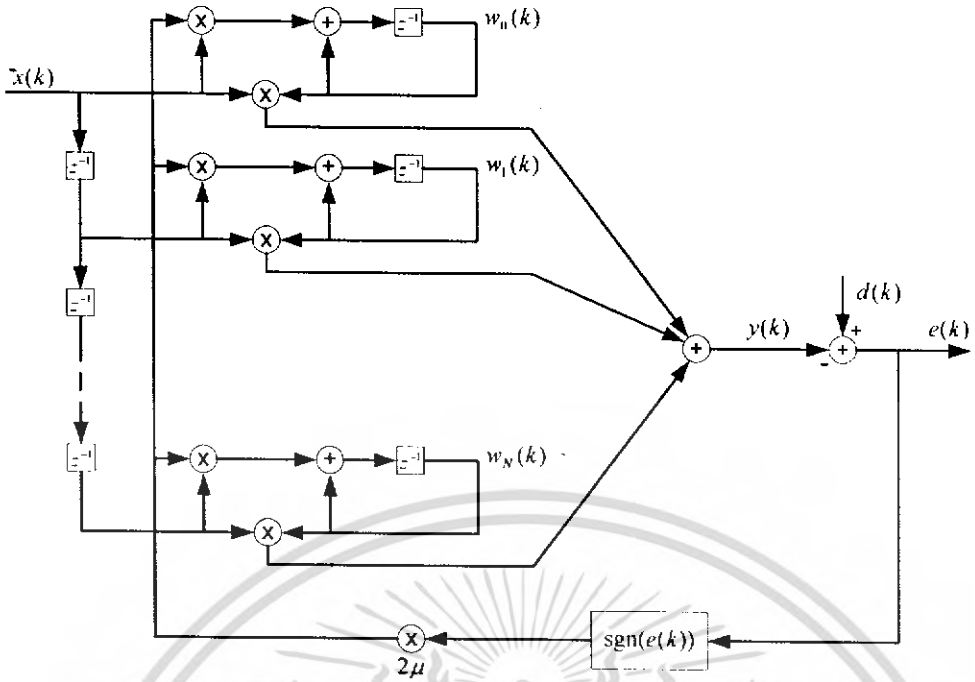
โดยสมการที่ (3.3) เป็นสมการปรับค่าสัมประสิทธิ์ของฟิลเตอร์ สำหรับวิธี sign – error algorithm

$$w(k+1) = w(k) + 2\mu e(k) \operatorname{sgn}(x(k)) \qquad (3.5)$$

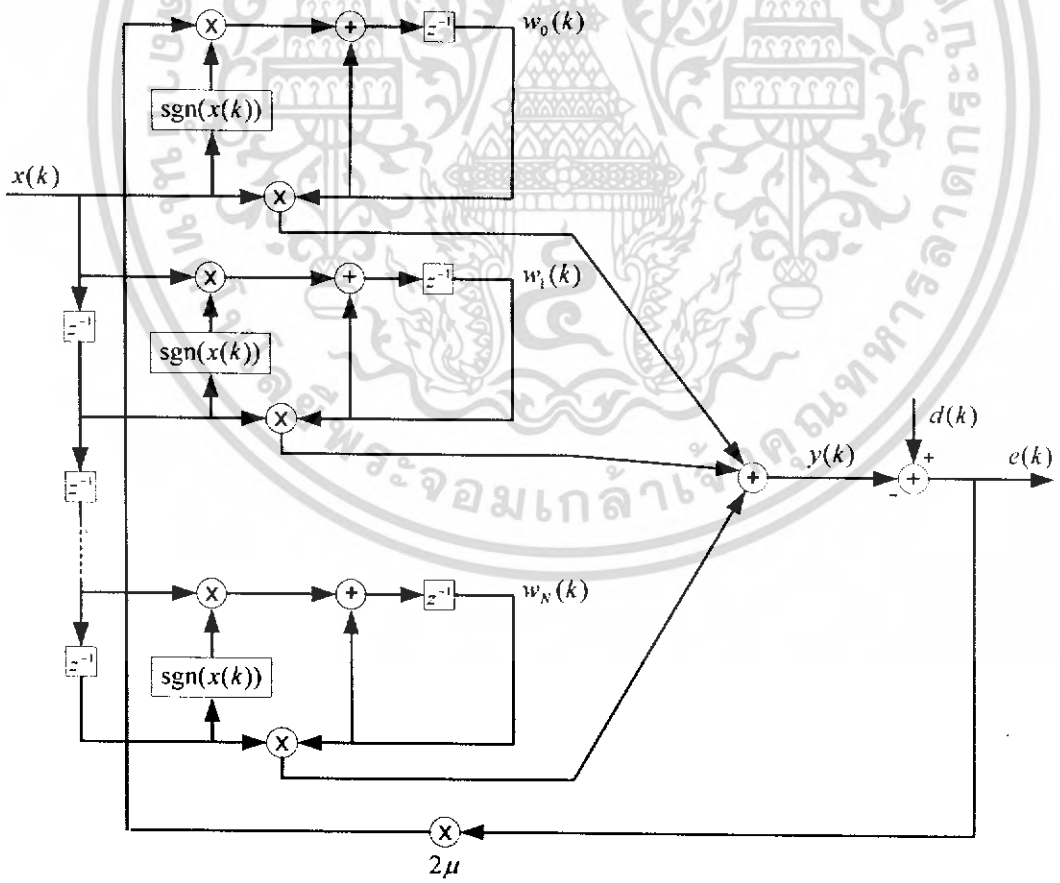
$$\operatorname{sgn}(x(k)) = \begin{cases} 1 & , x(k) > 0 \\ 0 & , x(k) = 0 \\ -1 & , x(k) < 0 \end{cases} \qquad (3.6)$$

โดยสมการที่ (3.5) เป็นสมการปรับค่าสัมประสิทธิ์ของฟิลเตอร์ สำหรับวิธี sign – data algorithm โดยสามารถที่จะแสดงโครงสร้างของ adaptive filter ชนิด Quantized – error algorithm ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงโครงสร้างของ adaptive filter ชนิด sign – error algorithm



รูปที่ 3.4 แสดงโครงสร้างของ adaptive filter ชนิด sign – data algorithm

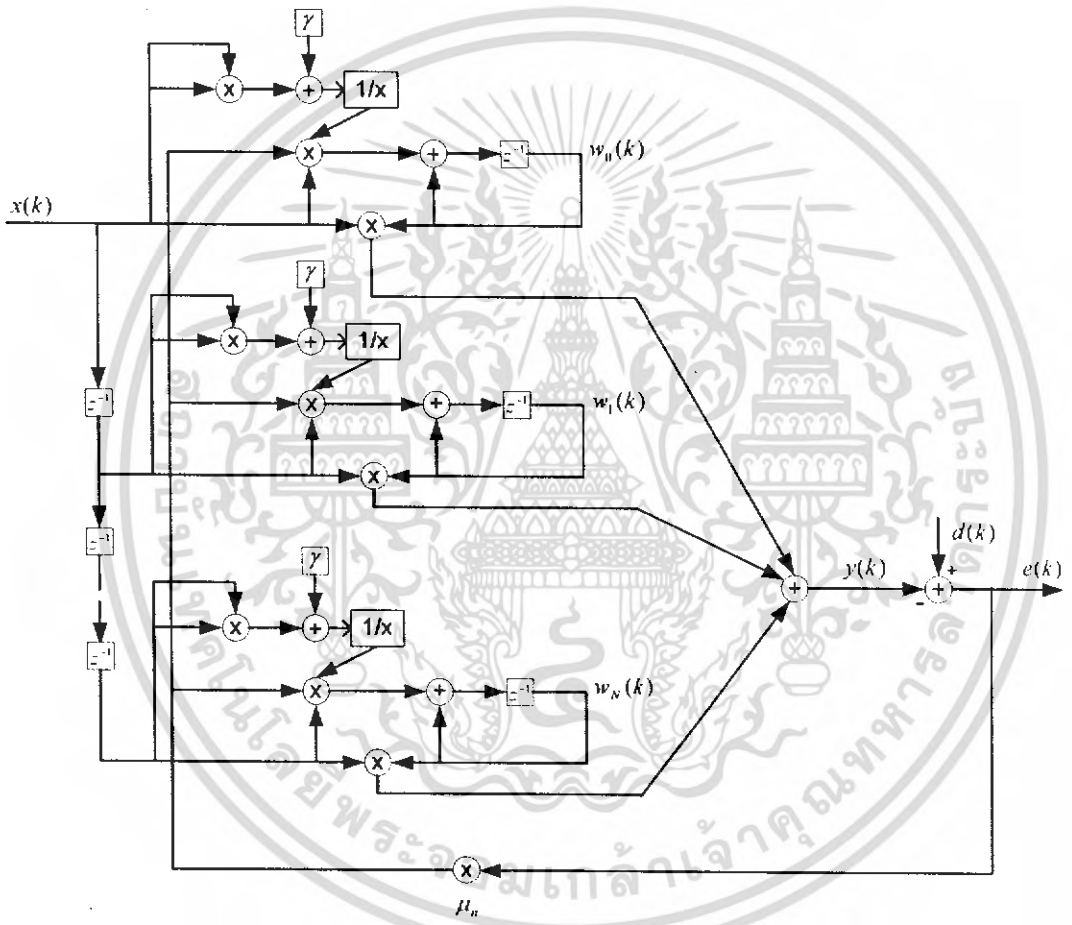
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 Normalized LMS algorithm (NLMS)

ในการทดลองสำหรับวิธี Normalized LMS algorithm จะสามารถหาค่าเอาท์พุท (output) จากตัวกำจัดเสียงสะท้อนจากสมการที่ (3.1)โดยในการปรับค่าสัมประสิทธิ์ของฟิลเตอร์สามารถหาได้ดังสมการ

$$w(k+1) = w(k) + \frac{\mu_n}{\gamma + x^T(k)x(k)} e(k)x(k) \qquad (3.7)$$

โดยสามารถที่จะแสดงโครงสร้างของ adaptive filter ชนิด NLMS algorithm ได้คือ



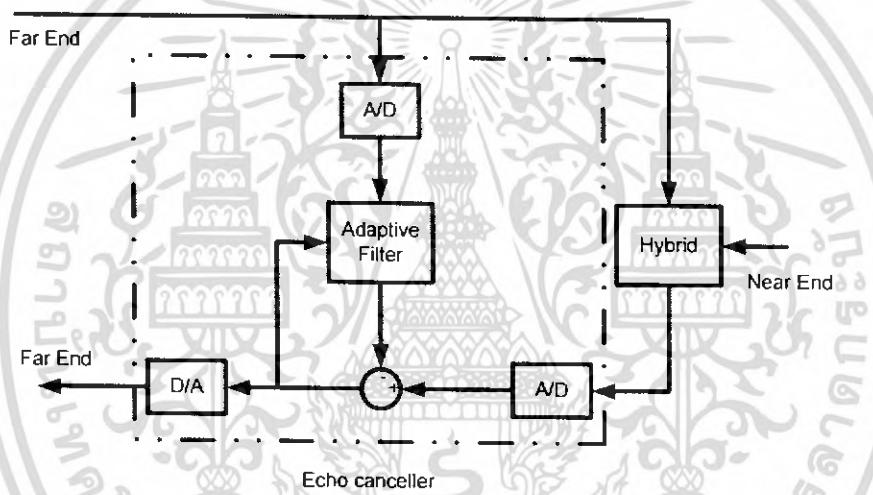
รูปที่ 3.5 แสดงโครงสร้างของ adaptive filter ชนิด NLMS algorithm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 แนวคิดในการออกแบบวงจรโดยใช้ FPGA

ในการออกแบบโครงสร้างของตัวกำจัดเสียงสะท้อนจะเป็นการนำเอาโมเดลทางคณิตศาสตร์ที่ได้จากการจำลองโดยโปรแกรม MATLAB มาสร้างบนอุปกรณ์ฮาร์ดแวร์ โดยจะทำการออกแบบในแต่ละส่วนแยกจากกัน จากนั้นจะทำการทดสอบการทำงานในระดับฟังก์ชันจนมีความถูกต้องตรงตามต้องการก่อน แล้วจึงนำแต่ละส่วนมาเชื่อมต่อเข้าด้วยกันเป็นระบบที่สมบูรณ์ ดังนั้นระบบที่ออกแบบจึงมีลักษณะเป็นระบบที่ทำการซิงโครนัสกันโดยใช้สัญญาณนาฬิกา

โดยในการทำงานสัญญาณก่อนที่จะเข้ามาในส่วนของ FPGA จะต้องทำการแปลงให้เป็นสัญญาณดิจิตอลโดยวงจรแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล (A/D Converter) หลังจากนั้นก็จะนำสัญญาณที่ได้ไปประมวลผลเพื่อทำการคำนวณหาค่าของเอาท์พุตออกมาจาก FPGA โดยสัญญาณดิจิตอลอยู่จะต้องทำการแปลงให้เป็นสัญญาณอนาลอกโดยใช้วงจรแปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาลอก (D/A Converter) จึงจะได้สัญญาณเอาท์พุตที่ต้องการ



รูปที่ 3.6 การเชื่อมต่อของวงจรกำจัดสัญญาณเสียงสะท้อน

จากรูปที่ 3.6 จะพบว่าโครงสร้างของวงจรกำจัดสัญญาณเสียงสะท้อนจะประกอบไปด้วยส่วนประกอบหลักที่สำคัญ 2 ส่วน คือ

1) วงจรที่แปลงสัญญาณอนาลอกของสัญญาณเสียง ให้เป็นสัญญาณแบบดิจิตอลและแปลงกลับจากสัญญาณแบบดิจิตอลให้เป็นสัญญาณอนาลอก

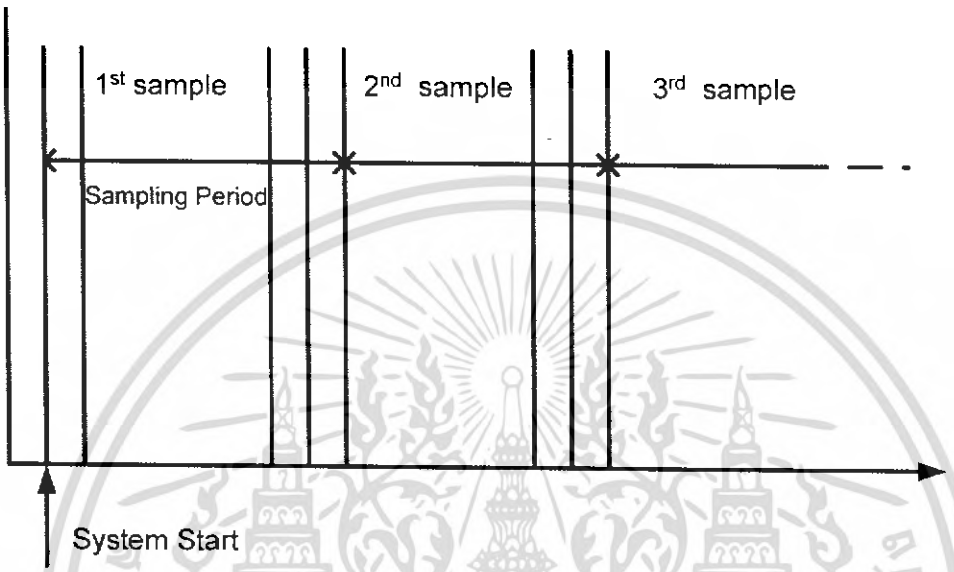
2) วงจรกรองปรับตัวได้แบบ FIR ซึ่งเป็นส่วนที่เป็นหัวใจหลักของระบบ และวงจรทางคณิตศาสตร์ที่จะทำการกำจัดสัญญาณเสียงสะท้อน

เนื่องจากการทำงานของโครงสร้างทั้งสองส่วนจะมีการทำงานร่วมกัน และข้อมูลที่เข้าสู่ระบบมีลักษณะเป็นอนุกรมเวลา ดังนั้นในการออกแบบสร้างระบบงานย่อยๆ ภายในวงจรกรองปรับตัวได้แบบ FIR จึงจำเป็นจะต้องมีการแบ่งโครงสร้างของแต่ละส่วนให้ทำงานอย่างสัมพันธ์กัน

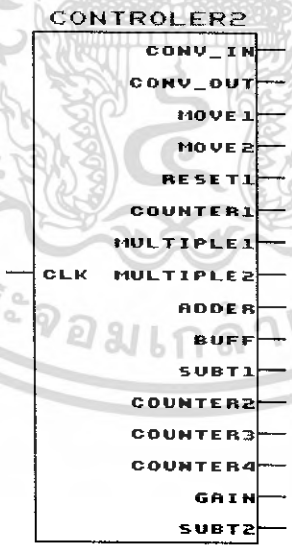
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.1 ส่วนของการควบคุม

ในการทำงานโดยใช้ FPGA ส่วนที่สำคัญที่สุดคือส่วนของการควบคุม เพราะว่าในการใช้งานจริงจะต้องคำนึงถึงเวลาที่ใช้ด้วย เช่น ช่วงเวลาไหนที่จะทำการรับข้อมูล ช่วงเวลาไหนที่จะทำการประมวลผล หรือช่วงเวลาไหนที่สามารถส่งข้อมูลออกไปได้ เป็นต้น ซึ่งหากทำการคำนวณเรื่องเวลาต่างๆ เหล่านี้ผิดพลาด จะทำให้การทำงานไม่เป็นไปตามที่คำนวณเอาไว้ก็เป็นได้



รูปที่ 3.7 แสดงช่วงเวลาที่ใช้ในการประมวลผลในแต่ละ Sample



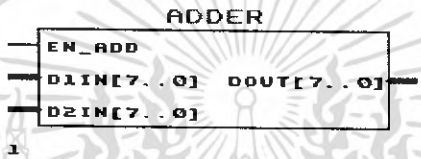
รูปที่ 3.8 แสดงบล็อกไดอะแกรมของวงจรในส่วนควบคุม (controller)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

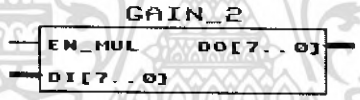
โดยในที่นี้ ข้อมูลที่ใช้ในการทำการทดลองนี้เป็นข้อมูลเสียง ซึ่งมีความถี่ประมาณ 300 Hz ถึง 3400 Hz สิ่งหนึ่งที่จะต้องพิจารณาก็คือค่าความถี่ที่ใช้ในการสุ่มค่าสัญญาณ (Sampling Rate)โดยค่าความถี่ที่ใช้ในการสุ่มสัญญาณจะต้องมีค่ามากกว่าหรือเท่ากับ 2 เท่าขององค์ประกอบความถี่สูงสุดของสัญญาณนั้น และช่วงเวลาที่ใช้ในการสุ่มค่าสัญญาณจะเป็นตัวกำหนดเวลาการประมวลผลของวงจรรวม ซึ่งจะต้องจัดการวางแผนในหน่วยการควบคุมให้ดีเพื่อให้ได้ผลการทดลองเป็นไปตามที่วางแผนไว้

3.6.2 หน่วยประมวลผลทางคณิตศาสตร์

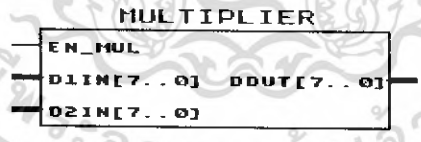
ในส่วนนี้จะมีหน้าที่ในการนำค่าข้อมูลต่างๆ มาทำการประมวลผลทางคณิตศาสตร์ นั่นก็คือการ บวก ลบ คูณ หรือหาร ตามที่ได้ตั้งการทำงานไว้ โดยในการทำการประมวลผลนั้น ข้อมูลที่นำมาใช้ยังคงเป็นข้อมูลประเภทดิจิตอล สิ่งที่ต้องคำนึงถึงในส่วนนี้คือ หลักการนำข้อมูลมาใช้ว่าเป็นข้อมูลแบบไหน เช่นเป็นข้อมูลแบบมีเครื่องหมาย หรือเป็นข้อมูลแบบทูคอมพลีเมนต์ (2's complement) เป็นต้น



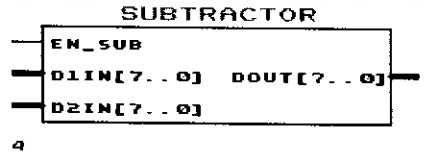
รูปที่ 3.9 แสดงบล็อกไดอะแกรมของวงจรบวก (adder)



รูปที่ 3.10 แสดงบล็อกไดอะแกรมของวงจรขยาย (gain)



รูปที่ 3.11 แสดงบล็อกไดอะแกรมของวงจรคูณ (multiplier)

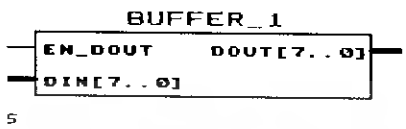


รูปที่ 3.12 แสดงบล็อกไดอะแกรมของวงจรลบ (subtractor)

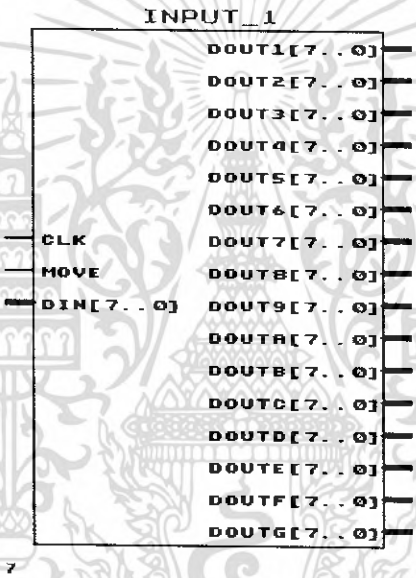
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.3 ส่วนของระบบการจัดเก็บข้อมูล

โดยในการจะทำการประมวลผลสัญญาณ จะมีส่วนหนึ่งที่มีหน้าที่คอยจัดการกับสัญญาณที่ส่งเข้ามา หรือส่งออกไปจากบอร์ด FPGA และก่อนที่จะส่งไปยังส่วนประมวลผลอื่นๆ ซึ่งส่วนนั้นก็คือส่วนที่ใช้ในการจะจัดเก็บข้อมูล โดยในการทำงานจะแบ่งเป็น ส่วนที่ทำการเก็บค่าของสัญญาณที่ถูกส่งเข้ามา และส่วนที่ทำหน้าที่ในการเก็บค่าสัมประสิทธิ์ของฟิวเตอร์

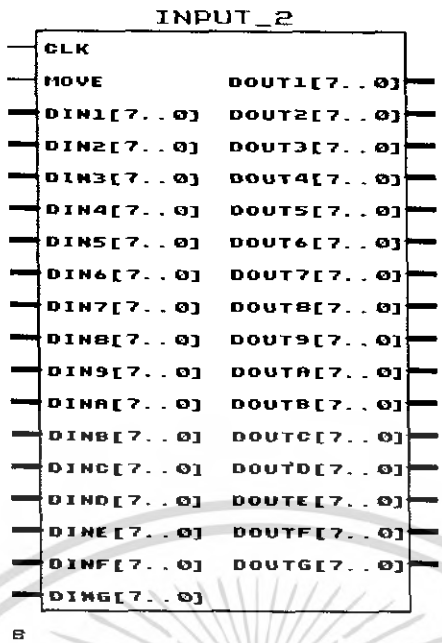


รูปที่ 3.13 แสดงบล็อกไดอะแกรมของวงจรบัฟเฟอร์ (buffer)



รูปที่ 3.14 แสดงบล็อกไดอะแกรมของวงจรที่ใช้ในการเก็บค่าสัญญาณ

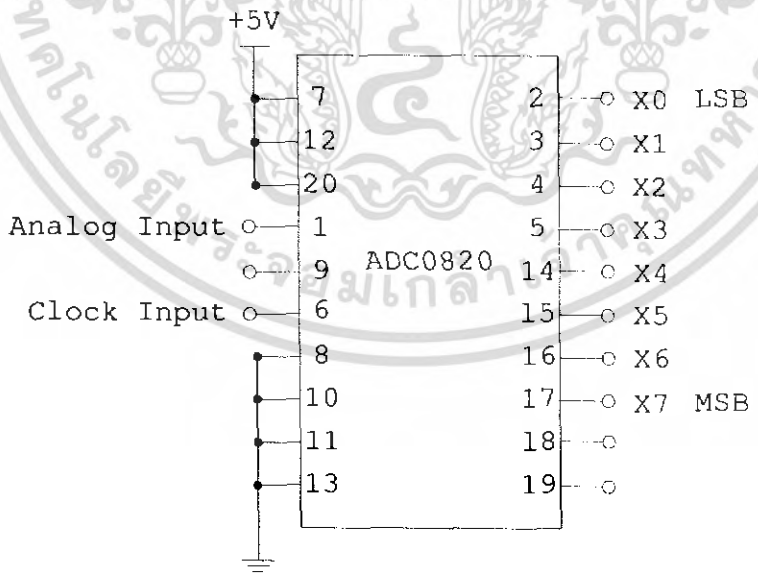
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงบล็อกไดอะแกรมของวงจรที่ใช้ในการเก็บค่าสัมประสิทธิ์ของฟิวเตอร์

