



ภาควิชาวิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาโท

ชื่อหัวข้อ การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ตโดยอุปกรณ์ FPGAs
Speech Compression with Wavelet Transform by FPGAs Device

ชื่อนักศึกษา

1. นายประจักษ์	วันดี	รหัสประจำตัว	42035308
2. นายอัศวิน	มีเงิน	รหัสประจำตัว	42035322
3. นายอาคม	เซ่งเถียน	รหัสประจำตัว	42035323

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา อาจารย์อำพล ทองระอา

อาจารย์ที่ปรึกษาร่วม อาจารย์กิติพงศ์ มะโน

คณะกรรมการสอบปริญญาโท	ลายมือชื่อ
1. อาจารย์อำพล ทองระอา	
2. อาจารย์กิติพงศ์ มะโน	
3. อาจารย์สุชิน อางหาญ	
4. อาจารย์ไพบุลย์ พวงวงศ์ตระกูล	
5. อาจารย์สุระชัย พิมพ์สวัสดิ์	

วัน/เดือน/ปีที่สอบ วันเสาร์ที่ 25 พฤศจิกายน พ.ศ. 2543 เวลา 11.00 น.

สถานที่สอบ ห้อง ค.310 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว
ลงนาม.....
(ผศ.วิสุทธิ์ อธิพรธรรม)
หัวหน้าภาควิชาครุศาสตร์วิศวกรรม
วันที่ 26 เดือน มกราคม พ.ศ. 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์

การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต

โดยอุปกรณ์ FPGAs

**SPEECH COMPRESSION WITH WAVELET TRANSFORM
BY FPGAs DEVICE**



นายประจักษ์ วันดี
นายอัศวิน มีเงิน
นายอาคม เช่งเถียน

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เลขหมู่.....
เลขทะเบียน 40178
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้เผยแพร่ประโยชน์ด้านการค้า
วัน, เดือน, ปี ส.ค. 2544

b. 41092531
i.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณีไปใช้

ปริญญานิพนธ์

เรื่อง การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดยอุปกรณ์ FPGAs
Speech Compression with Wavelet Transform by FPGAs Device

วัตถุประสงค์

1. เพื่อศึกษากระบวนการลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดยอุปกรณ์ FPGAs
2. เพื่อออกแบบวงจรลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดยอุปกรณ์ FPGAs
3. เพื่อสร้างวงจรลดขนาดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ต โดยอุปกรณ์ FPGAs
4. เพื่อนำไปใช้ในการบันทึกเสียงพูดได้จริง

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจหลักการลดขนาดข้อมูลเสียงพูดโดยใช้เวฟเล็ตได้
2. ได้วงจรถลดขนาดข้อมูลเสียงพูด โดยใช้การแปลงเวฟเล็ต
3. ได้เครื่องบันทึกข้อมูลเสียงพูด
4. ใช้ในงานบันทึกข้อมูลเสียงพูดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Speech Compression with Wavelet Transform by FPGAs Device	
Students	Mr.Prachak	Wandee
	Mr.Asavin	Meengen
	Mr.Arcom	Sengtien
Advisor	Mr.Amphon	Thongra-ar
Co-Advisor	Mr.Kitipong	Mano
Education Level	Bachelor of Science in industrial Education	
Program in	Electronics and Computer	
Academic	2543	

ABSTRACT

This thesis presents a design and implementation of speech compression with wavelet transform by FPGAs device. It uses wavelet theory for design to hardware. It divides signal or function and cut unused data which it will do with high frequency. In circuit uses sampling frequency at 16 KHz and keep coefficient of wavelet transform in external memory. It can keep 64 Kbytes for speech data. The result from simulation show that this method is correct.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่านขอขอบคุณอาจารย์อำพล ทองระอาและอาจารย์กิติพงศ์ มะโน สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม คณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่าน ที่ให้ความช่วยเหลืออนุเคราะห์เครื่องมือและอุปกรณ์ต่างๆที่ใช้ในการทำงาน รวมทั้งยังให้คำแนะนำ แนวความคิด ความรู้และทักษะในการทำโครงการครั้งนี้ อีกทั้งแนวทางในการแก้ไขปัญหาในการจัดทำปริญญานิพนธ์ ขอขอบพระคุณบิดา มารดาอันเป็นที่เคารพเป็นอย่างยิ่งที่ให้การสนับสนุนด้านงบประมาณทางการศึกษาและเป็นผู้ที่ให้กำลังใจด้วยดีตลอดมา ตั้งแต่อดีตจนถึงปัจจุบัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมา และความสำคัญของปริิณญานิพนธ์	1
1.2 ขอบเขตของปริิณญานิพนธ์	2
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 บทนำ	4
2.2 ทฤษฎีเวฟเล็ต	5
2.3 พื้นฐานของทฤษฎีเวฟเล็ต	8
2.4 ตัวแปลงเวฟเล็ตและคำจำกัดความ	10
2.5 การแปลงเวฟเล็ตผกผัน	11
2.6 การกระจายของพลังงานในเวฟเล็ต โดเมน	12
2.7 อนุกรมเวฟเล็ตต่อเนื่อง	13
2.8 อนุกรมเวฟเล็ตเต็มหน่วยเวลา	15
2.9 รูปแบบการแปลงเวฟเล็ตเต็มหน่วยเวลา	15
2.10 Multi Resolution Wavelet Transform	20
2.11 แนะนำภาษา VHDL	20
2.12 ข้อได้เปรียบของการใช้ VHDL	20
2.13 โครงสร้างภายในอุปกรณ์ FPGAs	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.14 ส่วนอินพุตและเอาต์พุตของอุปกรณ์ FPGAs	23
2.15 รายละเอียดการใช้งานของอุปกรณ์ FPGAs	24
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	26
3.1 ความคิดเบื้องต้น ในการสร้างอุปกรณ์บีบอัดข้อมูลเสียงโดยสมการเวฟเล็ตและ ประมวลผลด้วย FPGAs	26
3.2 วงจรในการสร้างอุปกรณ์บีบอัดข้อมูลเสียง โดยสมการเวฟเล็ตและประมวลผล ด้วย FPGAs	27
3.2.1 วงจรภายนอก	27
3.2.2 วงจรภายใน	33
3.3 วงจรแปลงสมการเวฟเล็ต DWT	37
บทที่ 4 การทดลอง และผลการทดลอง	45
4.1 ผลการทดลอง VHDL และ Schematic ของอุปกรณ์ FPGAs	45
4.1.1 ส่วนควบคุมสัญญาณแอดเดรส	45
4.1.2 ส่วนของวงจรตีโค้ดสัญญาณเวลาเพื่อไปแสดงผลยัง 7-Segment	46
4.1.3 ส่วนของการแปลง DWT และจัดเก็บข้อมูลในหน่วยความจำภายนอก	47
4.1.4 ส่วนของการแปลงกลับของสมการเวฟเล็ตผกผันของข้อมูลในหน่วย ความจำ	48
4.2 ผลการทดลองวัดสัญญาณต่างๆจากอุปกรณ์ภายนอก	49
4.2.1 การแปลงสัญญาณแอนาลอกเป็นดิจิตอลและแปลงกลับดิจิตอลเป็น สัญญาณแอนาลอก	49
4.2.2 การแปลงสัญญาณแอนาลอกเป็นดิจิตอลและแปลงกลับดิจิตอลเป็น สัญญาณแอนาลอกโดยผ่านการเก็บในหน่วยความจำภายนอก	50
4.2.3 ผลการทดลอง การลดรูปข้อมูลด้วยสมการเวฟเล็ต	51
บทที่ 5 บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา	53
5.1 บทสรุป	53

สารบัญ (ต่อ)

เรื่อง	หน้า
5.2 ปัญหาและแนวทางแก้ไข	54
5.3 แนวทางการพัฒนา	56
ภาคผนวก ก โปรแกรม	57
ภาคผนวก ข วงจรและแผนวงจรพิมพ์	62
ภาคผนวก ค รายละเอียดข้อมูลและคุณสมบัติของอุปกรณ์	66
บรรณานุกรม	75
ประวัติผู้แต่ง	76



สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 คุณสมบัติของ FPGAs ประเภทต่างๆ	22
ตารางที่ 2.2 ประมาณการนับเกตของเกตพื้นฐาน	23
ตารางที่ 2.3 โหมดต่างๆ ของการคอนฟิกูเรชัน	24
ตารางที่ 4.1 การนับแบบไบนารีและการนำไปแสดงผลออก 7-Segment	46



สารบัญรูป

รูป	หน้า
รูปที่ 2.1 เวฟเล็ตแม่	5
รูปที่ 2.2 รูปสัญญาณของ $y = \text{sine}(5x)$	6
รูปที่ 2.3 รูปสัญญาณของ $y = \text{Exp}(-0.5^x)$	6
รูปที่ 2.4 Scaling and translation of mother wavelet	7
รูปที่ 2.5 การแปลงเวฟเล็ตของสัญญาณ $f(x)$ ให้อยู่โดเมนของเวฟเล็ต g	9
รูปที่ 2.6 ขบวนการแปลงเวฟเล็ต	11
รูปที่ 2.7 ขบวนการแปลงกลับเวฟเล็ต	12
รูปที่ 2.8 Time Scale Resolution ของอนุกรมเวฟเล็ต	14
รูปที่ 2.9 กระบวนการ DWTS ที่มีการ Scaling เท่ากับสอง	16
รูปที่ 2.10 การแตกกระจายเวฟเล็ตด้วย DWTS block	17
รูปที่ 2.11 กระบวนการ IDWTS ที่มีค่าของการ Scaling เท่ากับสอง	18
รูปที่ 2.12 การรวมกลับเวฟเล็ตด้วย IDTWS Block	19
รูปที่ 3.1 ความคิดเบื้องต้นในการออกแบบวงจร Speech Compression	26
รูปที่ 3.2 วงจรปริโมโครโฟน	27
รูปที่ 3.3 วงจรโลว์พาสฟิลเตอร์	28
รูปที่ 3.4 ตำแหน่งขาต่างๆของไอซี DAC 0808	28
รูปที่ 3.5 วงจรใช้งานจริงของไอซี DAC 0808	29
รูปที่ 3.6 การต่อขาต่างๆใช้งานของไอซี ADC 908	30
รูปที่ 3.7 วงจรที่ใช้ในการติดต่อระหว่างอุปกรณ์ FPGAs กับอุปกรณ์และหน่วยความจำภายนอก	31
รูปที่ 3.8 บล็อกไดอะแกรมของวงจรภายในอุปกรณ์ FPGAs	33
รูปที่ 3.9 วงจรควบคุมสัญญาณอแดคเตรส	34
รูปที่ 3.10 การต่อวงจรรีเซ็ตการทำงาน	35
รูปที่ 3.11 วงจรกำเนิดสัญญาณนาฬิกา	35
รูปที่ 3.12 วงจรนับแสดงผลออกทาง 7-Segment	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูป	หน้า
รูปที่ 3.13 วงจรแปลงสมการเวฟเล็ต DWT	40
รูปที่ 3.14 โปรแกรมการแปลงเวฟเล็ต	41
รูปที่ 3.15 วงจรส่วนของการแปลงกลับสมการเวฟเล็ต IDWT	42
รูปที่ 3.16 โปรแกรมการแปลง Invert Wavelet	43
รูปที่ 3.17 บอร์ดอุปกรณ์ FPGAs ที่ต่อใช้งาน	44
รูปที่ 3.18 วงจรที่ทดสอบการทำงานทั้งหมด	44
รูปที่ 4.1 สัญญาณของส่วนควบคุมสัญญาณแอดเดรส	45
รูปที่ 4.2 สัญญาณของวงจรดีโคดเพื่อแสดงผลไปยัง 7-Segment	46
รูปที่ 4.3 สัญญาณส่วนของการแปลง DWT และจัดเก็บข้อมูลในหน่วยความจำภายนอก	47
รูปที่ 4.4 สัญญาณส่วนของการแปลงกลับสมการเวฟเล็ต	48
รูปที่ 4.5 สัญญาณอินพุตเทียบกับสัญญาณที่ผ่านการแปลงเป็นดิจิตอลและแปลงกลับแล้ว	49
รูปที่ 4.6 สัญญาณอินพุตเทียบกับสัญญาณที่อ่านออกมาจากหน่วยความจำภายนอก	50
รูปที่ 4.7 สัญญาณอินพุตเทียบกับสัญญาณเอาต์พุตที่ผ่านการแปลงสมการเวฟเล็ตแล้ว	51
รูปที่ ข.1 วงจรปริโมโครโฟน	63
รูปที่ ข.2 วงจรโลว์พาสฟิลเตอร์	63
รูปที่ ข.3 วงจรใช้งานจริงของไอซี DAC 0808	64
รูปที่ ข.4 การต่อขาต่างๆ ใช้งานของไอซี ADC 908	64
รูปที่ ข.5 วงจรที่ใช้ในการติดต่อระหว่างอุปกรณ์ FPGAs กับอุปกรณ์และหน่วยความจำภายนอก	65

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญญานิพนธ์

ในยุคของข้อมูลข่าวสารอย่างในปัจจุบันเทคโนโลยีสมัยใหม่ได้มีการพัฒนาไปอย่างกว้างขวาง ทั้งในด้านของคอมพิวเตอร์และในด้านของอิเล็กทรอนิกส์ ใครที่มีความรู้และสามารถเข้าถึงระบบการสื่อสารในรูปแบบต่างๆ ได้มากกว่าผู้อื่นก็มีโอกาสที่จะประสบความสำเร็จได้มากกว่าแน่นอน ข่าวสารไม่ว่าจะเป็นข้อมูลภาพหรือเสียงล้วนแต่มีประโยชน์และมีความสำคัญเป็นอย่างมาก ทั้งในด้านความบันเทิงและการติดต่อสื่อสารแต่ก็ยังคงมีปัญหาขึ้นมาเกี่ยวกับงานด้านการจัดเก็บข้อมูลที่มีข้อจำกัดในเรื่องปริมาณของข้อมูลภาพและเสียงที่มีขนาดใหญ่มาก จะทำให้เกิดปัญหาการส่งข้อมูลเหล่านั้นในระยะทางไกลๆ อีกทั้งยังจะทำให้เกิดการสูญเสียข้อมูลในระหว่างทำการส่งสัญญาณอีกด้วย ซึ่งนอกจากจะเป็นอุปสรรคในการสื่อสารข้อมูลแล้วยังเป็นปัญหาในเรื่องของการสิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูลอีกด้วย ดังนั้นในการจัดทำโครงการครั้งนี้ได้นำเสนอวิธีการลดรูปข้อมูลเสียงพูดโดยใช้สมการเวฟเล็ตมาทำการแปลงข้อมูลเสียงพูดให้อยู่ในรูปของวงจรอิเล็กทรอนิกส์ แล้วนำเอาโปรแกรม Xilinx มาใช้ในการออกแบบวงจรเพื่อที่จะสร้างวงจรอิเล็กทรอนิกส์ขึ้นมาและให้สามารถใช้งานได้จริง โดยเป็นการเขียนโปรแกรมและนำเอาตัวโปรแกรมไปสร้างเป็นวงจรเพื่อบรรจุลงในอุปกรณ์ FPGAs (Field Programmable Gate Arrays) ซึ่งอุปกรณ์ชนิดนี้มีความสามารถที่จะสร้างเป็นวงจรขนาดใหญ่ได้ อีกทั้งยังสามารถรวมและยุบวงจรที่มีขนาดใหญ่ ๆ ให้อยู่ภายในอุปกรณ์เพียงตัวเดียวโดยใช้โปรแกรม XACT Step ช่วยในการคำนวณโหลดลงในตัว FPGAs

ดังนั้นเพื่อความสมบูรณ์ในการสร้างโครงการนี้ได้มีการนำเอาทั้งส่วนของ Schematic และส่วนของภาษา VHDL มาใช้ในการออกแบบวงจรอิเล็กทรอนิกส์ซึ่งจะได้วงจรบีบอัดข้อมูลเสียงพูด ตามชื่อหัวข้อโครงการ การบีบอัดข้อมูลเสียงพูดด้วยการแปลงเวฟเล็ตโดยอุปกรณ์ FPGAs ซึ่งต้องต่อร่วมกับอุปกรณ์ภายนอกบ้างเพียงเล็กน้อยเพราะเป็นส่วนที่ไม่สามารถสร้างขึ้นภายในอุปกรณ์ FPGAs ได้ การนำเอาไปใช้งานจริงสามารถทำงานได้เหมือนอุปกรณ์อิเล็กทรอนิกส์ชนิดอื่นๆอีกทั้งยังสะดวกสบายประหยัดเนื้อที่ในการออกแบบวงจรด้วย

1.2 ขอบเขตของปริิญาานิพนธ์

1. สามารถลดขนาดข้อมูลเสียงพูดได้
2. สามารถจัดเก็บข้อมูลเสียงพูดลงในหน่วยความจำขนาด 64 Kbyte ได้
3. สามารถแปลงข้อมูลเสียงพูดที่ลดขนาดแล้วกลับเป็นสัญญาณเสียงได้
4. มีอัตราการสุ่มสัญญาณเสียงพูดไม่ต่ำกว่า 16 KHz
5. ใช้อุปกรณ์ FPGA เป็นตัวประมวลผลการบีบอัดข้อมูลเสียงพูด

1.3 เนื้อหาโดยสังเขป

ทฤษฎีเวฟเล็ตเป็นวิธีการเช่นเดียวกันกับระบบตัวเลขที่ใช้แทนจำนวนของสิ่งต่างๆ ได้ เช่น ผลเฉลยของสมการดิฟเฟอเรนเชียล, การเต้นของหัวใจ, การไหลเวียนของโลหิต, สภาพภูมิอากาศ เป็นต้น การแปลงเวฟเล็ตจึงเป็นเครื่องมือทางด้านคณิตศาสตร์อย่างหนึ่งที่สามารถนำไปประยุกต์ใช้งานได้กับสิ่งต่างๆ โดยมีวัตถุประสงค์เพื่อใช้อธิบายระบบหรือสัญญาณเหล่านั้นได้ดีกว่าวิธีการอื่นๆ ทฤษฎีการแปลงเวฟเล็ตสามารถอธิบายสิ่งหนึ่งสิ่งใดได้โดยการ แยกสิ่งเหล่านั้นออกเป็นส่วนประกอบย่อยๆที่มีความสัมพันธ์กัน โดยส่วนย่อยๆเหล่านั้นคือ เวฟเล็ตแม่ที่ถูกสเกลและเลื่อนตำแหน่ง

ขบวนการแยกสัญญาณออกเป็นส่วนๆหรือการหาสัมประสิทธิ์เราจะเรียกว่า " การแตกกระจายเวฟเล็ต " หรือเรียกว่า " การแปลงเวฟเล็ต " และในทางกลับกันก็ต้องมีการรวมกลับของสัญญาณ เพื่อให้ได้สัญญาณที่เหมือนเดิมเรียกว่า " การรวมกลับเวฟเล็ต(Wavelet reconstruction)" หรือ " การแปลงกลับเวฟเล็ต"

โดยเนื้อหาภายในปริิญาานิพนธ์ฉบับนี้ได้แบ่งออกเป็นบทต่างๆ เพื่อให้เกิดความสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทประกอบด้วยเนื้อหาที่สำคัญดังนี้

บทที่ 2 ทฤษฎีและหลักการ ซึ่งกล่าวถึงทฤษฎีของเวฟเล็ต การแปลงไปและแปลงกลับเวฟเล็ตรวมทั้งในส่วนของภาษา VHDL กับการออกแบบบิเล็กทรอนิกส์, โครงสร้างภายในของ FPGAs, รายละเอียดการใช้งานของอุปกรณ์ FPGAs, ขั้นตอนของการออกแบบและสังเคราะห์วงจร, ขั้นตอนการจำลองการทำงาน และ ขั้นตอนการลงโปรแกรมลงบนอุปกรณ์ FPGAs โดยใช้ซอฟต์แวร์ XDM ของบริษัทไซลิงค์ (Xilinx)

บทที่ 3 การออกแบบ และการสร้างซึ่งจะกล่าวถึง ความคิดเบื้องต้นในการออกแบบ วงจรบีบอัดข้อมูลเสียงพูด, ลักษณะการออกแบบ, วิธีการออกแบบ, และขั้นตอนการออกแบบ ผังการทำงานภายในอุปกรณ์ทั้งหมด

บทที่ 4 การทดลองและผลการทดลอง กล่าวถึงผลการทดลองโปรแกรม VHDL, การดาวน์โหลดโปรแกรมลงบนบอร์ด FPGAs และผลการทดลองหลังจากประกอบวงจรเสร็จ

บทที่ 5 ข้อสรุปและข้อเสนอแนะ กล่าวถึงปัญหา, แนวทางการแก้ไข, และแนวทางในการพัฒนาเพื่อสามารถนำไปประยุกต์ใช้ได้อย่างมีประสิทธิภาพ

ในภาคผนวกแสดงรายละเอียดของโปรแกรม และคุณสมบัติของอุปกรณ์ที่ใช้ในการจัดทำโครงการดังนี้

ภาคผนวก ก. โปรแกรมการทำงานของวงจรในส่วนของภาษา VHDL

ภาคผนวก ข. วงจรและแผ่นวงจรพิมพ์

ภาคผนวก ค. รายละเอียดข้อมูลและคุณสมบัติของอุปกรณ์



บทที่ 2

ทฤษฎีและหลักการ

2.1 บทนำ

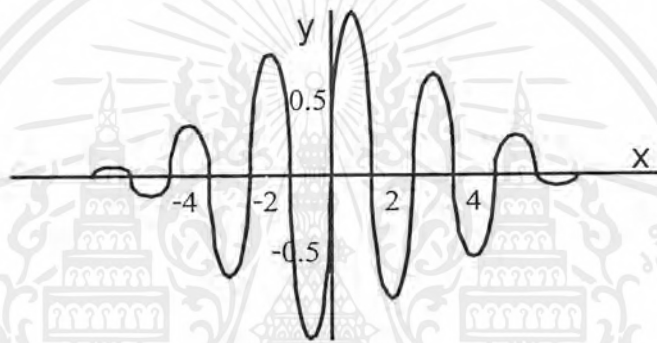
สัญญาณหรือฟังก์ชันใด ๆ ก็ตามสามารถจะแสดงให้อยู่ในรูปของการรวมเชิงเส้นของฟังก์ชันมูลฐาน ต่างๆ ได้ ซึ่งทำให้สามารถทำการวิเคราะห์หรือแก้ปัญหาได้อย่างสะดวก สามารถยกตัวอย่างได้เช่น การใช้ฟังก์ชันไซน์และโคไซน์เป็นฟังก์ชันมูลฐานในการวิเคราะห์ในระบบฟูเรียร์ เป็นต้น

การวิเคราะห์เวฟเล็ทก็เป็นการแสดงฟังก์ชันใด ๆ ให้อยู่ในรูปของการรวมเชิงเส้นของฟังก์ชันมูลฐานเช่นเดียวกัน แต่เพราะด้วยลักษณะพิเศษของฟังก์ชันมูลฐานเวฟเล็ทซึ่งมีลักษณะของการออสซิลเลท ตามแกนแนวนอนและมีแอมพลิจูดที่ลดลงสู่ศูนย์ทั้งทางด้านบวกและด้านลบอย่างรวดเร็วต่างกับฟังก์ชันไซน์และโคไซน์ซึ่งแผ่ตามแกนแนวนอนไปสู่อนันต์ จึงทำให้เหมาะที่จะใช้ในการวิเคราะห์สัญญาณที่มีลักษณะไม่คงที่ ลักษณะของความไม่ต่อเนื่องและลักษณะของปลายแหลมคมได้เป็นอย่างดี เนื่องจากในโดเมนของเวฟเล็ทจะประกอบไปด้วยสัมประสิทธิ์ของการกระจายหรือตัวคูณของฟังก์ชันมูลฐานเวฟเล็ทที่มีขนาดเล็กและเข้าสู่ศูนย์อย่างรวดเร็ว ซึ่งในรูปโดเมนของฟูเรียร์ของฟังก์ชันดังกล่าวจะต้องประกอบด้วยสัมประสิทธิ์ของการกระจายหรือตัวคูณของฟังก์ชันไซน์และโคไซน์เป็นจำนวนมากมายถึงอนันต์ แต่เมื่อมองในแง่ของการแปลงเพื่อเข้ารหัสแล้วการแปลงเวฟเล็ทจึงเป็นการเปลี่ยนที่มีประสิทธิภาพที่สูงมากเมื่อใช้กับสัญญาณที่มีลักษณะไม่คงที่ ลักษณะของความไม่ต่อเนื่องและลักษณะของปลายแหลมคมและประโยชน์ที่สำคัญอีกประการหนึ่งของการแปลงเวฟเล็ทก็คือ ความเปลี่ยนแปลงที่เกิดขึ้นภายในโดเมนของเวฟเล็ทจะมีผลกับทำให้เกิดการเปลี่ยนแปลงในโดเมนเดิมเป็นช่วง (Interval) เท่านั้น ซึ่งเป็นผลเนื่องมาจากวิธีการวิเคราะห์แบบมัลติเรโซลูชัน

2.2 ทฤษฎีเวฟเล็ต

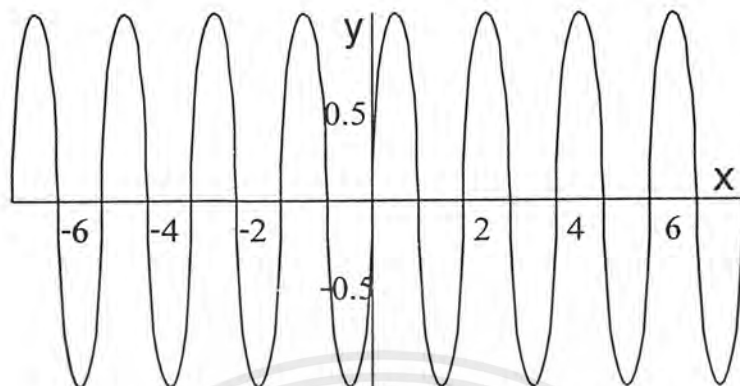
ทฤษฎีเวฟเล็ต เป็นกระบวนการทางคณิตศาสตร์ที่ใช้อธิบายถึงโมเดลของสัญญาณระบบ หรือกระบวนการ แล้วกำหนดให้เซตของสัญญาณที่มีลักษณะเฉพาะเป็นองค์ประกอบพื้นฐานโดยจะเรียกว่า "เซตของเวฟเล็ต" สมาชิกของเวฟเล็ตเซตจะมีคุณสมบัติที่สำคัญคือเป็นสัญญาณที่จะเกิดขึ้นมาจากต้นแบบเดียวกัน ต้นแบบของสัญญาณในเซตของเวฟเล็ตนี้เรียกว่า "เวฟเล็ตแม่"

เวฟเล็ต จะมีคุณสมบัติของสัญญาณที่เกิดในช่วงเวลาที่สั้นๆ หรือกล่าวคือ เวฟเล็ตเป็นสัญญาณที่เกิดขึ้นอย่างต่อเนื่องและมีขนาดที่ลดลงสู่ศูนย์ในทั้งสองด้าน

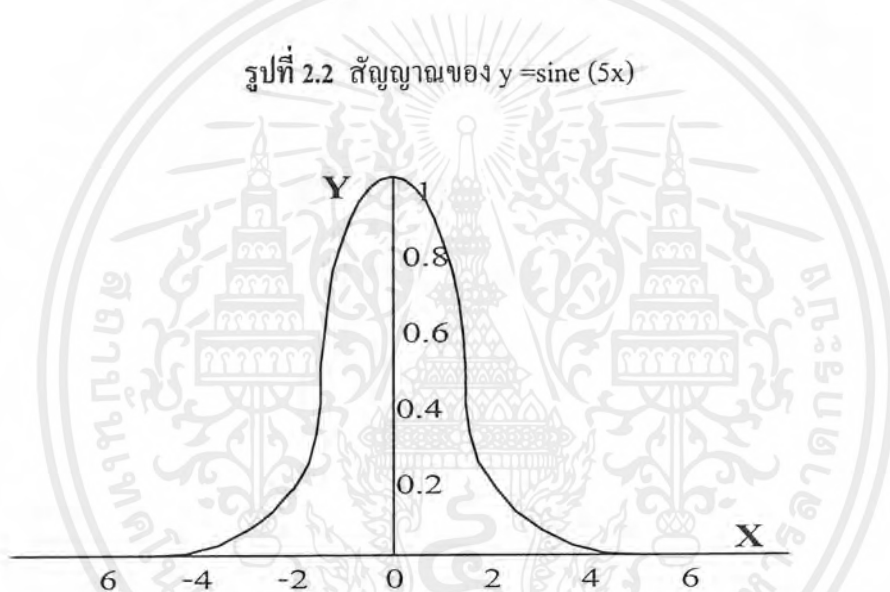


รูปที่ 2.1 เวฟเล็ตแม่

สัญญาณที่ 2.1 จะประกอบด้วยลักษณะทั้งสองด้านคือการเกิดขึ้นอย่างต่อเนื่อง ซึ่งเกิดจากสัญญาณไซน์ดังในรูปที่ 2.2 และการเข้าสู่ศูนย์ของสัญญาณในทั้งสองด้านจาก Window function ดังรูปที่ 2.3 เมื่อนำเอา function ทั้งสองมาคูณเข้าด้วยกันจะได้เป็นสัญญาณของเวฟเล็ตดังในรูปที่ 2.1



รูปที่ 2.2 สัญญาณของ $y = \sin(5x)$



รูปที่ 2.3 สัญญาณของ $Y = \text{Exp}(-0.5^2)$

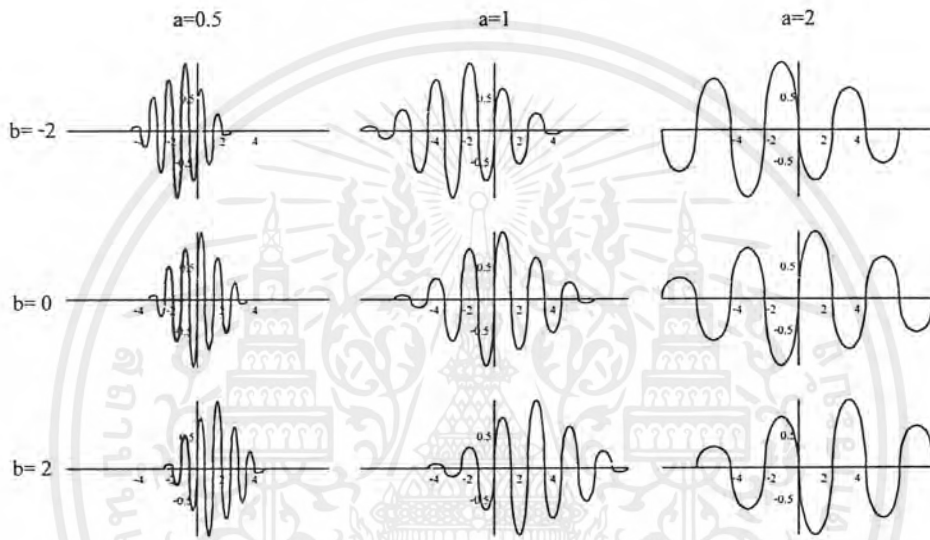
จากเวฟเล็ตแม่ที่ได้ จะถูกนำไปสร้างเป็นสมาชิกตัวอื่นๆ ในเซตของเวฟเล็ตที่ต้องการนำไปอธิบายถึงสัญญาณ กระบวนการ หรือระบบต่างๆ โดยสมาชิกอื่นๆ นี้ได้มาจากเวฟเล็ตแม่ที่ถูก " สเกล " (Scaling :a) และ " เลื่อนตำแหน่ง "(Translator :b) ไปตามแกนของเวลา ถ้ากำหนด $g(t)$ เป็น function ของ Mother wavelet แล้วการสเกลด้วยพารามิเตอร์ของ "a" และ การเลื่อนตำแหน่งด้วยพารามิเตอร์ของ "b" จะสามารถหาได้จากสมการที่ 2.1

$$g_{h,a}(t) = g(t-b)/a \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ a คือ ค่าอัตราส่วนของการสเกล
 b คือ การเลื่อนตำแหน่ง

ซึ่งการทำ Scaling และ Translation ตามแกนของเวลานี้ จะรวมเรียกว่า " Affine Operation " (การทำ Linear Mapping ร่วมกับการทำ Translation) และเพื่อสัญญาณที่ถูก " สเกล " และมีค่าของสัญญาณ ที่เท่าเดิม จึงจะต้องทำการ Normalization สัญญาณที่ได้



รูปที่ 2.4 การสเกลและการเปลี่ยนแปลงของเวฟเล็ดแม่

จากรูปที่ 2.4 จะสามารถสังเกตเห็นว่า ลักษณะจำนวนลูกคลื่นสัญญาณนั้นจะเหมือนกันทั้งหมด สมาชิกแต่ละตัวจะแตกต่างกันที่ ขนาดและตำแหน่งทางแกนเวลาเท่านั้น

2.3 พื้นฐานของทฤษฎีเวฟเล็ต

ทฤษฎีเวฟเล็ตก็เป็นวิธีการอย่างหนึ่งเช่นเดียวกับระบบตัวเลขที่ใช้แทนจำนวนสิ่งของต่างๆได้ เช่น ผลเฉลยของสมการดิฟเฟอเรนเชียล การเดินของหัวใจ การไหลเวียนของโลหิต สภาพภูมิอากาศ เป็นต้น

ทฤษฎีการแปลงเวฟเล็ต เป็นเครื่องมือทางด้านคณิตศาสตร์อย่างหนึ่งที่สามารถนำไปประยุกต์ใช้งานได้กับสิ่งต่างๆ โดยมีวัตถุประสงค์ที่จะใช้อธิบายระบบ หรือสัญญาณเหล่านั้นได้ดีกว่าวิธีการอื่นๆ การที่จะนำทฤษฎีของเวฟเล็ตไปใช้งาน จึงอยู่บนพื้นฐานที่ว่า " การนำมาใช้งานนั้นมีประสิทธิภาพเพียงใด " ซึ่งจะอธิบายสิ่งหนึ่งสิ่งใดได้โดยการ แยกสิ่งเหล่านั้นออกเป็น ส่วนประกอบย่อยๆที่มีความสัมพันธ์กัน โดยส่วนย่อยๆเหล่านั้นคือ เวฟเล็ตแม่ที่ถูกสเกล แล้วทำการเลื่อนตำแหน่ง

ขบวนการที่ทำการแยกสัญญาณออกเป็นส่วนๆ หรือการหาค่าสัมประสิทธิ์นี้เราจะเรียกว่า " การแตกกระจายเวฟเล็ต " หรือ " การแปลงเวฟเล็ต " และในทางกลับกันก็ต้องมีการรวมกลับของสัญญาณ เพื่อให้ได้สัญญาณเดิมเราจะเรียกว่า " การรวมกลับเวฟเล็ต " หรือ " การแปลงกลับเวฟเล็ต "

การแปลงเวฟเล็ตอย่างต่อเนื่องของฟังก์ชัน $f(x)$ จะสามารถแสดงด้วยสมการทางคณิตศาสตร์คือ

$$Wg [f(x)](a,b) = |a|^{-1/2} \int f(x)g^*((x-b)/a)dx \quad (2.2)$$

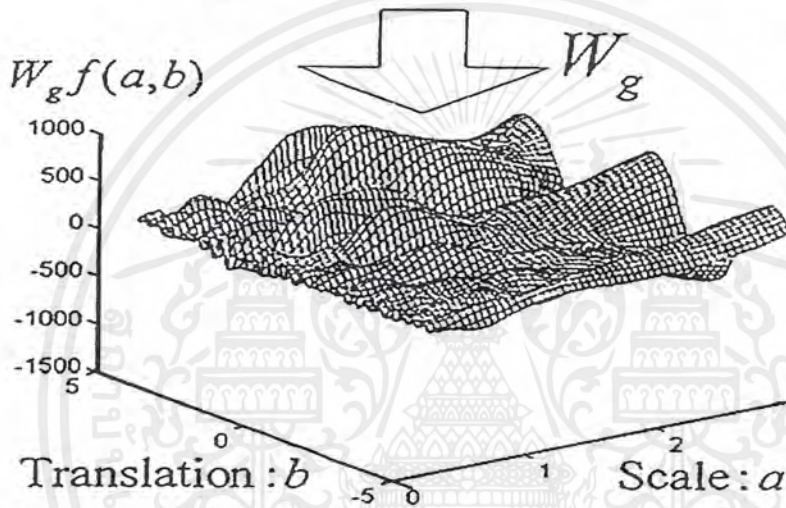
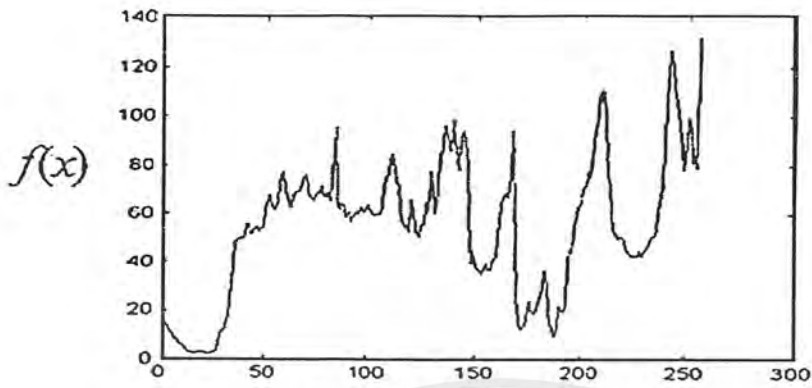
เมื่อ $g(x)$ เป็นฟังก์ชันของเวฟเล็ตแม่ และ $g^*(x)$ คือ Complex conjugate ของ $g(x)$

a Scaling

b Translation

$Wg f(x)(a,b)$ แทนค่าของสัมประสิทธิ์ที่ได้

จากขั้นต้นจะเห็นว่าการแปลงเวฟเล็ตจะเป็นการส่งผ่านฟังก์ชัน ที่มีตัวแปรอิสระ x ไปยังปฐมภูมิของฟังก์ชันที่เป็น 2 มิติของตัวแปรอิสระคือ a และ b เรียงตามลำดับ ค่าสัมประสิทธิ์ของเวฟเล็ตที่ a,b นี้จะ แสดงถึง ค่าสหพันธ์ระหว่างเวฟเล็ตแม่ที่สเกล a และตำแหน่ง b กับสัญญาณอินพุต f ที่ตำแหน่งเดียวกันโดยถ้าสัญญาณที่ได้ทั้งสองนี้มีความเหมือนหรือใกล้เคียงกันมากค่าสัมประสิทธิ์ที่ได้จะมีค่ามากตามไปด้วย โดยสัมประสิทธิ์ที่ได้ $Wg [f(x)](a,b)$ จะเรียกว่า " โดเมนของเวฟเล็ต (Wavelet Domain) "



รูปที่ 2.5 การแปลงเวฟเล็ตของสัญญาณ $f(x)$ ให้อยู่โดเมนของเวฟเล็ต g

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ตัวแปลงเวฟเล็ดและคำจำกัดความ

ในการกำหนดฟังก์ชันที่จะสามารถนำไปใช้เป็นเวฟเล็ดแม่ได้นั้นจะต้องมีคุณสมบัติของการแกว่งไป-มา และการลู่เข้าศูนย์ โดยจะฟังก์ชันเป็นที่สามารถหาพลังงานได้จะใช้เป็นเวฟเล็ดได้เมื่อ

$$Cg = \int_{-\infty}^{\infty} |G(w)|^2 |w| dw < \infty \quad (2.3)$$

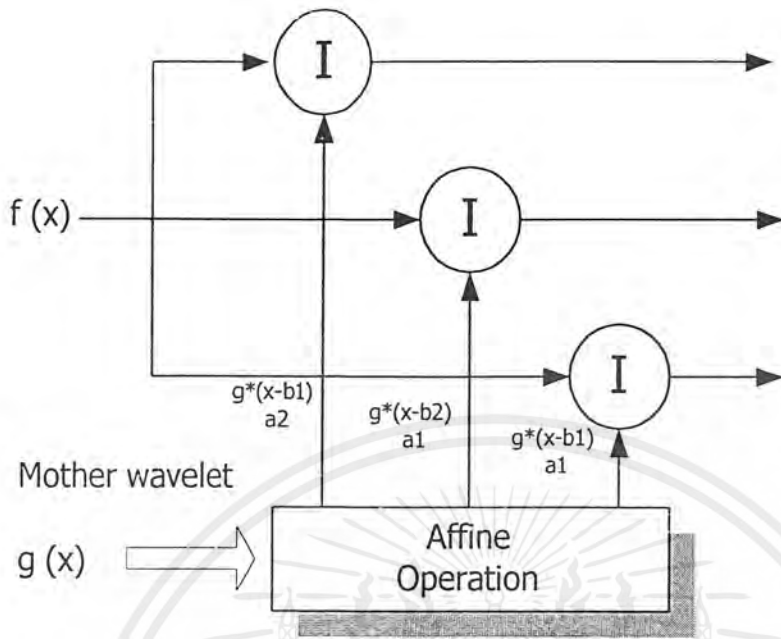
เมื่อ $G(w)$ Fourier transform ของ g

Cg ค่าพลังงานของ g

การที่ค่าพลังงานของเวฟเล็ดแม่ที่มีค่าเป็นหนึ่งหน่วยจะทำให้เกิดคุณสมบัติที่สำคัญอย่างหนึ่งคือ ค่าพลังงานของสัมประสิทธิ์ที่ได้ในเวฟเล็ด โดเมนจะเท่ากับค่าพลังงานของสัญญาณที่อยู่ในโดเมนของเวลา หรือปฐมภูมิ และยังทำให้ค่าสัมประสิทธิ์เวฟเล็ดที่ (a, b) ใดๆ สามารถที่จะนำมาเปรียบเทียบกันได้ สำหรับการแปลงเวฟเล็ดอย่างต่อเนื่องนั้น คุณสมบัติที่ต้องการฟังก์ชันที่นำมาเป็นเวฟเล็ดแม่ จะมีเพียงดังที่กล่าวข้างต้น ซึ่งการเลือกใช้เวฟเล็ดขึ้นอยู่กับการประยุกต์ใช้

เซตของการแปลงเวฟเล็ดนี้จะสามารถเป็นตัวแทนของสัญญาณและจากการแปลงกลับเวฟเล็ดแม่ตัวเดิมก็สามารถใช้ในการแปลงกลับได้

ค่าสัมประสิทธิ์เวฟเล็ดเปรียบเสมือนตัวประกอบย่อยของสัญญาณ ในการหาค่าสัมประสิทธิ์ของสัญญาณส่วนประกอบนั้นจะถูกฉายไปยังส่วนประกอบย่อย ซึ่งผลลัพธ์ที่ได้ก็คือค่าจำนวนจริงหรือจำนวนเชิงซ้อนเรียกว่าค่าสัมประสิทธิ์เวฟเล็ดของสัญญาณหรือฟังก์ชัน f ที่มีความสัมพันธ์กับเวฟเล็ดแม่ g



รูปที่ 2.6 ขบวนการแปลงเวฟเล็ต

นอกจากที่กล่าวมาแล้วยังมีการเปรียบเทียบสัญญาณที่เรียกว่า (Correlation) เป็นการเปรียบเทียบว่าสัญญาณทั้งสองมีความเหมือนกันมากเพียงใดสัญญาณ f จะถูกนำมาเปรียบเทียบกับสัญญาณเวฟเล็ตแม่ g ที่ถูกสเกลและถูกเลื่อนออกไป แสดงตามรูปที่ 2.6 Correlation จะแสดงด้วย I ที่อยู่ภายในวงกลม

ในการแปลงตามรูปที่ 2.6 นั้น เรียกอีกอย่างหนึ่งว่า Analysis filter โดยกระบวนการดังกล่าวจะทำการแตกกระจายสัญญาณออกเป็นส่วนย่อย ในการแปลงกลับเวฟเล็ตนั้นสัญญาณเหล่านี้จะถูกนำกลับมารวมกันอีกครั้งหนึ่ง

2.5 การแปลงกลับเวฟเล็ต

การแปลงกลับเวฟเล็ตเป็นการทำเพื่อให้ฟังก์ชันเดิมออกมา ซึ่งจะแสดงตามรูปที่ 2.7 และในการแปลงกลับเวฟเล็ตจะแสดงตามสมการ

$$W^1 g : Wgf(a,b) \rightarrow f(x)$$

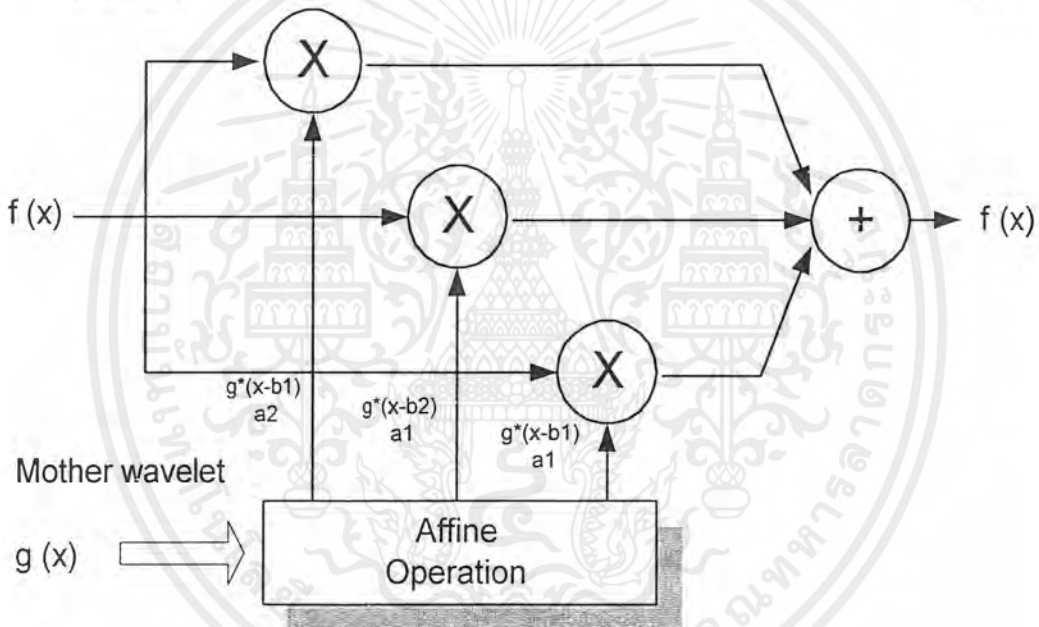
$$f(x) = (1/c) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Wgf(a,b) (1/\sqrt{|a|}) g((x-b)/a) da db / a^2 \tag{2.4}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแปลงกลับเวฟเล็ทจะสร้างฟังก์ชันสัญญาณต้นฉบับขึ้นมาจากการรวมกันของสัญญาณเวฟเล็ทแม่ g คูณกับค่าถ่วงน้ำหนัก(ค่าสัมประสิทธิ์เวฟเล็ท) กรณีที่การแปลงเวฟเล็ทนี้ใช้เวฟเล็ทแตกต่างไปจากตัวของเวฟเล็ทแม่ที่ใช้ตอนการแปลงเวฟเล็ท ผลลัพธ์ที่ได้จะเป็นสัญญาณหรือฟังก์ชันที่มีความแตกต่างกันออกไปจากฟังก์ชันของต้นฉบับเดิม

$$f(x) = (1/c) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Wgf(a,b) (1/\sqrt{|a|}) g((x-b)/a) da db/a^2 \neq f(x)$$

$$f(x) \neq g(x) \tag{2.5}$$



รูปที่ 2.7 ขบวนการแปลงกลับเวฟเล็ท

2.6 การกระจายของพลังงานในเวฟเล็ทโดเมน

จากการที่สัมประสิทธิ์ของเวฟเล็ทเป็นตัวแทนของสัญญาณได้รูปแบบหนึ่งแล้ว มันจะมีคุณสมบัติเฉพาะตัวเช่นเดียวกันสัญญาณนั้นๆ คุณสมบัติที่สำคัญของสัญญาณก็คือค่าพลังงานของสัญญาณนั้นๆนั่นเอง ค่าพลังงานของสัญญาณจะแทนด้วย (Ef) และจะสามารถคำนวณได้จากสมการ

$$Ef = \int_{-\infty}^{\infty} |f(t)|^2 dt \tag{2.6}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ef คือ ค่าพลังงานของสัญญาณ