3.6.4 การออกแบบและโครงสร้างการทำงานของชุดวงจร Analog to Digital Converter

เนื่องจากวงจรกำจัดเสียงสะท้อนจะต้องรับข้อมูลที่เป็นสัญญาณเสียงที่มีความถี่อยู่ในย่าน 0.3 – 3.4 kHz จึงจำเป็นจะต้องมีวงจรที่ทำการแปลงสัญญาณเสียงแบบอนาลอกให้เป็นดิจิตอล ในการออกแบบได้เลือกใช้ไอซีเบอร์ ADC0820 และสัญญาณดิจิตอลเอาท์พุตที่ได้จะถูกส่งไปยังวงจรอินพุตบัฟเฟอร์ของ FPGA โดยตรงเมื่อเสร็จสิ้นการทำงาน

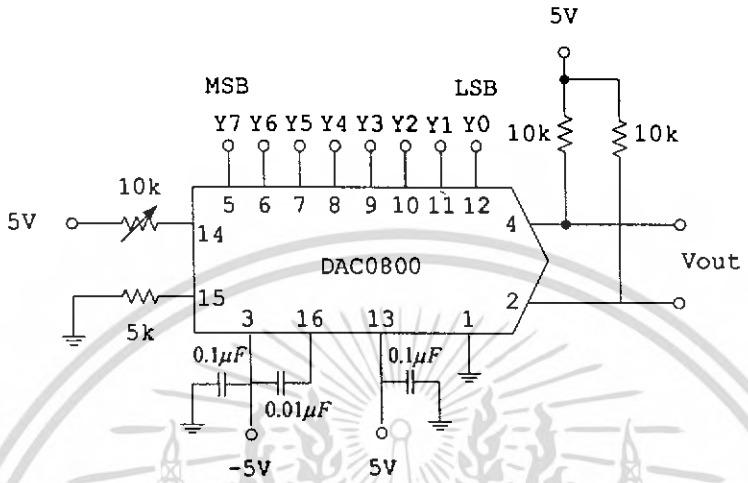


รูปที่ 3.16 แสดงวงจรแปลงสัญญาณเชิงอนาลอกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.5 การออกแบบและโครงสร้างการทำงานของชุดวงจร Digital to Analog Converter

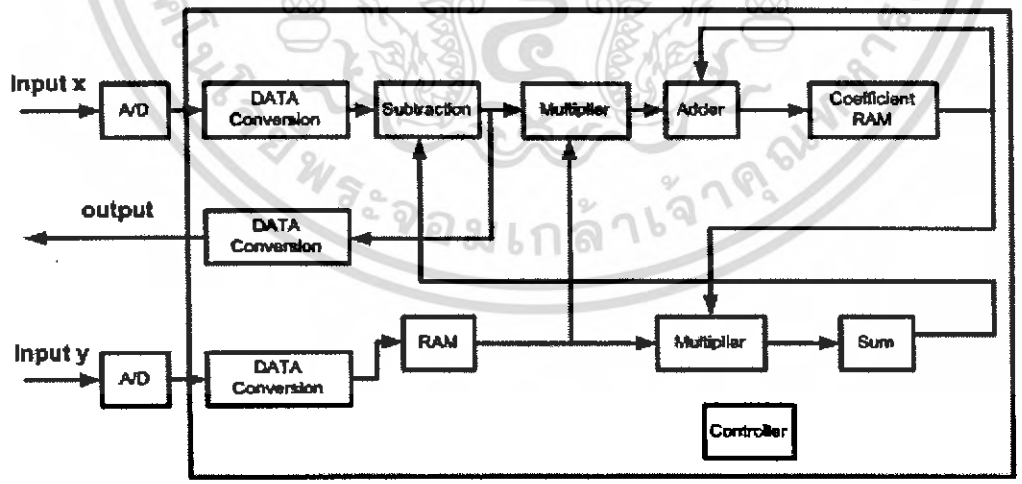
วงจรการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกจะทำการแปลงสัญญาณผลลัพธ์ที่ได้จากวงจรกำจัดเสียงสะท้อนที่อยู่ในรูปของสัญญาณดิจิตอลให้เป็นสัญญาณอนาลอก ในการออกแบบได้ใช้ไอซี DAC0800



รูปที่ 3.17 แสดงวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณเชิงอนาลอก

3.6.6 แบบแผนวงจรรวมของวงจรกำจัดเสียงสะท้อน

แบบแผนวงจรรวมของวงจรกำจัดเสียงสะท้อนจะได้จากการนำเอาแต่ละระบบย่อยๆ มาเชื่อมต่อกันให้ทำงานอย่างเป็นระบบ สามารถที่จะแสดงได้ดังรูปที่ 3.18

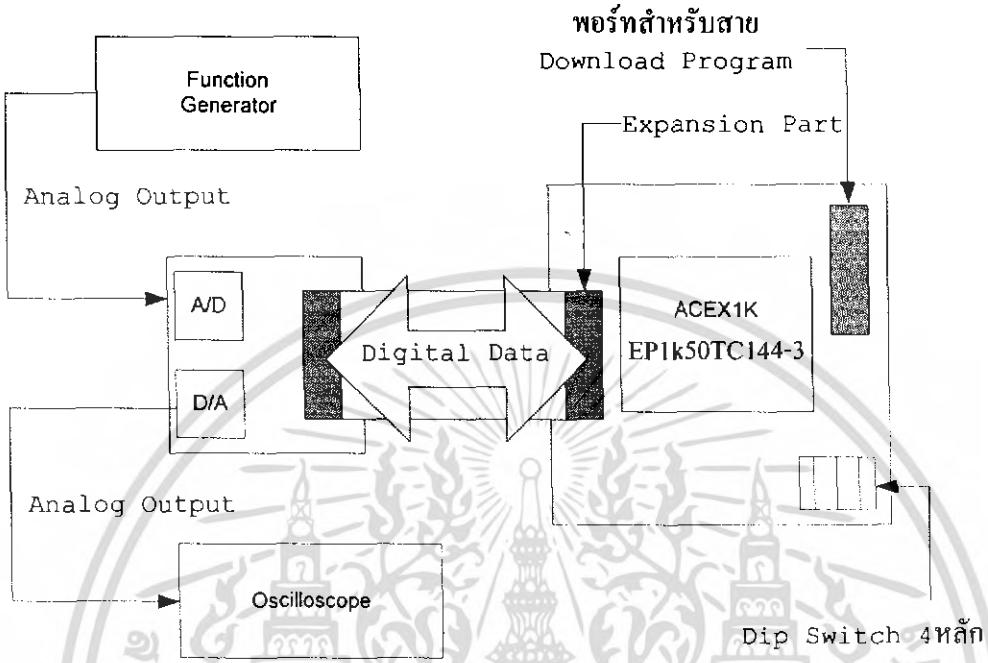


รูปที่ 3.18 แบบแผนวงจรรวมของระบบการทำงานที่จะถูกนำไปทำคอนฟิกูเรชันลงบนชิพ FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.7 การออกแบบการทดลองสำหรับการวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบ

ออกแบบวงจรรวมโดยใช้อุปกรณ์ทั้ง 3 ส่วนข้างต้นมาต่อรวมกันดังรูปที่ 3.18 เพื่อนำไปวัดคุณสมบัติของวงจรกรองสัญญาณที่ออกแบบไว้



รูปที่ 3.19 แสดงการเตรียมอุปกรณ์สำหรับการทดลองวัดคุณลักษณะของวงจรกรองสัญญาณที่ออกแบบขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

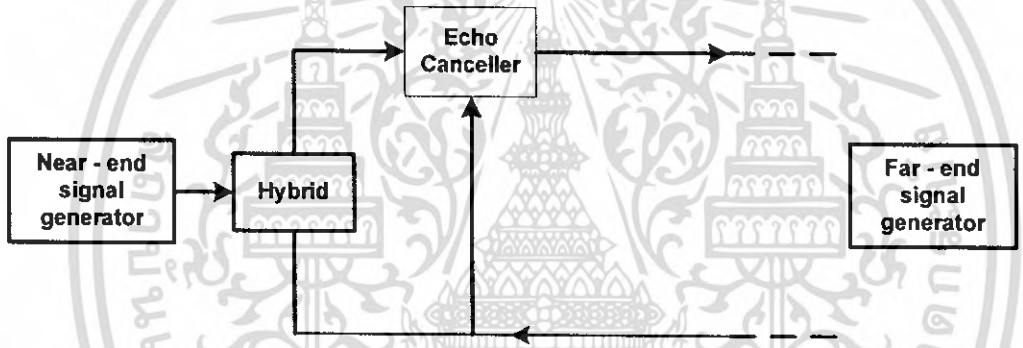
บทที่ 4
การทดลองและผลการทดลอง

4.1 บทนำ

ในบทนี้จะเป็นการนำเสนอผลการทดลองที่ได้ แบ่งได้เป็น 2 ส่วน คือ ส่วนผลการทดลองที่ได้จากการจำลองการทำงานด้วยโปรแกรม MATLAB และอีกส่วนคือส่วนของผลการทดลองที่ได้จากบอร์ด FPGA โดยจะเป็นการทดลองเพื่อกำจัดสัญญาณเสียงสะท้อนโดยใช้ adaptive filter

4.2 รูปแบบที่ใช้ในการทดลอง

ในการทำการจำลองผลการทดลอง จะทำการสร้างสัญญาณเสียงผู้พูดด้านไกล และสัญญาณเสียงผู้พูดด้านใกล้ โดยจะใช้ adaptive filter เป็นตัวกำจัดเสียงสะท้อนที่เกิดหลังจากผ่านวงจรไฮบริดจ์ในที่นี้ก็คือจะเป็นวงจรผลรวม ซึ่งสามารถแสดงวงจรได้ดังรูปที่ 4.1



รูปที่ 4.1 แสดงรูปของวงจรที่ใช้สำหรับการจำลองผลการทำงาน

4.3 ผลการทดลองเมื่อใช้ LMS Algorithm

ในหัวข้อนี้จะเป็นการแสดงผลการทดลองโดยใช้โปรแกรม MATLAB เพื่อจำลองผลการทำงานของตัวกำจัดเสียงสะท้อนโดยใช้ adaptive filter แบบ LMS Algorithm โดยในการจำลองผลการทำงานจะสนใจผลการทดลองเมื่อทำการเปลี่ยนค่า μ ว่ามีผลอย่างไรต่อฟิลเตอร์ โดยเมื่อพิจารณาจากสมการการปรับค่าสัมประสิทธิ์ของฟิลเตอร์ ดังสมการที่ (4.1) พบว่าค่า μ มีส่วนเกี่ยวข้องกับการเปลี่ยนค่าสัมประสิทธิ์ของฟิลเตอร์

$$w(k+1) = w(k) + 2\mu e(k)x(k) \qquad (4.1)$$

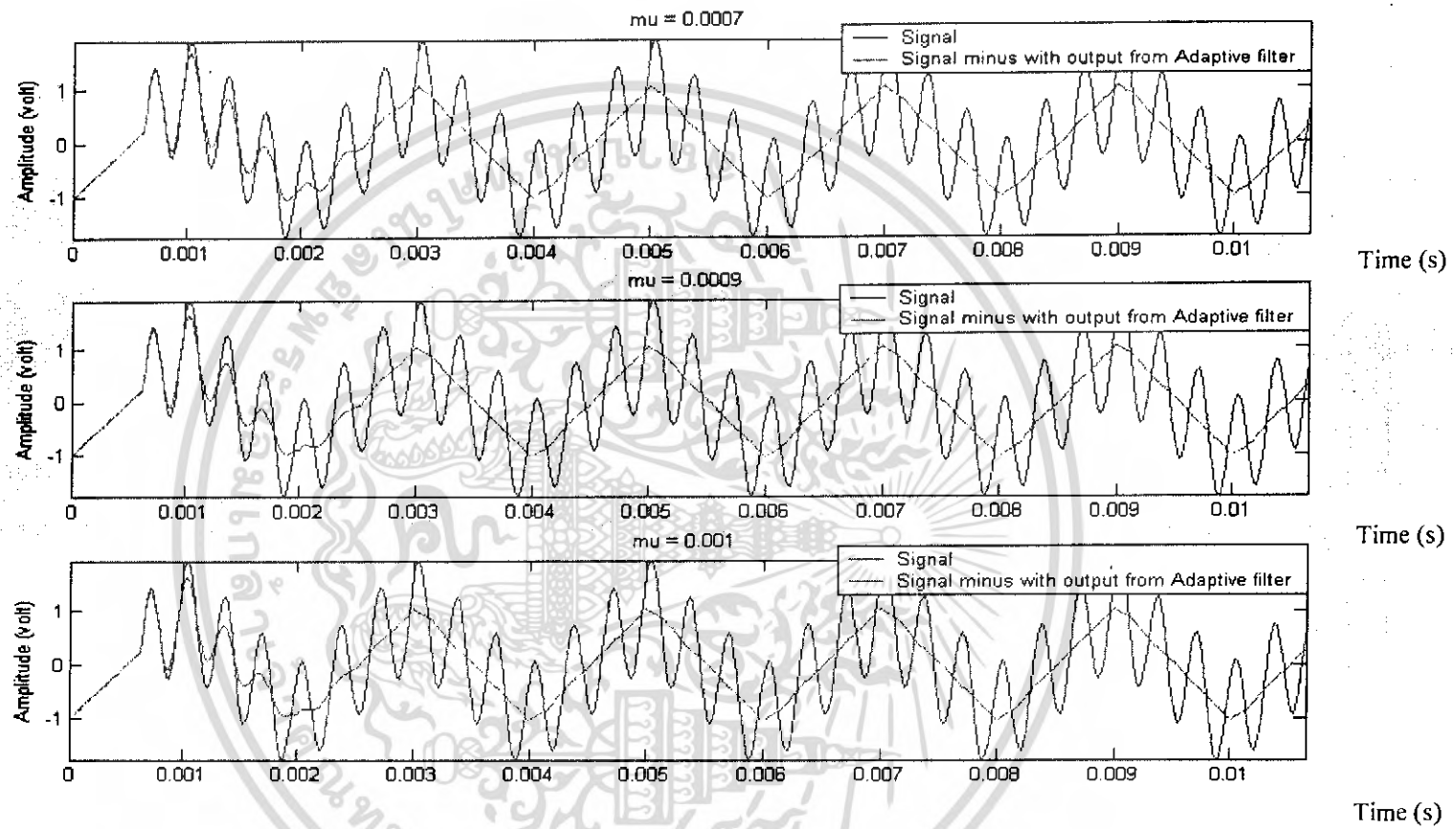
โดยในการทำการทดลองจะต้องทำการกำหนดค่าเริ่มต้นในการที่จะเขียนโปรแกรมเพื่อจำลองการทำงานก่อน โดยสามารถที่จะกำหนดได้ดังตารางที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรม
ของ Adaptive filter แบบ LMS Algorithm

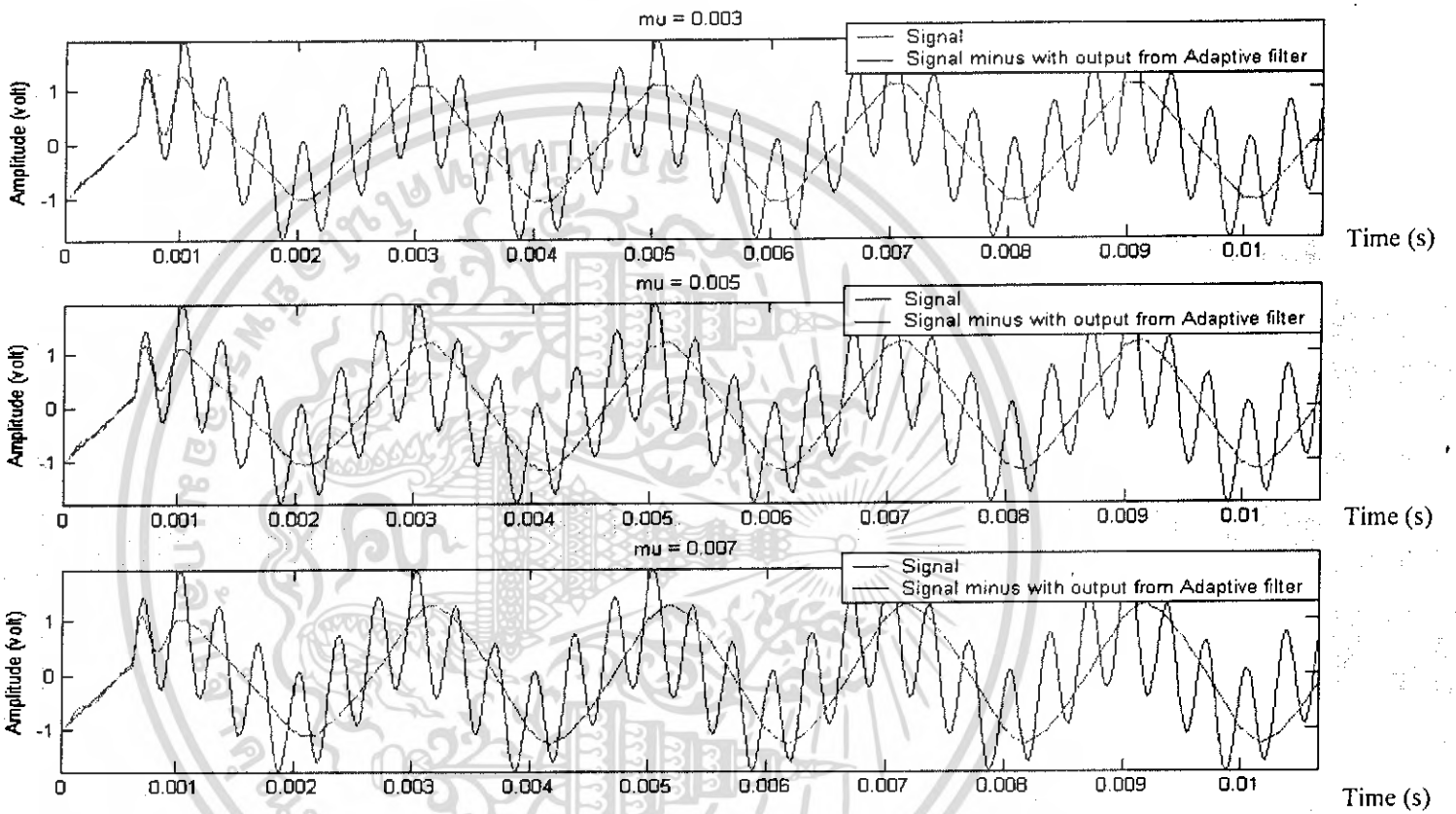
ค่าเริ่มต้นของ LMS Algorithm
ค่าเริ่มต้น $x(0) = w(0) = [0\ 0\ \ldots\ 0]^T$ เมื่อค่า $k \geq 0$ $e(k) = d(k) - x^T(k)w(k)$ $w(k+1) = w(k) + 2\mu e(k)x(k)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



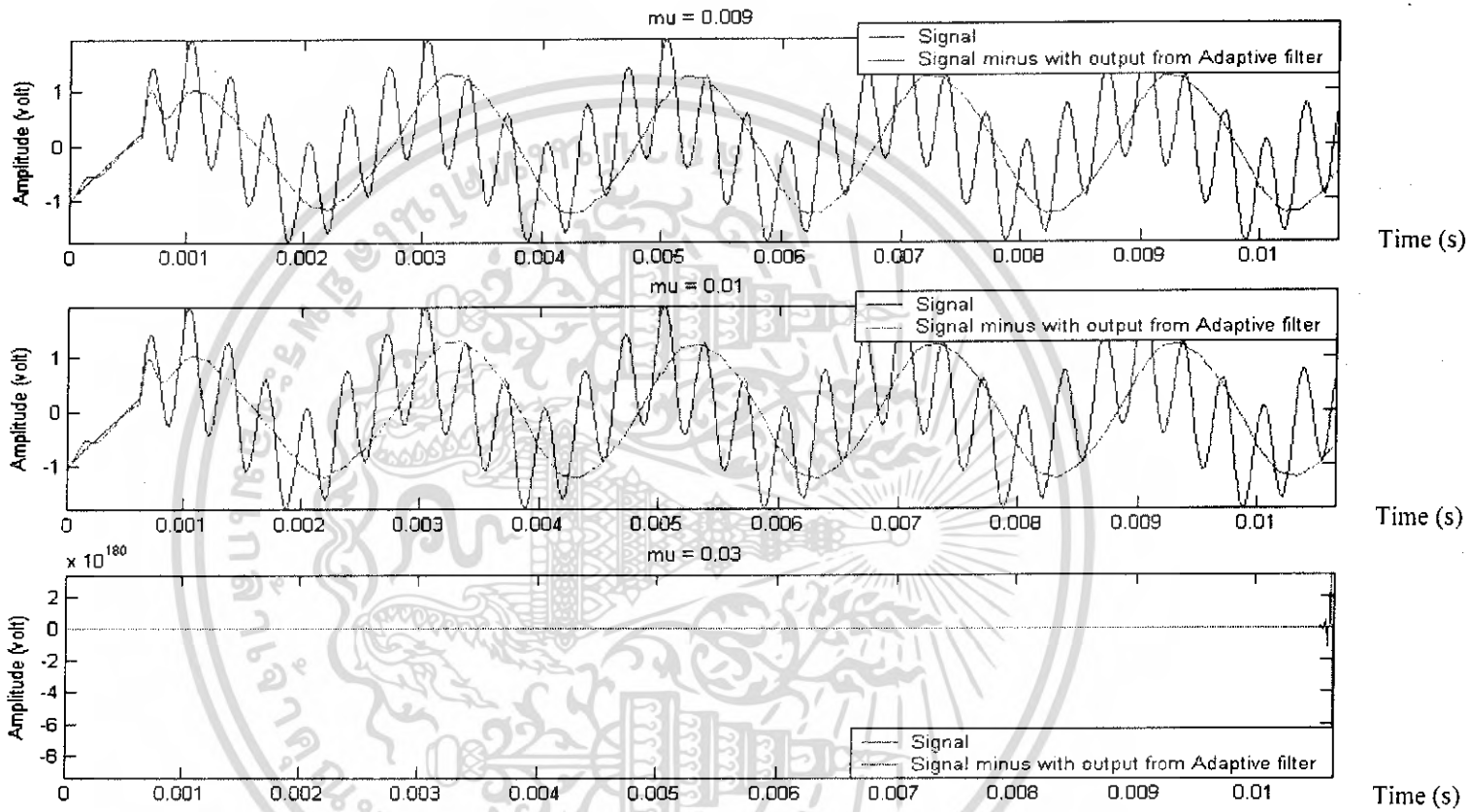
รูปที่ 4.2 แสดงผลการทดลองของ Adaptive filter แบบ LMS Algorithm

(รูปบน) ค่า $\mu = 0.0007$ (รูปกลาง) ค่า $\mu = 0.0009$ (รูปล่าง) ค่า $\mu = 0.001$



รูปที่ 4.3 แสดงผลการทดลองของ Adaptive filter แบบ LMS Algorithm

(รูปบน) ค่า $\mu = 0.003$ (รูปกลาง) ค่า $\mu = 0.005$ (รูปล่าง) ค่า $\mu = 0.007$



รูปที่ 4.4 แสดงผลการทดลองของ Adaptive filter แบบ LMS Algorithm

(รูปบน) ค่า $\mu = 0.009$ (รูปกลาง) ค่า $\mu = 0.01$ (รูปล่าง) ค่า $\mu = 0.03$

จากรูปที่ 4.2 ถึง 4.4 เป็นการแสดงผลการจำลองการทำงาน โดยทำการเปรียบเทียบผลที่ได้จากวงจร adaptive filter กับสัญญาณเสียงสะท้อน พบว่า สัญญาณที่ได้จาก adaptive filter สามารถที่จะกำจัดสัญญาณเสียงสะท้อนที่เกิดขึ้น หรือสามารถกำจัดสัญญาณเสียงผู้พูดด้านไกลที่มารวมกับสัญญาณเสียงผู้พูดด้านใกล้ได้ โดยที่แต่ละรูปจะแสดงถึงผลของค่า μ ที่มีผลต่อ adaptive filter พบว่า ค่า μ ที่นำมาใช้ได้จะมีค่าอยู่ในช่วงประมาณ 0 ถึง 0.1 โดยที่ค่า μ มีค่าน้อยๆ จะทำให้การประมาณค่าใกล้เคียงกับค่าที่ต้องการ แต่ความรวดเร็วในการติดตามสัญญาณก็จะทำได้ช้าเช่นกัน ดังรูปที่ 4.2 ในทางกลับกันที่ค่า μ มีค่ามากจะทำให้ได้การติดตามสัญญาณที่ต้องการทำได้ดี ดังรูปที่ 4.4 และที่ค่า $\mu > 0.1$ พบว่าจะเกิดความผิดพลาดในการติดตามสัญญาณที่ต้องการ