$f(t)$ คือ สัญญาณ

ใน Wavelet transform domain ค่าความถี่ซึ่งหมายถึงการสเกลค่าพลังงานไปตามสมการ

$$\Delta Ef = |Wgf(a,b)|^2 dadb/a^2 \quad (2.7)$$

และค่าพลังงานรวมเป็น

$$Ef = (1/c) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} Wgf(a,b) (1/|a|) g((x-b)/a) dadb/a^2 \quad (2.8)$$

2.7 อนุกรมเวฟเล็ตต่อเนื่อง (Continuous Time Wavelet Series)

ในอนุกรมเวฟเล็ตต่อเนื่องจำนวนค่าสัมประสิทธิ์ที่ได้จะมีจำนวนที่จำกัด(ในกรณีของเวฟเล็ตต่อเนื่องจะมีค่าเป็นอนันต์)ค่าสัมประสิทธิ์จะแบ่งออกเป็นส่วนๆตามค่าของสเกลที่กำหนดโดยตัวเลขจำนวนเต็ม m และค่าของ Translater จะถูกกำหนดโดยตัวเลขจำนวนเต็ม n ดังนั้นค่า Scale(a) และค่า Translater (b) จะเป็น

$$a = a_0^m \quad (2.9)$$

$$b = nb_0 a_0^m \quad (2.10)$$

เมื่อ a_0 คือ ช่วงของ Scale

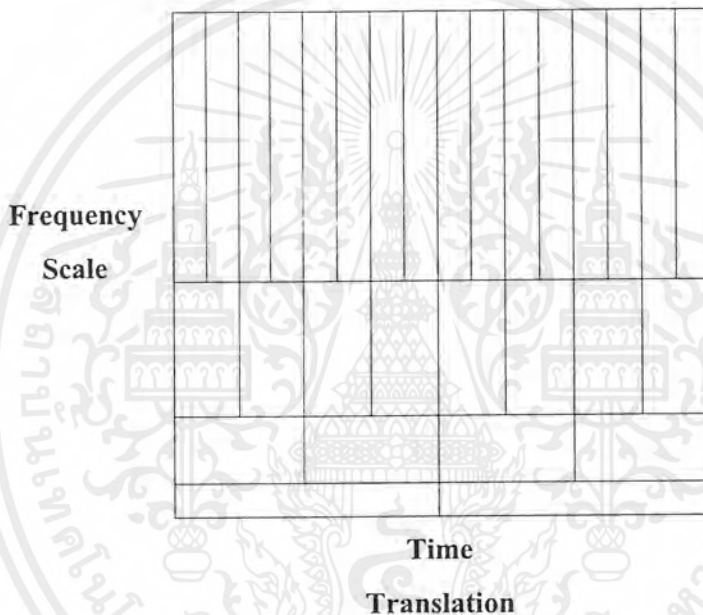
b_0 คือ ช่วงของ Translater

เมื่อนำมาแทนสมการ 2.3 แล้ว สมการของอนุกรมเวฟเล็ตต่อเนื่อง จะได้เป็น

$$Wgf(m,n) = a_0^{-m/2} \int_{-\infty}^{\infty} f(x)g(a_0^{-m} x - n b_0) dx \quad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากสมการข้างต้นนั้นยังใช้การ Intergrate อยู่เนื่องมาจากสัญญาณ $f(x)$ และเวฟเล็ตแม่ $g(x)$ นั้นยังเป็นเวลาแบบต่อเนื่องและเช่นเดียวกับการแปลงเวฟเล็ตต่อเนื่อง ค่าของสัมประสิทธิ์ก็จะเป็นตัวแทนของสัญญาณ Wavelet Domain แต่ในเวฟเล็ตจะใช้แต่ค่าของสเกลที่อยู่ในช่วงบวกเท่านั้น



รูปที่ 2.8 Time Scale Resolution ของอนุกรมเวฟเล็ต

จากตัวอย่างรูปที่ 2.4 ที่ผ่านมามีค่าของสเกลมีค่ามากเวฟเล็ตที่ได้จะเป็นเวฟเล็ตแม่ ที่ถูกยืดออกตามแกนเวลา ซึ่งคาบเวลายาวนานขึ้น จะทำให้ความละเอียดตามแกนเวลาลดลง ดังแสดงตามรูปที่ 2.8 และสำหรับที่ค่าของสเกลมีค่าน้อย เวฟเล็ตที่ได้จะมีขนาดลดลงตามแกนของเวลา คาบเวลาที่ลดลงมีผลทำให้ความละเอียดตามแกนของเวลามากขึ้น แต่ความละเอียดทางแกนความถี่จะลดลง ดังช่องในส่วนบนของรูป

และนอกจากนี้รูปที่ 2.8 ยังแสดงถึงการเกิดและลำดับของค่าสัมประสิทธิ์เวฟเล็ตด้วย จากที่ได้กล่าวมาแล้วก่อนหน้านี้ ช่องสี่เหลี่ยมแต่ละช่องจะเป็นตัวแทนของเวฟเล็ตแม่ที่ scale และ translation ต่างๆกัน ซึ่งจะนำไปสู่ค่าของค่าสัมประสิทธิ์เวฟเล็ตนั้นๆ ค่าของเวฟเล็ตที่ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัมประสิทธิ์เวฟเลตระดับของสเกลที่มีค่ามากก็จะใช้เวลาในการประมวลผลนานจากเวฟเลตแม่ที่ยืดออกนั่นเอง ดังนั้นค่าสัมประสิทธิ์ที่สัมพันธ์กับระดับของสเกลที่มากขึ้น จะได้จำนวนสัมประสิทธิ์ต่อหน่วยเวลาที่น้อยกว่า สัมประสิทธิ์ที่ค่าของ scale น้อยๆลงไป เนื่องจากขนาดของเวฟเลตแม่ที่สั้นเข้าจึงเวลาในการประมวลผลน้อยลง และเมื่อพิจารณาตามแกนของเวลาจะพบว่าค่าของสัมประสิทธิ์ที่ระดับสเกลที่มากจะมีค่าซ้ำเดิม ในขณะที่สัมประสิทธิ์ที่ระดับสเกลน้อยๆจะเปลี่ยนไปเรื่อยๆ ดังนั้นเพื่อเป็นการลดการซ้ำของสัมประสิทธิ์ที่เกิดขึ้น จึงต้องตัดส่วนที่ซ้ำกันออกไป ซึ่งเรียกว่า Sub Sampling นั่นเอง

2.8 อนุกรมเวฟเลตเต็มหน่วยเวลา (Discret Time Wavelet series)

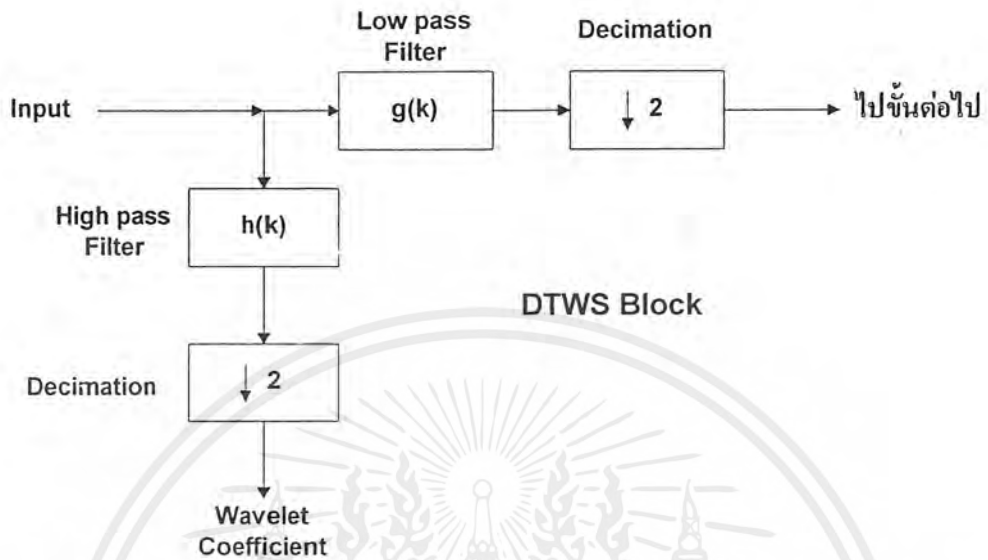
นอกเหนือจากการแปลงเวฟเลตที่มีค่าของ Scale และ Translation แบบเต็มหน่วยแล้ว ยังมีค่าของแกนเวลาที่สามารถเป็นแบบเต็มหน่วยได้เช่นกันเช่น ลำดับของตัวเลข สามารถนำมาใช้กับเวฟเลตได้ ซึ่งเรียกว่าเป็นอนุกรมเวฟเลตเต็มหน่วยเวลา (DTWS) การแปลงแบบ DTWS มีสมการของการแปลงดังนี้

$$W_h f(m,n) = a^{-m/2} \sum_{k=-\infty}^{\infty} f(k)h(a_0^{-m}k - nb_0) \quad (2.12)$$

สมการข้างต้น $h(k)$ แทนตัวเวฟเลตแม่เพื่อให้สอดคล้องกับทฤษฎีการกรองสัญญาณ และจากการที่ $h(k)$ เป็นแบบเต็มหน่วยเวลา ดังนั้นที่ค่าของ k ที่ไม่เป็นจำนวนเต็ม ค่าของ $h(k)$ ก็จะได้การประมาณค่า

2.9 รูปแบบการแปลงเวฟเลตเต็มหน่วยเวลา

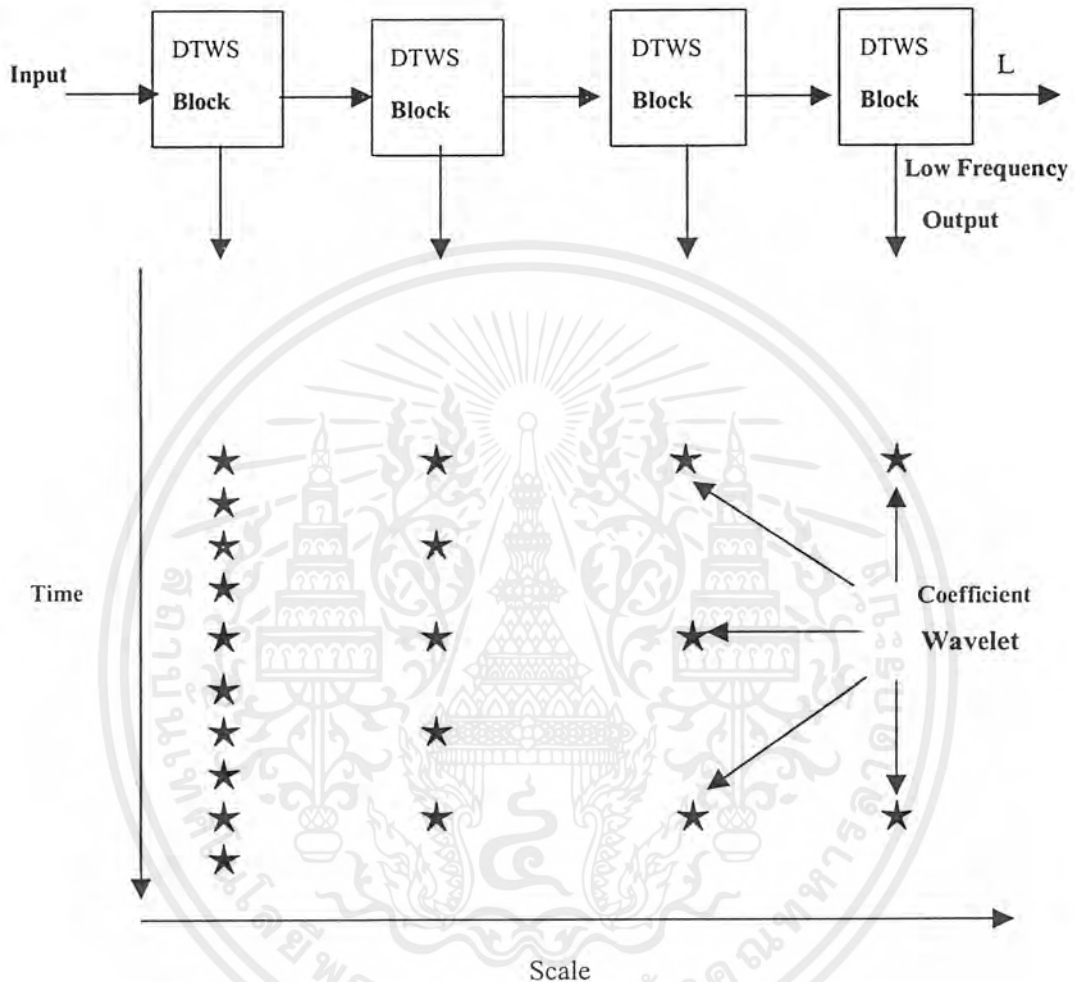
รูปแบบการแปลงเวฟเลตแบบเต็มหน่วยเวลาจะประกอบด้วยค่าใน Time Domain และ Scale Translation Domain (Wavelet Domain) จะเป็นจำนวนเต็มหน่วยที่ค่าอัตราส่วนการย่อของแกนเวลา (scale) เท่ากับสอง แล้วการทำ scaling ด้วยค่าอัตราส่วนของสองนี้สามารถทำได้ง่าย และมีประสิทธิภาพโดยการตัด sampling ของข้อมูลนั้นออกไปครึ่งหนึ่ง วิธีการนี้เรียกว่า Decimation หรือ Subsampling ด้วยค่าของสองรูปแบบทั่วไปของการแปลงเวฟเลตที่มีค่าของการย่อขนาดเป็นค่ากำลังสองก็จะได้อีกคือ



รูปที่ 2.9 กระบวนการ DTWS ที่มีการ Scaling เท่ากับสอง

จากขบวนการตามรูปที่ 2.9 สามารถนำไปใช้สำหรับการแปลงเวฟเล็ตแบบ Multiresolution, Orthogonal และ Perfect Reconstruction – Quadrature Mirror Filter (PRQMF) ซึ่งในแต่ละแบบจะมีคุณสมบัติของการ Filter ที่แตกต่างกัน

ตัวกรองจะถูกแบ่งเป็นตัวกรองความถี่ต่ำผ่านและตัวกรองความถี่สูงผ่านที่กำหนดด้วยค่าของสัมประสิทธิ์ $g(k)$ และ $h(k)$ ตามลำดับ ซึ่งค่าสัมประสิทธิ์ของตัวกรองนี้ก็จะสัมพันธ์กับเวฟเล็ตแม่ที่ใช้ร่วมกับการแปลงเวฟเล็ตนั่นเอง ขบวนการของการแตกกระจายเวฟเล็ตจะแสดงตามรูปที่ 2.10

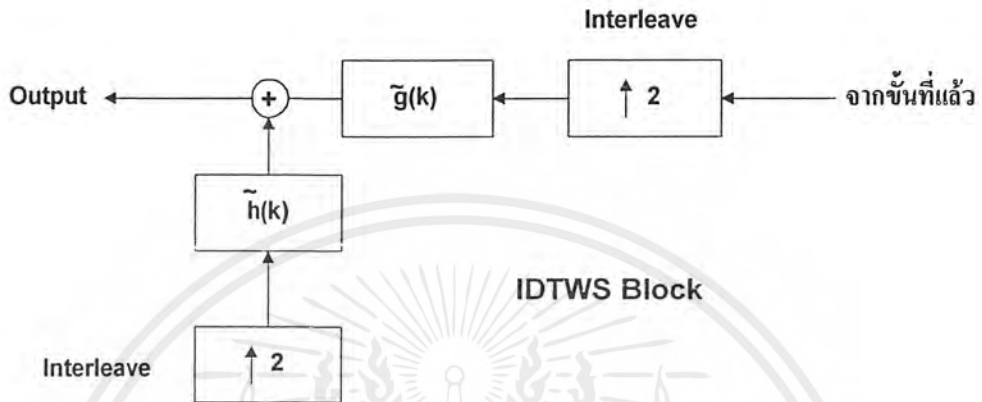


รูปที่ 2.10 การแตกกระจายเวฟเลตด้วย DTWS block

ขบวนการทั้งหมดก็จะเกิดขึ้นมาจากการนำเอาบล็อกย่อยๆ ของ DTWS ที่มีค่าของการ scaling เท่ากับสองมาต่อเข้าด้วยกัน สัญญาณเอาต์พุตที่ได้จากบล็อกแรก ในส่วนของตัวกรองความถี่ต่ำผ่านจะไปเป็นสัญญาณอินพุตของส่วนต่อไป

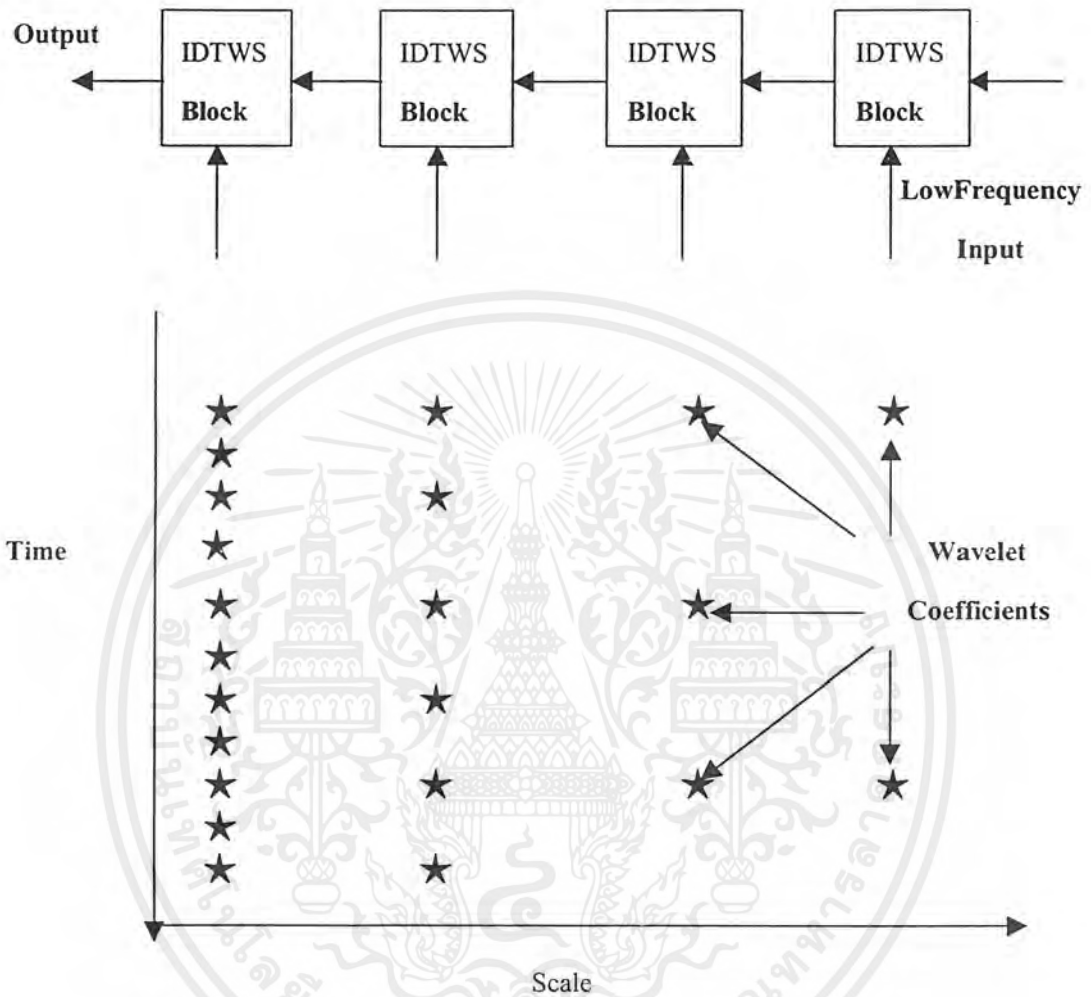
จากรูปที่ 2.10 จะเห็นได้ว่าสัญญาณอินพุตถูกป้อนเข้าทางด้านซ้ายมือค่าสัมประสิทธิ์เวฟเลตจะได้ออกมาทางด้านล่าง ซึ่งมีจำนวนลดลงเรื่อยๆ และสุดท้ายส่วนของสัญญาณความถี่ต่ำที่เหลือออกมาทางด้านขวามือนั่นเอง สัญญาณและค่าสัมประสิทธิ์ที่ได้ออกมาจะเป็นตัวแทนของ

สัญญาณใน Time domain นั้นเองการแปลงกลับเวฟเล็ตหรือ “การรวมกลับเวฟเล็ต” แบบเต็มหน่วยเวลา สามารถทำตามวิธีตามรูปที่ 2.11 และ 2.12



รูปที่ 2.11 กระบวนการ IDTWS ที่มีค่าของการ Scaling เท่ากับสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การรวมกลับเวฟเล็ตด้วย IDTWS block

จากรูปจะมีการทำงานที่กลับกันกับการทำงานของ IDTWS จะรับสัญญาณความถี่ต่ำและค่าสัมประสิทธิ์เวฟเล็ตเพื่อนำมาสังเคราะห์หรือสร้างสัญญาณใน Time domain ออกมา สำหรับการ Decimation หรือ Subinterleave หรือ Up Sampling ด้วยอัตราส่วนของสองนี้ จะถูกแทนด้วยกระบวนการที่กลับกัน คือกระบวนการ Interleave หรือ Up Sampling ที่จะแทรกข้อมูล 0 เข้าไปในแต่ละ Sampling

2.10 Multi Resolution Wavelet Transform

จากการแปลงเวฟเลตแบบที่เงื่อนไขเฉพาะต่างๆจะเห็นว่าการแปลงเวฟเลตแบบ Multi Resolution Wavelet Transform จะเป็นกรณีที่ใช้นิยมใช้กันมากที่สุด ซึ่งในการทำ Multi Resolution Wavelet Transform นั้นฟังก์ชันที่นำมาเป็นเวฟเลตแม่อาจจะมีคุณสมบัติแบบ Non Orthogonal หรืออื่นๆก็ได้

โดยที่ข้อกำหนดเบื้องต้นของตัวเวฟเลตแม่หรือตัวกรองความถี่สูงผ่านมันจะต้องมีความสัมพันธ์กับฟังก์ชันที่เป็น Scaling Function หรือตัวกรองความถี่ต่ำผ่านด้วยการทำ Multi Resolution Wavelet Transform นี้มีคุณสมบัติของ ค่าสัมประสิทธิ์ ที่ได้ออกแบบมาเป็นแบบ Paramidal Structure อยู่ในตัว ซึ่งคุณสมบัตินี้ไม่จำเป็นสำหรับการแปลงเวฟเลตโดยทั่วไป แต่ด้วยคุณสมบัติ Paramidal Structure นี้ทำให้การแปลงเวฟเลตที่ระดับสเกลต่างๆ สามารถที่จะทำได้โดยใช้ Scaling Function (Low Pass Filter) และ Wavelet Function (High Pass Filter) เพียงตัวเดียวได้ ซึ่งสามารถทำงานได้ง่ายในทางปฏิบัติด้วยการทำงานซ้ำ

2.11 แนะนำภาษา VHDL

VHDL เป็นภาษาบรรยายทางฮาร์ดแวร์สำหรับการออกแบบในระดับสูงทางอิเล็กทรอนิกส์ (Electronics) มีความสามารถในการเลียนแบบ (Simulation), การสังเคราะห์ (Synthesis), และ การทดสอบ (Testing) ในการจำลองระบบอิเล็กทรอนิกส์ดิจิทัลโดยลักษณะการทำงานของ VHDL สามารถทำงานได้ทั้งแบบขนาน และลำดับ

2.12 ข้อได้เปรียบของการใช้ VHDL

1. การออกแบบมีคุณภาพที่สูง
2. ออกแบบวงจรที่มีความซับซ้อนมากๆได้
3. ใช้เวลาในการออกแบบสั้นกว่า
4. ค้นหาข้อผิดพลาดและเปลี่ยนแปลงแก้ไขได้ง่าย

2.13 โครงสร้างภายในของอุปกรณ์ FPGAs

FPGAs จัดเป็นวงจรรวมเฉพาะกิจชนิดหนึ่งที่สามารถจะโปรแกรมเป็นวงจรเชิงเลขใดๆ ก็ได้ เช่นเดียวกับ EPLD ต่างกันที่ EPLD โปรแกรมลงบน EPROM ภายในและสามารถโปรแกรมใหม่ได้หลังจากนำไปลบด้วยแสง UV แต่ใน FPGAs นั้นสามารถที่จะโปรแกรมลงบนสแตติกแรมภายในด้วยข้อมูลที่อยู่นอก และสามารถโปรแกรมใหม่ได้โดยการรีเซตด้วยสัญญาณไฟฟ้านอกจากนั้น FPGAs ยังประหยัดและมีความจุวงจรสูง (จำนวนเกตมาก) ได้อีกด้วย

วงจรรวมชนิดที่ใช้ในโครงงานนี้ผลิตโดยบริษัทไซลิงค์ (Xilinx) ซึ่งเป็นบริษัทที่ทำการค้นคว้าร่วมกับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นกลุ่มของเกตไว้จำนวน 600-25,000 เกต ดังแสดงในตารางที่ 2.1 การที่ต้องการบอกขนาดของวงจรรวมเป็นจำนวนเกตเพราะจะได้รู้ว่าขนาดของ วงจรที่ได้ออกแบบไว้สามารถโปรแกรมลงบนวงจรรวม FPGAs ได้หรือไม่

FPGAs มีโครงสร้างภายในใกล้เคียงกับสถาปัตยกรรมของเกตอะเรย์ (GAL, Gate Array Logic) มาก สามารถโปรแกรมและลบคอนฟิกูเรชัน (Configuration) ภายในสแตติกแรม (Static RAM) ได้ โดยใช้กระแสไฟฟ้าซึ่งทำการโปรแกรมได้ โดยดึงข้อมูลฐานสิบหกมาจากภายนอก เช่น Parallel EPROM หรือ PROM ต่างกับ EPLD, PAL ที่มี EPROM อยู่ในตัวภายใน FPGAs จะจัดเรียงเป็นลอจิกเซลล์ล้อมรอบภายนอกด้วยอินพุต-เอาต์พุตเซลล์ FPGAs ตัวแรกที่ผลิตโดยบริษัท ไซลิงค์คือเบอร์ XC2064 (2000 Family) ประกอบด้วยเซลล์เรียงกันเป็นเมตริกซ์ (Matrix) เป็นจำนวน 64 เซลล์ หลังจากนั้นผลิต FPGAs ตระกูล 3000 และ 4000 ซึ่งมีโครงสร้างซับซ้อนขึ้นสามารถเพิ่มจำนวนเกตได้มากขึ้นและดีขึ้น แต่ละเซลล์เรียกว่า CLB (Configurable Logic Block)

ตารางที่ 2.1 คุณสมบัติของ FPGAs ตระกูลต่างๆ

FPGAs	App.Gate Count	Max I/Os	Flip-Flops	RAM bits	Available CLBs
XC2064	1,000	58	122	0	64
XC2018	1,500	74	174	0	100
XC3020/3120	1,800	64	256	0	64
XC3030/3130	2,700	80	360	0	100
XC3042/3142	3,700	96	480	0	144
XC3064/3164	5,500	120	688	0	244
XC3090/3190	7,500	144	928	0	320
XC3195	9,000	176	1,320	0	484
XC4002A	2,000	64	256	2,048	64
XC3003/4003A	3,000	80	360	3,200	100
XC4000H	3,000	160	200	3,200	100
XC4004A	4,000	960	480	4,608	144
XC4005/4005A	5,000	122	616	6,072	196
XC4005H	5,000	192	392	6,272	196
XC4006	6,000	128	768	8,192	256
XC4008	8,000	144	936	10,368	324
XC4010	10,000	160	1,120	12,800	400
XC4013	13,000	192	1,536	18,432	576
XC4025	25,000	256	2,560	32,768	1,024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ประมาณการนับเกตของเกตพื้นฐาน

Gate	Equivalent gate count	Gate	Equivalent gate count
INV	1	RS Latch	3
NAIINR2	1	D Latch	4
NAIINR3	2	D Latch with CLR	5
NAIINR4	2	D Latch with PRE	5
NAIINR6	5	D Latch with PRE/CLR	6
NAIINR8	6	D F/F	6
NAIINR9	7	D F/F with CLR	7
NAIINR12	8	D F/F with PRE	7
NAIINR16	11	D F/F with PRE/CL	8
BUFF	2	JK F/F with CLR	9
ANIIOR2	2	JK F/F with PRE	12
ANIIOR3	2	JK F/F with PRE/CL	13
ANIIOR4	3	T F/F with CLR	8
XOR2	3	T F/F with PRE	8
XNOR2	3	T F/F with PRE/CL	9

หมายเหตุ : NAIINR2 หมายถึงเกต NAND2 หรือ เกต NOR2

2.14 ส่วนอินพุตและเอาต์พุตของอุปกรณ์ FPGAs

รอบนอกของ FPGAs จะประกอบด้วย IOBs ประมาณ 64 ถึง 144 ตัว ซึ่งขึ้นอยู่กับตระกูลของ FPGAs ซึ่ง IOBs จะเป็นตัวเชื่อมต่อระหว่างภายในกับภายนอกของวงจรถ่ายรูปของ FPGAs ลักษณะของ IOBs จะมีลักษณะ 2 ทิศทาง สามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตก็ได้

2.15 รายละเอียดการใช้งานของอุปกรณ์ FPGAs

FPGAs สามารถทำงานได้หลายลักษณะ โดยกำหนดได้ที่ขาสัญญาณ M0, M1, และ M2 ดังแสดงในตารางที่ 2.3 ในลักษณะมาสเตอร์พาราเรล (Master Parallel Mode) รับโปรแกรมคอนฟิกทีละ 1 ไบต์ (Byte) จากหน่วยความจำภายนอกที่เป็นแบบขนาน โดยสามารถรับโปรแกรมคอนฟิก (Config) จากแอดเดรส (Address) ต่ำหรือสูงก่อนก็ได้ การต่อใช้งานในลักษณะเพริเฟอร์ล (Peripheral) จะรับโปรแกรมคอนฟิกทีละ 1 ไบต์จากไมโครโปรเซสเซอร์ โดยสามารถโต้ตอบกันได้ว่าพร้อมหรือไม่ที่จะรับข้อมูล การต่อลักษณะสเลฟซีเรียล (Slave Serial) จะรับโปรแกรมคอนฟิกทีละ 1 บิต (Bit) จากไมโครโปรเซสเซอร์ตามสัญญาณอินพุต CCLK ส่วนการต่อลักษณะมาสเตอร์ซีเรียล (Master Serial) จะรับโปรแกรมคอนฟิกทีละ 1 บิต จากหน่วยความจำภายนอกที่เป็นแบบอนุกรม

ตารางที่ 2.3 โหมดต่าง ๆ ของการคอนฟิกกูเรชั่น

Mode	M2	M1	M0	CCLK	Data
Master Serial	0	0	0	Output	Bit-Serial
Slave Serial	1	1	1	Input	Bit-Serial
Master Parallel up	1	0	0	Output	Byte-Wide,00000up
Master Parallel down	1	1	0	Output	Byte-Wide,3FFF down
Peripheral Synchr	0	1	1	Input	Byte-Wide
Peripheral Asynchr	1	0	1	Output	Byte-Wide
Reserved	0	1	0	-	-
Reserved	0	0	1	-	-

จากความต้องการในการสร้างให้สามารถใช้กระแสไฟฟ้าต่ำจากลักษณะการต่อใช้งานที่มีอยู่ทั้ง 5 แบบ จึงมีเพียง 2 แบบเท่านั้นที่เหมาะสม คือ แบบมาสเตอร์ซีเรียลและแบบสเลฟซีเรียล ส่วนแบบมาสเตอร์พาราเรลต้องใช้ EPROM 27CXXX ซึ่งกินกระแสมากกว่า PROM XC17XXX เหมาะในการทดสอบต้นแบบก่อน เมื่อวงจรต้นแบบทำงานได้ถูกต้องแล้วจึงทำการอัดโปรแกรมลง PROM อีกทีหนึ่งเพราะว่าในแบบพาราเรลนั้น EPROM สามารถโปรแกรมได้ใหม่ต่างกับ PROM ที่โปรแกรมได้เพียงครั้งเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน FPGAs ด้วยการต่อในลักษณะสเตฟซีเรียลและมาสเตอร์ซีเรียล เมื่อเริ่มทำการจ่ายไฟเข้าตัว FPGAs จะทำการลบข้อมูลหน่วยความจำที่ใช้ในคอนฟิก (Configuration Memory) ตรวจสอบลักษณะการต่อคอนฟิกว่าเป็นลักษณะใดในตารางที่ 2.3 ว่าเป็นแบบอนุกรมหรือขนาน หลังจากนั้นจะเริ่มทำการโปรแกรมคอนฟิกสัญญาณ DONE/PROGRAM เป็น '0' ซึ่งอยู่ในระหว่างโปรแกรม และเมื่อข้อมูลในคอนฟิกที่รับมาจากภายนอกเต็มหน่วยความจำที่ใช้งานอยู่ในคอนฟิกและความยาวของข้อมูลตรงกับที่ส่วนหัวของข้อมูลในคอนฟิกก็จะปรากฏสัญญาณ DONE/PROGRAM จะเป็น '1' ซึ่งหมายถึง โปรแกรมทำการคอนฟิกเสร็จสิ้นแล้ว

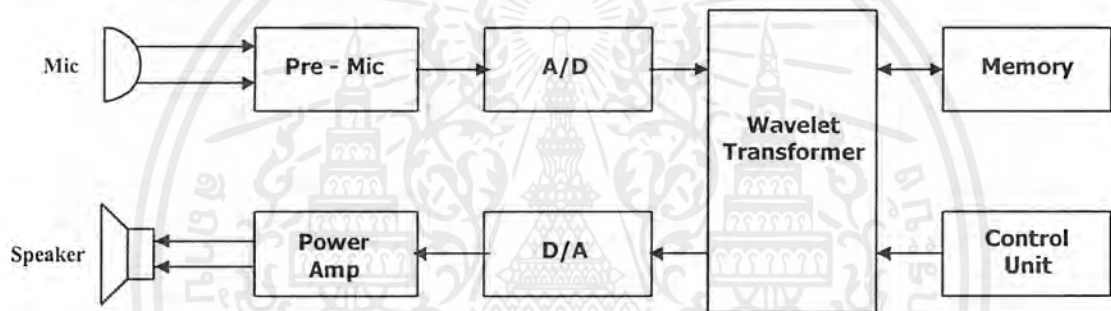


บทที่ 3

การออกแบบ การสร้างและการทำงาน

3.1 ความคิดเบื้องต้นในการสร้างอุปกรณ์บีบอัดข้อมูลเสียงโดยสมการเวฟเล็ต และประมวลผลด้วย FPGAs

อุปกรณ์บีบอัดข้อมูลเสียงพูดโดยสมการเวฟเล็ตด้วยอุปกรณ์ FPGAs นี้สามารถออกแบบได้โดยใช้ทั้งส่วนที่เป็นฮาร์ดแวร์และส่วนที่เป็นซอฟต์แวร์ร่วมกัน โดยมีหลักการทำงานดังนี้



รูปที่ 3.1 ความคิดเบื้องต้นในการออกแบบวงจร Speech Compression

จากผังงานรูปที่ 3.1 สัญญาณเสียงจะถูกแปลงให้เป็นสัญญาณไฟฟ้าโดยไมโครโฟนผ่านภาคปริโมคเพื่อขยายสัญญาณให้มีความแรงของสัญญาณเพียงพอที่จะส่งต่อไปยังภาค Analog to Digital Converter (A/D) เพื่อแปลงสัญญาณเสียงที่ได้ให้เป็นสัญญาณดิจิทัลขนาด 8 บิต เพื่อทำการประมวลผลโดยสมการเวฟเล็ต ในบล็อก Wavelet Transformer ซึ่งเป็นการนำสัญญาณมาลดรูปด้วยสมการเวฟเล็ตเพื่อให้สามารถเก็บข้อมูลได้เพิ่มขึ้น ในขนาดหน่วยความจำที่เท่าเดิม จากนั้นจึงส่งไปยังภาค Memory หรือการจัดเก็บเข้าไปยังหน่วยความจำที่กำหนด

จากนั้นเมื่อต้องการอ่านสัญญาณเสียงที่ทำการจัดเก็บ ก็ทำได้โดยการอ่านข้อมูลจากหน่วยความจำส่งกลับไปยังภาค Wavelet Transformer เพื่อแปลงกลับสัญญาณโดยวิธีการแปลงกลับเวฟเล็ต หรือ การนำข้อมูลที่เก็บไว้แบบลดขนาดข้อมูลมาแปลงให้มีสัญญาณคงเดิม ด้วยวิธีการแทรกค่าข้อมูลที่หายไปด้วยข้อมูลอื่นๆมาทดแทนส่วนที่ถูกตัดทิ้งไป โดยวิธีการของการแปลงเวฟเล็ตนั่นเอง จากนั้นจึงส่งต่อไปสัญญาณยังภาค Digital to Analog Converter (D/A) ซึ่งจะเป็นการแปลงข้อมูลดิจิทัล 8 บิตที่ผ่านการประมวลผลมาแปลงให้เป็นสัญญาณแอนะล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

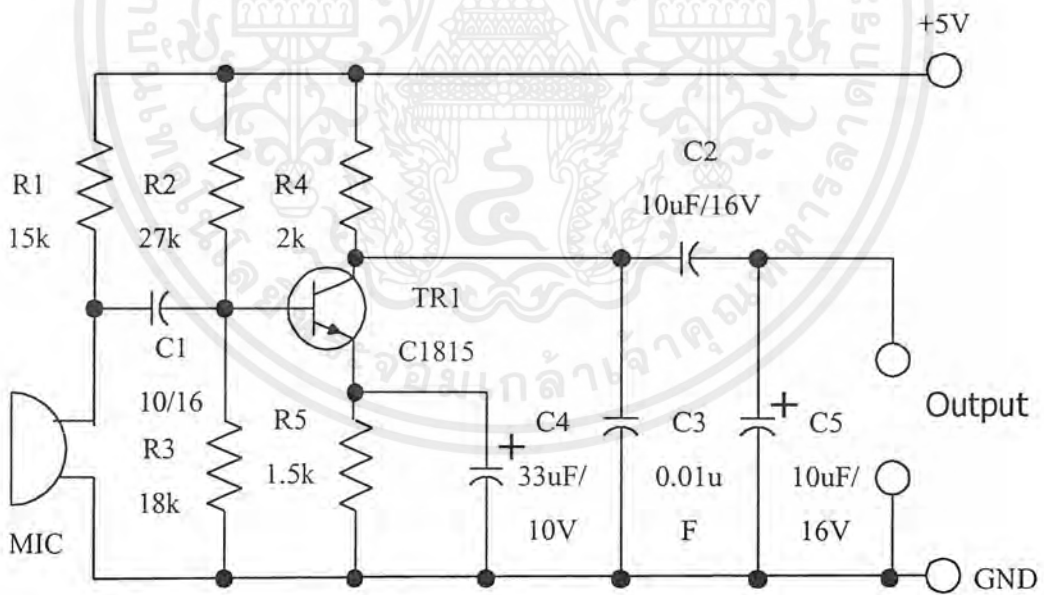
หรือสัญญาณเสียงตามเดิม แล้วจึงส่งไปยังภาค Power Amplifier เพื่อทำการขยายกำลังให้มีขนาดสัญญาณแรงพอที่จะไปขับลำโพง (Speaker) ได้เพียงพอดังตามความต้องการ

3.2 วงจรในการสร้างอุปกรณ์บีบอัดข้อมูลเสียงโดยสมการเวฟเล็ดและ ประมวลผลด้วยอุปกรณ์ FPGAs

วงจรทั้งหมดมีทั้งส่วนที่เป็นวงจรรภายนอก และวงจรในส่วนที่เป็นภายในอุปกรณ์ FPGAs ซึ่งจะเลือกใช้ตามความเหมาะสมและความสะดวกในการใช้งาน ซึ่งเพื่อเป็นการประหยัดฮาร์ดแวร์ และลดความยุ่งยากของวงจรรวมทั้งขนาดของวงจร จึงได้สร้างเป็นวงจรรภายในเป็นหลัก และได้แยกออกมาเป็นวงจรรภายนอกเพียงส่วนหนึ่งเท่านั้น

3.2.1 วงจรรภายนอก

วงจรรปรีไมโครโฟน

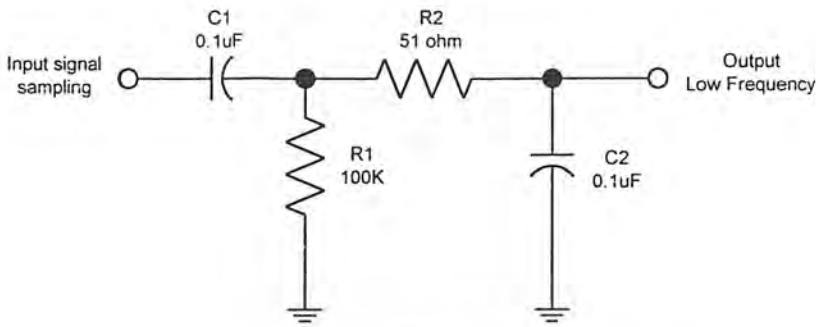


รูปที่ 3.2 วงจรรปรีไมโครโฟน

จากรูปที่ 3.2 TR1 ต่อเป็นวงจรรขยายแบบคลาส A สัญญาณจาก MIC จะผ่าน C1 มาเข้าขา B ของ TR1 เพื่อทำการขยายสัญญาณออกทางขา C ของ TR1 สัญญาณที่ทำการขยายแล้วจะผ่าน

C2 ไปที่จุด Output และ R1 จะทำหน้าที่เป็น R โหลดของ MIC C3 ทำหน้าที่บายพาสความถี่สูง
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

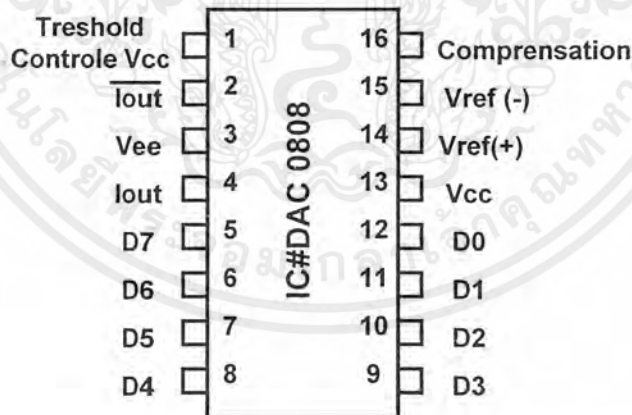
วงจรโลว์พาสฟิลเตอร์



รูปที่ 3.3 วงจรโลว์พาสฟิลเตอร์

จากรูปที่ 3.3 เป็นวงจรซึ่งจะกรองความถี่ต่ำผ่านซึ่งอินพุตที่เป็นความถี่สูงจะถูกแยกไฟตรงเมื่อผ่าน C1 และ R1 เพราะ C1 จะมีคุณสมบัติบล็อกไฟตรง และ R1 จะทำหน้าที่ดึงกระแสลงกราวด์ ส่วน R2 และ C2 จะเป็นวงจร โลว์พาสฟิวเตอร์นั่นเองเพราะว่า C2 จะทำหน้าที่ดึงสัญญาณความถี่สูงลงกราวด์เหลือเพียงความถี่ต่ำออกเอาท์พุตไป