4.4 ผลการทดลองเมื่อใช้ Quantized – error Algorithm

ในหัวข้อนี้จะนำเสนอผลการทดลองที่ได้จากวิธี Quantized – error Algorithm โดยสามารถแบ่งได้เป็น 2 แบบ คือ Sign – error Algorithm และ Sign – data Algorithm โดยในการจำลองผลการทำงานจะทำการศึกษาว่าเมื่อทำการเปลี่ยนค่า μ ตามสมการที่ (4.2) และ (4.4) จะมีผลอย่างไรต่อเอาท์พุตที่ได้จาก Adaptive filter และจำลองผลการทดลองโดยใช้โปรแกรม MATLAB

$$w(k+1) = w(k) + 2\mu \operatorname{sgn}(e(k))x(k) \qquad (4.2)$$

$$\operatorname{sgn}(e(k)) = \begin{cases} 1 & , e(k) > 0 \\ 0 & , e(k) = 0 \\ -1 & , e(k) < 0 \end{cases} \qquad (4.3)$$

$$w(k+1) = w(k) + 2\mu e(k) \operatorname{sgn}(x(k)) \qquad (4.4)$$

$$\operatorname{sgn}(x(k)) = \begin{cases} 1 & , x(k) > 0 \\ 0 & , x(k) = 0 \\ -1 & , x(k) < 0 \end{cases} \qquad (4.5)$$

โดยสมการที่ (4.2) และ (4.3) ใช้สำหรับการหาค่าโดยใช้ sign – error algorithm และสมการที่ (4.4) และ (4.5) ใช้สำหรับการหาค่าโดยใช้ sign – data algorithm

โดยในการทำการทดลองจะต้องทำการกำหนดค่าเริ่มต้นในการที่จะเขียนโปรแกรมเพื่อจำลองการทำงานก่อน โดยสามารถที่จะกำหนดได้ดังตารางที่ 4.2 และ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

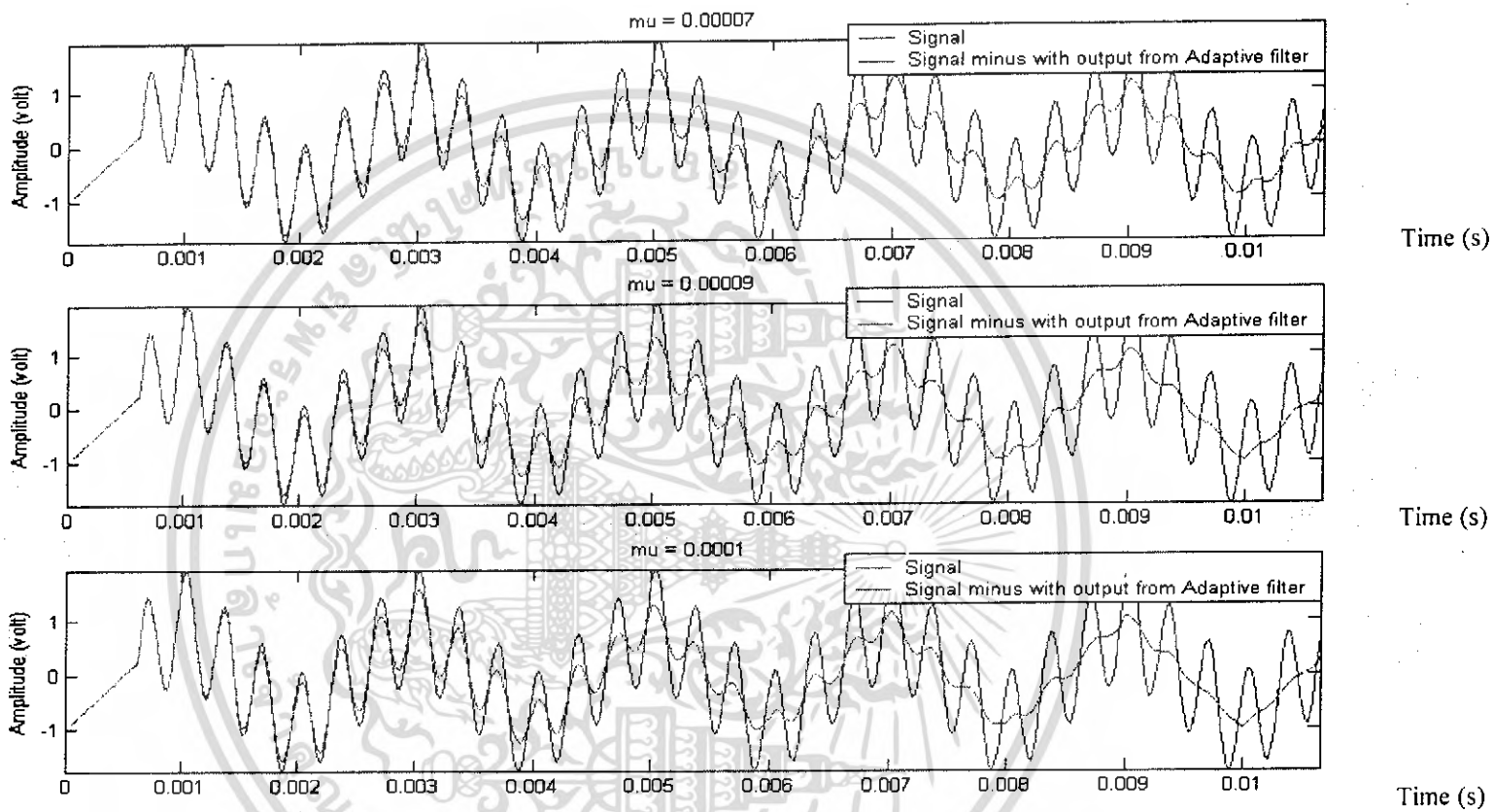
ตารางที่ 4.2 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ Sign – Error Algorithm

ค่าเริ่มต้นของ Sign – Error Algorithm
ค่าเริ่มต้น $x(0) = w(0) = [0\ 0\ \ldots\ 0]^T$ เมื่อค่า $k \geq 0$ $e(k) = d(k) - x^T(k)w(k)$ $\rho = \text{sgn}[e(k)]$ $w(k+1) = w(k) + 2\mu\rho x(k)$

ตารางที่ 4.3 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ Sign – Data Algorithm

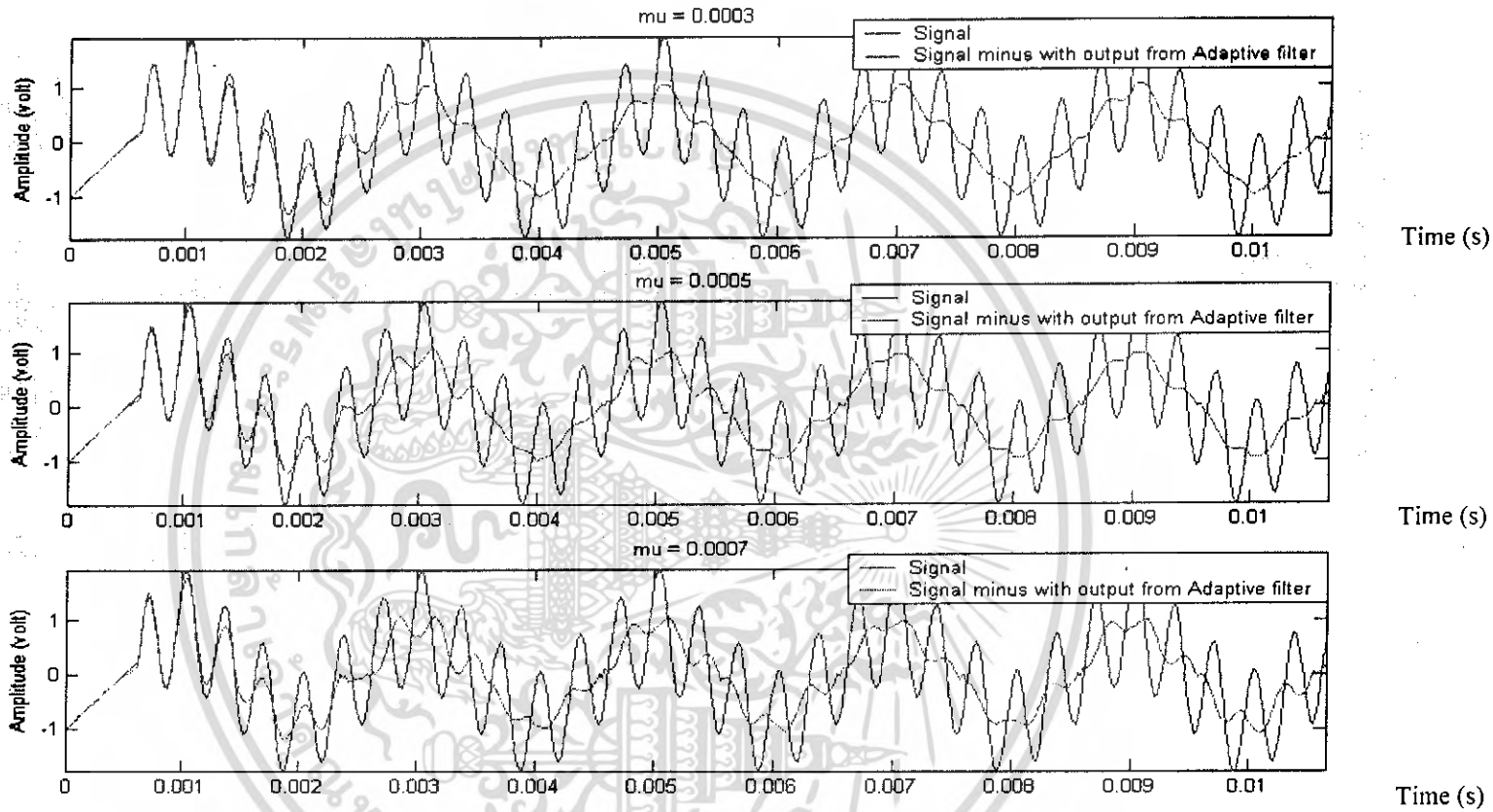
ค่าเริ่มต้นของ Sign – Data Algorithm
ค่าเริ่มต้น $x(0) = w(0) = [0\ 0\ \ldots\ 0]^T$ เมื่อค่า $k \geq 0$ $e(k) = d(k) - x^T(k)w(k)$ $\rho = \text{sgn}[x(k)]$ $w(k+1) = w(k) + 2\mu\rho e(k)$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



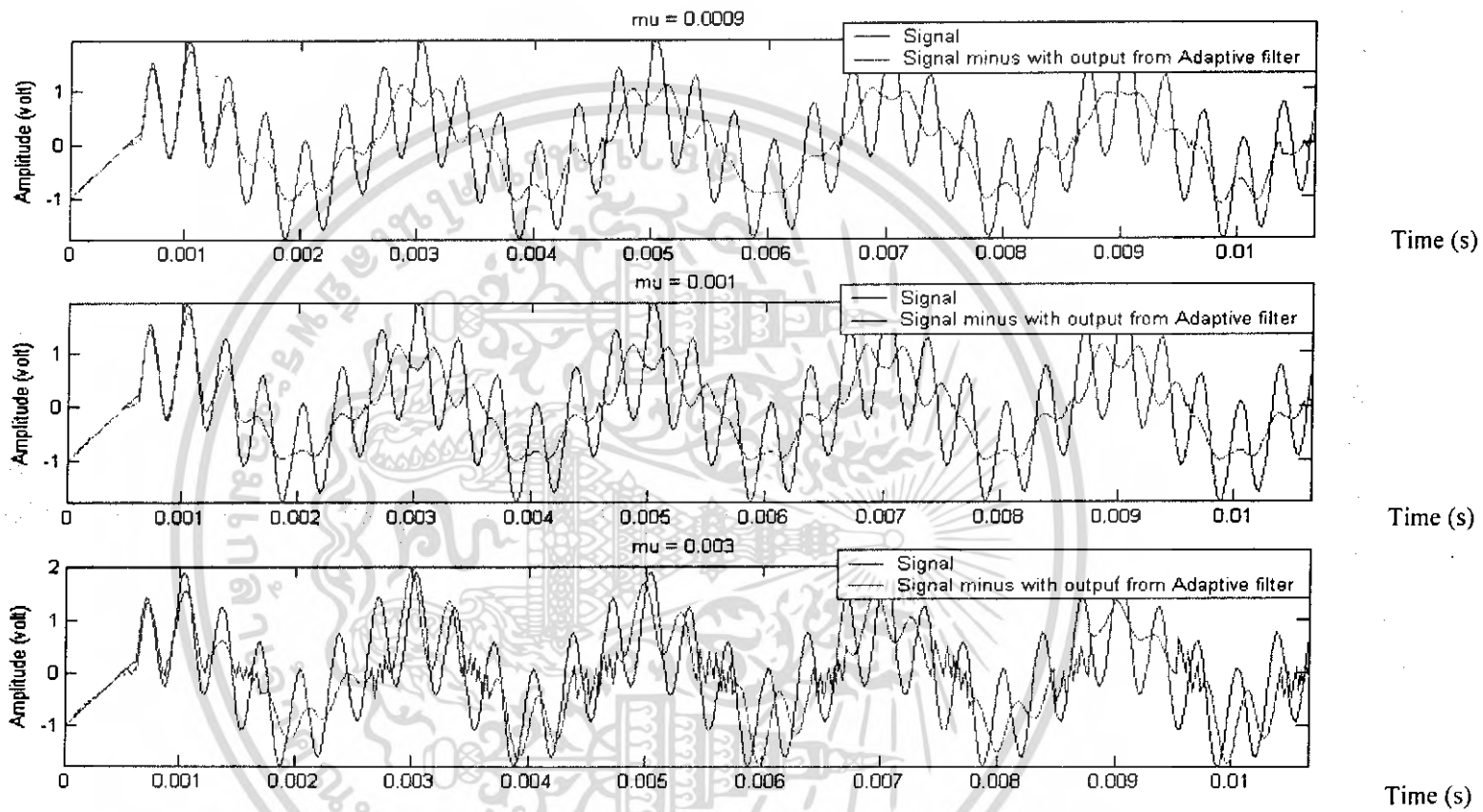
รูปที่ 4.5 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Error Algorithm

(รูปบน) ค่า $\mu = 0.00007$ (รูปกลาง) ค่า $\mu = 0.00009$ (รูปล่าง) ค่า $\mu = 0.0001$



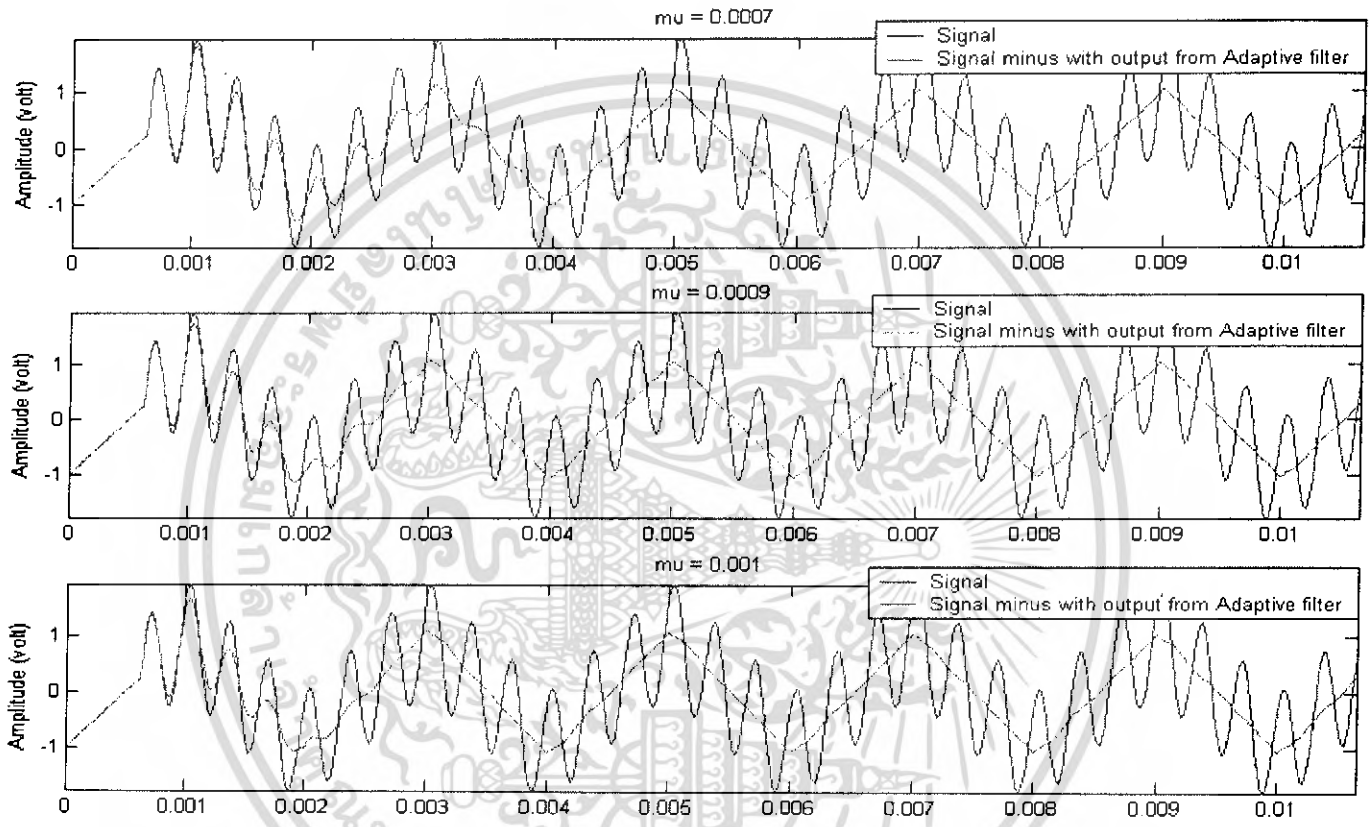
รูปที่ 4.6 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Error Algorithm

(รูปบน) ค่า $\mu = 0.00003$ (รูปกลาง) ค่า $\mu = 0.00005$ (รูปล่าง) ค่า $\mu = 0.0007$



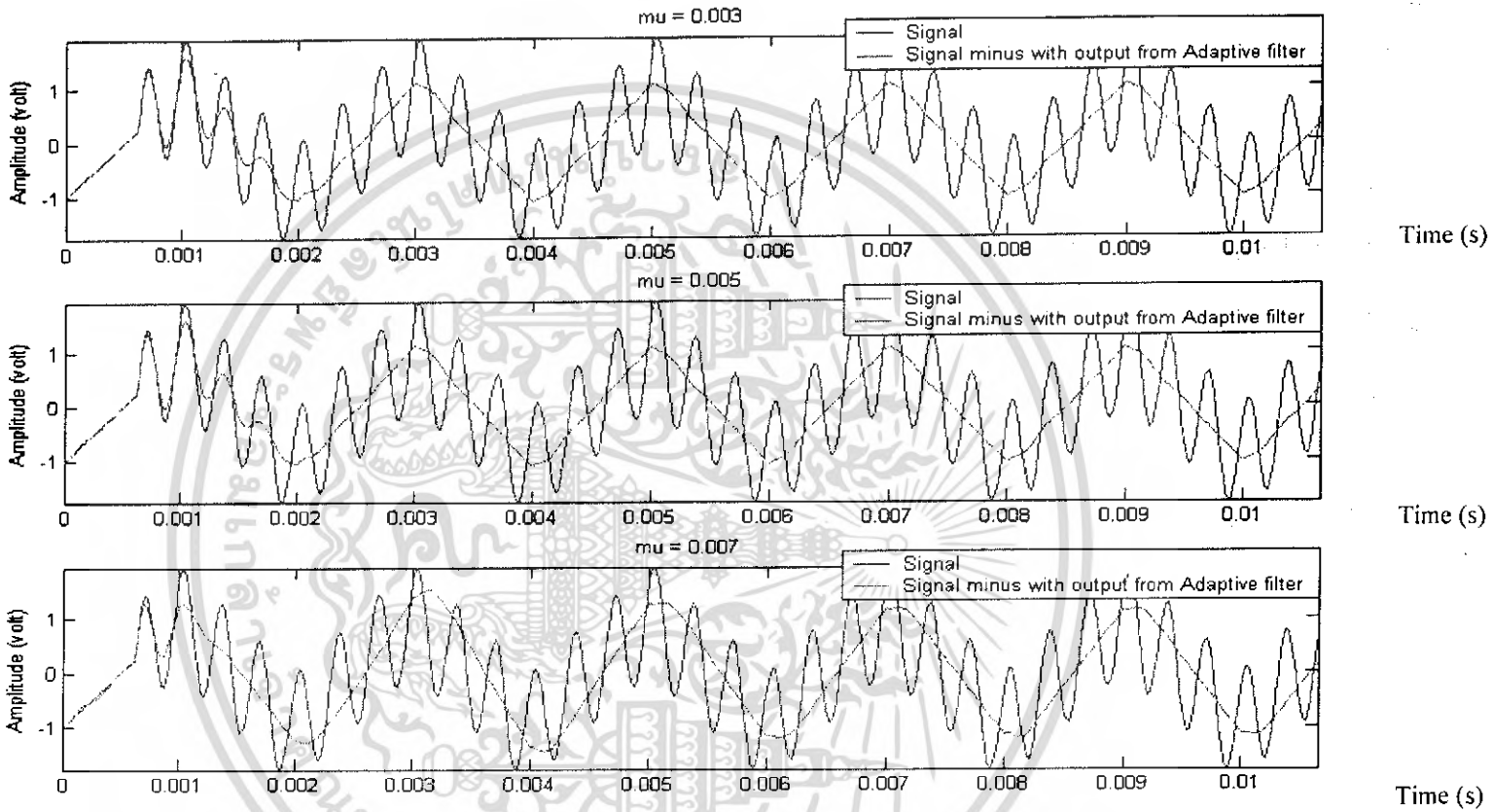
รูปที่ 4.7 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Error Algorithm

(รูปบน) ค่า $\mu = 0.0009$ (รูปกลาง) ค่า $\mu = 0.001$ (รูปล่าง) ค่า $\mu = 0.003$



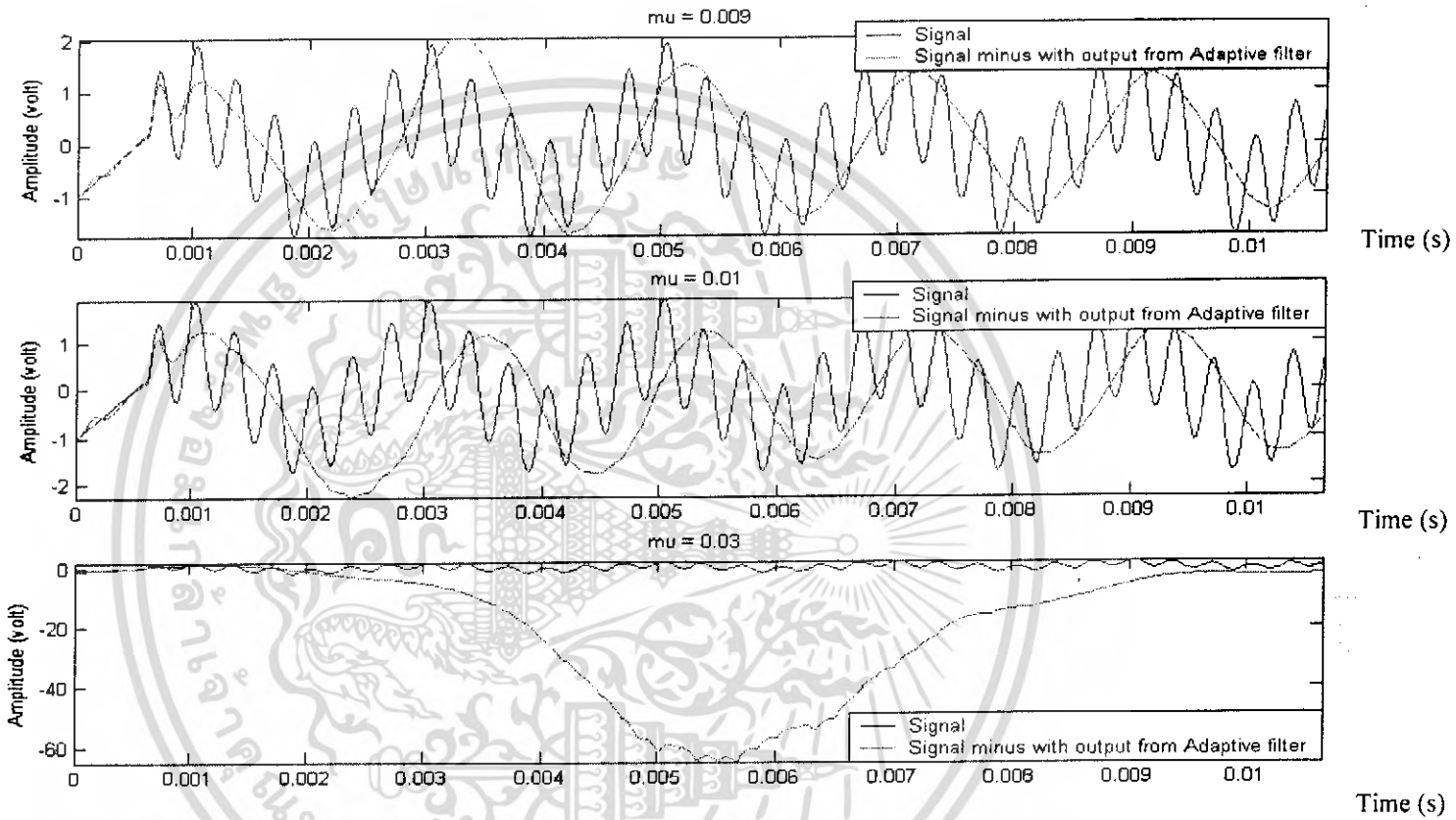
รูปที่ 4.8 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Data Algorithm