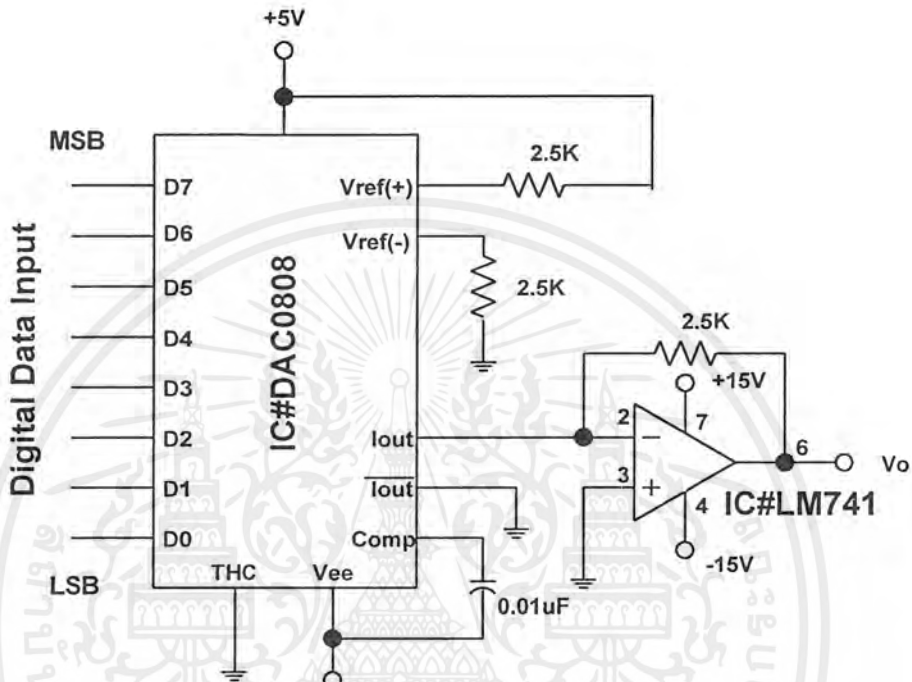
วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนาลอก โดยใช้ IC#DAC0808



รูปที่ 3.4 ตำแหน่งขาต่างๆของไอซี DAC 0808

DAC 0808 เป็น ไอซีแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนาลอก ซึ่งมีแหล่งกำเนิดกระแสอ้างอิงอยู่ในตัวไอซี โดยในรูปเป็นการแสดงการจัดขาของ DAC 0808 ขา 5 ถึงขา 12

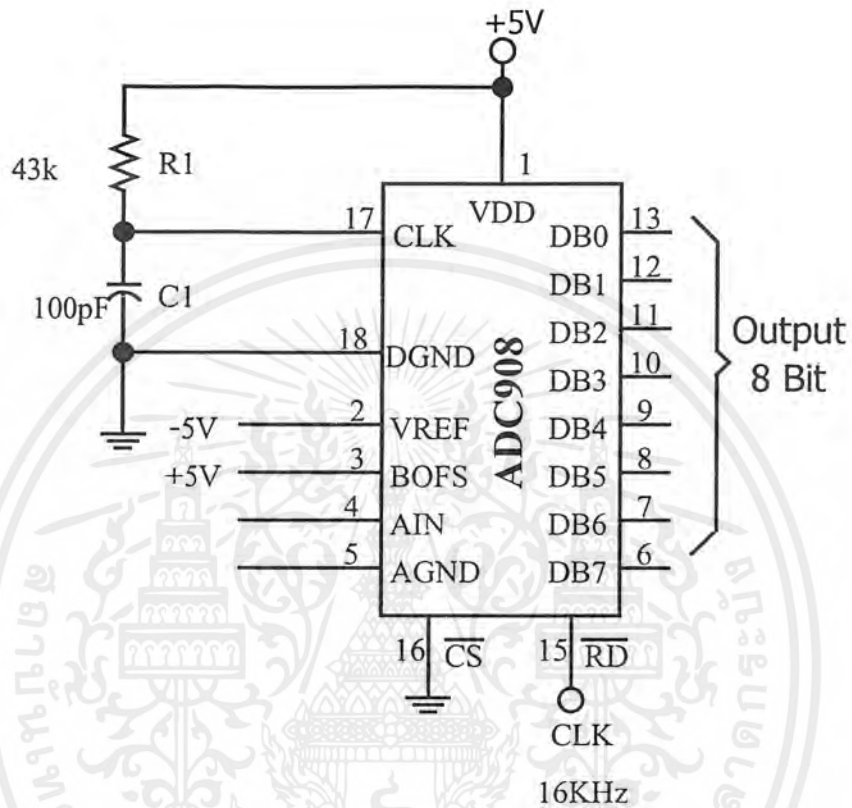
เป็นขารับข้อมูลอินพุตแบบดิจิทัลขนาด 8 บิต ข้อมูลที่ป้อนเข้ามาจะถูกแปลงเป็นกระแส Iout ออกทางขา 4



รูปที่ 3.5 วงจรใช้งานจริงของไอซี DAC 0808

จากรูปที่ 3.5 แสดงวงจรใช้งานเบื้องต้นของ DAC 0808 ที่ต่อเป็นตัวแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนาล็อก ตัวความต้านทาน ขา Vref (+) คือ ตัวความต้านทานปรับค่าได้เพื่อกำหนดค่ากระแสอ้างอิงเท่ากับ 2 มิลลิแอมป์และตัวต้านทาน ขา Vref (-) จะมีค่าเท่ากับตัวต้านทานตัวแรกคือ $2.5\text{ K}\Omega$ ต่อเพื่อชดเชยกระแสอินพุตภายในไอซี กระแสเอาต์พุตออกที่ขา 4 นำไปแปลงเป็นแรงดันด้วย ไอซีออปแอมป์

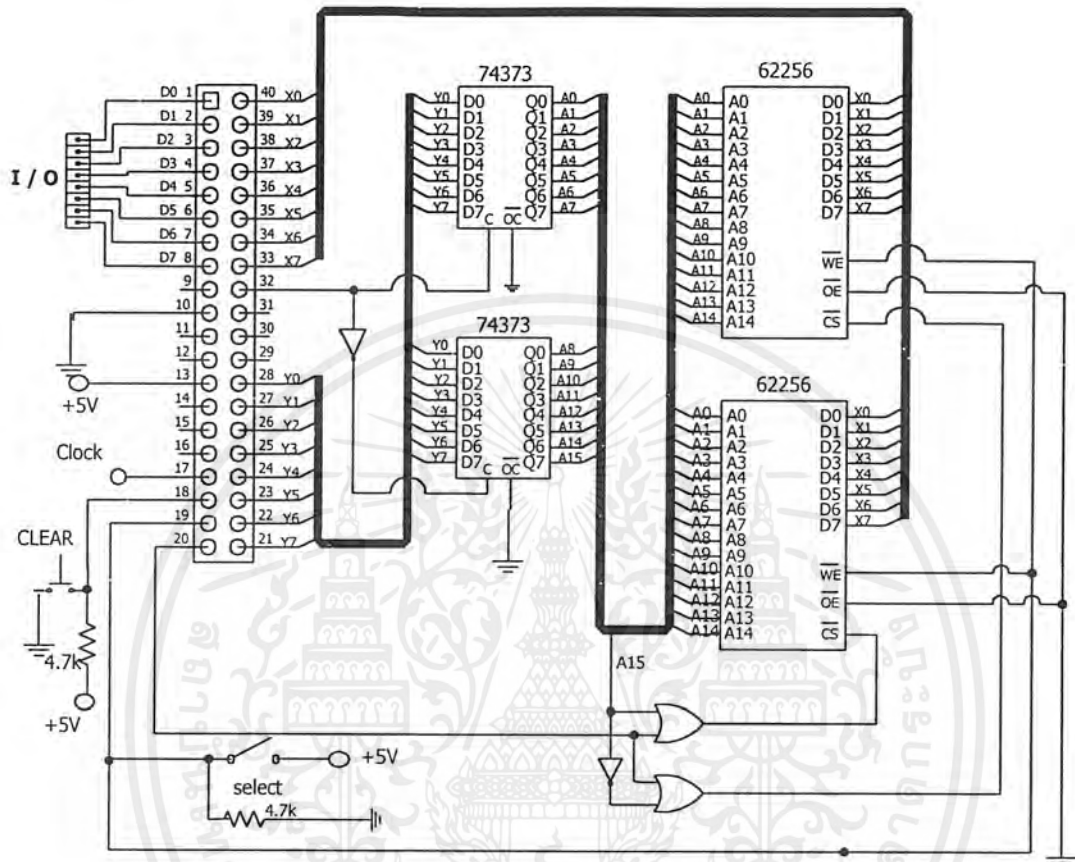
วงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัล โดยใช้ IC#ADC908



รูปที่ 3.6 การต่อขาต่างๆ ใช้งานของไอซี ADC 908

ADC908 เป็นไอซีแปลงสัญญาณดิจิทัล เป็นสัญญาณแอนาลอก ได้สัญญาณข้อมูลจำนวน 8 บิต ซึ่งอาศัยแหล่งกำเนิดสัญญาณแชนเปลลิง CLK จากภายนอกมาเป็นสัญญาณอ้างอิง โดยสามารถใช้สัญญาณภายนอกได้สูงถึง 50 MHz แต่การใช้งานจริงใช้ความถี่แชนเปลลิงเพียงแค่ 16 KHz ที่เลือกใช้ไอซีตัวนี้ก็เพราะมีสัญญาณเอาต์พุตที่ดีกว่าไอซีตัวอื่น ที่ความถี่แชนเปลลิงเท่ากัน เช่น ไอซี ADC 0804

วงจร Interface Ram กับอุปกรณ์ FPGAs



รูปที่ 3.7 วงจรที่ใช้ในการติดต่อระหว่างอุปกรณ์ FPGAs กับอุปกรณ์และหน่วยความจำภายนอก

จากรูปที่ 3.7 เป็นการติดต่อวงจรภายนอกเข้ากับอุปกรณ์ FPGAs หรืออุปกรณ์ภายในนั้นเอง โดยที่ขา 1-8 ของ Connector จะต่อกับสัญญาณอินพุตซึ่งเป็นสัญญาณดิจิทัล 8 บิต (D0-D7) ที่ได้จากการแปลงสัญญาณแอนาลอกมา (สัญญาณเสียง) โดยไอซี ADC908 คือเมื่ออุปกรณ์ FPGAs ต้องการรับอินพุตก็จะรับเข้ามาจากขา 1-8 นี้เอง

ขา 10 ของ Connector จะต่อกับกราวด์เพื่อเป็นกราวด์สำหรับเปรียบเทียบสัญญาณของอุปกรณ์ FPGAs

ขา 13 เป็นขา Vcc 5 โวลต์เพื่อจ่ายไฟเลี้ยงให้กับอุปกรณ์ FPGAs

ขา 17 เป็นขา Clock คือจะเป็นขาที่ส่งสัญญาณนาฬิกาออกมาจากอุปกรณ์ FPGAs เพื่อเป็นตัวควบคุมการทำงานให้มีการทำงานที่พร้อมกัน และนอกจากนั้นยังเป็นสัญญาณในการแซมปลิ่งให้กับวงจรแปลงสัญญาณแอนาลอกเป็นสัญญาณดิจิทัลด้วยเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 18 เป็นขาที่ต่อสำหรับเคลียร์การทำงานของอุปกรณ์ FPGAs หรือสวิตช์เคลียร์นั่นเอง โดยขานี้จะได้รับลอจิก 0 หรือลอจิก 1 จากการกดสวิตช์ ซึ่งต่อค่าความต้านทานอนุกรมเข้ากับแหล่งจ่ายไว้หากสวิตช์ไม่ได้กด ขา 18 ก็จะได้รับลอจิก 1 แต่ถ้าหากกดสวิตช์ก็จะได้รับลอจิก 0 แทนแล้วอุปกรณ์ FPGAs จะทำการเช็คที่ขา 18 นี้เองว่าจะเคลียร์การทำงานหรือไม่

ขา 19 เป็นขาที่ต่อสำหรับการเลือกโหมดการทำงานติดต่อกับแรมภายนอกโดยหากสลับสวิตช์ Select ขา 19 ก็จะได้รับลอจิก 0 หรือลอจิก 1 เหมือนกับขา 18 เช่นกัน จากนั้นอุปกรณ์ FPGAs ก็ทำการเช็คที่ขา 19 นี้เองว่าจะทำการอ่านหรือเขียนข้อมูลจากแรมภายนอก

ขา 20 เป็นขาที่ต่อเพื่อควบคุมขา CS (Chip Select) ของแรมทั้งสองตัว ด้วยการเข้ารหัสหรือควบคุมการทำงานของแรมนั่นเอง คือถ้าหากขา 20 ป้อนลอจิก 1 ออกมา ขา CS ทั้งสองตัวก็จะได้รับลอจิก 1 ทำให้แรมทั้งสองตัวไม่ทำงานนั่นเอง แต่ถ้าหากขา 20 นี้ป้อนลอจิก 0 ออกมาแรมทั้งสองตัวก็จะผลัดกันทำงาน ซึ่งจะสลับการทำงานโดยการควบคุมของแอดเดรสไบต์สูงอีกทีหนึ่ง

ขา 21-28 เป็นขาข้อมูลที่เป็นแอดเดรสทั้งหมด 16 บิต โดยที่สามารถทำงานเป็นแอดเดรส 16 บิตได้ทั้งที่มีอยู่ 8 ขา โดยผ่านไอซี 74373 เป็นบัฟเฟอร์ที่จะคอยเลือกว่าจะแบ่งข้อมูลส่วนไหนเป็นแอดเดรสไบต์สูงและแอดเดรสไบต์ต่ำซึ่งจะมีขาควบคุมการทำงานของไอซี 74373 ที่ขา 32 ของ Connector อีกทีหนึ่ง จะได้แอดเดรส A0-A15 ทั้งหมด 2 ไบต์ในการควบคุมการจัดการกับหน่วยความจำภายในแรม 62256

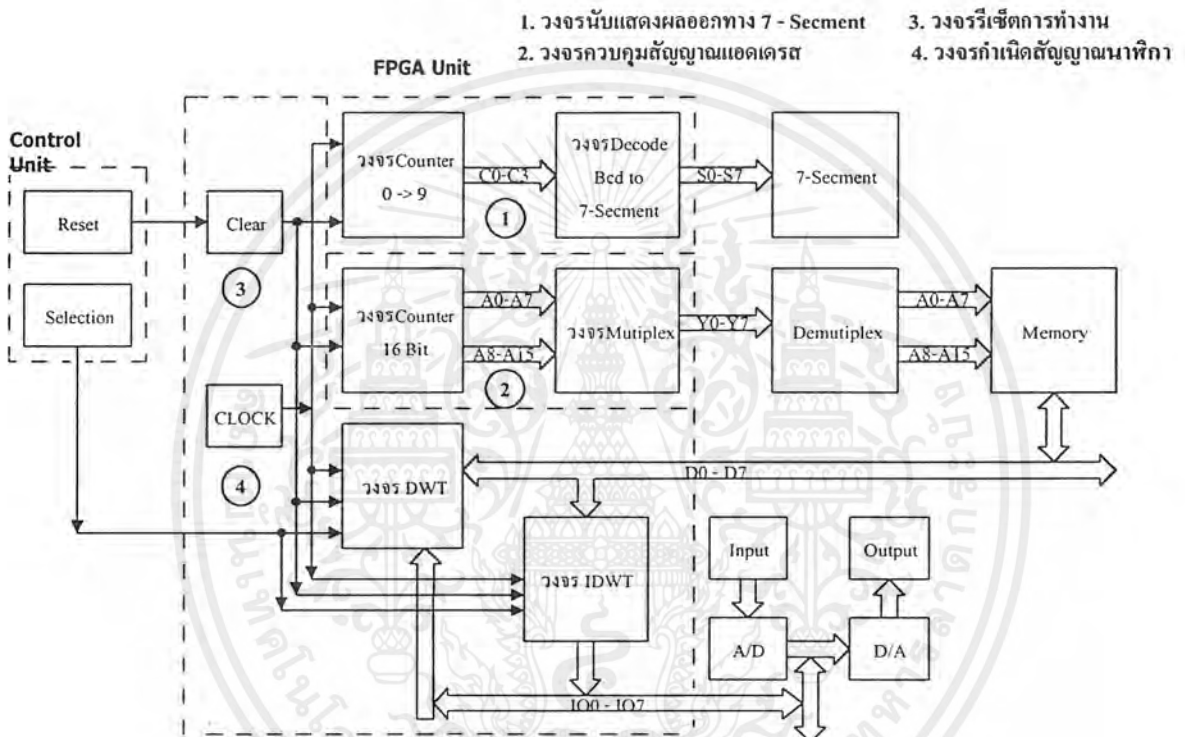
สำหรับไอซี แรม 62256 ซึ่งเป็น แรมขนาด 32K ทั้ง 2 ตัว จะได้รับแอดเดรสตัวละ 15 บิต และจะสลับกันทำงานคือเมื่อตัวหนึ่งเต็มอีกตัวหนึ่งก็จะทำงานต่อ โดยจะส่งข้อมูลออกไปหรือเขียนข้อมูลเข้ามาก็จะถูกควบคุมจากขา 19 ซึ่งเป็นสวิตช์ Select ควบคุมซึ่งจะมีขาข้อมูลอยู่ตัวละ 8 ขา (D0-D7) ที่ใช้ในการอ่านและเขียนข้อมูลนั่นเอง

ขา 32 เป็นขาควบคุมเลือกการทำงานของไอซี 74373 ว่าจะเลือกแอดเดรสจากขา Connector ขา 21-28 ว่าช่วงไหนควรจะเป็นแอดเดรสไบต์สูงและช่วงไหนควรเป็นแอดเดรสไบต์ต่ำ

ขา 33-40 เป็นขาข้อมูลที่จะส่งไปเก็บในแรม หรือข้อมูลที่จะอ่านมาจากแรมนั่นเองมีขนาด 8 บิต (D0-D7) เพื่อจะนำไปทำการประมวลผลจากอุปกรณ์ FPGAs อีกทีหนึ่ง

3.2.2 วงจรภายใน

ส่วนของวงจรภายในอุปกรณ์ FPGAs นั้นจะมีทั้งส่วนที่เป็น Schematic และ VHDL ซึ่งจะสามารถสร้างขึ้นได้ภายในอุปกรณ์ FPGAs ซึ่งการทำงานจะถูกแบ่งออกเป็นภาคการทำงานต่างๆตาม บล็อกไดอะแกรมรูปที่ 3.8



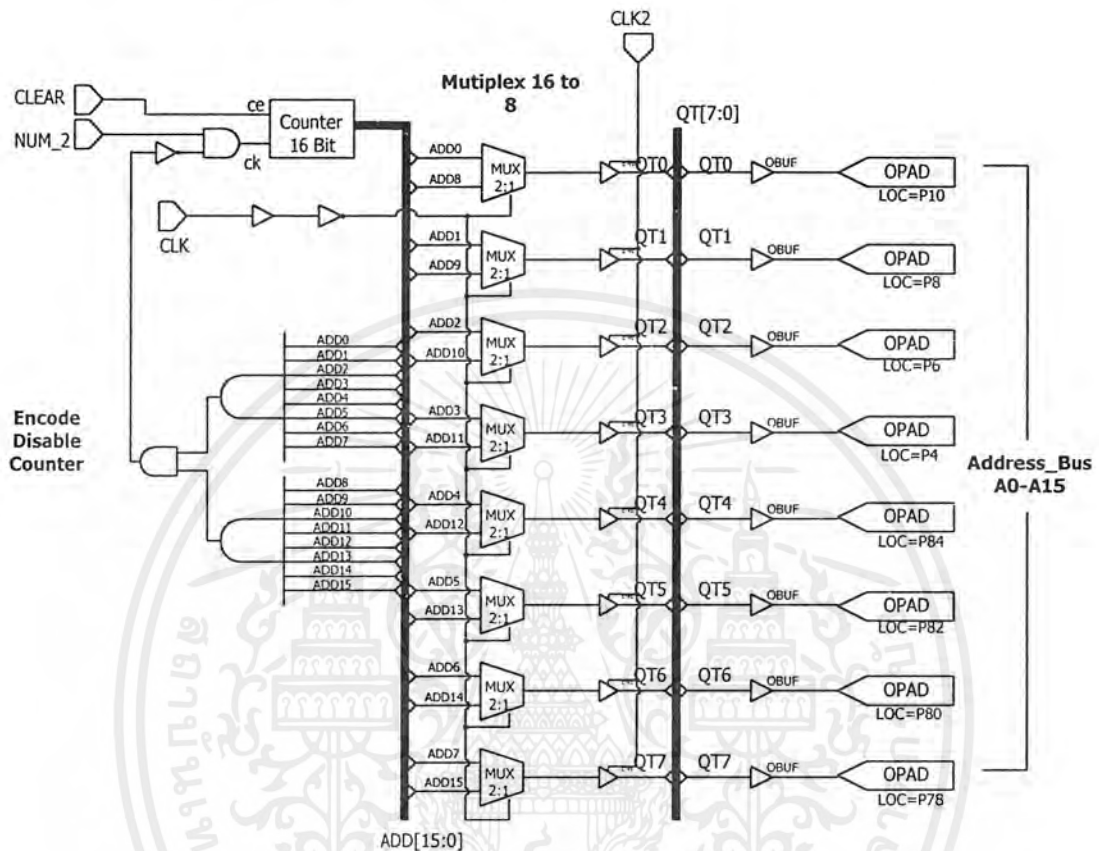
รูปที่ 3.8 บล็อกไดอะแกรมของวงจรภายในอุปกรณ์ FPGAs

จากรูปที่ 3.8 วงจรภายในอุปกรณ์ FPGAs จะรับข้อมูลผ่านทางบัสอินพุตและเอาต์พุต แล้วส่งข้อมูลไปยังบล็อก DWT ซึ่งบล็อกนี้จะทำการคำนวณด้วยสมการเวฟเล็ต จะได้ข้อมูลที่ทำการลดรูปแล้วจึงนำไปเก็บไว้ในหน่วยความจำ ซึ่งตำแหน่งของหน่วยความจำจะถูกกำหนดจากวงจรควบคุมสัญญาณแอดเดรสโดยส่วนของวงจรมีทำการนับเลขจำนวน 16 บิต คือตั้งแต่ 0000H – FFFFH

ในส่วนของการแปลงข้อมูลกลับข้อมูลจะถูกอ่านออกมาจากหน่วยความจำและส่งเข้าไปยังบล็อก IDWT หรือ บล็อกแปลงกลับสมการเวฟเล็ต เพื่อทำการส่งค่าที่เก็บเข้ามาในตอนแรกออกไปยังเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรถอบคุมสัญญาณแอดเดรส



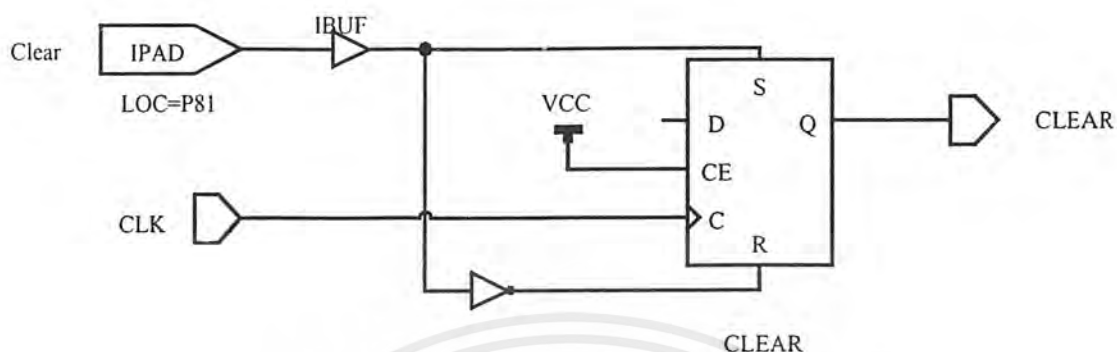
รูปที่ 3.9 วงจรถอบคุมสัญญาณแอดเดรส

จากรูปที่ 3.9 เมื่อส่วน Counter 16 บิต ได้รับสัญญาณนาฬิกา จะทำให้วงจรเริ่มนับตั้งแต่ 0000H – FFFFH และเมื่อวงจรมันถึงค่า FFFFH จะทำให้ส่วนของสัญญาณ Counter 16 บิต หยุดทำงานเนื่องจากส่วน Encode Disable Counter จะมีลอจิกเป็น 1 ทำให้สัญญาณนาฬิกาไม่มีผลต่อวงจรมันอีก หรือวงจรมันหยุดรับสัญญาณนาฬิกาของมันเอง