(รูปบน) ค่า $\mu = 0.0007$ (รูปกลาง) ค่า $\mu = 0.0009$ (รูปล่าง) ค่า $\mu = 0.001$



รูปที่ 4.9 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Data Algorithm

(รูปบน) ค่า $\mu = 0.003$ (รูปกลาง) ค่า $\mu = 0.005$ (รูปล่าง) ค่า $\mu = 0.007$



รูปที่ 4.10 แสดงผลการทดลองของ Adaptive filter แบบ Sign – Data Algorithm

(รูปบน) ค่า $\mu = 0.009$ (รูปกลาง) ค่า $\mu = 0.01$ (รูปล่าง) ค่า $\mu = 0.03$

จากรูปที่ 4.5 ถึง 4.7 เป็นการแสดงผลการจำลองการทำงานของ adaptive filter ชนิด sign – error algorithm และรูปที่ 4.8 ถึง 4.10 เป็นการแสดงผลการจำลองการทำงานของ adaptive filter ชนิด sign – data algorithm โดยทำการเปรียบเทียบผลที่ได้จากวงจร adaptive filter กับสัญญาณเสียงสะท้อน โดยที่แต่ละรูปจะแสดงถึงผลของค่า μ ที่มีผลต่อ adaptive filter พบว่า ค่า μ ที่นำมาใช้ได้จะมีค่าอยู่ในช่วงประมาณ 0 ถึง 0.001 สำหรับ sign – error algorithm และ ค่า μ ที่อยู่ในช่วงประมาณ 0 ถึง 0.01 สำหรับ sign – data algorithm โดยที่ค่า μ มีค่าน้อยๆ จะทำให้การประมาณค่าใกล้เคียงกับค่าที่ต้องการ แต่ความรวดเร็วในการติดตามสัญญาณก็จะทำได้ช้าเช่นกัน ในทางกลับกันที่ค่า μ มีค่ามากจะทำให้ได้การติดตามสัญญาณที่ต้องการทำได้ดีและที่ค่า $\mu > 0.01$ พบว่าจะเกิดความผิดพลาดในการติดตามสัญญาณที่ต้องการสำหรับ sign – error algorithm เช่นเดียวกันที่ค่า $\mu > 0.1$ สำหรับ sign – data algorithm

4.5 ผลการทดลองเมื่อใช้ Normalized LMS Algorithm

ในหัวข้อนี้จะนำเสนอผลการทดลองที่ได้จากวิธี Normalized LMS Algorithm โดยในการจำลองผลการทำงานจะทำการศึกษาว่าเมื่อทำการเปลี่ยนค่า μ ตามสมการที่ (4.6) จะมีผลอย่างไรต่อเอาท์พุตที่ได้จาก Adaptive filter และจำลองผลการทดลองโดยใช้โปรแกรม MATLAB

$$w(k+1) = w(k) + \frac{\mu_n}{\gamma + x^T(k)x(k)} e(k)x(k) \qquad (4.6)$$

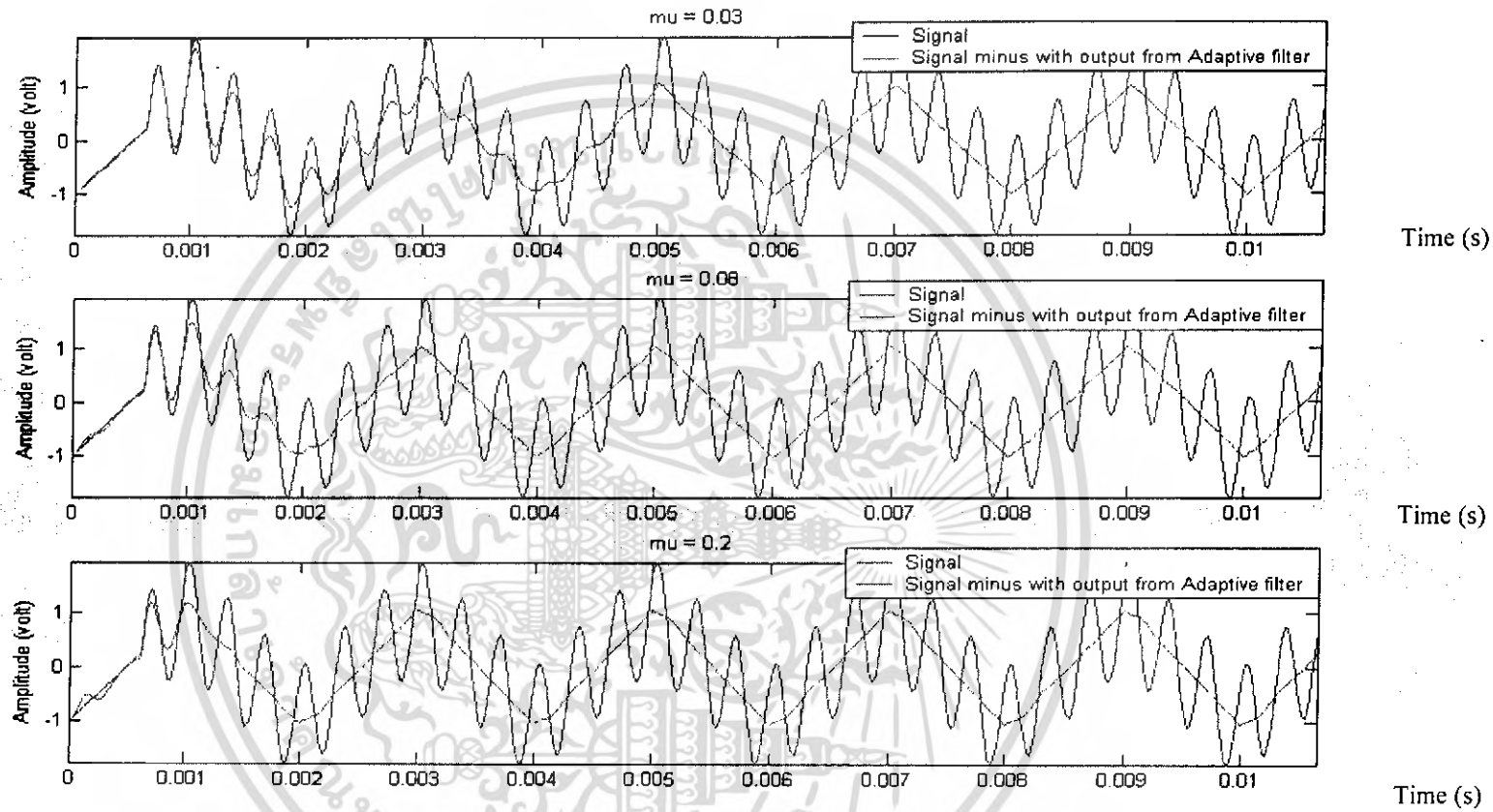
โดยในการทำการทดลองจะต้องทำการกำหนดค่าเริ่มต้นในการที่จะเขียนโปรแกรมเพื่อจำลองการทำงานก่อน โดยสามารถที่จะกำหนดได้ดังตารางที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

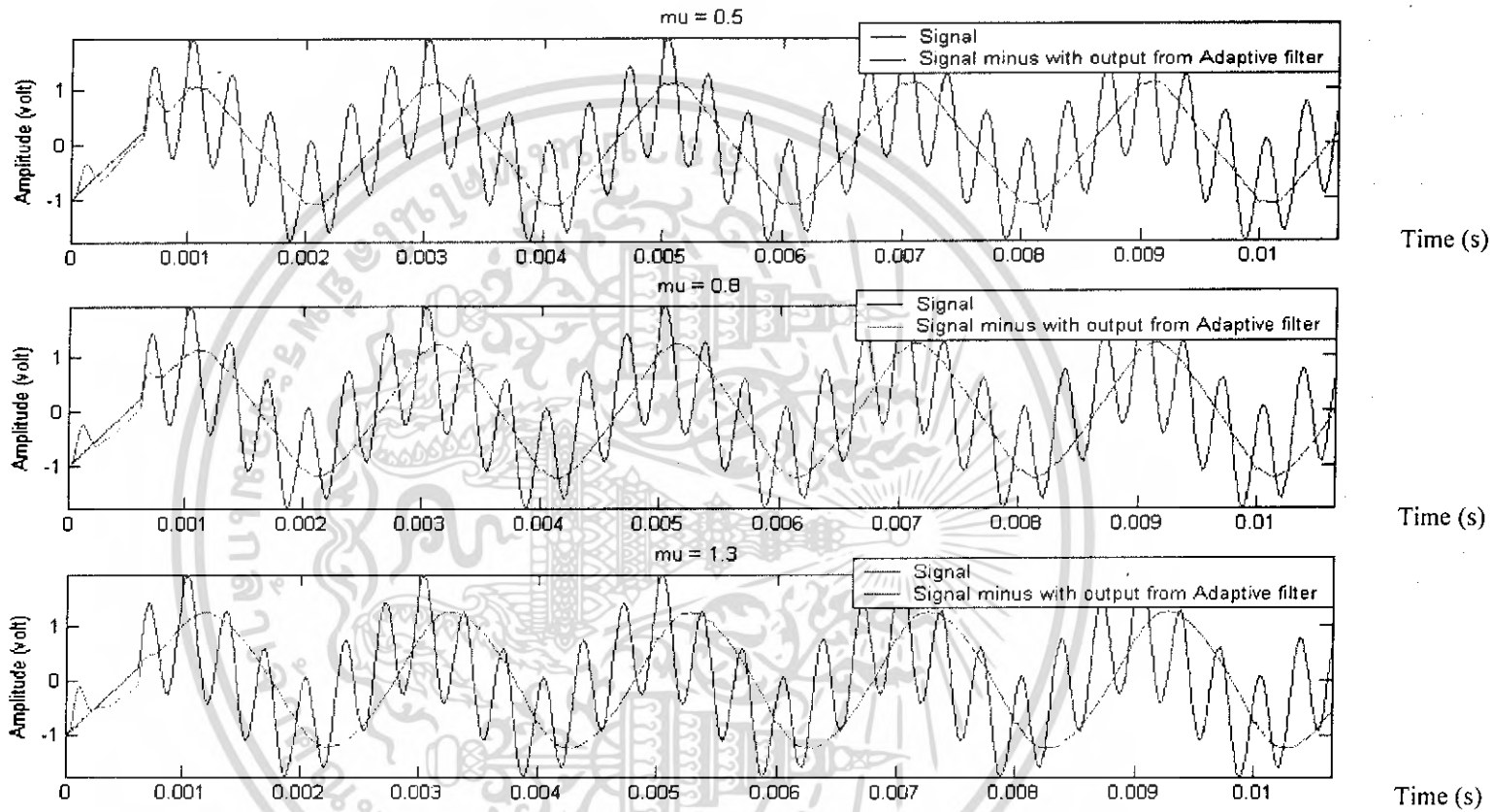
ตารางที่ 4.4 แสดงค่าเริ่มต้นที่ใช้ในการเขียนโปรแกรมของ Adaptive filter แบบ Normalized LMS Algorithm

ค่าเริ่มต้นของ Normalized LMS Algorithm
ค่าเริ่มต้น $x(0) = w(0) = [0\ 0\ ...\ 0]^T$ เลือกค่า μ_n โดย $0 < \mu_n < 2$ γ คือ ค่าคงที่ที่มีค่าน้อยๆ เมื่อค่า $k \geq 0$ $e(k) = d(k) - x^T(k)w(k)$ $w(k+1) = w(k) + \frac{\mu_n}{\gamma + x^T(k)x(k)} e(k)x(k)$

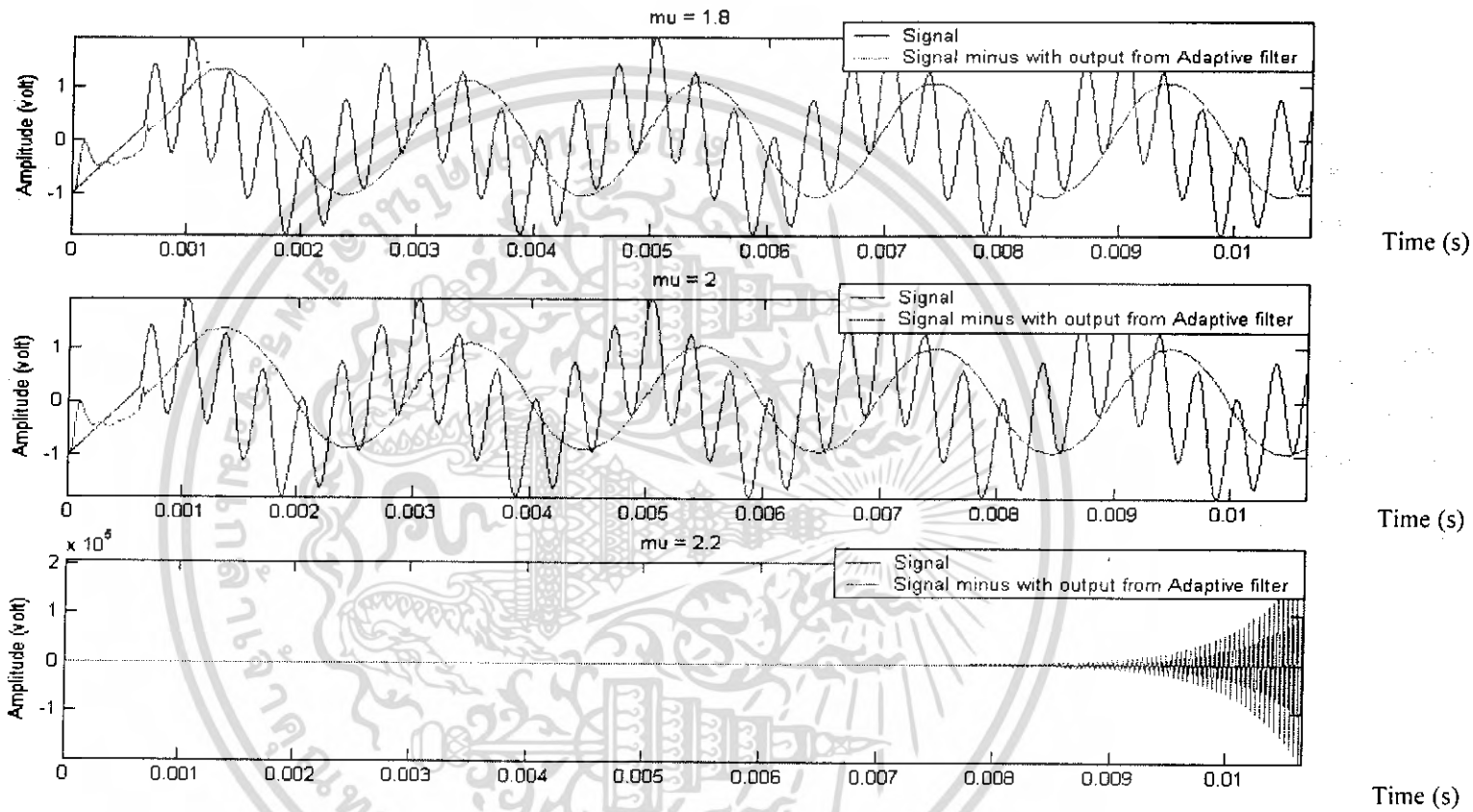
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงผลการทดลองของ Adaptive filter แบบ Normalized LMS Algorithm เมื่อ $\gamma = 5$ (รูปบน) ค่า $\mu = 0.03$ (รูปกลาง) ค่า $\mu = 0.08$ (รูปล่าง) ค่า $\mu = 0.2$



รูปที่ 4.12 แสดงผลการทดลองของ Adaptive filter แบบ Normalized LMS Algorithm เมื่อ $\gamma = 5$ (รูปบน) ค่า $\mu = 0.5$ (รูปกลาง) ค่า $\mu = 0.8$ (รูปล่าง) ค่า $\mu = 1.3$

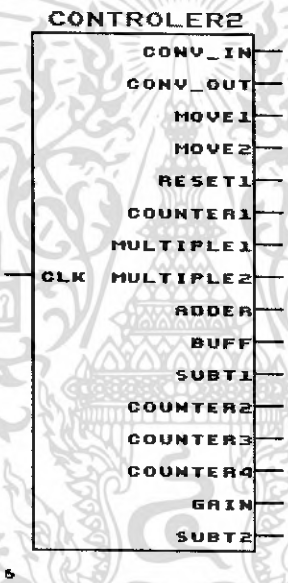


รูปที่ 4.13 แสดงผลการทดลองของ Adaptive filter แบบ Normalized LMS Algorithm เมื่อ $\gamma = 5$ (รูปบน) ค่า $\mu = 1.8$ (รูปกลาง) ค่า $\mu = 2$ (รูปล่าง) ค่า $\mu = 2.2$

จากรูปที่ 4.11 ถึง 4.13 เป็นการแสดงผลการจำลองการทำงานของ adaptive filter ชนิด Normalized LMS algorithm โดยทำการเปรียบเทียบผลที่ได้จากวงจร adaptive filter กับสัญญาณเสียงสะท้อน โดยที่แต่ละรูปจะแสดงถึงผลของค่า μ ที่มีผลต่อ adaptive filter พบว่า ค่า μ ที่นำมาใช้ได้จะมีค่าอยู่ในช่วงประมาณ 0 ถึง 2 โดยที่ค่า μ มีค่าน้อยๆ จะทำให้การประมาณค่าใกล้เคียงกับค่าที่ต้องการ แต่ความรวดเร็วในการติดตามสัญญาณก็จะทำได้ช้าเช่นกัน ในทางกลับกันที่ค่า μ มีค่ามากจะทำให้ได้การติดตามสัญญาณที่ต้องการทำได้ดี และที่ค่า μ > / พบว่าจะเกิดความผิดพลาดในการติดตามสัญญาณที่ต้องการ

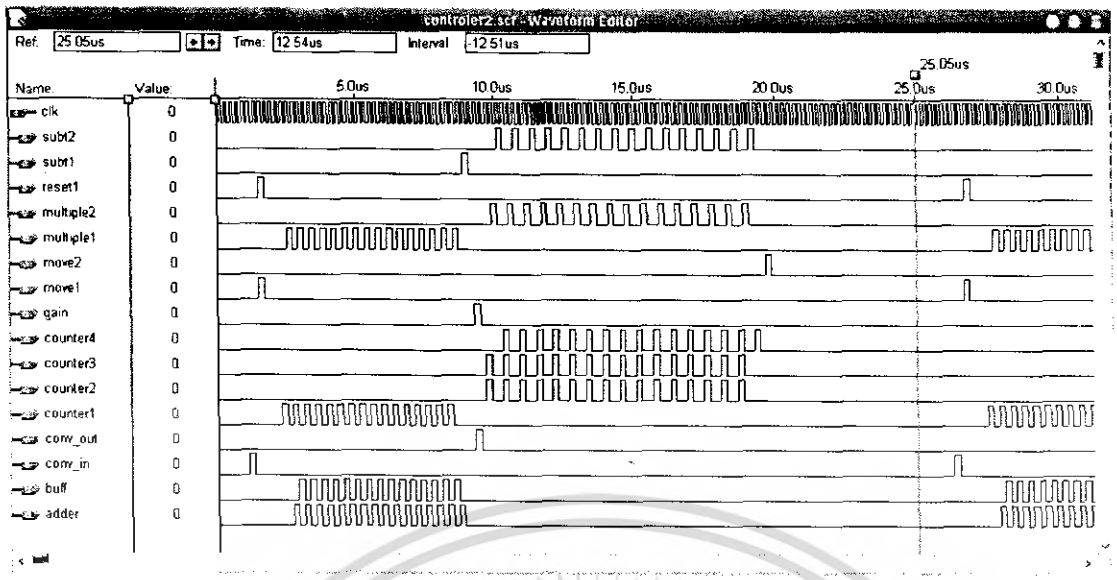
4.6 ผลการทดลองของวงจรที่ใช้ในการควบคุม

วงจรนี้เป็นส่วนที่มีความสำคัญมากโดยจะเป็นส่วนที่ทำหน้าที่สร้างสัญญาณควบคุมต่างๆ เพื่อควบคุมการทำงานของวงจรรวมทั้งหมด



รูปที่ 4.14 แสดงบล็อกไดอะแกรมของวงจรในส่วนควบคุม (controller)

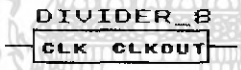
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดง timing diagram การทำงานของวงจรในส่วนควบคุม (controller)

4.7 ผลการทดลองของวงจรหารความถี่

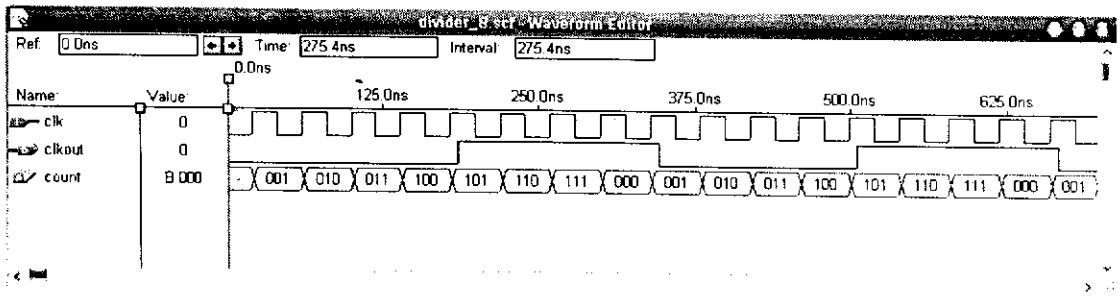
เนื่องจากความถี่ของคริสตัล (crystal) ในบอร์ด FPGA มีความถี่สูงมาก จึงจำเป็นที่จะต้องหารความถี่ลงเพื่อใช้ในบอร์ด FPGA เพื่อการประมวลผล และอีกส่วนหนึ่งเพื่อเป็นความถี่ที่ใช้ในการสุ่มค่าสัญญาณ



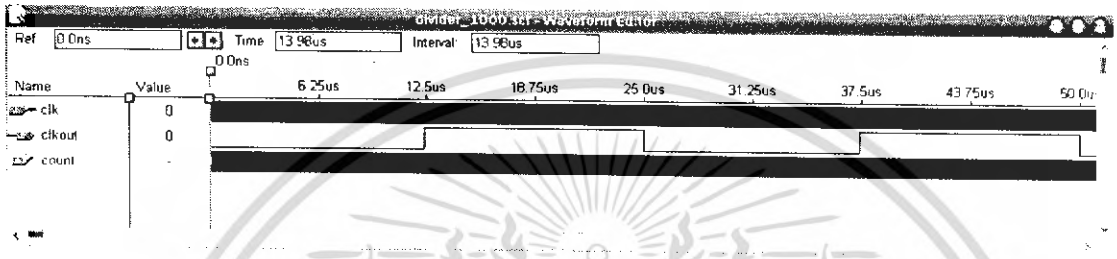
รูปที่ 4.16 แสดงบล็อกไดอะแกรมของวงจรหาร 8

รูปที่ 4.17 แสดงบล็อกไดอะแกรมของวงจรหาร 1000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



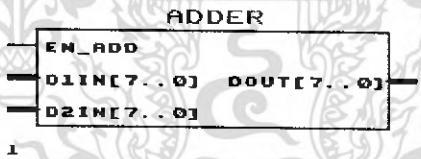
รูปที่ 4.18 แสดง timing diagram การทำงานของวงจรหาร 8