ส่วนแอดเดรสบัสจะใช้งานเพียง 8 เส้นเท่านั้น ซึ่งความจริงแล้วจะมีสัญญาณแอดเดรสอยู่ถึง 16 เส้น เนื่องจากข้อจำกัดของอุปกรณ์ FPGAs มีจำนวนจำกัด ไม่เพียงพอต่อการใช้งาน จึงต้องออกแบบให้ใช้งานร่วมกันโดยใช้การมัลติเพล็กซ์เข้าช่วย ซึ่งจะสลับส่งข้อมูลแอดเดรสไบต์ต่ำออกมาก่อน ADD0 – ADD7 จากนั้นจึงจะสลับกันส่งแอดเดรสไบต์สูง ADD 8 – ADD15 ตามออกมาอีกทีหนึ่ง โดยจะมีการส่งสัญญาณควบคุมไปเป็นตัวเลือกว่าสัญญาณแอดเดรสช่วงไหนเป็นสัญญาณไบต์สูงหรือไบต์ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

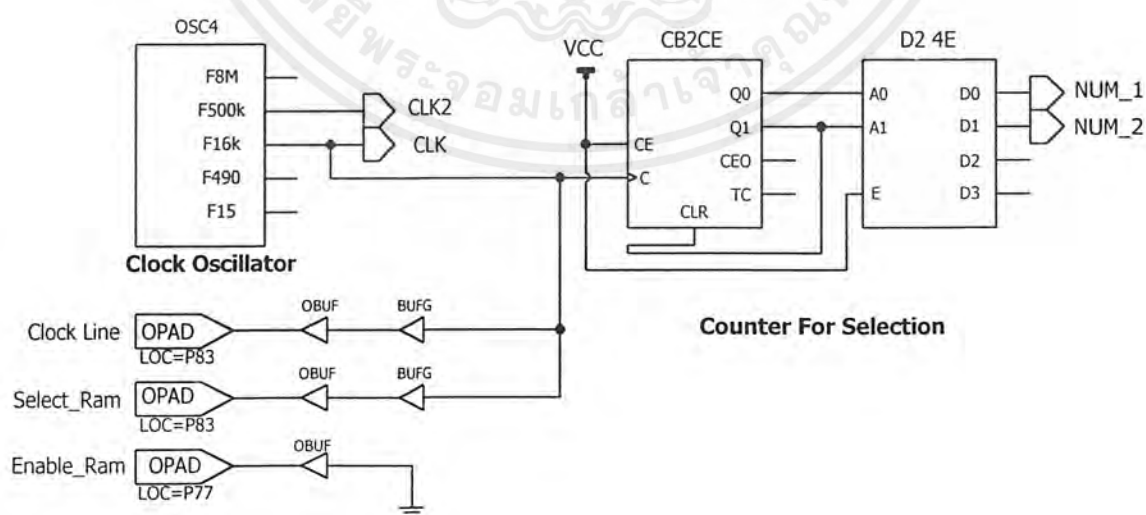
วงจรรีเซ็ตการทำงาน



รูปที่ 3.10 การต่อวงจรรีเซ็ตการทำงาน

จากรูปที่ 3.10 ใช้การทำงานของ D Flip-Flop โดยใช้ขาเซต และรีเซ็ตในการทำงาน โดยหากสัญญาณที่ขาเคลียร์ซึ่งต่อกับอุปกรณ์ภายนอกมีค่าเป็นลอจิก 1 จะไม่มีผลกับวงจรก็เพราะว่าขาเซตจะได้ลอจิก 1 ทำให้ได้ลอจิก 1 ออกขาเอาต์พุต Q ขาเคลียร์ของอุปกรณ์ทุกตัวก็จะไม่เคลียร์เนื่องจากจะแอกทีฟที่ลอจิก 0 แต่หากได้สัญญาณลอจิก 0 เข้ามาเมื่อผ่านอุปกรณ์อินเวอร์เตอร์ที่ขา รีเซ็ตก็จะทำงาน ที่ลอจิก 1 นั่นเอง ทำให้มีเอาต์พุต Q เป็นลอจิก 0 อุปกรณ์ภายในทุกตัวก็จะรีเซ็ตตัวเองทั้งหมด

วงจรกำเนิดสัญญาณนาฬิกา



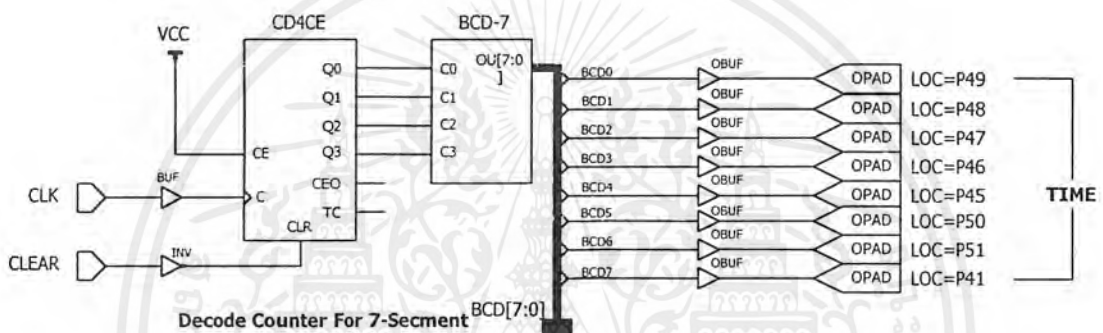
รูปที่ 3.11 วงจรกำเนิดสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.11 เป็นวงจรที่ใช้ในการกำเนิดสัญญาณนาฬิกา มีความถี่ 16 KHz เพื่อใช้เป็นสัญญาณอ้างอิงในอุปกรณ์ FPGAs และใช้ร่วมกับวงจรที่อยู่ภายนอกอุปกรณ์ FPGAs ด้วย และในวงจรนี้ยังส่งสัญญาณควบคุม หน่วยความจำภายนอกออกไปด้วย

อีกทั้งยังส่งสัญญาณควบคุม NUM1 และ NUM2 ไปยังภาค DWT (การแปลงเวฟเล็ด) เพื่อควบคุมให้ภาค DWT ทำงานได้ตรงกับความต้องการ

วงจรรับแสดงผลออกทาง 7-Segment



รูปที่ 3.12 วงจรรับแสดงผลออกทาง 7-Segment

จากรูปที่ 3.12 ตัวนับ CD4CE จะได้รับสัญญาณนาฬิกา CK เข้ามาแล้วจะนับออกมาเป็นรหัสไบนารี 4 บิต จากนั้นจึงทำการถอดรหัสออกมาด้วยอุปกรณ์ BCD-7 เพื่อทำการส่งไปแสดงผลยัง 7-Segment ที่ต่ออยู่ภายนอกให้นับเป็นเวลาในการแสดงผลเป็นตัวเลขซึ่งจะบอกได้ว่าเก็บข้อมูลเสียงมาเวลาเท่าไร และส่งข้อมูลเสียงที่ทำการผ่านอุปกรณ์ FPGAs และจะประมวลผลด้วยสมการเวฟเล็ด ออกไปเป็นเวลาเท่าไร

วงจรแปลงสมการเวฟเล็ต DWT

1. หลักการคำนวณโดยใช้สมการเวฟเล็ตลดรูปตัวแปร

1.1 การแปลงเวฟเล็ต

ในการแปลงเวฟเล็ตจะทำการจัดแบ่งข้อมูลออกเป็นส่วนๆ โดยที่ความถี่สูงจะพิจารณาที่ความละเอียดสูงกว่าที่ความถี่ต่ำ โดยที่ช่วงในการพิจารณาความถี่สูงก็จะกว้างกว่าที่ความถี่ต่ำด้วย คุณสมบัตินี้ทำให้ Wavelet Transform เหมาะกับงานบางด้านเท่านั้น เช่นงานประมวลผลสัญญาณ เนื่องจากสายตามนุษย์จะแยกแยะภาพที่มีรายละเอียดสูงได้ยาก ซึ่งก็หมายความว่าความถี่สูง สายตามนุษย์แยกแยะความแตกต่างของความถี่ได้ไม่ชัดเจน

สำหรับในการประมวลผลสัญญาณเสียงพูด มนุษย์จะมีความไวเสียงในแต่ละช่วงความถี่ซึ่งอาจกล่าวได้ว่าไม่แตกต่างกัน ซึ่งเป็นผลให้การแปลงเวฟเล็ตไม่สามารถให้ผลดีในงานประมวลผลสัญญาณเสียงพูด Wickerhauser จึงได้นำเสนอวิธีการแปลงเวฟเล็ตที่มีอีกแบบหนึ่งซึ่งเหมาะกับสัญญาณที่ต้องการรายละเอียดสูงในทุกๆ ช่วงความถี่ที่เท่ากันเช่น สัญญาณเสียงพูดของมนุษย์โดยเรียกว่า Wavelet Packet Transform

การแปลงเวฟเล็ตเทคนิคนี้มีหลักการเดียวกับการทำการแปลงเวฟเล็ตเพียงแต่จะทำการแปลงทุกๆ ชั้นทั้งในส่วนที่ผ่านจาก ตัวกรองความถี่สูงและตัวกรองความถี่ต่ำ อาจแสดงดังนี้

ข้อมูลเริ่มต้นเป็น	X0	X1	X2	X3	X4	X5	X6	X7
ชั้นความละเอียดที่ 1	S0	S1	S2	S3	D0	D1	D2	D3
ชั้นความละเอียดที่ 2	SS0	SS1	SD0	SD1	DS0	DS1	DD0	DD1
ชั้นความละเอียดที่ 3	SSS0	SSD0	SDS0	SDD0	DSS0	DSD0	DDS0	DDD0

โดยที่ D คือรายละเอียด

และ S คือข้อมูลหาย

การทำการแปลงนั้นจะเลือกตัดส่วนที่มีความสำคัญน้อย ซึ่งสามารถตัดออกไปโดยไม่มีผลกระทบต่อสัญญาณโดยรวมมากนัก สำหรับการเลือกพิจารณาส่วนประกอบว่าส่วนใดมีความสำคัญน้อย สามารถตัดออกได้มีเกณฑ์อยู่หลายอย่าง ได้แก่

1. ใช้การเปรียบเทียบขนาดของสัญญาณกับค่า threshold
2. ใช้การวัดพลังงานของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ใช้การวัด entropy ของสัญญาณ
4. นำขนาดของส่วนประกอบทั้งหมดเรียงลำดับ และเก็บไว้เฉพาะส่วนประกอบที่มีขนาดสูงสุด n ตัวแรก โดยที่ n จะถูกกำหนด โดย entropy

การลดรูปขนาดของข้อมูลโดยใช้สมการของเวฟเล็ต

จากสมการ Convolution

$$A(n) = \sum_{k=0}^{n-1} f(k) h(n-k) \quad , \quad D(n) = \sum_{k=0}^{n-1} f(k) g(n-k)$$

กำหนดค่าสัมประสิทธิ์แม่ที่ใช้ในการแปลง WAVELET ใช้ Daubechies 1 ค่าสัมประสิทธิ์คือ 0.5 แต่เพื่อให้ง่ายต่อการคำนวณจึงคูณ 2 เข้าไปผลที่ได้

ในส่วนของการแปลง WAVELET เราจะกำหนดค่าสัมประสิทธิ์เป็นดังนี้

$$G(s) \text{ ค่าเป็น } (1, 1)$$

$$H(s) \text{ ค่าเป็น } (-1, 1)$$

ในส่วนของการแปลงกลับ WAVELET เราจะกำหนดค่าสัมประสิทธิ์เป็นดังนี้

$$G(s) \text{ ค่าเป็น } (1, 1)$$

$$H(s) \text{ ค่าเป็น } (1, -1)$$

ยกตัวอย่างเช่น

หากค่า DATA INPUT ค่าเป็น (70,-50) ทำการ Convolution 3 ครั้งเนื่องจากสูตร

จำนวนครั้ง = (จำนวน DATA INPUT + จำนวนของค่าสัมประสิทธิ์ WAVELET) -1
จากสูตรจะได้

$$(2+2)-1 = 3$$

จากสมการ Convolution เราจะคำนวณ ได้ดังนี้

$$A(0) = G1D1$$

$$A(1) = G1D2 + G2D1$$

$$A(2) = G1D3 + G2D2 + G3D1$$

วิธีทำ

$$\text{ที่ } G(s) = (1, 1)$$

$$A(0) = (1)(70) = 70$$

$$A(1) = (1)(-50) + (1)(70) = 20$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$A(2) = (1)(0) + (1)(-50) + (0)(70) = -50$$

$$\text{ที่ } H(s) = (-1, 1)$$

$$D(0) = (-1)(70) = -70$$

$$D(1) = (-1)(-50) + (1)(70) = 120$$

$$D(2) = (-1)(0) + (1)(-50) + (0)(70) = -50$$

จากนั้นทำการ DOWN SAMPLING โดยการเลือกเอาลำดับที่ 2 หารแล้วลงตัวมาใช้จะได้

$$\text{LOW} = (0), (20)$$

$$\text{HIGH} = (0), (120)$$

แล้วทำการแปลงกลับจากค่าสัมประสิทธิ์ที่กำหนดในส่วนของ การแปลงกลับซึ่งก็คือ

$$G(s) \text{ กำหนดค่าให้เป็น } (1, 1)$$

$$H(s) \text{ กำหนดค่าให้เป็น } (1, -1)$$

วิธีทำ

$$\text{ที่ } G(s) = (1, 1)$$

$$X(0) = (1)(0) = 0$$

$$X(1) = (1)(20) + (1)(0) = 20$$

$$X(2) = (1)(0) + (1)(20) + (0)(0) = 20$$

$$\text{ที่ } H(s) = (1, -1)$$

$$Y(0) = (1)(0) = 0$$

$$Y(1) = (1)(120) + (-1)(0) = 120$$

$$Y(2) = (1)(0) + (-1)(120) + (0)(0) = -120$$

จากต้นสมการทำการคูณ 2 เอาไว้เพื่อให้ได้ค่าที่แท้จริงเพราะฉะนั้น OUTPUT ต้องทำการหาร 2

$$\text{OUTPUT}(0) = (X+Y) / 2 = (0+0) / 2 = 0$$

$$\text{OUTPUT}(1) = (X+Y) / 2 = (20+120) / 2 = 70$$

$$\text{OUTPUT}(2) = (X+Y) / 2 = (20+(-120)) / 2 = -50$$

สังเกตว่าเมื่อทำการแปลงกลับคืนแล้วจะได้ค่าเท่าเดิม

การลดขั้นตอนการแปลงสมการของเวฟเล็ด

จากสมการเวฟเล็ด จะเห็นว่า การ Convolution และการ Down Sampling สามารถลดขนาดสมการเวฟเล็ดได้ ดังตัวอย่างต่อไปนี้ (1,1) กับข้อมูล (1,2)

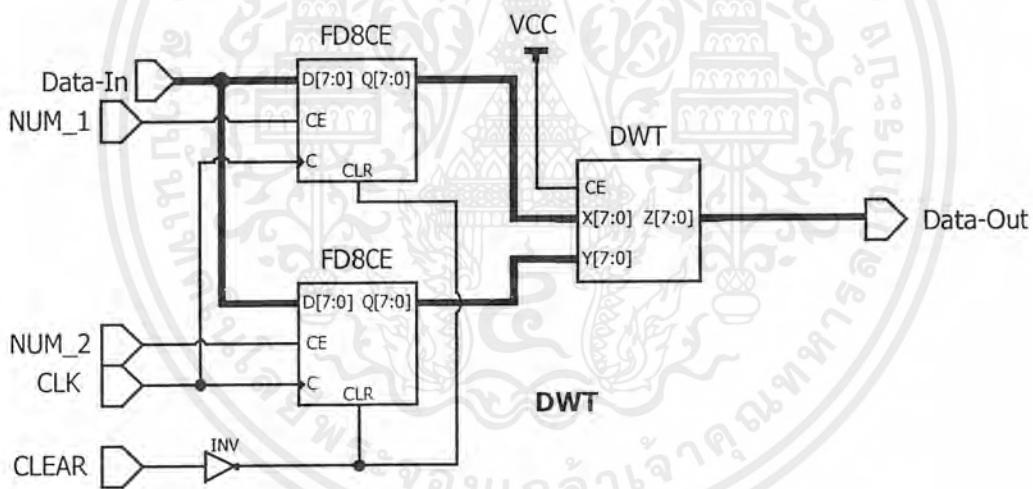
$$G(0) = 1 * 1 = 1$$

$$G(1) = (1 * 1) + (2 * 1) = 3$$

$$G(2) = (1 * 0) + (2 * 1) + (3 * 1) = 5$$

ทำการ Down Sampling จะได้ค่าของ $G(1)$ คือ (3) จะเห็นว่าค่าที่ได้จะเกิดจากการที่ข้อมูล (1,2) จะถูกนำมาบวกกัน ดังนั้นจะได้สมการใหม่คือ $Z = (x + Y)$ นั่นเอง

ส่วนของการแปลงกลับเวฟเล็ด จะพบว่าเมื่อเราลดรูปโดยตัดสัมประสิทธิ์ด้านสูงออกจะทำให้ลดรูปได้ครึ่งหนึ่ง จึงทำให้เราไม่ต้องทำการ Convolution สัมประสิทธิ์ด้านสูงแต่การแปลงกลับสัญญาณยังต้องนำข้อมูลที่ได้ออกมาหารด้วย 2



รูปที่ 3.13 วงจรแปลงสมการเวฟเล็ด DWT

จากรูปที่ 3.13 เป็นวงจรการแปลงสัญญาณเสียงให้อยู่ในรูปของสมการเวฟเล็ด โดยวงจรจะรับข้อมูลมาทีละ 2 ไบต์ เพื่อที่จะนำไปประมวลผลด้วยสมการเวฟเล็ด ในบล็อก DWT เพื่อจะนำมาแยกสัมประสิทธิ์ แล้วเลือกเก็บเฉพาะสัมประสิทธิ์ที่ต้องการจัดเก็บตามวิธีการของเวฟเล็ด เพื่อจะได้เก็บเฉพาะข้อมูลที่จำเป็น ทำให้ข้อมูลที่จะเก็บมีจำนวนน้อยลง สัญญาณเสียงที่จะเก็บก็จะสามารถจัดเก็บได้นานขึ้นนั่นเอง ซึ่งภายในบล็อก DWT จะถูกเขียนเป็นสมการด้วยภาษา VHDL ดังรูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการแปลง Wavelet

```

library IEEE;
use IEEE.std_logic_1164.all;

entity a1 is
    port (
        x: in INTEGER range -128 to 127;
        y: in INTEGER range -128 to 127;
        z: inout INTEGER range -128 to 127;
        ce: in BIT
    );
end a1;

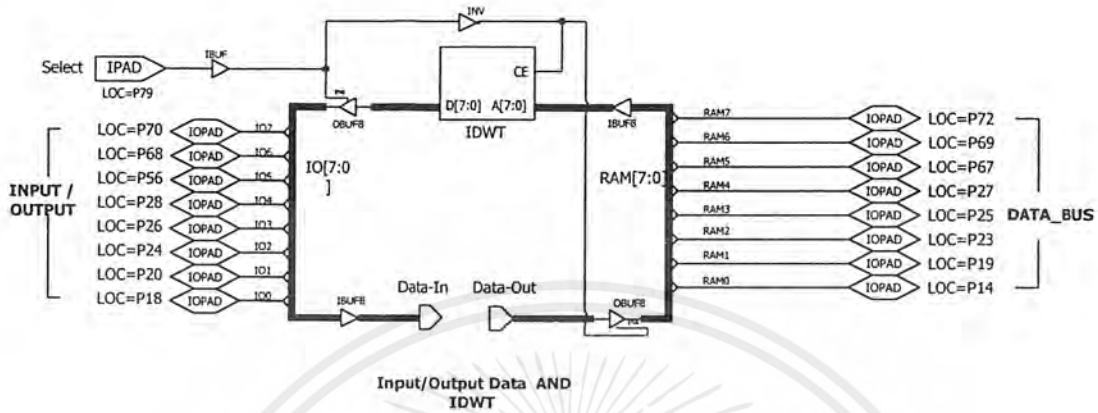
architecture a1_arch of a1 is
begin
    process(ce)
    begin
        if ce='1'
        then z<=x+y ;
        end if;
    end process;
end a1_arch;

```

รูปที่ 3.14 โปรแกรมการแปลงเวฟเล็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรรูปแปลงกลับสมการเวฟเล็ด IDWT



รูปที่ 3.15 วงจรส่วนของการแปลงกลับสมการเวฟเล็ด IDWT

จากรูปที่ 3.15 เมื่อขา Select ได้รับลอจิก 0 จะทำให้วงจรอ่านข้อมูลเข้าไปยังอุปกรณ์ FPGAs ผ่านทาง IO [7:0] บัส และส่งข้อมูลที่ได้ออกไปยังภาค DWT (การแปลงด้วยสมการเวฟเล็ด) ผ่านทางขา Data-In ผลลัพธ์ที่ได้จากการแปลงจะถูกส่งกลับมาทางขา Data-Out และส่งไปเก็บยังหน่วยความจำภายนอกเพื่อทำการจัดเก็บข้อมูลที่เป็นสัมประสิทธิ์ที่เลือกไว้

เมื่อขา Select ได้รับลอจิก 1 ข้อมูลจะถูกอ่านมาจากหน่วยความจำภายนอกเข้ามายังอุปกรณ์ FPGAs ข้อมูลจะถูกส่งเข้าไปยังบล็อก IDWT เพื่อทำการแปลงกลับสมการของเวฟเล็ด และจะได้ออกมาเป็นสัญญาณเสียงที่ผ่านการลดรูปด้วยสมการของเวฟเล็ดแล้ว จากนั้นข้อมูลจะถูกส่งออกไปแสดงผลยังอุปกรณ์ภายนอกผ่านทางบัส IO [7:0] ซึ่งบล็อก IDWT จะถูกเขียนเป็นสมการด้วยภาษา VHDL ดังรูปที่ 3.16

โปรแกรมการแปลง Invert Wavelet

```

library IEEE;
use IEEE.std_logic_1164.all;

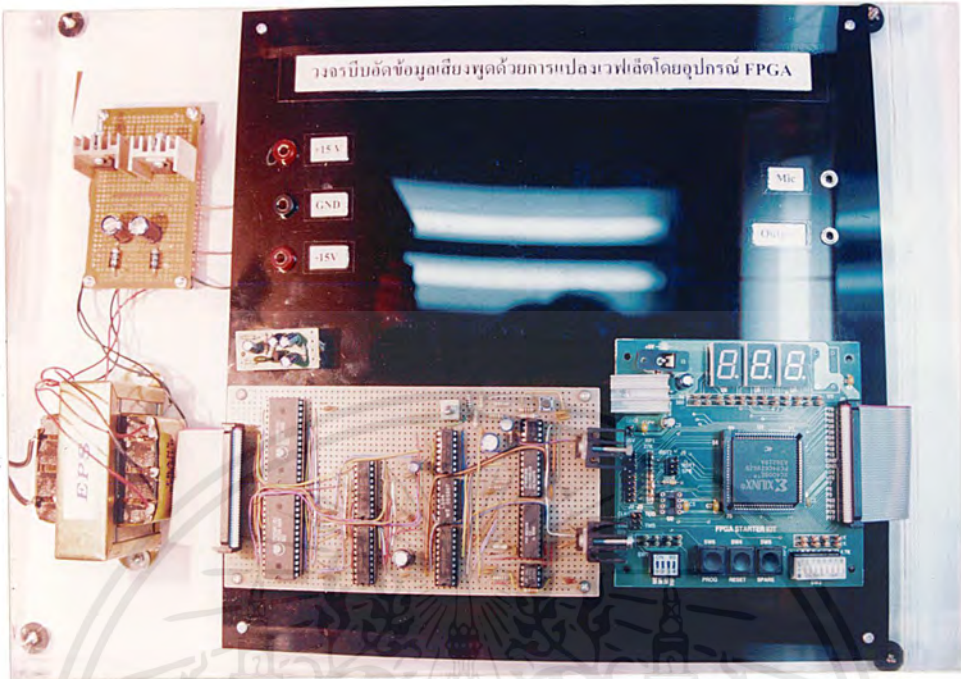
entity Untitled is
    port (
        a: in STD_LOGIC_VECTOR (7 downto 0);
        d: out STD_LOGIC_VECTOR ( 7 downto 0);
        ce: in STD_LOGIC
    );
end Untitled;

architecture Untitled_arch of Untitled is
begin
    d(7)<=a(0)and ce;
    d(6)<=a(7);
    d(5)<=a(6);
    d(4)<=a(5);
    d(3)<=a(4);
    d(2)<=a(3);
    d(1)<=a(2);
    d(0)<=a(1);
end Untitled_arch;

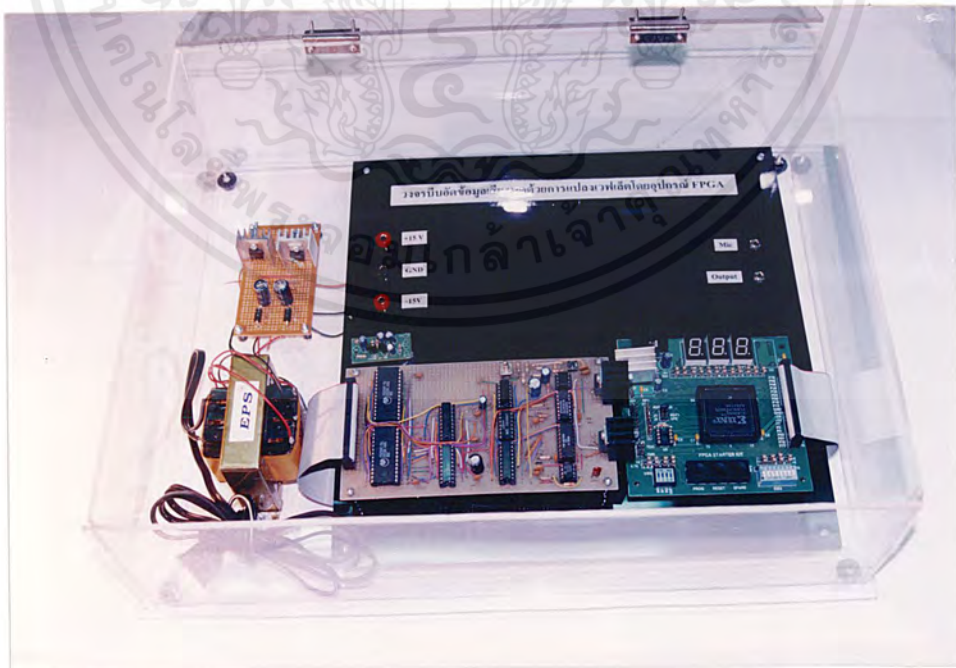
```

รูปที่ 3.16 โปรแกรมการแปลง Invert Wavelet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 บอร์ดอุปกรณ์ FPGAs ที่ต่อใช้งาน



รูปที่ 3.18 วงจรที่ทดสอบการทำงานทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

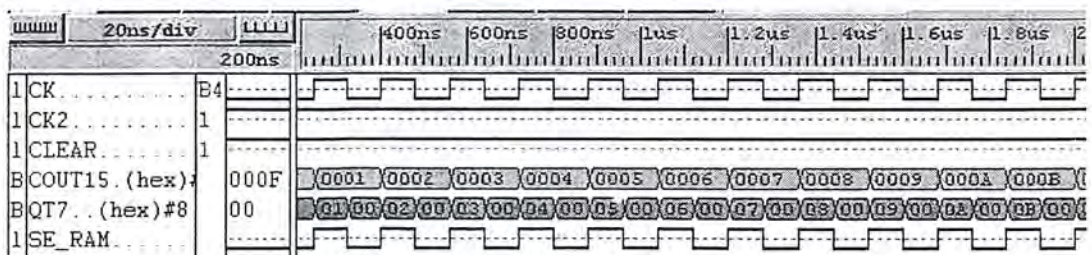
การทดลองและผลการทดลอง

จากการออกแบบอุปกรณ์บีบอัดข้อมูลเสียงด้วยอุปกรณ์ FPGAs โดยใช้สมการเวฟเล็ต หลังจากการออกแบบมา ได้แบ่งวงจรที่ออกแบบเป็น 2 ส่วนใหญ่ๆ ได้แก่ ส่วนที่เป็นวงจรภายในอุปกรณ์ FPGAs และส่วนที่เป็นอุปกรณ์เชื่อมต่อเพิ่มเติมภายนอก โดยการจะทำให้อุปกรณ์หรือวงจรทั้งหมดทำงานร่วมกันได้นั้นต้องมีการทดสอบการทำงานของแต่ละวงจรและแต่ละภาค แล้วทำการตรวจวัดสัญญาณอินพุตและเอาต์พุตให้มีการทำงานตรงตามที่ได้การออกแบบมา ซึ่งวงจรภายในของอุปกรณ์ FPGAs ก็จะเป็นส่วนที่สร้างมาจากสมการ VHDL และ Schematic ส่วนวงจรภายนอกก็เป็นส่วนเป็นวงจรแปลงสัญญาณแอนาล็อกเป็นสัญญาณดิจิทัล วงจรแปลงกลับสัญญาณดิจิทัลเป็นสัญญาณแอนาล็อก วงจรโลว์พาสฟิลเตอร์ และวงจรต่างๆที่ทำการติดต่อกับหน่วยความจำภายนอก

4.1 ผลการทดลอง VHDL และ Schematic ของอุปกรณ์ FPGAs

จากบทที่ 3 ได้ออกแบบการทำงานโดยแบ่งเป็นวงจรภายนอกและวงจรภายในอุปกรณ์ FPGAs โดยในส่วนของวงจรภายในได้จำลองการทำงานแต่ละส่วน และทดสอบการทำงานโดยวิธีการป้อนสัญญาณต่างๆ ให้แต่ละภาคทั้งภาคควบคุมสัญญาณแอดเดรสและภาควงจรถีโคคสัญญาณเวลาเพื่อไปแสดงผลยัง 7-Segment ภาคของการแปลงข้อมูลกับสมการเวฟเล็ต รวมทั้งการติดต่อกับหน่วยความจำภายนอก และการแปลงกลับเวฟเล็ตซึ่งได้ผลการทดลองดังนี้

4.1.1 ส่วนควบคุมสัญญาณแอดเดรส

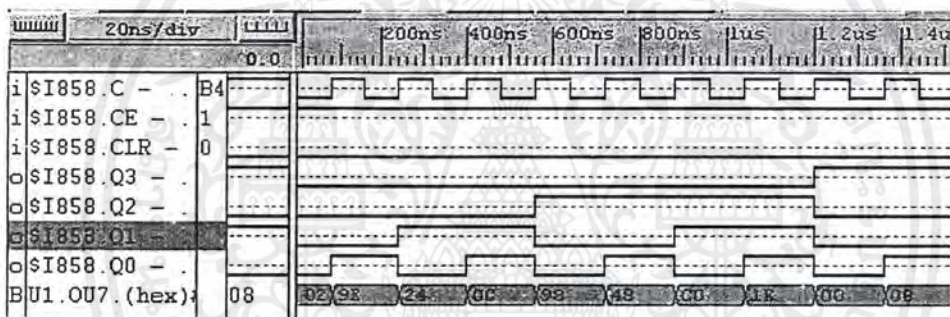


รูปที่ 4.1 สัญญาณของส่วนควบคุมสัญญาณแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 เป็นการจำลองการทำงานโดยทดลองป้อนสัญญาณนาฬิกาเข้าที่ขา CK และกำหนดให้ขา CLEAR เป็นลอจิก 1 จะทำให้เกิดสัญญาณที่ขา COUT15 เป็นสัญญาณแค่น์เตอร์ 16 บิต จาก 0000H ถึง FFFFH ซึ่งจะทำหน้าที่เป็นสัญญาณแอดเดรสนั่นเอง จากนั้นจะส่งไปยังวงจรมัลติเพล็กซ์ซึ่งจะทำให้เอาท์พุทที่ได้ออกไปยังขา QT7 โดยจะทำการส่งแอดเดรสไบต์ต่ำออกไปก่อน ในขณะที่ขา SE_RAM เป็นลอจิก 1 แล้วจะทำการส่งแอดเดรสไบต์สูงตามออกไป ในขณะที่ขา SE_RAM เป็นลอจิก 0 โดยจะนำสัญญาณแอดเดรสทั้งไบต์ต่ำและไบต์สูงนี้ไปใช้ควบคุมการโอนถ่ายข้อมูลกับหน่วยความจำภายนอก

4.1.2 ส่วนของวงจรถิโคคสัญญาณเวลาเพื่อไปแสดงผลยัง 7- Segment



รูปที่ 4.2 สัญญาณของวงจรถิโคคเพื่อแสงผลไปยัง 7- Segment

จากรูปที่ 4.2 เป็นสัญญาณจากวงจรมับ 4 บิต ซึ่งนำไปดีโคคเพื่อที่จะนำไปควบคุมการแสดงผลของ 7- Segment นับเวลาของการเก็บข้อมูลและการแสดงผลออกทางภายนอก โดยสัญญาณจะออกจากขา U1.OU7 ซึ่งการดีโคคแสดงได้ตามตารางที่ 4.1 ดังนี้

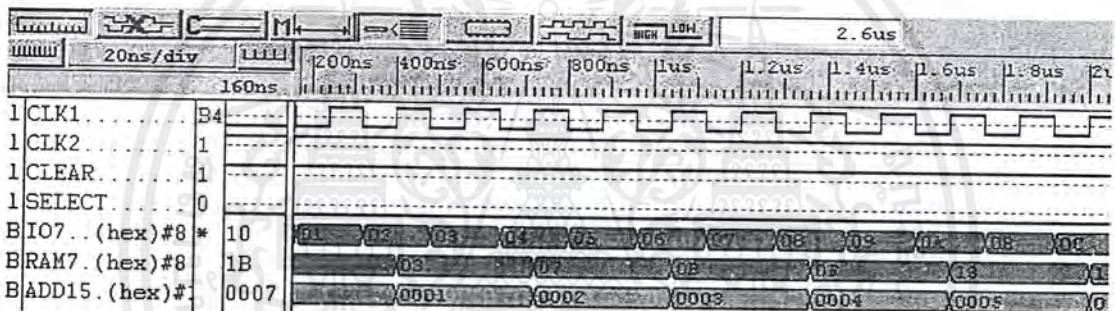
ตารางที่ 4.1 การนับแบบไบนารีและการนำไปแสดงผลออก 7- Segment

หมายเลข	รหัส Binary	รหัส 7 - Segment
0	0000	02
1	0001	9E
2	0010	24
3	0011	0C
4	0100	98

ตารางที่ 4.1 (ต่อ) การนับแบบไบนารีและการนำไปแสดงผลออก 7-Segment

หมายเลข	รหัส Binary	รหัส 7-Segment
5	0101	48
6	0110	C0
7	0111	1E
8	1000	00
9	1001	08

4.1.3 ส่วนของการแปลง DWT และจัดเก็บข้อมูลในหน่วยความจำภายนอก



รูปที่ 4.3 สัญญาณส่วนของการแปลง DWT และจัดเก็บข้อมูลในหน่วยความจำภายนอก

จากรูปที่ 4.3 เป็นผลการทดลองของการแปลงสมการเวฟเล็ด จะเห็นว่าเมื่อข้อมูลเข้ามาที่ขา IO7 วงจรจะทำการหน่วงสัญญาณข้อมูลให้สามารถรับข้อมูลได้ 2 ครั้ง หลังจากนั้นวงจรจะทำการแปลงเวฟเล็ด และส่งข้อมูลที่ได้ออกไปยังขา RAM 7 ซึ่งเป็นขาที่ใช้ติดต่อกับบัสข้อมูลของหน่วยความจำภายนอก และขาที่ควบคุมบัสแอดเดรสก็จะถูกเพิ่มค่าขึ้นทีละหนึ่งจะสังเกตได้ที่ขา ADD15 เพื่อรอรับข้อมูลที่แอดเดรสต่อไป

จากรูปที่ 4.3 เราทำการคำนวณได้โดยใช้สมการเวฟเล็ด ที่ลดขั้นตอนแล้วคือ $z = x + y$ โดยให้ x คือข้อมูลตัวที่ 1 และให้ y เป็นข้อมูลตัวที่ 2 สัญญาณข้อมูลนี้มาจากขา IO7 ดังนั้น ในแอดเดรสที่

$$0000H = 00 + 00 = 00H$$

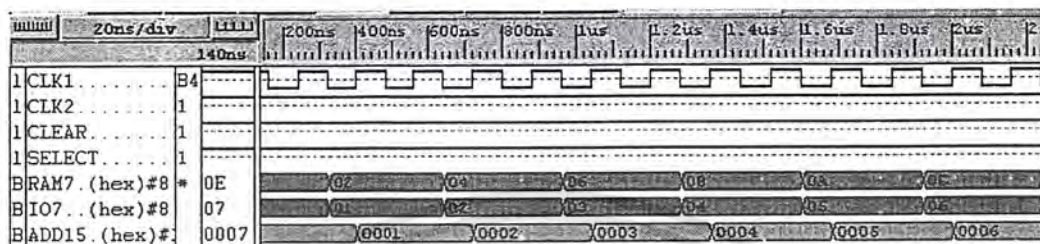
$$0001H = 01 + 02 = 03H$$

$$0002H = 03 + 04 = 07H$$

$$0003H = 05 + 06 = 0BH$$

วงจรมีจะคำนวณไปเรื่อยจนถึง แอดเดรสที่ FFFFH เอาที่พืทที่ได้จะปรากฏที่ขา RAM7

4.1.4 ส่วนของการแปลงกลับสมการเวฟเล็ทของข้อมูลในหน่วยความจำ



รูปที่ 4.4 สัญญาณส่วนของการแปลงกลับสมการเวฟเล็ท

จากรูปที่ 4.4 จะพบว่าข้อมูลจากหน่วยความจำจะผ่านบัสข้อมูลมายังขา RAM7 และจะถูกแปลงกลับโดยสมการแปลงกลับเวฟเล็ท และจะถูกนำออกไปยังขา IO7 เพื่อส่งต่อไปยังอุปกรณ์ภายนอก จากนั้นบัสแอดเดรสจะถูกเพิ่มค่าขึ้นทีละหนึ่งเพื่อนำข้อมูลต่อไปมาทำการแปลงกลับเวฟเล็ทให้ได้สัญญาณตามเดิม

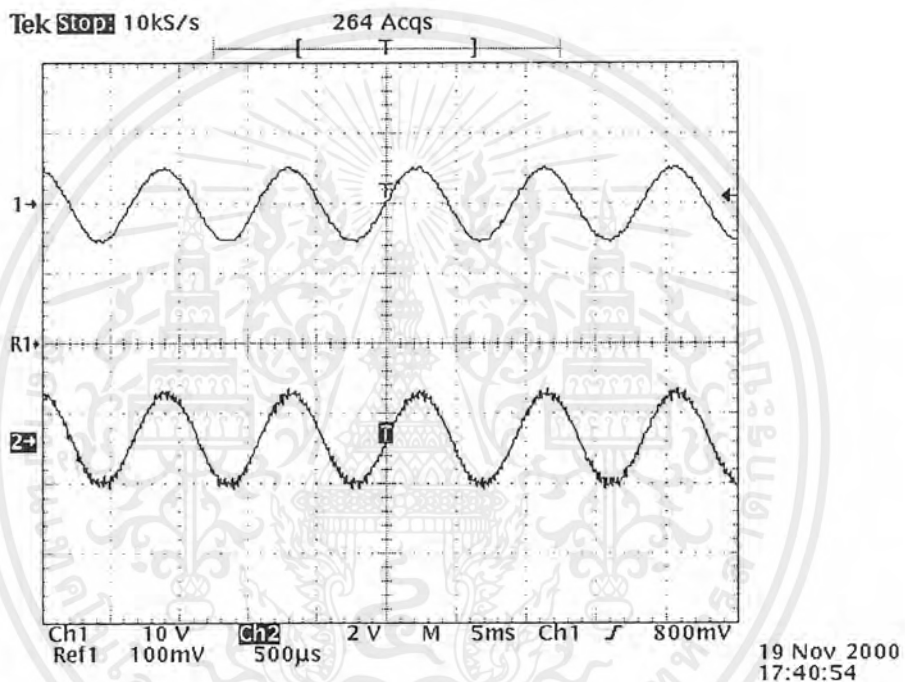
จากรูปที่ 4.4 การแปลงกลับเวฟเล็ทเราจะใช้วิธีการหาร 2 กลับให้ข้อมูลเป็นดั้งเดิมโดยอ่านจากขา RAM7 โดยผ่านแอดเดรสตามที่ขา ADD15 กำหนดรูปแบบการคำนวณจะเป็นดังนี้

$$\text{ขา IO7} = (\text{RAM7}) / 2$$

ขาแอดเดรสที่	0000H	01 = 02 / 2
	0001H	02 = 04 / 2
	0002H	03 = 06 / 2
	0003H	04 = 08 / 2

4.2 ผลการทดลองวัดสัญญาณต่างๆจากอุปกรณ์ภายนอก

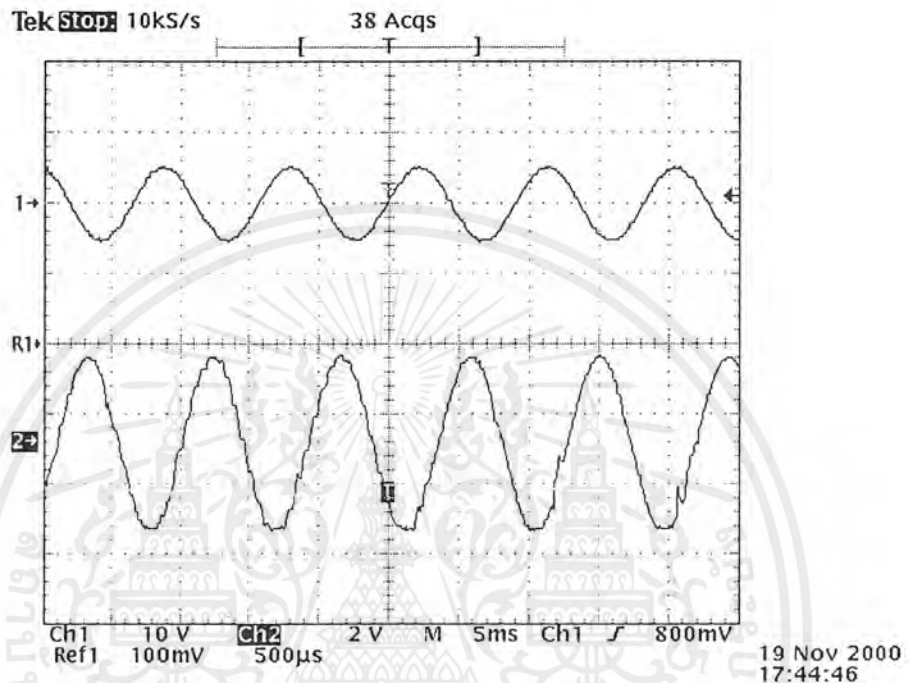
4.2.1 การแปลงสัญญาณแอนาลอกเป็นดิจิตอลและแปลงกลับดิจิตอลเป็นสัญญาณแอนาลอก



รูปที่ 4.5 สัญญาณอินพุตเทียบกับสัญญาณที่ผ่านการแปลงเป็นดิจิตอลและแปลงกลับแล้ว

จากรูปที่ 4.5 สัญญาณด้านบนคือสัญญาณไซน์ หรือสัญญาณแอนาลอกที่ใช้แทนสัญญาณเสียงในการทดสอบ ที่นำมาผ่านวงจรแปลงสัญญาณแอนาลอกให้เป็นสัญญาณดิจิตอลขนาด 8 บิต โดยใช้ไอซี ADC 908 จากนั้นก็จะผ่านวงจรแปลงสัญญาณดิจิตอลขนาด 8 บิตให้กลายเป็นสัญญาณแอนาลอก โดยที่จะยังมีสัญญาณแซมเปิ้ลถึงความถี่สูงคอยรบกวนอยู่ จึงต้องผ่านวงจรโลว์พาสฟิลเตอร์เพื่อดึงสัญญาณความถี่สูงลงกราวด์ ทำให้ได้เอาต์พุตที่มีสัญญาณใกล้เคียงกับสัญญาณอินพุต ดังรูปข้างล่างแต่จะยังเหลือสัญญาณความถี่สูงอยู่บ้างเล็กน้อย