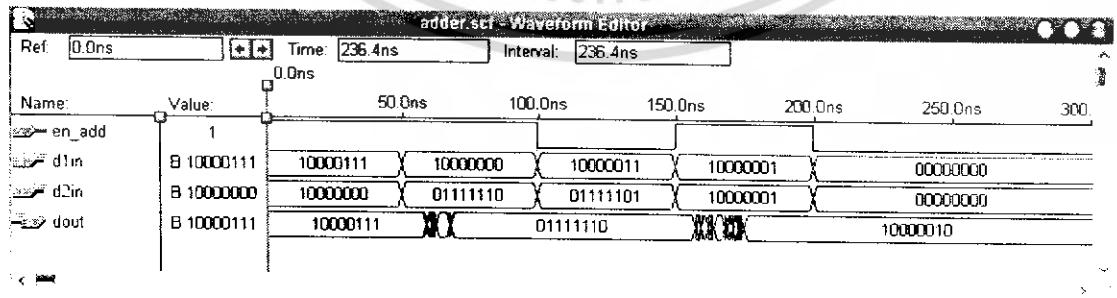
รูปที่ 4.19 แสดง timing diagram การทำงานของวงจรหาร 1000

4.8 ผลการทดลองของวงจรบวก

เป็นวงจรที่ใช้ในการบวกข้อมูลที่เข้ามา โดยในการบวกจะต้องรอสัญญาณที่จะสั่งให้ทำงาน ข้อมูลที่จะผ่านเข้าวงจรจะเป็นข้อมูลขนาด 8 บิต และให้ผลลัพธ์เป็นข้อมูลขนาด 8 บิต



รูปที่ 4.20 แสดงบล็อกไดอะแกรมของวงจรบวก (adder)

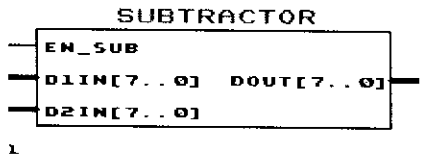


รูปที่ 4.21 แสดง timing diagram การทำงานของวงจรบวก (adder)

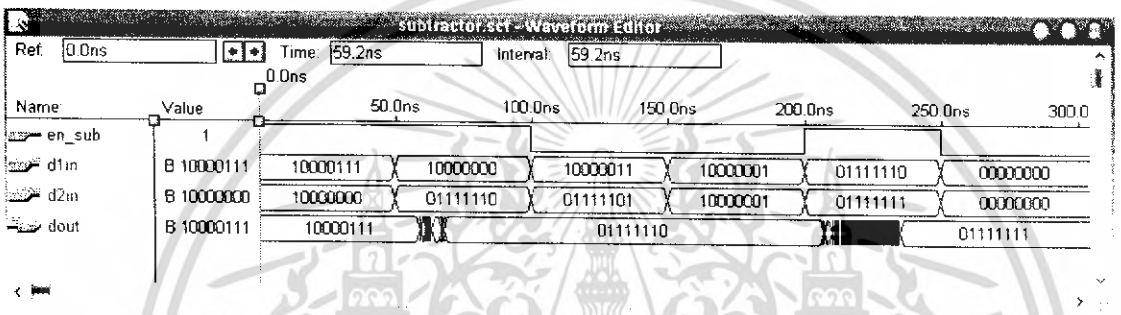
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.9 ผลการทดลองของวงจรลบ

เป็นวงจรที่ใช้ในการลบข้อมูลที่เข้ามา โดยในการลบจะต้องรอสัญญาณที่จะสั่งให้ทำงาน และข้อมูลที่จะผ่านเข้าวงจรจะเป็นข้อมูลขนาด 8 บิต และให้ผลลัพธ์เป็นข้อมูลขนาด 8 บิต



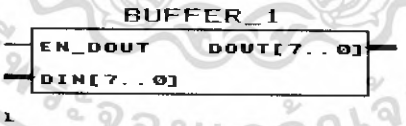
รูปที่ 4.22 แสดงบล็อกไดอะแกรมของวงจรลบ (subtracter)



รูปที่ 4.23 แสดง timing diagram การทำงานของวงจรลบ (subtracter)

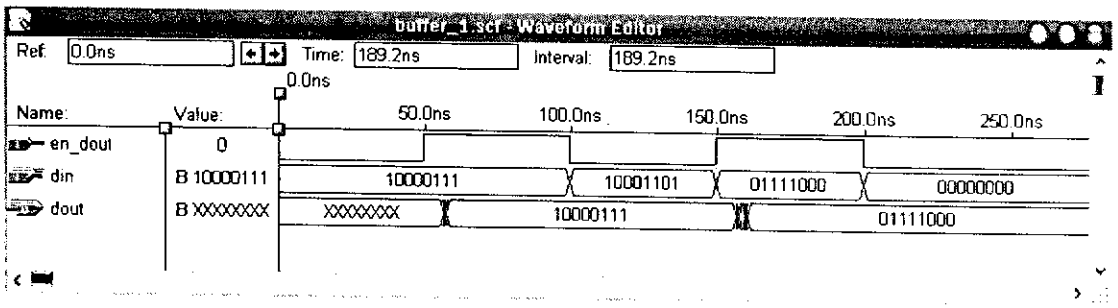
4.10 ผลการทดลองของวงจรบัฟเฟอร์

วงจรนี้จะทำหน้าที่ในการเก็บค่าข้อมูลไว้ จนกว่าจะมีสัญญาณที่ส่งมาให้ปล่อยข้อมูล จึงจะทำการปล่อยค่าข้อมูลนั้นออกไป โดยข้อมูลที่เข้ามาและออกจากวงจรนี้จะมีขนาด 8 บิต



รูปที่ 4.24 แสดงบล็อกไดอะแกรมของวงจรบัฟเฟอร์ (buffer)

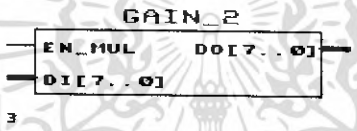
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



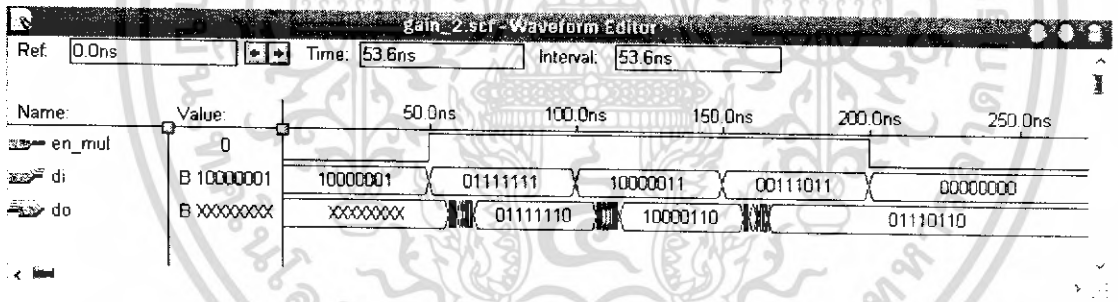
รูปที่ 4.25 แสดง timing diagram การทำงานของวงจรบัฟเฟอร์ (buffer)

4.11 ผลการทดลองของวงจรขยาย

วงจรส่วนนี้จะทำการคูณค่าข้อมูลที่เข้ามาด้วยค่าคงที่ที่ได้ตั้งไว้ โดยในที่นี้คือ 2 โดยข้อมูลที่เข้าและออกจากวงจรจะมีขนาด 8 บิต



รูปที่ 4.26 แสดงบล็อกไดอะแกรมของวงจรขยาย (gain)

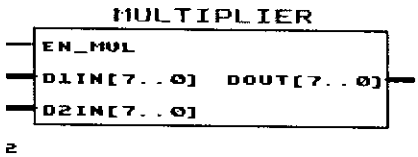


รูปที่ 4.27 แสดง timing diagram การทำงานของวงจรขยาย (gain)

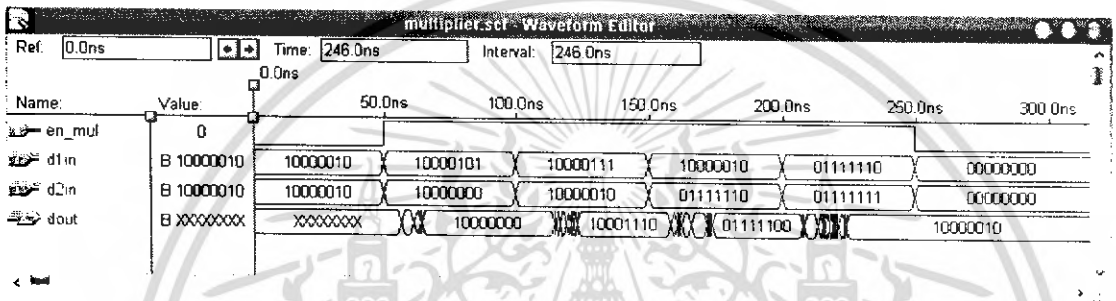
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.12 ผลการทดลองของวงจรคูณ

วงจรนี้จะทำหน้าที่คูณข้อมูลที่เข้ามา โดยในการคูณจะต้องรอสัญญาณคูณก่อนจึงจะสามารถที่จะทำการคูณได้ โดยข้อมูลที่เข้ามาและออกจากวงจรนี้จะมีขนาด 8 บิต



รูปที่ 4.28 แสดงบล็อกไดอะแกรมของวงจรคูณ (multiplier)

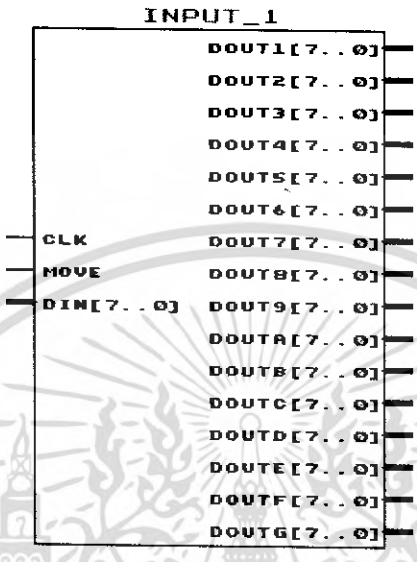


รูปที่ 4.29 แสดง timing diagram การทำงานของวงจรคูณ (multiplier)

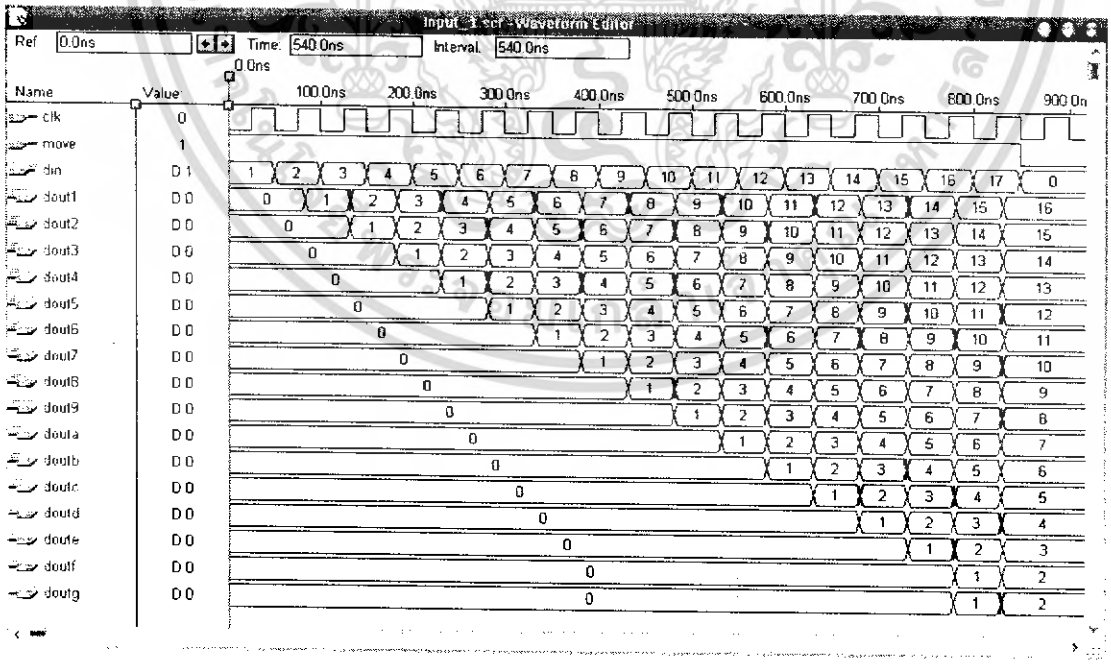
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.13 ผลการทดลองของวงจรเก็บค่าสัญญาณ

วงจรนี้จะทำการเก็บค่าข้อมูลที่เข้ามาไว้ โดยจะสามารถที่จะเก็บค่าข้อมูลไว้ได้ 8 ค่าข้อมูล โดยจะส่งค่าข้อมูลเหล่านั้นออกก็ต่อเมื่อได้รับสัญญาณการปล่อยค่า และข้อมูลที่เข้าและออกจากวงจรจะมีขนาด 8 บิต



รูปที่ 4.30 แสดงบล็อกไดอะแกรมของวงจรเก็บค่าสัญญาณ

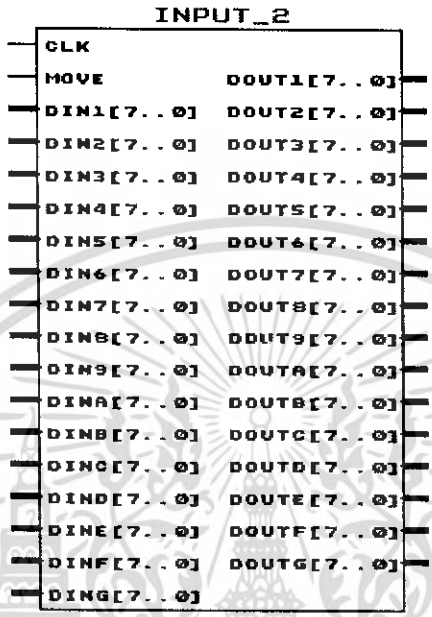


รูปที่ 4.31 แสดง timing diagram การทำงานของวงจรเก็บค่าสัญญาณ

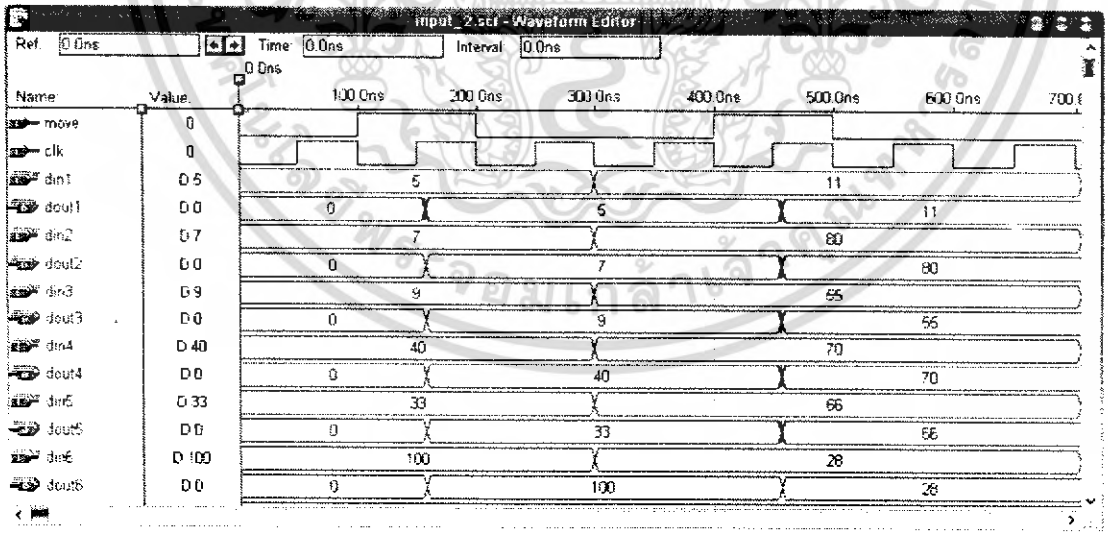
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.14 ผลการทดลองของวงจรเก็บค่าสัมประสิทธิ์

วงจรนี้จะทำหน้าที่ในการเก็บค่าสัมประสิทธิ์ฟิลเตอร์ โดยจะสามารถที่จะเก็บค่าข้อมูลไว้ได้ 8 ค่าข้อมูล โดยจะส่งค่าข้อมูลเหล่านั้นออกก็ต่อเมื่อได้รับสัญญาณการปล่อยค่า และข้อมูลที่เข้าและออกจากวงจรจะมีขนาด 8 บิต



รูปที่ 4.32 แสดงบล็อกไดอะแกรมของวงจรเก็บค่าสัมประสิทธิ์ของฟิลเตอร์

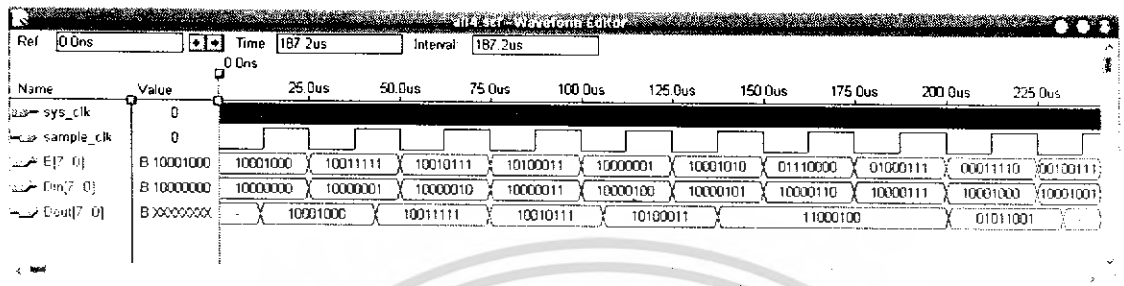


รูปที่ 4.33 แสดง timing diagram การทำงานของวงจรเก็บค่าสัมประสิทธิ์ของฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

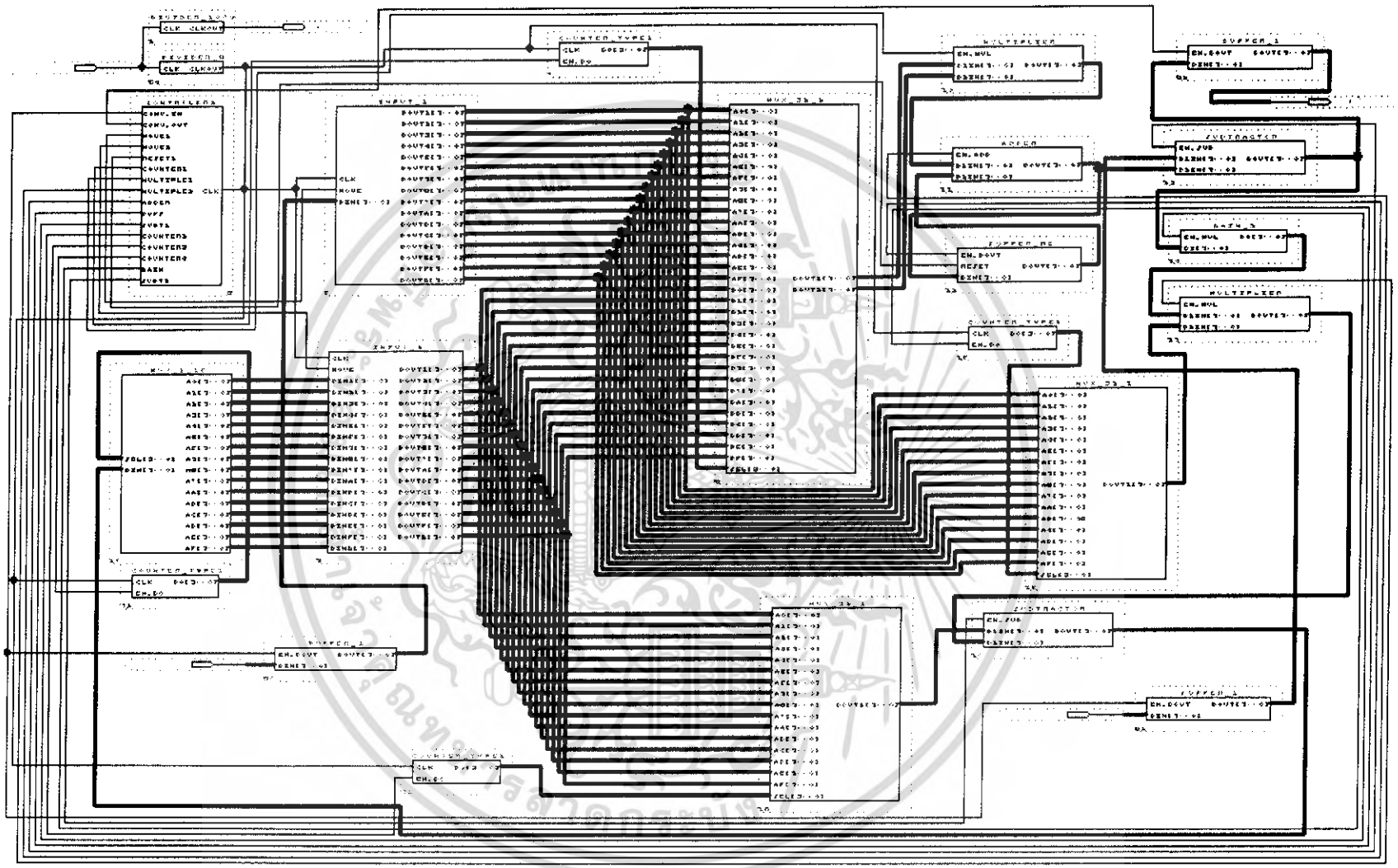
4.15 ผลการทดลองของวงจรรวม

ในส่วนนี้จะเป็นการนำเอาแต่ละส่วนมาประกอบรวมเข้าด้วยกัน โดยที่จะควบคุมการทำงานโดยสัญญาณจากตัวควบคุม ที่จะส่งมาควบคุมในแต่ละส่วนให้เป็นไปตามที่ได้ออกแบบไว้ โดยที่ค่าข้อมูลที่เข้าและออกจากวงจรรวมจะมีขนาด 8 บิต



รูปที่ 4.34 แสดง timing diagram การทำงานของวงจรรวม

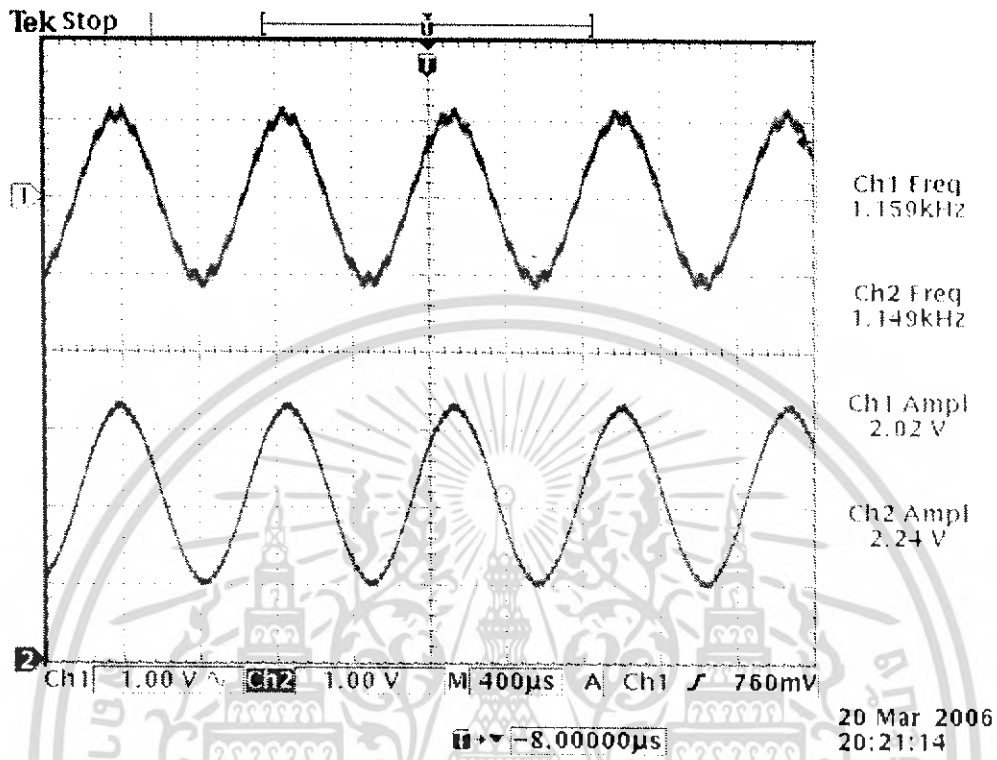
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 แสดงบล็อกไดอะแกรมของวงจรรวม

4.16 ผลการทดลองของวงจร A/D และ D/A Converter

ในส่วนนี้เป็นวงจรที่จะทำหน้าที่ในการแปลงสัญญาณ



รูปที่ 4.36 (บน) รูปสัญญาณก่อนที่จะผ่านวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล
(ล่าง) รูปสัญญาณหลังจากผ่านวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิเคราะห์ผลการทดลอง

5.1 สรุปผลการทดลองจากการซิมมูเลชั่นด้วยโปรแกรม MATLAB

จากการทดลองโดยใช้โปรแกรม MATLAB มาใช้ในการซิมมูเลชั่นสามารถที่จะแสดงให้เห็นถึงผลการทำงานในแต่ละวิธีการที่ได้นำมาเสนอ โดยสามารถที่จะแจกแจงได้ดังนี้

- จากทั้ง 4 วิธีการที่ได้นำเสนอพบว่าในแต่ละวิธีการจะมีช่วงของค่า μ ที่สามารถนำมาใช้ได้ โดยถ้าค่า μ ที่ใช้น้อยกว่าช่วงของค่า μ ที่ใช้ได้จะทำให้การติดตามสัญญาณได้ไม่เร็ว แต่ถ้าใช้ค่า μ ที่ใช้มากกว่าช่วงของค่า μ ที่ใช้ได้ จะทำให้เกิดค่าความผิดพลาดขึ้นมามาก ดังนั้นในการเลือกค่า μ จึงต้องทำการพิจารณาให้ดี
  - วิธี LMS algorithm พบว่าช่วงของค่า μ ที่ใช้ได้จะมีค่าอยู่ระหว่าง 0 - 0.1
  - วิธี sign – error algorithm พบว่าช่วงของค่า μ ที่ใช้ได้จะมีค่าอยู่ระหว่าง 0 - 0.001
  - วิธี sign – data algorithm พบว่าช่วงของค่า μ ที่ใช้ได้จะมีค่าอยู่ระหว่าง 0 - 0.01
  - วิธี NLMS algorithm พบว่าช่วงของค่า μ ที่ใช้ได้จะมีค่าอยู่ระหว่าง 0 – 2
- จากแต่ละวิธีที่นำเสนอมีทั้งข้อดีและข้อเสียที่แตกต่างกัน โดยวิธีแรกจะเป็นวิธีเริ่มต้นก็คือวิธี LMS algorithm ส่วนวิธี sign – error algorithm และวิธี sign – data algorithm เป็นวิธีที่จะมาช่วยลดความยุ่งยากในการคำนวณแต่ก็ทำให้เกิดความผิดพลาดตามมาด้วย โดยวิธี sign – data algorithm จะให้ผลที่ดีกว่าวิธี sign – error algorithm เพราะว่าเป็นการคำนวณที่ยึดเอาผลของค่าสัญญาณเป็นที่ตั้ง ส่วนวิธี NLMS algorithm จะให้ช่วงของค่า μ ที่ใช้ได้มีค่าช่วงที่กว้างกว่าวิธีอื่น แต่ก็ทำให้การคำนวณยุ่งยากตามไปด้วย

5.2 สรุปผลการทดลองที่ได้จากบอร์ด FPGA

จากที่ได้ผลซิมมูเลชั่นจากโปรแกรม MATLAB มาแล้ว หลังจากนั้นนำมาสร้างเป็นชิ้นงานจริงโดยอาศัยบอร์ด FPGA จากโปรแกรมที่เขียนขึ้นโดยใช้ภาษา VHDL แล้วทำการ burn ลงบนบอร์ด FPGA พบว่าหลังจาก burn แล้วบอร์ดไม่สามารถทำงานได้ตามที่ได้ออกแบบไว้ ซึ่งเป็นสาเหตุมาจาก

- Code ที่ทำการเขียนขึ้นมานั้นยังมีจุดบกพร่องอยู่ทำให้ไม่ได้ผลตามที่ได้ออกแบบไว้
- ในการออกแบบเป็นการออกแบบโดยใช้แนวคิดที่คิดขึ้นเองซึ่งไม่เป็นหลักสากล อาจจะเป็นเหตุผลหนึ่งซึ่งทำให้ไม่ได้ผลการทดลองตามที่ต้องการ
- ในตัวของภาษาที่ใช้เขียนมีความละเอียดอ่อนในการทำงานมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางแก้ไขปัญหาที่เกิดขึ้น

ในการออกแบบควรจะใช้หลักการคิดที่เป็นสากล และต้องคำนึงถึงความละเอียดอ่อนในการใช้บอร์ด FPGA ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] Simon Haykin , "Adaptive Filter Theory ", Prentice – Hall , Inc,1991.

[2] S. Haykin and B. widrow , "Least – Mean – Square Adaptive Filter ",John Wiley and Sons Inc,2003.

[3] Fredrik Gustafsscn," Adaptive Filter and Change Detection" ,John Wiley and Sons Inc,2000.

[4] B.Farhang – Boroujeny," Adaptive Filter Theory and Applications" ,John Wiley and Sons ltd,1998.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
การประยุกต์ใช้งานของ FPGA

1. บทนำ

อุปกรณ์ FPGA เป็นอุปกรณ์ทางด้านอิเล็กทรอนิกส์ที่มีการออกแบบให้มีโครงสร้างการทำงานแบบดิจิตอลโดยให้ผู้ใช้งานสามารถกำหนดหรือจัดคุณสมบัติการทำงาน (Configuration) ได้ตามความต้องการ แต่อย่างไรก็ตามภายใต้เงื่อนไขของแบบแผนโครงสร้าง (Paradigm) ของ FPGA แต่ละชนิดก็จะพบว่าในแต่ละแบบแผนโครงสร้างก็ให้คุณลักษณะที่โดดเด่นแตกต่างกันไปจึงอาจจะมีความเหมาะสมต่องานที่แตกต่างกันตามไปด้วย ดังนั้นการนำเอาอุปกรณ์ FPGA ไปใช้งานสิ่งที่ผู้ใช้ต้องคำนึงถึงก็คือชนิดและสมบัติของ FPGA เพื่อให้การนำไปใช้งานได้เกิดประโยชน์อย่างสูงสุด

1.1 แบบแผนโครงสร้างของ FPGA (FPGA paradigm)

เมื่อมีการเริ่มต้นสร้าง Digital electronics ก็ได้มีการพยายามค้นหาแนวทางในการออกแบบวงจรและสิ่งที่จำเป็นต่อการออกแบบที่สุดคือส่วนประกอบที่สำคัญ 3 ส่วนของวงจรที่จะต้องการมีคือ

1) มีตัวดำเนินการข้อมูล (Data operators) โดยเกทจะถูกสร้างขึ้นเพื่อให้สามารถใช้งานตามวัตถุประสงค์ของการทำงานได้อย่างสมบูรณ์ในการส่งผ่าน (State transform) สถานะทางอินพุตไปเป็นสถานะทางเอาต์พุต

2) มีส่วนของอุปกรณ์สำหรับเก็บข้อมูล (Storage element) ดังเช่น แลทช์ (Latches) ฟลิปฟลอป (Flip-Flop) ซึ่งจะทำหน้าที่เก็บผลลัพธ์แต่ละสถานะ (state operand)

3) มีการเชื่อมต่อของสาย (Wires) ระหว่างจุดมีระบบบัสเพื่อเชื่อมต่อสื่อสาร ส่งค่าระหว่างอุปกรณ์ที่เก็บข้อมูลกับส่วนของตัวดำเนินการข้อมูล

โดยทั่วไปสารึ่งตัวนำจะแสดงให้เห็นถึงความสามารถในการเป็นอุปกรณ์ขั้นพื้นฐานที่มีอยู่อย่างหลากหลายทั้งในด้านแนวความคิดและรูปแบบ ในอดีตการดำเนินการกับอุปกรณ์เหล่านี้จะต้องใช้อยู่บนระบบของแผงวงจรชั่วคราว (Breadboard) เป็นการเชื่อมต่อขาของเกทและฟลิปฟลอปโดยตรงเพื่อให้อุปกรณ์เหล่านี้ทำงานร่วมกันบนแผ่นวงจรพิมพ์ (Printed circuit board) แต่อย่างไรก็ตาม เมื่อไอซี (Integrated circuit) ได้มีวิวัฒนาการเพิ่มขึ้น ทำให้ไม่สามารถเข้าถึงการทำงานของไอซีได้โดยตรง แต่จะเป็นการทำงานในลักษณะกลุ่มวงจรลอจิก เช่น กลุ่มของ TTL แต่เมื่อได้มีการนำไอซีรูปแบบเดิมที่คุ้นเคยมาใช้งานอีกโดยนำกลุ่มฟังชั่นทางลอจิก มาทำการโปรแกรมเพื่อเชื่อมต่อสายไฟในระหว่างการผลิต ดังเช่น อุปกรณ์ประเภทที่มีฟังชั่นการทำงานเฉพาะทาง (Application specific integrated-circuit , ASIC) ก็ทำให้สามารถเข้าถึงการทำงานของไอซีได้โดยตรง แต่วิธีการนี้ยังมีต้นทุนที่สูงมาก และต้องใช้ระยะเวลาการดำเนินงานที่ยาวนานนับตั้งแต่การออกแบบไปจนถึงจบขั้นตอนกระบวนการผลิต

เมื่อมีการพัฒนาการ FPGA ขึ้นมาโดยเป็นอุปกรณ์ที่รวมเอาอุปกรณ์พื้นฐานสามชนิดเข้าไว้ด้วยกัน คืออุปกรณ์เก็บข้อมูล อุปกรณ์ลอจิก และสายสัญญาณตัวนำ ที่สามารถทำการโปรแกรมเพื่อสร้างให้เป็นวงจรที่ทำงานตามกำหนดได้ด้วย ดังนั้น FPGA จึงเป็นทางออกของการแก้ปัญหาที่เป็นที่เป็นจุดอ่อนของอุปกรณ์ทั้งแบบชิ้นส่วนมาตรฐาน (Standard parts) และอุปกรณ์แบบ ASIC ที่สำคัญคือมันยอมให้มีการ

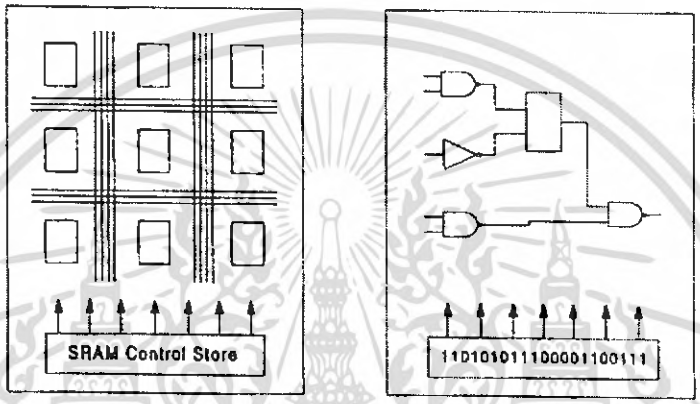
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าถึงได้ทุกส่วนของอุปกรณ์ที่อยู่ในตัวไอซี จึงก่อให้เกิดวิวัฒนาการแบบใหม่ของเทคโนโลยีพื้นฐานทางอิเล็กทรอนิกส์ส่งผลให้การออกแบบระบบสามารถมีทางเลือกได้ถึง 3 ทางคือ

1) การใช้ชิ้นส่วนพื้นฐานโดยใช้อุปกรณ์ที่มีฟังก์ชั่นตามมาตรฐานของผู้ผลิตที่มีในตลาด ซึ่งจะทำให้มีต้นทุนการผลิตต่ำ

2) การใช้ ผู้ใช้สามารถกำหนดฟังก์ชั่น การทำงานตามความต้องการของตนเองได้ แต่ก็มีต้นทุนการผลิตสูง

3) การใช้ ผู้ใช้กำหนดฟังชันการทำงานตามความต้องการของตนเอง แต่ทำให้มีต้นทุนการผลิตต่ำได้ในจำนวนการผลิตจำนวนหนึ่ง



รูปที่ ก.1 แนวความคิดทางโครงสร้างของสอง FPGA

แนวคิดที่เป็นโครงสร้างง่ายๆ ของ Reprogrammable FPGA นั้นดังแสดงดังรูปที่ 2.10 โดยจะมีตัวที่เป็น Static RAM มีหน่วยของ Function unit และสายตัวนำที่สามารถกำหนดได้โดยกำหนดใน Control store เพื่อระบุชิ้นส่วนของวงจรเกท และการเชื่อมต่อต่าง ๆ ส่วนประเภทหรือตระกูลของ FPGA จะถูกกำหนดโดยอาศัยความแตกต่างของโครงสร้างทางสถาปัตยกรรมภายในซึ่งเป็นหน่วยเล็กๆที่เรียกว่า Function unit กับข่ายโครงสร้างการเชื่อมต่อภายใน (Interchip wiring organization) เนื่องจากวงจรที่ถูกสร้างขึ้นโดย FPGA จะเป็นการกำหนดค่าภายในของ SRAM control store การนำไปประยุกต์ใช้งานของ FPGA จึงมีหลายแนวทางคือ

1) การสร้างชิ้นงานแบบทันที (Instantaneous implementation) จะมีลักษณะเหมือนกับการเชื่อมต่อกันบนหน่วยความจำโดยตรง (Memory-wiring) ทำให้ใช้เวลาสร้างสั้นมากชิ้นงานที่ออกแบบสามารถนำไปใช้งานได้ทันที ซึ่งแตกต่างจากการออกแบบโดยใช้ VLSI ที่ต้องใช้เวลาหลายสัปดาห์ในการผลิตงานตามแบบที่ได้ออกแบบไว้ จึงมีความได้เปรียบในการสร้างงานต้นแบบ

2) มีความยืดหยุ่นในการกำหนดการทำงานของโครงสร้าง (Dynamic reconfiguration) เนื่องจากสถาปัตยกรรมบางส่วนของ FPGA สามารถที่จะทำการโปรแกรมได้ใหม่ในขณะที่กำลังทำงาน (Run time) ซึ่งเป็นช่วงเวลาที่ FPGA กำลังทำงานตามคำสั่งของโปรแกรม ดังนั้นชิพจึงสามารถถูกนำไปใช้งานได้ในฟังชันอื่นที่แตกต่างจากคำสั่งในตอนแรก แต่ก็จะขึ้นอยู่กับความสามารถของวงจรที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างหลักและความพอเพียงของทรัพยากรในการปรับเปลี่ยนฟังชันการทำงานของมันเองด้วย

3) มีความปลอดภัยในการรักษาข้อมูลของแบบชิ้นงาน (Design security) เนื่องจากข้อมูลการทำคอนฟิกกูเรชันของ FPGA จะไม่ปรากฏให้เห็นถ้าไม่มีการจ่ายพลังงานให้กับชิพ เป็นการเพิ่มขอบเขตของการนำไปประยุกต์ใช้งานร่วมกับระบบรักษาความปลอดภัยของข้อมูลร่วมกับระบบอื่นที่ผู้ออกแบบได้สร้างขึ้น

4) มีความยืดหยุ่นในการพัฒนาระบบ (Field programmability) ระบบงานที่สร้างโดยใช้ FPGA สามารถที่จะทำการปรับปรุงและตรวจสอบได้ในภาคสนามโดยการดำเนินการของผู้ใช้ระบบเอง (Operator) หรือจากการควบคุมระยะไกล (Telemeter) เพื่อแก้ไขข้อบกพร่องซ่อมแซมส่วนที่เสียหายหรือเพิ่มเติมฟังชันการทำงานใหม่ๆเข้าไป

1.2 การออกแบบและสร้างชิ้นงานโดยใช้ FPGA

1.2.1 (Design and Implementation using FPGAs)

การออกแบบเพื่อใช้งาน FPGA จะมีขั้นตอนการออกแบบแยกออกจากขั้นตอนของการแมพ (Mapping) อย่างชัดเจน เพราะว่าชิ้นงานต้นแบบอาจจะต้องใช้ FPGA มากกว่าหนึ่งตัวซึ่งบ่อยครั้งแบบที่ได้จะเป็นการออกแบบระบบที่มีขนาดใหญ่เพียงระบบเดียว เครื่องมือที่เป็นตัวกลาง (Medium) ในการเชื่อมต่องานทั้งสองส่วนก็มักจะมีผลกระทบมากต่อการออกแบบโดยใช้ FPGA ด้วยเช่นกัน แต่อย่างไรก็ตามการออกแบบมักจะไม่สามารถทำให้มีรูปแบบที่ง่ายต่อการทำความเข้าใจและอาจจะต้องใช้ความพยายามอย่างมากเมื่อเป็นงานที่มีความคิดที่ค่อนข้างแปลกใหม่ดังนั้นแนวคิดที่เกี่ยวกับการออกแบบจึงมักจะมุ่งไปในเรื่องของปัญญาปลีกย่อยต่าง ๆ ที่นำมาใช้ในการออกแบบ เช่น การลดรูปวงจรลอจิก (Logic minimization) การออกแบบด้วยเทคนิค State machine หรืออาจจะมุ่งไปที่การลดขนาดของแบบให้เป็นสมการทางคณิตศาสตร์เพื่อให้ได้วิธีการปัญหาที่สมบูรณ์ เพื่อเป็นการแก้ปัญหานี้จึงต้องมีการแยกขั้นตอนการออกแบบกับการสร้างชิ้นงานออกจากกันทำให้พิจารณาได้ว่า ตัวแบบในความหมายที่เป็นนามธรรมจะหมายถึงสิ่งที่แสดงการเชื่อมต่อกันของส่วนประกอบต่าง ๆ ที่มีการกำหนดคุณสมบัติไว้อย่างชัดเจนทุกชิ้นส่วน ชิ้นส่วนบางชิ้นอาจจะมีการเรียกใช้ซ้ำ ๆ กัน และอาจจะถูกประกอบจากชิ้นส่วนปลีกย่อยอื่นๆ โครงสร้างของการออกแบบในลักษณะนี้จะพบเห็นได้ในทุกๆแขนงงานทางวิศวกรรม เช่น ในงานอิเล็กทรอนิกส์โครงสร้างนี้พบในการเชื่อมต่อของสายไฟระหว่างเกทในงานทางเครื่องกลจะพบได้ในการจับยึดด้วยหมุดหรือขอเกี่ยวที่จับยึดระหว่างคานต่างๆเป็นต้น

อย่างไรก็ตามแบบงานที่นำไปสร้างเป็นชิ้นงานได้จริงนั้น ชิ้นส่วนแบบนามธรรมซึ่งมีการกำหนดในเชิงโครงสร้างไว้แล้วจะต้องสามารถแมพลงบนชิ้นส่วนที่เป็นทางกายภาพที่มีอยู่ในเครื่องมือตัวกลางในการออกแบบนั้นได้ ถ้ากระบวนการนี้สามารถทำได้ก็จะทำให้เกิดชิ้นงานขึ้นมาได้จริง ด้วยเหตุนี้จึงสามารถกำหนดคุณลักษณะเชิงปริมาณของแบบเพื่อให้เกิดความมั่นใจในคุณภาพของแบบงานได้โดยจะพิจารณาจากลักษณะต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

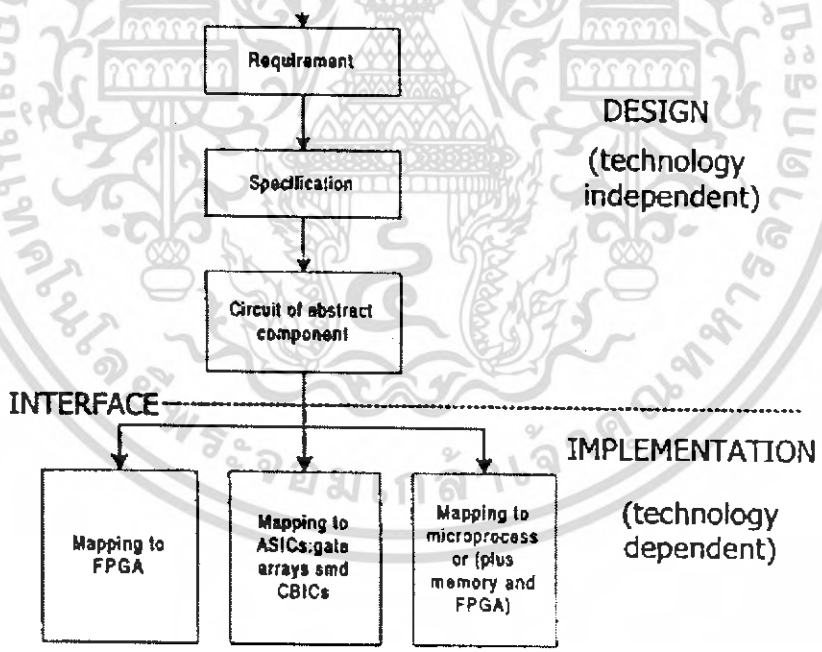
1) พื้นที่ใช้งาน (Space) แสดงด้วยจำนวนของอุปกรณ์ที่ต้องใช้ ส่งผลก่อนพื้นที่ใช้งานของบอร์ดซึ่งจะเป็นการช่วยให้สามารถคำนวณหาต้นทุนของชิ้นงานได้

2) ความเร็ว (Speed) แสดงด้วยความเร็วในการทำงานของงานต้นแบบที่สร้างขึ้นซึ่งก็จะเป็นการกำหนดสมรรถนะที่ต้องการของงานต้นแบบ

3) พลังงานและความต้องการกำลังงาน (Energy and power demand) การทำงานของชิ้นงานจะทำให้เกิดความสูญเสียรูปของความร้อน จำเป็นต้องลดเพื่อให้เกิดความสูญเสียน้อยที่สุด แต่อาจจะมีผลกระทบโดยตรงต่อการจัดการในเรื่องของพื้นที่ใช้งานของบอร์ด

4) ระยะเวลาที่ใช้ในการทำงานที่เหมาะสม (Timeliness) เป็นระยะเวลาที่ต้องใช้ทั้งหมดก่อนชิ้นงานจะผลิตออกสู่ตลาด

ในการออกแบบและสร้างชิ้นงานจะมีการดำเนินงานดังแสดงในรูปที่ ก.2 จากโฟลว์ชาร์ทการดำเนินงานจะเห็นงานแบ่งแยกส่วนกันอย่างชัดเจนในเรื่องของการออกแบบที่มีแนวทางการทำงานของผู้ออกแบบไม่ต้องขึ้นอยู่กับเทคโนโลยีการผลิต กับอีกส่วนหนึ่งที่เป็นขั้นตอนของการสร้างชิ้นงานจริงที่แนวทางการทำงานจะขึ้นอยู่กับเทคโนโลยีที่เลือกใช้ โดยอาจจะแบ่งเป็นแนวทาง สำหรับการออกแบบได้หลายรูปแบบตามลักษณะการสร้างชิ้นงาน เช่น ชิ้นงานสร้างบน FPGA ชิ้นงานสร้างบนเกทอาร์เรย์ (Gate arrays) และชิ้นงานที่สร้างบน cell-base ASIC เป็นต้น



รูปที่ ก.2 ความสัมพันธ์ระหว่างการออกแบบและการสร้างชิ้นงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของกระบวนการออกแบบแสดงให้เห็นแนวโน้มของหลักเกณฑ์ของการออกแบบ เพราะว่ามันจะต้องป้องกันไม่ให้มีการออกแบบในลักษณะที่ทำให้ไม่สามารถสร้างเป็นชิ้นงานจริงได้ แต่การจำแนกว่างานใดเป็นการออกแบบในลักษณะที่ทำให้ไม่สามารถสร้างเป็นชิ้นงานจริงได้ แต่การจำแนกว่างานใดเป็นการออกแบบที่ดีนั้นยังไม่มีความชัดเจนมากนักเพราะยังขึ้นอยู่กับเทคโนโลยีที่เลือกใช้ ซึ่งผู้ออกแบบต้องมีความกล้าที่จะกำหนดรูปแบบเทคโนโลยีเพื่อจะหาจุดที่ดีที่สุดของความต้องการ โดยคำนึงถึงความเป็นไปได้ในเชิงเศรษฐศาสตร์ด้วยดังนั้นข้อมูลบางส่วนของขั้นตอนการสร้างต้องถูกนำไปพิจารณาในขณะที่เป็นขั้นตอนการออกแบบโดยสามารถอยู่ในรูปแบบของกฎการออกแบบ (Design rules) ข้อแนะนำในการปฏิบัติหรือรูปแบบอื่นๆที่ได้จากทักษะและประสบการณ์ในของผู้ออกแบบ

2. รูปแบบที่ใช้ในการสร้างชิ้นงาน (Implementation styles)

ถึงแม้ว่าในการสร้างชิ้นงานของระบบอิเล็กทรอนิกส์จะสามารถขยายจำนวนระดับชั้นทางกายภาพ (Physical hierarchy) ที่สามารถผลิตได้จริงทั้งในส่วนของอุปกรณ์และแผ่นวงจรพิมพ์นั้นเป็นที่แพร่หลายจนเป็นอุปกรณ์มาตรฐาน ความก้าวหน้าในเรื่องของอุปกรณ์และเทคโนโลยีของบอร์ดทำให้เกิดรูปแบบต่างๆในการสร้างชิ้นงาน ซึ่งก็พอที่จะจำแนกได้เป็นยุคของเทคโนโลยีด้านต่างๆเช่นยุคของ SSL/MSI LSI และ VLSI ที่เติบโตอย่างรวดเร็วมากในช่วง 30 ปีที่ผ่านมา

2.1 ยุคกลุ่มตระกูลของลอจิกยุคแรกเริ่ม

ยุคนี้จะอยู่ในช่วงของปี 1960 จนถึง 1970 ซึ่งจะเป็นแบบฉบับของ Texas Instrument และ Transistor-Transistor Logic (TTL) และกลุ่มของ National Semiconductor (กลุ่มของ 4000 series) ที่มักนิยมเรียกกันว่า Small-scale integration (SSI) รวมไปถึงลอจิกที่อยู่ในรูปเกท ฟลิปฟลอปอุปกรณ์ด้าน Register transfer counter และ Arithmetic-logic unit (ALU) ที่อยู่ในระดับชิพ ด้วยเมื่อนำมาใช้งานในระดับของบอร์ด อุปกรณ์เหล่านี้จะมีจำนวนที่ไม่มากนักในแต่ละบอร์ดรวมถึงขนาดของเส้นทางของข้อมูล (Data-path size) ที่มีขนาดเล็ก เช่นขนาด 8 บิต เป็นต้น

2.2 ยุคอุปกรณ์ LSI และ VLSI

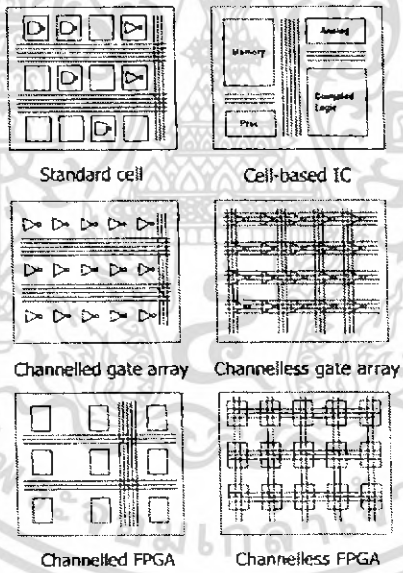
ในปี 1970 ได้มีการคิดค้น Dynamic RAM ของบริษัท Intel และปิดยุคด้วยการที่ตัวชิพแบบ Microprocessor ขนาด 32 บิต ได้กลายเป็นอุปกรณ์มาตรฐาน ดังเช่น Motorola 68000 เป็นต้น ความก้าวหน้าของเทคโนโลยีได้ทำให้เกิดการเปลี่ยนแปลงที่สำคัญสองประการคือ ขนาดของตลาดและจำนวนที่มากมายของตลาด เช่น ตลาดของ SRAM Micro controller ในยุคนี้อุปกรณ์ทางลอจิกของยุคแรกยังถูกใช้งานอย่างต่อเนื่องและมีปริมาณที่มากขึ้นอยู่ภายในชิพเดียวกันขนาดของบอร์ดก็จะโตมากขึ้นและมักจะให้ทุกอย่างอยู่บนบอร์ดเดียวกัน มีการพยายามลดต้นทุนการผลิตลด โดยลดจำนวนขา การเชื่อมต่อ และมีการกำหนดขาการเชื่อมต่อบนแผงวงจรพิมพ์ ด้วยอุปกรณ์ Connector เพื่อการนำระบบย่อยเหล่านี้ไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ยุคของอุปกรณ์ ASIC

เริ่มจากในปี 1980 เมื่อได้มีการพัฒนาอุปกรณ์ให้มีความแน่นหนาในชิพได้มากขึ้นควบคู่ไปกับการทำให้มีความซับซ้อนขึ้นในรูปแบบการทำงานจนต้องใช้เทคโนโลยีที่ใช้ในการผลิตมาสร้างอุปกรณ์ให้เป็น ASIC แทนการใช้อุปกรณ์มาตรฐาน เช่น อุปกรณ์ Mask-programmable gate arrays อุปกรณ์ Cell-based IC (CBIC) อุปกรณ์ Programmable array Logic (PAL) อุปกรณ์ Field-programmable array logic array (FPAL) รวมไปถึงอุปกรณ์ที่สามารถโปรแกรมได้รุ่นแรกๆที่ปรากฏให้เห็นได้ในระหว่างยุคที่ใช้ ASIC เป็นเทคโนโลยีพื้นฐานเพื่อการผลิต

การจัดแบ่งกลุ่มประเภทของ ASIC โดยทั่วไปแล้วได้มีการแบ่งเป็น 3 กลุ่ม คือ เกทอาร์เรย์, Cell-based ICs (CBIC) และ Programmable Logic device (PLD) ส่วน FPGA นั้นจะถูกจัดอยู่ในกลุ่มของ PLD จะมีความแตกต่างกันทั้งรูปแบบและโครงสร้างและความต้องการในกระบวนการผลิต เช่น CBIC มีความต้องการทุกขั้นตอนในกระบวนการสร้างขึ้นเป็นตัวไอซี และยังต้องใช้เวลาที่ยาวนานกว่ามาก เมื่อเปรียบเทียบกับเกทอาร์เรย์ที่ต้องการเพียงขั้นตอนของการทำ metallization ในกระบวนการผลิตเท่านั้น ในการนำ PLD ไปใช้งานสามารถที่จะดำเนินการเพียงบันทึกบิตข้อมูลลงบนชิพเพื่อทำการหลอม Anti fuse เท่านั้นในรูปที่ ก.3 แสดงให้เห็นถึงสัญลักษณ์และสถาปัตยกรรมภายในของ ASIC



รูปที่ ก.3 สัญลักษณ์และสถาปัตยกรรมภายในของ ASIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) อุปกรณ์เกทอาร์เรย์ โครงสร้างเกทอาร์เรย์ในยุคแรก จะคล้ายสถาปัตยกรรมแบบมีช่องสายสัญญาณเชื่อมต่อ (Channeled architecture) ที่มีลักษณะเป็นแถว และมีบางส่วนอยู่ในแถว คอลัมน์ของกลุ่มเกทที่ถูกกั้นด้วยช่องสายสัญญาณเชื่อมต่อ (Wiring channel) ที่มีขนาดความจุคงที่ภายในชิพจะมีระดับความหนาแน่นประมาณพันเกท แต่ก็พอเพียงต่อการนำไปใช้เป็นส่วนของวงจรควบคุมระบบ ซึ่งขึ้นอยู่กับความต้องการในแต่ละงาน เช่น ในส่วนของข้อมูลและหน่วยความจำ เป็นต้น นอกจากนี้เกทอาร์เรย์ยังมีวิวัฒนาการต่อไปจนกระทั่งอยู่ในรูปแบบที่ไม่มีช่องสายสัญญาณเชื่อมต่อ (Channel less) หรืออยู่ในรูปแบบ "Sea-of-gate" และมีโครงสร้างของสารกึ่งตัวนำเป็นแบบ Multilevel metallization โดยโครงสร้างเช่นนี้จึงทำให้มีความเป็นอิสระสูงในการที่จะเชื่อมต่อวงจรกันของเกทด้วยตัวนำแบบที่มีสายเชื่อมต่อหลายชั้น (Multiple level wire) ที่อยู่ในชิพได้เป็นอย่างดี

2) อุปกรณ์ Cell-based ICs การออกแบบโดยใช้ Cell-Based จะมีการเตรียมไลบรารีของ Proven cell ไว้ให้ผู้ใช้ โดยทั่วไปแล้วมักจะกำหนดให้มีความสูงคงที่แต่จะสามารถปรับความกว้างได้ขึ้นอยู่กับความกว้างที่ใช้งาน โดยในช่วงปี 1980 ได้มีการพัฒนากลุ่มเซลล์มาตรฐาน (Standard cell families) ที่มีฟังก์ชั่นซับซ้อนมาก และเทคโนโลยีทางด้านซิลิคอนที่ดีขึ้นทำให้เพิ่มความสามารถของการออกแบบกลุ่มของฟังก์ชั่นได้ดีขึ้น แต่กระบวนการสร้างชิพ (Fabrication) ก็จะต้องทำทุกขั้นตอนให้เสร็จอย่างสมบูรณ์ทั้งการทำ mask และ process อุปกรณ์ชนิดนี้จะเป็น ASIC ที่มีความหนาแน่นสูงมากและมีสมรรถนะสูงที่สุด

3) อุปกรณ์ Programmable Logic device การถือกำหนดของ PLD สามารถถือได้ว่าเกิดจากอุปกรณ์ประเภท Monolithic memory และทำให้เกิดการสร้าง PAL ขึ้นเป็นครั้งแรกในปี 1978 ซึ่งมันมีลักษณะคล้ายกับ FPLA ในยุคแรกก็ไม่ประสบผลสำเร็จนักในรูปแบบของอุปกรณ์แบบ Two-level product จนกระทั่งถึงปี 1980 แนวคิดพื้นฐานของ PLD ได้ถูกพัฒนาขึ้นเป็น Complex PLD ที่อยู่ในรูปแบบของ FPGA และโครงสร้างของ

2.4 ยุคของ Programmable Logic

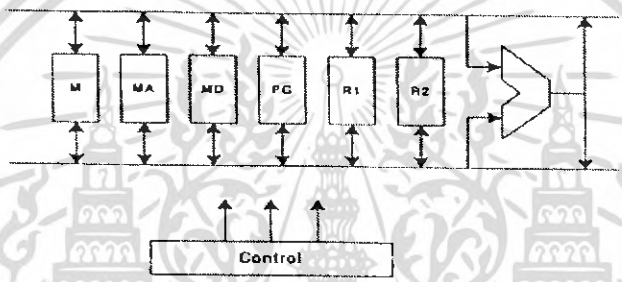
การเกิดขึ้นของอุปกรณ์ประเภท Complex programmable ดังเช่น FPGA ทำให้สามารถที่จะคาดการณ์ได้ว่าระบบที่ใช้ไมโครโปรเซสเซอร์สามารถที่จะถูกแทนได้ด้วย SRAM-programmable FPGA แม้กระทั่งการคำนวณแบบ Von Neumann engine ที่สนับสนุนการคำนวณแบบใช้งานได้เอนกประสงค์ก็สามารถที่จะดำเนินการได้เป็นอย่างดีบน FPGA

3. รูปแบบการออกแบบ (Design styles)

เมื่อนำงานที่ออกแบบได้ไปสร้างเป็นชิ้นงานในทางปฏิบัติก็ยังมีสิ่งที่ยุ่งยาก คือการสร้างวงจรของอุปกรณ์ที่มีลักษณะเป็นแบบนามธรรม (Abstract components) ที่จะต้องสร้างขึ้นตามรายละเอียด (Specification) ที่กำหนด เนื่องจากเป็นการทำงานเกี่ยวกับระบบดิจิตอล ดังนั้นข้อมูลอาจอยู่ในรูปอัลกอริทึมที่มีรูปแบบเป็นทางการหรือไม่เป็นทางการก็ได้ การแปลผล (Interpretation) ของอัลกอริทึมจึงมีผลกระทบต่อการใช้ประโยชน์บางอย่างจากข้อมูลของการออกแบบอย่างหลีกเลี่ยงไม่ได้ อัลกอริทึมหนึ่งๆ สามารถเป็นได้ทั้งกลุ่มของกฎ (Set of rules) หรือกลุ่มของขั้นตอนการทำงานเพื่อทำการแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Transforming data) และดำเนินการคำนวณ (Executing a computation) โดยปกติแล้วตัวอัลกอริทึมจะมีลักษณะเป็น Static description ที่จำเป็นต้องใช้เครื่องมือ (Machine) ในการแปลผลเพื่อแสดงการทำงานวงจรของอุปกรณ์ที่มีลักษณะเป็นนามธรรมก็เป็นเครื่องมือที่แสดงถึงการทำงานตามอัลกอริทึม ดังนั้นงานที่หนักของการออกแบบจึงอยู่ที่การเลือกกลุ่มอุปกรณ์ที่ต้องสัมพันธ์กันทั้งฮาร์ดแวร์และซอฟท์แวร์ในการเชื่อมต่ออุปกรณ์เหล่านี้เข้าด้วยกัน กระบวนการนี้จะเรียกว่า สถาปัตยกรรมระบบ (System paradigm) ซึ่งจะมีรูปแบบที่เกิดขึ้น 2 แนวทาง คือ ตัวแบบแผนทางซอฟท์แวร์ (Software paradigm) ที่อาศัยตัวแบบการคำนวณของ Von Neumann และตัวแบบแผนทางฮาร์ดแวร์ (Hardware paradigm) สถาปัตยกรรมที่มีการใช้ FPGA จะมีลักษณะเป็นตัวแบบแผนประเภทใดประเภทหนึ่งหรือถูกสร้างจากอุปกรณ์ที่มีลักษณะเป็นตัวแบบแผนประเภทใดประเภทหนึ่งก็ได้ เมื่อสร้างโดยตัวแบบแผนทางซอฟท์แวร์ อัลกอริทึมจะถูกแปลโดยตัวประมวลผล (Process) แล้วแมพรหัส



รูปที่ ก.4 แสดงต้นแบบของเครื่องคำนวณแบบ Von Neumann engine

ในรูปที่ ก.4 แสดงต้นแบบของเครื่องคำนวณแบบ Von Neumann engine ที่ประกอบไปด้วยหน่วยความจำ (Memory : M) รีจิสเตอร์ต่างๆ (Memory address , Memory data , Program counter) และ ALU เมื่อสร้างโดยตัวแบบแผนฮาร์ดแวร์อัลกอริทึมจะถูกแมพลงบน Storage unit และ Function unit ด้วยลักษณะการที่เหมือนกัน นอกจากนี้ระดับที่สูงที่สุดของมโนภาพของแบบงาน (Design abstraction) อาจจะทำให้ส่งผลดีต่อความสามารถในการออกแบบกับงานที่ออกแบบได้จริง โดยเฉพาะงานที่ต้องการความรีบด่วนมากกว่าปกติกระบวนการออกแบบที่ดีนี้ก็จะช่วยทำให้ผู้ใช้ต้องปรับปรุงตรวจสอบงานเพียงเล็กน้อยเท่านั้น เพื่อให้งานเป็นไปตามกฎการออกแบบ เช่น การเลือกสถาปัตยกรรมแบบ Pipelined จะทำให้ความสามารถของระบบมีสภาพที่ดีขึ้นถึง 100% แทนการเลือกเทคโนโลยีที่สร้างวงจร (Circuit technology) ซึ่งจะทำให้ระบบมีสภาพที่ดีขึ้นเพียงประมาณ 10% เท่านั้น

เพื่อแสดงแนวทางสถาปัตยกรรมทั้งสองแบบ โดยจะพิจารณาการออกแบบอย่างง่ายทั้งสองแบบของอัลกอริทึมการหาผลคูณของเวคเตอร์ที่มีสมการดังนี้

$$\sum_{i=0}^{n-1} a_i * b_i \qquad (ก-1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ตัวแบบแผนทางซอฟท์แวร์

การออกแบบภายใต้แบบแผนทางซอฟท์แวร์ จะพบว่างานของการออกแบบคือ การเขียนกลุ่มคำสั่ง(Program) สำหรับอัลกอริทึม ดังนี้

```
sum = 0
for i=0 to n-1 do
sum = sum + a[i]* b[i]
```

จากนั้นคอมไพล์ด้วยเครื่องคำนวณแบบ Von Neumann และใช้อุปกรณ์ในรูปที่ ก.4 ด้วยการสมมุติว่าเป็นโครงสร้างอย่างง่ายแทนตำแหน่งลงบนแผนผังการทำงานโดยรีจิสเตอร์ R1 ให้ทำหน้าที่เป็น accumulator และให้ R2 เป็นรีจิสเตอร์พื้นฐาน (base register) ดังนั้นการจัดหน่วยความจำจะแสดงได้เป็น

โปรแกรมที่จะทำการคำนวณฟังก์ชันแสดงดังรูปที่ ก.5

```
        CLR     R1
        STR     R1,R2,2     sum = 0
        STR     R1,R2,3     i = 0

loop    LOAD    R1,R2,4     R1=a0
        ADD     R1,R2,3     R1=a0+1
        LOAD    R1,R1,0     R1=a[i]
        STR     R1,R2,6     tmp=a[i]

        LOAD    R1,R2,5     R1=a0
        ADD     R1,R2,3     R1=b0+1
        LOAD    R1,R1,0     R1=b[i]

        MLT     R1,R2,6     R1=b[i]*tmp
        ADD     R1,R2,2     R1=sum+ditto
        STR     R1,R1,2     i = 0

        LOAD    R1,R2,3     R1= i
        ADDIM   R1,1        R1=R1+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

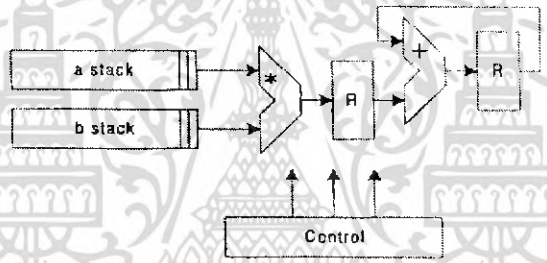
```
STR     R1,R2,3

CMP     R1,R2,1     compare i : n
BNE     loop        branch if i ≠ n
```

รูปที่ ก.5 ตัวอย่างโปรแกรมการคำนวณโดยใช้ Von Neumann

3.2 ตัวแบบแผนทางฮาร์ดแวร์ (Hardware Paradigm)

การออกแบบภายใต้แบบแผนทางฮาร์ดแวร์ ตัวแบบจะมีพื้นฐานอยู่บนการเชื่อมต่อภายในของฮาร์ดแวร์สแตคจำนวน 2 ชุด มี Unit function 2 หน่วย และมีรีจิสเตอร์ 2 ตัว ข้อมูลจะแสดงในรูปบิตอนุกรม (bit-serial) หรือบิตขนาน (bit-parallel) นอกจากนั้นฮาร์ดแวร์ สแตคจะมีขนาดที่จำกัดชัดเจน และสามารถถูกแบ่งแยกการทำงานออกจากกันได้อย่างอิสระ การออกแบบเช่นนี้สามารถสร้างได้จากชิ้นส่วนมาตรฐานที่ระดับ MSI



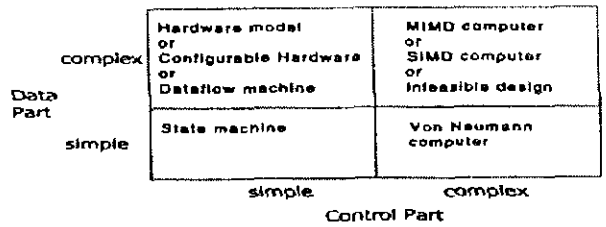
รูปที่ ก.6 หน่วยประมวลผลของ Vector product

3.3 ตัวแบบแผนทางฮาร์ดแวร์ที่สามารถคอนฟิกูเรชันได้

(Configurable hardware paradigm)

จากตัวอย่างการออกแบบทั้งสองแนวทาง จะเป็นลักษณะที่ง่ายของสถาปัตยกรรมแต่ก็สามารถทำให้เห็นถึงความแตกต่างกันได้มากเมื่อพิจารณาเส้นทางของข้อมูล (Data path) และโครงสร้างการควบคุมของแต่ละสถาปัตยกรรม สำหรับตัวแบบทางซอฟท์แวร์จะมีหน่วยคำนวณอย่างง่ายบนหน่วยความจำพื้นฐานของเครื่อง รีจิสเตอร์ไฟล์ และ ALU ที่ถูกแสดงความซับซ้อนไว้ด้วยรหัสส่วนของการควบคุม (Compiled code) ได้อย่างชัดเจน ส่วนตัวแบบแผนทางซอฟท์แวร์จะมีเส้นทางข้อมูลซึ่งซับซ้อนกว่าเนื่องจากประกอบไปด้วยชิ้นส่วน 6 ชุด เชื่อมต่อกันด้วยโครงสร้างในรูปแบบการควบคุมที่เป็นไปตามลำดับ สำหรับความซับซ้อนของข้อมูลและโครงสร้างการควบคุมสามารถเลือกได้โดยดูจากรูปที่ ก.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.7 ความซับซ้อนของข้อมูลและโครงสร้างการควบคุมสามารถเลือก

การออกแบบด้วยตัวแบบแผนทางฮาร์ดแวร์จะมีลักษณะเป็นเครื่องที่ใช้งานเฉพาะทางมีการทำงานตามอัลกอริทึมด้วยความเร็วสูงและมีค่าใช้จ่ายในการสร้างชิ้นงานต้นแบบอย่างน้อยหนึ่งชิ้นที่ค่าหนึ่งในทางกลับกันตัวแบบแผนทางฮาร์ดแวร์จะให้คำตอบที่สมรรถนะต่ำกว่าเนื่องจากต้องเสียเวลาในการแปลโปรแกรมแต่คำตอบที่ได้จะมีค่าใช้จ่ายถูกกว่าเพราะเครื่องคอมพิวเตอร์มีการผลิตเป็นจำนวนมากจึงมีราคาถูกและนำมาใช้ได้อีก เมื่อมีการสร้าง FPGA ที่ใช้งานได้อย่างกว้างขวางสามารถโปรแกรมซ้ำได้และนำมาใช้ได้อีก สามารถสร้างได้เป็นจำนวนมาก ดังรูปแบบของฮาร์ดแวร์ที่ทำการคอนฟิกูเรชันให้มีความสามารถในการคำนวณจึงถูกสนับสนุนมากขึ้น เนื่องจากการมีข้อได้เปรียบในเรื่องของสมรรถนะที่อัลกอริทึมสามารถถูกแมพไปยังฮาร์ดแวร์ได้โดยตรง ดังนั้นเมื่อผลิตเป็นปริมาณมากก็จะทำให้ต้นทุนต่ำที่ถูกได้เช่นกัน ตารางที่ ก.1 แสดงภาพรวมของการออกแบบสถาปัตยกรรมของตัวแบบแผนทางฮาร์ดแวร์

และซอฟท์แวร์แต่ในสภาพการณ์จริงสถาปัตยกรรมที่ดีที่สุดจะมีองค์ประกอบได้ทั้งสองตัวแบบ

ตารางที่ ก.1 เปรียบเทียบแนวทางต่างๆ ของภารออกแบบ

Feature	Software	Hardware	Configurable Hardware
Instruction interpretation	Integral number of cycles	None	None
Gate delay	Low	Low	High
Spatial demand	PCB	IC	PCB
Cost (1 off)	Commodity	High NRE	Commodity
Architural efficiency	Fixed	Tailored architecture	Tailored architectures
Excution efficiency	Fixed units	Tailored excutation units	Tailored architectures
Bit efficiency	Fixed	Variable word size	Variable word size
Data reprecentation	Fixed	Variable word size	Variable word size
Reusability	Yes	No	Yes

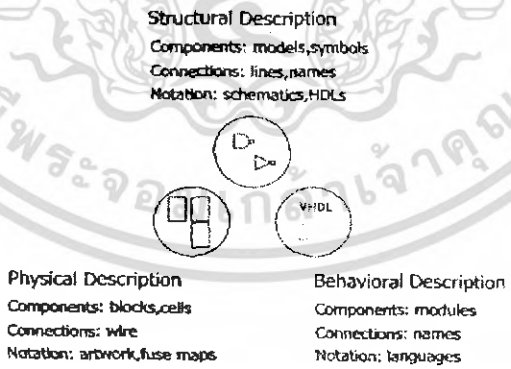
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. วิธีการออกแบบ (Design methodologies)

วิธีการออกแบบระบบด้วยการใช้อุปกรณ์ FPGA เพื่อให้ได้ชิ้นงานต้นแบบที่เหมาะสมจะต้องพิจารณาสิ่งต่างๆที่มีอยู่หลายประการคือ

4.1 การอธิบายแบบ (Describing a design)

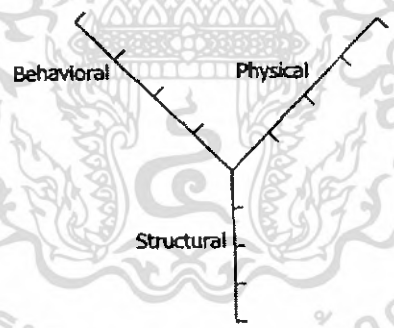
การอธิบายแบบมีหลายรูปแบบโดยเฉพาะเมื่อเทคโนโลยีความก้าวหน้ามากขึ้นและความซับซ้อนของแบบมีมากขึ้นแผ่นแบบของแผนภาพ (Diagramming templates) และบอร์ดภาพร่าง (Drawing boards) ได้ถูกแทนด้วยโปรแกรมการทำงานของระบบออกแบบที่เรียกว่า CAD ทำให้การอธิบายวงจรพื้นฐานที่เป็นอุปกรณ์ เช่น ทรานซิสเตอร์ รีจิสเตอร์ และคาปาซิสเตอร์ อยู่ในรูปแบบของฐานข้อมูล ที่ถูกรวบรวมไว้ในรูปของแผนภาพลำดับขั้น (Hierarchical schematics) นอกจากนี้ก็มีการใช้โปรแกรมภาษาที่อธิบายฮาร์ดแวร์ (Hardware description language program : HDL) ใช้ตัวแบบจำลอง (Simulation model) การออกแบบแผนงานทางศิลป์ (Layout artwork) และแบบแผนทดสอบ (Test pattern) ต่าง ๆ ทำให้เห็นถึงความแตกต่างของการให้คำอธิบายแบบ (Design description) กับการบันทึกหมายเหตุ (Notation) โดยความหมายของการให้คำอธิบายแบบที่ชัดเจนนั้นข้อมูลจะต้องถูกออกแบบหรือสร้างให้อยู่ใน 3 กลุ่มต่อไปนี้ คือ กลุ่มโครงสร้าง (Structural domain) กลุ่มพฤติกรรม (Behavioral domain) และกลุ่มกายภาพ (Physical domain) ดังแสดงแผนภาพของเวนน์และลักษณะเฉพาะของส่วนประกอบภายในการอธิบายแบบแต่ละกลุ่ม การออกแบบที่อยู่ในระดับลอจิกไดอะแกรมได้ การอธิบายแบบในพฤติกรรม (Behavioral description) สามารถแสดงเป็นกลุ่มของสมการทางลอจิก (Logic equation)ได้ และการอธิบายแบบด้วยทางกายภาพ ก็จะเป็นกลุ่มของแผนภาพเซลล์มาตรฐาน (Standard cell layout) ของ ASIC แต่ละตัว (หรือ Fuse map ของ PAL) ส่วนบันทึกหมายเหตุนั้นมีจุดมุ่งหมายเพื่อใช้ในการบรรยายสิ่งเหล่านี้ โดยบันทึกหมายเหตุนั้นมีจุดมุ่งหมายเพื่อใช้ในการบรรยายสิ่งเหล่านี้ โดยบันทึกหมายเหตุที่แตกต่างกันก็อาจจะใช้สร้างส่วนย่อยๆที่แตกต่างกันในแต่ละส่วนของตัวแบบงาน



รูปที่ ก.8 กลุ่มต่าง ๆ ของการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้แบบงานที่มีความซับซ้อนจะถูกสร้างโดยการอธิบายแบบในรูปแบบต่างๆจากหลายระดับในทางปฏิบัติแบบงานในเชิงวิศวกรรมมักเป็นการดำเนินงานแบบแบ่งส่วนงานให้แยกย่อยก่อนการเข้าไปแก้ปัญหา ทำให้แบบแผนทางความคิด (Conceptual) การทำงานของระบบที่มีขนาดใหญ่ถูกแบ่งให้เป็นหน่วยที่เล็กลงจนกระทั่งเหมาะสมต่อการสร้างเป็นชิ้นงานจริงในระดับอุปกรณ์ทางกายภาพได้ ตัวอย่างของระดับแบบแผนความคิด (Conceptual levels) ของมโนภาพของแบบงานมักจะประกอบไปด้วย แผนผังวงจร อุปกรณ์ลอจิก อุปกรณ์ Register transfer และ Processor-Memory-Switch (PMS) ในระดับที่เป็นทางกายภาพก็จะประกอบไปด้วยอุปกรณ์สารกึ่งตัวนำ และแผ่นวงจรพิมพ์ เป็นต้น ดังนั้นการมีเพียงมิติเดียวของการอธิบายแบบดังรูปที่ ก.8 จึงมีลักษณะที่เข้าใจง่ายและสามารถถูกปรับเปลี่ยนหรือแก้ไขเพื่อให้มีการอธิบายแบบในระดับที่แตกต่างแยกย่อยไปได้ เท่าที่ต้องการ โดยไม่จำกัดจำนวนลำดับชั้นของการอธิบาย ด้วยเหตุนี้จึงเป็นวิธีที่ดีสำหรับการนึกภาพและดูการเลื่อนไหล (Flow) ของกระบวนการออกแบบ ในรูปที่ ก.9 แสดงถึงรูปกราฟความสัมพันธ์กันบน 3 แกน (Gajski-Kuhn diagram) โดยให้ Y แทนการอธิบายแบบของทั้งสามกลุ่ม แต่ละแกนจะแทนสเกลระดับของการออกแบบ ในแผนภาพนี้ขั้นตอนของกระบวนการออกแบบจะถูกแสดงในรูปของการผ่านจุด (transition) จากบนแกนหนึ่งไปยังจุดที่อยู่บนแกนอื่น เช่น การสร้างแผนภาพจากลสอจิกไดอะแกรม ซึ่งมีลักษณะการออกแบบในเชิงพฤติกรรม เป็นต้น ก็จะมีกระบวนการไหลของการออกแบบ (Design process flow) เป็นเหมือนกับการหมุนรอบแกนทั้งสาม (Spiral path) จากจุดเริ่มต้นของกราฟเป็นแบบเชิงพฤติกรรม แล้วพัฒนาต่อไปเป็นโครงสร้างและเป็นแบบทางกายภาพในที่สุด นอกจากนี้การอธิบายแบบทั้งสามกลุ่มจะมีรูปแบบเฉพาะที่แตกต่างกัน คือ



รูปที่ ก.9 แผนภาพ Gajski-Kuhn diagram

1) **Structural design** เป็นแบบที่แสดงในเชิงโครงสร้างการเชื่อมต่อของชิ้นส่วนต่างๆเข้าด้วยกันอย่างง่าย เช่น การใช้สายไฟเชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์เข้าไว้ด้วยกัน การใช้หมุดย้ำและขอเกี่ยวเพื่อยึดโครงสร้างอุปกรณ์เข้าไว้ด้วยกัน เป็นต้น ซึ่งพบว่าตำแหน่งทางโครงสร้างถูกกำหนดโดยสถาปัตยกรรมที่ทำให้ระบบมีลักษณะเป็นไปตามที่ต้องการ เช่น สมรรถนะ ความสมบูรณ์ของโครงสร้าง (Structural integrity) และอื่นๆ สำหรับแบบที่แสดงในเชิงโครงสร้างเพื่อใช้งานในด้าน FPGA นั้นจะใช้โปรแกรมประเภท schematic drawing tool

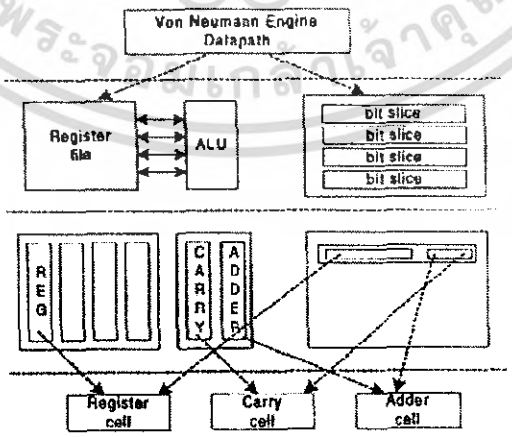
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) Physical design เป็นแบบที่แสดงในเชิงกายภาพที่จะต้องมีการรวมเอาการสร้างข้อมูลที่พอเพียงสำหรับการออกแบบเพื่อการผลิตรวมถึงการทดสอบไว้ด้วย แบบเชิงกายภาพสำหรับ FPGA จำเป็นจะต้องมีการกำหนดหรือนิยามฟังก์ชันสำหรับลอจิกบล็อกและส่วนของการเชื่อมต่อที่เชื่อมต่ออยู่ภายในด้วย เหมือนกับการสร้างแผนภาพไว้ในการทำ VLSI ส่วนข้อมูลของการผลิตชิ้นงานจะอยู่ในรูปของแฟ้มข้อมูลแบบไบนารีสำหรับการทำโปรแกรมเครื่องมือที่ใช้ออกแบบเชิงกายภาพจะต้องมีทั้ง Symbolic editors, Generator และซอฟท์แวร์สำหรับกำหนดเส้นทางและตำแหน่งแบบอัตโนมัติ (Automatic place-and-route software)

3) Behavioral design เป็นแบบที่แสดงในเชิงพฤติกรรมรายละเอียดของแบบชิ้นงานจะอยู่ในรูปของฟังก์ชันการทำงานโดยอาจถูกรวมไว้ในตัวแบบจำลอง (Simulation model) ในโปรแกรมในกลุ่มของสมการลอจิกหรือในตารางความจริง (Truth table) ก็ได้ แม้กระทั่งการอธิบายด้วยลอจิกไดอะแกรมก็สามารถแสดงในแบบเชิงพฤติกรรมได้ด้วยการใช้แบบแผนการเขียนภาพวาด (Drawing) อย่างง่ายๆ

4.2 การออกแบบรูปแบบที่มีรูปแบบเป็นลำดับขั้น (Hierarchical design)

การออกแบบรูปแบบที่มีรูปแบบเป็นลำดับขั้นหรือแบบเชิงโครงสร้างถูกค้นพบเมื่อความซับซ้อนในการออกแบบที่เป็น Full-custom มีมากขึ้นในด้านการผลิต ส่งผลกระทบต่อวิศวกรสร้างซอฟท์แวร์ที่จะต้องปรับวิธีการเขียนโปรแกรมโดยใช้เทคนิคการแบ่งย่อยงานแล้วเข้าไปแก้ปัญหาเพื่อผลิตซอฟท์แวร์ให้มีความซับซ้อนในการสร้างมากขึ้นตามไปด้วย แนวความคิดเกี่ยวกับการแยกส่วน (Modularization) การซ่อนสาระสำคัญ (Information hiding) การตรวจสอบอย่างเป็นขั้นตอน (Stepwise refinement) จึงถูกนำมาใช้เพื่อการออกแบบ VLSI ทำให้มีการนำเทคนิคการเขียนโปรแกรมมาประยุกต์ใช้กับการทำเลย์เอาท์ของ VLSI เพราะว่าโครงสร้างการเขียนโปรแกรมยอมให้มีการแยกรายละเอียดต่างๆ ของอาร์คเวิร์กได้ โดยเฉพาะอย่างยิ่งเมื่อมีการทำซ้ำหรือการทำงานอย่างมีเงื่อนไข ในรูปที่ ก.10 จะเป็นการแสดงถึงการออกแบบที่มีลักษณะเป็นลำดับขั้น



รูปที่ ก.10 ลำดับขั้นการแบ่งส่วนของ data path

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ ก.10 แสดงการแยกส่วนแบบมีลำดับขั้นของเส้นทางข้อมูลที่ถูกกำหนดขึ้นให้อยู่ในรูปของเครื่องคำนวณแบบ Von Neumann โดยแบบงานสามารถถูกแจกแจงให้เห็นรายละเอียดด้วยกลุ่มของแบบแผนวงจรที่ระดับต่ำสุดได้เป็นวงจรหรือลอจิกไดอะแกรมของเซลล์ต่างๆ ทั้งการแบ่งส่วนรายละเอียดจริงของแบบงานของความแตกต่างของแต่ละส่วนด้วยทางเลือกที่อยู่ในแผนผังการแบ่งส่วนที่อยู่ในแนวแกนนอนก็จะเป็นการให้ความสำคัญกับการออกแบบของ Bit slice หรือในแนวแกนตั้งก็จะเป็นการให้ความสำคัญของการออกแบบของ Functional unit จากรูปจะเป็นแบบแผนการแบ่งส่วนที่ทางเลือกจะเป็นการแสดงด้วยแผนผังการเชื่อมต่อของสาย ในตัวอย่างนี้เป็นบัสที่ใช้สื่อสารระหว่าง Operands และส่งผลลัพธ์ที่ได้ระหว่างรีจิสเตอร์ไฟล์กับ ALU การออกแบบที่ถูกต้องของแบบที่เป็นลำดับขั้นจะต้องแบ่งชิ้นส่วนของกลุ่มสายเหล่านี้ไปยังเซลล์ที่อยู่ด้านใต้ของเส้นแบ่งชั้นเสมอ เพราะจะแสดงให้เห็นถึงระดับความซับซ้อนที่แตกต่างกันในการเชื่อมต่อข้อมูลเพื่อนำไปจำลองการทำงาน

4.3 การออกแบบที่มีความเป็นอิสระจากเทคโนโลยี (Technology-independent design)

การออกแบบที่มีความเป็นอิสระจากเทคโนโลยีจะใช้ได้เฉพาะกับแบบงานที่เป็นการออกแบบขั้นสูงเท่านั้นโดยใช้โปรแกรมภาษาVHDL (Hardware Description Language) และจะแปลงรูปแบบของข้อมูลอย่างอัตโนมัติด้วยโปรแกรมสังเคราะห์เพื่อให้อยู่ในรูปของตัวกลาง (Intermediate form) ที่สามารถนำไปกำหนดอุปกรณ์เป้าหมายให้มีรูปแบบในการสร้าง (Implementation styles) ที่แตกต่างกันหรือผู้ผลิตที่แตกต่างกันได้ จึงมีลักษณะคล้ายกับเทคโนโลยีตัวแปลภาษาระดับสูง(High level language compiler technology) ที่เป็นตัวสร้างรหัสการทำงานจากข้อมูลในรูปแบบตัวกลางไปสร้างเป็นรหัสสำหรับเครื่องจักร (Machine) ที่ต้องการ สิ่งที่จะกำหนดว่าเป็นการออกแบบที่มีความเป็นอิสระจากเทคโนโลยีก็คือ มีการสูญเสียสูงในเรื่องของประสิทธิภาพการแมพจากแบบซึ่งเป็นการออกแบบขั้นสูงให้ไปเป็นชิ้นงานจริง และมีการสูญเสียจากการที่ผู้ออกแบบสามารถควบคุมค่าของเวลาที่ใช้ในการทำงานและขนาดของเกทในแบบงานได้โดยตรง ซึ่งเหตุผลทั้งคู่จะเป็นกุญแจที่นำไปสู่ความสำเร็จหรือไม่สำเร็จในการสร้างชิ้นงานจริงได้

4.4 การออกแบบด้วยวิธี Mead-Conway

การออกแบบด้วยวิธี Mead-Conway จะนิยมใช้ในการออกแบบวงจรVLSI มีลักษณะเป็นการออกแบบที่เป็นลำดับขั้นที่มุ่งเน้นใน เรื่องของการจัดการเกี่ยวกับความซับซ้อนในการสื่อสารภายในหรือการเชื่อมต่อของสายตัวนำในชิ้นงาน ซึ่งเป็นเรื่องที่สำคัญมากเพราะว่าต้นทุนของเวลา ที่ใช้ในการเคลื่อนย้ายข้อมูลไปรอบๆ วงจรของ VLSI อาจจะมีผลมากต่อการทำงานทั้งหมดของมัน วัตถุประสงค์ของเทคนิคการออกแบบคือเพื่อเพิ่มประสิทธิภาพของงานที่ออกแบบ ซึ่งจะประกอบไปด้วยหลักเกณฑ์ดังนี้

1) การเชื่อมต่อของสายจะเป็นกุญแจสำคัญในการออกแบบในการออกแบบขั้นสูง

2) ใช้กฎพื้นฐานในการออกแบบในการทำให้ส่วนที่เป็นนามธรรมหมดไปจากรายละเอียด

3) ใช้ Build block เป็นส่วนของทางเดินข้อมูลและ PLA (Two-level logic block) เนื่องจากมันมีโครงสร้างของการเชื่อมต่อของตัวนำที่ดี

วิธีการนี้จะจัดวางตำแหน่งของชิ้นส่วนต่างๆ ภายในวงจร VLSI โดยคำนึงถึงโครงสร้างและรูปแบบกายภาพเป็นหลัก การพยายามรักษารายละเอียดรูปแบบของโครงสร้างและรูปแบบทางกายภาพให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมือนเดิมที่สุดจะช่วยหลีกเลี่ยงการเกิดข้อบกพร่อง (bug) ที่อาจจะเป็นกับดักที่ซับซ้อนในการออกแบบ หากจำเป็นต้องมีการแยกโครงสร้างกับลำดับขั้นทางกายภาพออกจากกัน

เทคนิคนี้สามารถมาใช้ได้กับ FPGA ได้เป็นอย่างดีภายใต้เงื่อนไขคือ FPGA จะต้องมีความเหมือนกันอย่างแท้จริงกับตัวสื่อกลางหรือโปรแกรมที่ใช้สำหรับการออกแบบและสร้างวงจร อย่างในกรณีนี้จะเห็นได้ว่าสถาปัตยกรรมของ FPGA แบบ Fine-grain จะมีความคล้ายกันกับลอจิกอาร์เรย์ที่โปรแกรมได้เนื่องจากชิ้นส่วนในระดับเกทอาร์เรย์ของมันเป็นแบบเซลลูล่าร์อาร์เรย์มีความเหมาะสมกับอุปกรณ์แวดล้อมมากกว่าแบบสถาปัตยกรรมแบบ Coarse-grain channeled ซึ่งผู้จัดรูปแบบนั้นสามารถที่จะคาดหวังได้จากการเตรียมทางเลือกต่างๆในการสร้างได้โดยตรง

4.5 การออกแบบเพื่อให้ได้ความเร็วของการทำงานที่เหมาะสม

เนื่องจากการใช้ FPGA หรือสื่อกลางที่ช่วยในการสร้างชิ้นงานใด ๆก็ตามยังคงต้องมีการตัดสินใจในเรื่องเวลาการทำงานของตัวชิ้นงานเสมอโดย เป็นการพิจารณาเลือกระหว่างขนาดของพื้นที่ใช้งานภายในชิพที่จะทำให้มีเวลาในการทำงานได้ดีที่สุดกับสมรรถนะของชิ้นงาน ดังนั้นการเตรียมหรือเลือกลักษณะเฉพาะของ Storage element ในส่วนที่เป็นเซลล์พื้นฐานของ FPGA อาจจะเป็นสิ่งที่จำกัดขอบเขตทางเลือกต่างๆ ที่จะใช้ในการออกแบบซึ่งมีอยู่ 2 ลักษณะคือ

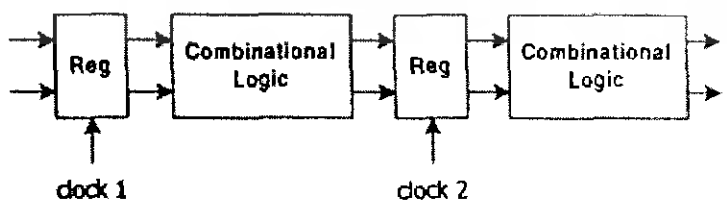
1) ระบบสัญญาณนาฬิกา (Clocked)

2) ระบบการกำหนดเวลา Self-timed

ดังมีรายละเอียดต่อไปนี้

1) ระบบสัญญาณนาฬิกา (Clocked)

สัญญาณนาฬิกาจะเป็นตัวกำหนดกฎเกณฑ์จังหวะการทำงานของทุกๆ เซลล์ให้ทำงานแบบซิงโครนัสกับสัญญาณนาฬิกาของระบบ ตัวอย่างของวงจรในรูปที่ 2.20 มีการทำงานแบบลำดับของวงจรคำนวณที่เป็นแบบส่งผ่านรีจิสเตอร์ในวงจรจะพบว่าช่วงเวลาของสัญญาณนาฬิกาเป็นสิ่งที่กำหนดการไหลของข้อมูลที่ผ่านระบบ ดังนั้นผลลัพธ์จากการคำนวณที่เสร็จสิ้นในช่วงหนึ่งๆ ด้วยวงจรที่เป็นคอมบิเนชั่นลอจิกจะถูกเก็บไว้ในรีจิสเตอร์ที่อยู่ตรงกลางก่อนที่จะส่งต่อไปให้ช่วงถัดไป ปัญหาสามารถที่จะเกิดขึ้นได้เมื่อมีการหน่วงเวลาบนสายสัญญาณที่อยู่ในระบบคือทำให้เกิดการเบี่ยงเบน (Skew) ของสัญญาณนาฬิกาส่งผลให้เกิดการเลื่อนเวลาในการทำงานของรีจิสเตอร์ ดังนั้นใน FPGA บางกลุ่มจึงได้มีการเตรียมระบบที่เป็นโครงข่ายพิเศษที่ครอบคลุมทั้งชิพไว้โดยเฉพาะเพื่อให้เกิดค่าของเวลาหน่วงน้อยที่สุดส่งผลให้การทำงานของระบบเป็นแบบซิงโครนัสมากที่สุดเท่าที่จะเป็นไปได้



รูปที่ ก.11 รูปแบบของ Register transfer model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าในกรณีที่เซลล์ของ FPGA ถูกใช้งานเป็นแบบ Smart-memory ที่อยู่ภายในระบบคอมพิวเตอร์ก็จะทำให้แบบแผนผังของระบบสัญญาณนาฬิกา (Clocked scheme) อาจจะเป็นเทคนิคที่ถูกเลือกมาใช้ได้ เพราะว่ามันจะยอมให้ไมโครโปรเซสเซอร์ทำการอ่านหรือเขียนบนจุดใดๆ ของวงจรที่สร้างขึ้นไว้ภายในได้โดยที่เซลล์ไม่ไปรบกวนการทำการคำนวณของระบบ เพราะเนื่องจากมันมีโครงสร้างสร้างเป็นแบบเซลล์อาร์เรย์จึงเป็นที่น่าสังเกตว่าการที่มีระบบสัญญาณนาฬิกาที่สามารถเลือกได้จะทำให้ผู้ใช้งานไม่ต้องกังวลเรื่องกำจัดสัญญาณรบกวนอันเนื่องมาจากการมีเวลาเหลื่อมกันของสัญญาณ (Timing hazard) และแนวทางการใช้สัญญาณนาฬิกาเพื่อให้วงจรสามารถทำงานได้อย่างถูกต้องมี 2 แนวทางคือ

1) การใช้ Single-phase clocking คือการใช้อุปกรณ์ที่มีการกระตุ้นการทำงานแบบ Master-slave storage element และมีสัญญาณนาฬิกาแบบ Single-phase ดังในรูปที่ ก.11 ที่สัญญาณนาฬิกาทั้งสองตัวต้องเป็นสัญญาณเดียวกัน รูปแบบเช่นนี้ได้มีการกำหนดเป็นกฎเกณฑ์ที่ใช้ในระบบ TTL แต่อย่างไรก็ตามอุปกรณ์ FPGA ก็สนับสนุนรูปแบบนี้ด้วยเช่นกัน

2) การใช้ Two-phase clocking เป็นการใช้สัญญาณนาฬิกาแบบ Two-phase ที่ไม่มีการเหลื่อมทับกัน (Overlap) ให้กับตัวแลชท์ การทำเช่นนี้ก็จะทำให้การไปประยุกต์ใช้ในวงจรในรูปที่ ก.11ได้ และการที่ไม่มีการเหลื่อมกันของสัญญาณนาฬิกาทั้งสองจะช่วยป้องกันการเคลื่อนย้ายข้อมูลมากกว่าหนึ่งช่วงในแต่ละวงรอบสัญญาณนาฬิกา ดังนั้นถ้าใน FPGA มีการเตรียมแลชท์ที่สามารถควบคุมได้ตรงจากสัญญาณนาฬิกาของระบบก็จะทำให้สามารถหลีกเลี่ยงการใช้สายสัญญาณนาฬิกาเพิ่มขึ้นมาสำหรับใช้ในการป้องกันสัญญาณนาฬิกาได้

2) ระบบการกำหนดเวลา Self-timed

ข้อกำหนดเรื่องเวลานี้หากในแต่ละเซลล์สร้างสัญญาณที่ชัดเจนของ “go” และ “done” ขึ้นมาโดยที่มีเส้นทางที่เป็นแบบขนานไปกับสัญญาณที่เป็นข้อมูลและมีการใช้การเปลี่ยนขอบ (Edge transition) ของตัวสัญญาณมาเป็นการทำ Handshake ซึ่งจะพบว่าเป็นแนวทางที่น่าสนใจมากเพราะว่ามันจะช่วยให้ลดปัญหาได้มากในเรื่องของเวลาในขณะที่มีความถูกต้องของการทำงานสูงมากอย่างไรก็ตามการสร้าง self-time FPGA มีต้นทุนที่สูงมากในเรื่องของขนาดพื้นที่ของเซลล์ เนื่องจากความซับซ้อนของเซลล์จะมีมากกว่าการสร้างด้วยเทคโนโลยีแบบซิงโครนัสในส่วนที่เป็นสายสัญญาณ “go” และ “done” และต้องมีจำนวนของเกทที่มากพอ ในการใช้งานหากจะต้องมีการแบ่งหรือแยกสายสัญญาณที่ตัวมัลติเพล็กเซอร์ ที่สำคัญคือฟังก์ชันการทำงานของมันจะต้องถูกคำนวณไว้ก่อนเพื่อเตรียมไว้สำหรับสิ่งที่จำเป็นต่อการควบคุมด้วย.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้