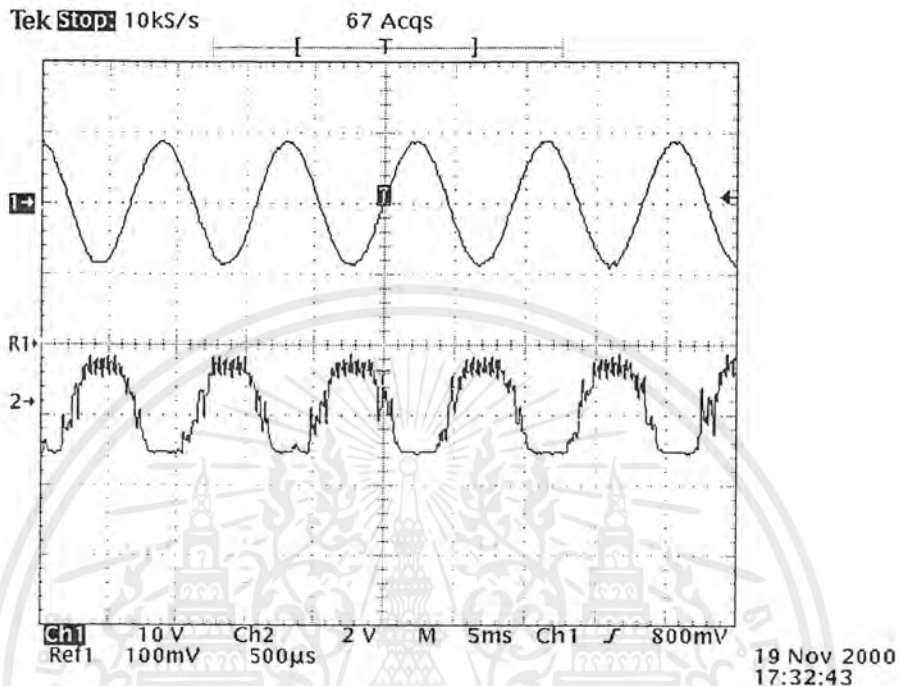
4.2.2 การแปลงสัญญาณแอนาล็อกเป็นดิจิตอลและแปลงกลับดิจิตอลเป็นสัญญาณแอนาล็อกโดยผ่านการเก็บในหน่วยความจำภายนอก



รูปที่ 4.6 สัญญาณอินพุตเทียบกับสัญญาณที่อ่านออกมาจากหน่วยความจำภายนอก

จากรูปที่ 4.6 เมื่อทำการป้อนสัญญาณไซน์ซึ่งเป็นสัญญาณแอนาล็อกในรูปแบบให้วงจร จะถูกนำไปแปลงเป็นสัญญาณดิจิตอลขนาด 8 บิต แล้วเมื่อได้รับสัญญาณควบคุมที่คอยควบคุม การเขียนข้อมูล หน่วยความจำภายนอกก็จะทำการเก็บข้อมูลขนาด 8 บิตไปไว้ในหน่วยความจำ จากนั้นหากได้รับสัญญาณควบคุมที่คอยควบคุมการอ่านข้อมูลที่ส่งมายังหน่วยความจำภายนอกก็ จะทำการส่งข้อมูลขนาด 8 บิตที่เก็บไว้ออกมา ผ่านวงจรแปลงสัญญาณดิจิตอลเพื่อเป็นสัญญาณ แอนาล็อกเพื่อให้ได้สัญญาณไซน์ตามเดิมและผ่านวงจรกรองความถี่สูงลงกราวด์ แล้วผ่าน กระบวนการโลว์พาสฟิลเตอร์อีกทีหนึ่ง จะได้สัญญาณไซน์ที่ทำการเก็บค่าไว้ในหน่วยความจำ ภายนอก ดังรูปที่ 4.6 (ล่าง) จะเห็นว่ามีความผิดเพี้ยนไปจากเดิมบ้างเล็กน้อย เนื่องจาก สัญญาณรบกวนต่างๆ และความผิดพลาดในการติดต่อรับส่งข้อมูลบ้าง

4.2.3 ผลการทดลอง การลดรูปข้อมูลด้วยสมการเวฟเล็ด



รูปที่ 4.7 สัญญาณอินพุตเทียบกับสัญญาณเอาต์พุตที่ผ่านการแปลงสมการเวฟเล็ดแล้ว

จากรูปที่ 4.7 เมื่อป้อนสัญญาณไซน์ซึ่งเป็นสัญญาณแอนาลอกในรูปแบบนาฬิกาบวจร จะถูกนำไปแปลงเป็นสัญญาณดิจิทัลขนาด 8 บิต จากนั้นจะถูกส่งไปส่วนของการประมวลผลด้วยสมการเวฟเล็ดซึ่งเป็นวงจรที่สร้างขึ้นภายในอุปกรณ์ FPGAs เพื่อทำการลดค่าของข้อมูล แล้วจะถูกส่งไปเก็บไว้ในหน่วยความจำภายนอก และเมื่อต้องการจะอ่านข้อมูลที่เก็บไว้ในหน่วยความจำก็จะเรียกข้อมูลที่เก็บไว้และถูกลดขนาดข้อมูลแล้วในหน่วยความจำออกมา เพื่อที่จะนำมาผ่านกระบวนการในการแปลงกลับสมการเวฟเล็ด เป็นสัญญาณดิจิทัลขนาด 8 บิต แล้วจึงส่งมาแปลงให้เป็นสัญญาณแอนาลอกรูปคลื่นไซน์ตามเดิมด้วยวงจรแปลงสัญญาณดิจิทัลเป็นแอนาลอก จากนั้นจะผ่านในส่วนกระบวนการของวงจรโลว์พาสฟิลเตอร์ดังกล่าว ซึ่งจะได้สัญญาณรูปคลื่นไซน์ตามรูปที่ 4.7 (ล่าง)

จะเห็นว่าสัญญาณมีรูปคลื่นที่ค่อนข้างจะแตกต่างกับสัญญาณเดิม สาเหตุก็เพราะสัญญาณจะผิดเพี้ยนไปบ้างจากการติดต่อกับหน่วยความจำภายนอก หรือจากการแปลงไปกลับระหว่างข้อมูลแอนาลอกและดิจิทัล และสัญญาณรบกวนต่างๆอาจมาจากอุปกรณ์ภายนอกได้

เพราะใช้อุปกรณ์ต่างๆ ไปมาเชื่อมต่อทำงานร่วมกัน แต่รูปคลื่นสัญญาณก็ยังคงแสดงให้เห็นว่าเป็นสัญญาณตัวเดิมอยู่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

5.1 บทสรุป

การออกแบบการสร้างอุปกรณ์เก็บและบีบอัดเสียงพูดโดยใช้สมการเวฟเล็ตได้นำเสนอวิธีการออกแบบวงจรควบคุมโดยใช้ภาษา VHDL เข้ามาช่วยในการออกแบบวงจรควบคุมแล้วจึงนำไปดาวน์โหลดลงในอุปกรณ์ FPGAs เพื่อทำการลดขนาดของวงจรลง ส่วนที่ไม่สามารถสร้างขึ้นภายในอุปกรณ์ FPGAs ได้ก็จะทำการต่อเป็นอุปกรณ์เพิ่มเติมภายนอก ประกอบไปด้วยส่วนแรก คือส่วนปริโมโครโฟนที่จะเปลี่ยนสัญญาณเสียงพูดเป็นสัญญาณไฟฟ้าให้มีขนาดแรงพอที่จะส่งไปยังส่วนของวงจรที่เปลี่ยนสัญญาณจากแอนาล็อกเป็นสัญญาณดิจิทัลขนาด 8 บิต จากนั้นก็จะทำการส่งสัญญาณดิจิทัลที่เป็นสัญญาณอินพุต 8 บิตส่งเข้าไปยังส่วนของภาคอุปกรณ์ภายในของ FPGAs เพื่อจะนำไปประมวลผลว่าจะทำการเขียนลงสู่หน่วยความจำแรมที่กำหนดตามสมการของเวฟเล็ตและทำการประมวลผลต่างๆตามขั้นตอนของสมการเวฟเล็ต โดยอุปกรณ์ภายในของ FPGAs จะถูกสร้างขึ้นตามสมการของเวฟเล็ต คือ การสร้างให้อุปกรณ์ภายในสามารถทำงานตรงกับสมการเวฟเล็ตนั่นเอง โดยต้องมีการลดรูปตรงตามสมการเวฟเล็ตไปเก็บในแรมภายนอก และเมื่อจะแปลงกับออกมาเป็นสัญญาณเสียงพูดหรือแอนาล็อกตามเดิมก็ต้องมีการอ่านข้อมูลออกมาจากแรมภายนอก แล้วทำการแปลงสัญญาณที่ลดรูปแล้วให้อยู่ในรูปเดิมโดยการเพิ่มสัมประสิทธิ์เข้าไปในข้อมูลที่ลดรูป ตามหลักการของเวฟเล็ต จากนั้นเมื่อได้สัญญาณดิจิทัลที่เป็นสัญญาณเอาต์พุต 8 บิต ก็นำไปเข้าวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนาล็อก แล้วจึงมาผ่านวงจรโลว์พาสฟิลเตอร์เพื่อกรองความถี่แซมปลิงออกอีกทีหนึ่ง จึงจะได้สัญญาณเสียงที่ทำการจัดเก็บเข้ามาในตอนแรก โดยที่จะทำการอ่านหรือเขียนข้อมูลจากแรม ก็สามารถทำได้โดยการเลือกสวิทช์ Select ที่ออกแบบไว้ควบคุมการติดต่อกับแรมภายนอก รวมทั้งสวิทช์เคลียร์การทำงานของ อุปกรณ์ FPGAs และให้แสดงผลออกมาใหม่ได้จากสวิทช์เคลียร์นี้เอง

ในส่วนของการซอฟต์แวร์หรืออุปกรณ์ภายใน FPGAs และส่วนของฮาร์ดแวร์ หรืออุปกรณ์ภายนอก FPGAs เมื่อทำการทดลองแล้วปรากฏว่าในส่วนของการซอฟต์แวร์นั้นสามารถทำงานได้ตามปกติ เมื่อนำมารวมเข้าเป็นวงจรฮาร์ดแวร์ที่สมบูรณ์สัญญาณที่ได้มีความผิดเพี้ยนจากเดิมไปบ้างเล็กน้อย ทั้งนี้เนื่องมาจากค่าความผิดพลาดของอุปกรณ์ทางด้านฮาร์ดแวร์นั่นเอง

5.2 ปัญหาและแนวทางแก้ไข

1. เนื่องจากว่าในการออกแบบวงจรด้วยภาษา VHDL ในบางครั้งในการสร้างวงจรมันการเดินสายสัญญาณในการเชื่อมต่ออุปกรณ์แต่ละตัวนั้นจะมีปัญหาที่อาจทำให้วงจรไม่ได้ต่อถึงกัน

แนวทางการแก้ไข

ควรจะมีการตรวจสอบหรือแก้ไขสายสัญญาณหรือสายบัสที่ทำหน้าที่ในการเชื่อมต่อระหว่างอุปกรณ์ว่าเชื่อมต่อกันดีหรือไม่ มิเช่นนั้นอาจจะทำให้วงจรทำงานไม่ครบวงจร

2. อุปกรณ์ แหล่งจ่ายไฟเลี้ยงวงจรมีความร้อนเกิดขึ้นในขณะที่ใช้งานตามปกติเป็นเวลานานๆ

แนวทางการแก้ไข

เมื่อมีการกำหนดให้ภายในอุปกรณ์ FPGAs มีการทำงานของวงจรที่มีขนาดใหญ่ หรือใช้ อุปกรณ์ต่อรวมจำนวนมากก็จะเกิดการใช้กระแสความร้อนเป็นสาเหตุหนึ่งที่ทำให้อุปกรณ์ต่างๆ ในส่วนของภาคจ่ายไฟเลี้ยงเกิดความร้อน ควรกำหนดวงจรและอุปกรณ์ใช้งานให้มีขนาดเหมาะสม รวมถึงการใช้ Heat Sink ที่ช่วยในการขจัดความร้อนได้ดี

3. ในส่วนของการออกแบบวงจรแปลงสัญญาณจากแอนะล็อกเป็นสัญญาณดิจิทัลนั้นเกิดปัญหา ในส่วนของการตอบสนองของการที่สัญญาณอินพุตและสัญญาณแอมพลีตูด้อยในช่วงเดียวกันหรือมีความแตกต่างกันน้อย เช่น เสียง 0-4 KHz แต่แอมพลีตูด้อยด้วยความถี่ 16 KHz ทำให้ความผิดพลาดของสัญญาณที่ออกมามีมาก

แนวทางการแก้ไข

หากสามารถเพิ่มความถี่แอมพลีตูด้อยจะทำให้คุณภาพเสียงที่จะได้จะมีคุณภาพดีขึ้น แต่ก็ต้องดู ความสามารถของหน่วยความจำที่สามารถจัดเก็บได้ ดังนั้นต้องเพิ่มความถี่แอมพลีตูด้อยในช่วงที่ยอมรับได้คือ ฟังเสียงรู้เรื่องและหน่วยความจำสามารถจัดเก็บได้

4. ในส่วนของการออกแบบวงจรแปลงสัญญาณจากดิจิทัลเป็นสัญญาณแอนะล็อกนั้นจะเกิด ปัญหาใน ส่วนของเอาต์พุตที่มีระดับของสัญญาณที่ต่ำ

แนวทางการแก้ไข

ให้ใช้ลำโพงที่มีเพาเวอร์แอมป์ขยายสัญญาณในตัวและสามารถปรับระดับได้ เช่น ลำโพงที่ ต่อร่วมกับคอมพิวเตอร์ทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เกิดสัญญาณรบกวนที่สัญญาณเอาต์พุต เนื่องจากสัญญาณแชนเปลิ่งยังคงหลงเหลืออยู่เมื่อผ่าน วงจร โลว์พาสฟิลเตอร์มาแล้ว

แนวทางในการแก้ไข

ให้เพิ่มค่าตัวเก็บประจุในภาคโลว์พาสฟิลเตอร์ สัญญาณที่เป็นความถี่สูงก็จะถูกบายพาสลงกราวด์ไปหมด แต่ไม่ควรใช้ค่าตัวเก็บประจุที่มีค่ามากเกินไป ถ้าไม่เช่นนั้นแล้วสัญญาณที่ออกมาจะเหลือเพียงสัญญาณความถี่ต่ำเพียงอย่างเดียว ทำให้สัญญาณเสียงที่ออกมามีเสียงที่ทึบหรือบีบมากเกินไป ดังนั้นในการเพิ่มค่าตัวเก็บประจุควรพิจารณาความเหมาะสม โดยอาจจะยอมให้สัญญาณความถี่สูงผ่านได้บ้าง เพื่อให้ได้สัญญาณเสียงที่มีคุณภาพดีขึ้นกว่าเสียงที่เบาของสัญญาณความถี่ต่ำเพียงอย่างเดียว

6. ในการสร้างสัญญาณนาฬิกาที่อาจเกิดการดำเนินงานที่ไม่พร้อมกัน การทำงานของอุปกรณ์แต่ละตัวจึงไม่ตรงกันและไม่แน่นอน

แนวทางในการแก้ไข

สร้างแหล่งกำเนิดสัญญาณนาฬิกาจากวงจรภายใน หรืออุปกรณ์ FPGAs แล้วใช้สัญญาณดังกล่าวไปควบคุมการทำงานของวงจรทั้งหมด รวมทั้งสัญญาณแชนเปลิ่งของภาคแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล ที่เป็นวงจรภายนอกด้วย

7. ในการทำงานของวงจรรวมแล้ว เมื่อทำงานเกิดความผิดพลาดเล็กน้อยทั้งนี้สาเหตุอาจจะเกิดจากสัญญาณรบกวนที่เกิดจากการติดต่อกับหน่วยความจำ RAM ภายนอก

แนวทางในการแก้ไข

ตรวจวัดสัญญาณที่ละจุด ว่าสัญญาณรบกวนมาจากตรงไหน เมื่อพบแล้วก็ทำการต่อตัวเก็บประจุรอมแหล่งจ่ายบริเวณนั้น เพื่อดึงสัญญาณรบกวนลงกราวด์ หรือทำการต่ออุปกรณ์ที่จะติดต่อกับหน่วยความจำภายนอกให้น้อยที่สุดก็จะทำให้สัญญาณรบกวนลดลงได้อีกมาก

5.3 แนวทางการพัฒนา

- 1) ในส่วนของการติดต่อกับหน่วยความจำหากต้องการเก็บสัญญาณเสียงให้มีเวลาหรือข้อมูลในการเก็บมากขึ้นก็สามารถทำการเพิ่มส่วนของหน่วยความจำภายนอกได้
- 2) ถ้าหากต้องการคุณภาพสัญญาณเสียงที่มีคุณภาพดีขึ้นกว่าเดิม สามารถทำได้โดยการเพิ่มความถี่แซมปลิง แต่จะทำให้เวลาในการจัดเก็บสัญญาณเสียงลดลง ก็สามารถเพิ่มเติมได้ด้วยการเพิ่มหน่วยความจำภายนอกได้อีกแต่ต้องออกแบบส่วนเข้ารหัสควบคุมการทำงานของไอซีหน่วยความจำภายนอกให้สามารถสลับกันทำงานได้อย่างเหมาะสม
- 3) สามารถลดขนาดข้อมูลได้มากกว่า 50 % โดยการแยกสัมประสิทธิ์เวฟเล็ตได้อีก แล้วจึงเพิ่มสัมประสิทธิ์ชดเชยก็จะทำให้เก็บข้อมูลได้เพิ่มมากขึ้นแต่คุณภาพของสัญญาณเสียงจะมีคุณภาพลดลงตามลำดับ เพราะสัมประสิทธิ์ที่จัดเก็บถูกลดจำนวนลง
- 4) พัฒนาวงจรแปลงสัญญาณแอนาลอกเป็นดิจิทัลให้มีคุณภาพดีขึ้นได้ ซึ่งจะขึ้นอยู่กับไอซีและวงจรที่ใช้งาน
- 5) พัฒนาวงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณแอนาลอกให้มีคุณภาพดีขึ้นได้ ซึ่งจะขึ้นอยู่กับไอซีและวงจรที่ใช้
- 6) สามารถเพิ่มจำนวนของไอซีที่ทำหน้าที่จ่ายแรงดันไฟเลี้ยงด้วยการต่อขนาน จะทำให้ลดภาระและกำลังตกคร่อมของไอซีจ่ายไฟเลี้ยงของวงจร และทำให้เกิดความร้อนลดลงได้
- 7) สามารถเพิ่มส่วนปรับระดับความแรงของสัญญาณเสียงได้ ด้วยการเพิ่มส่วนควบคุมแรงดันไฟเลี้ยงให้กับวงจรปริโมโครโฟนได้ จะทำให้เราสามารถที่จะควบคุมเสียงว่าจะต้องการเสียงดังหรือเสียงค่อยได้ตามต้องการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมถอดรหัส 7 Segment

```

library IEEE;
use IEEE.std_logic_1164.all;

entity bcd7 is
port (
    c0,c1,c2,c3: in bit;
    ou: out integer range 0 to 255
);
end bcd7;

architecture bcd7_arch of bcd7 is
begin
    process (c0,c1,c2,c3)
    begin
        if(c3 ='1'and c2 ='0' )and(c1 ='0'and c0 ='1' )then ou<=16;
        end if;
        if(c3 ='1'and c2 ='0' )and(c1 ='0'and c0 ='0' )then ou<=0;
        end if;
        if(c3 ='0'and c2 ='1' )and(c1 ='1'and c0 ='1' )then ou<=120;
        end if;
        if(c3 ='0'and c2 ='1' )and(c1 ='1'and c0 ='0' )then ou<=3;
        end if;
        if(c3 ='0'and c2 ='1' )and(c1 ='0'and c0 ='1' )then ou<=18;
        end if;
        if(c3 ='0'and c2 ='1' )and(c1 ='0'and c0 ='0' )then ou<=25;
        end if;
        if(c3 ='0'and c2 ='0' )and(c1 ='1'and c0 ='1' )then ou<=48;
        end if;
        if(c3 ='0'and c2 ='0' )and(c1 ='1'and c0 ='0' )then ou<=36;
        end if;
    end process;
end architecture bcd7_arch;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(c3='0'and c2='0' )and(c1='0'and c0='1' )then ou<=121;
        end if;
        if(c3='0'and c2='0' )and(c1='0'and c0='0' )then ou<=64;
        end if;
    end process;
end bcd7_arch;

```

โปรแกรม Counter 16 Bits

```

library IEEE;
use IEEE.std_logic_1164.all;

entity c8 is
    port (
        clk: in STD_LOGIC;
        en: in STD_LOGIC;
        cout: inout integer range 0 to 65535
    );
end c8;
architecture c8_arch of c8 is
    begin
        process (CLK, en)
            begin
                if en='0' then
                    cout <= 0;
                elsif CLK='1' and CLK'event then
                    cout <= cout + 1;
                end if;
            end process;
        end c8_arch;

```

โปรแกรมการแปลง Wavelet

```

library IEEE;
use IEEE.std_logic_1164.all;

entity a1 is
    port (
        x: in INTEGER range -128 to 127;
        y: in INTEGER range -128 to 127;
        z: inout INTEGER range -128 to 127;
        ce: in BIT
    );
end a1;

architecture a1_arch of a1 is
begin
    process(ce)
    begin
        if ce='1'
        then z<=x+y ;
        end if;
    end process;
end a1_arch;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมการแปลง Invert Wavelet

```

library IEEE;
use IEEE.std_logic_1164.all;

entity Untitled is
    port (
        a: in STD_LOGIC_VECTOR (7 downto 0);
        d: out STD_LOGIC_VECTOR ( 7 downto 0);
        ce: in STD_LOGIC
    );
end Untitled;

architecture Untitled_arch of Untitled is
begin
    d(7)<=a(0)and ce;
    d(6)<=a(7);
    d(5)<=a(6);
    d(4)<=a(5);
    d(3)<=a(4);
    d(2)<=a(3);
    d(1)<=a(2);
    d(0)<=a(1);
end Untitled_arch;

```

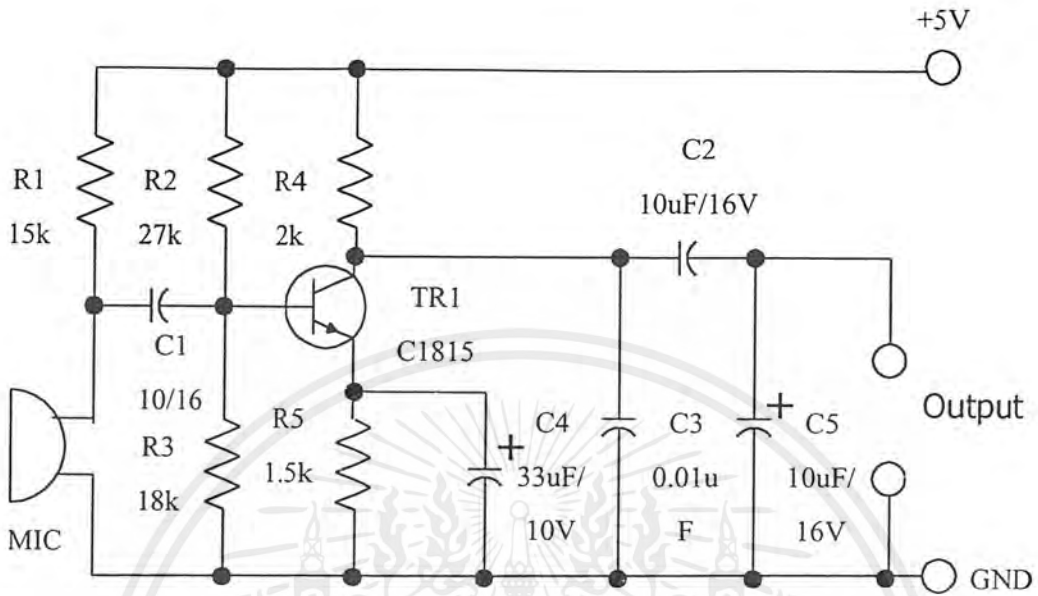
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



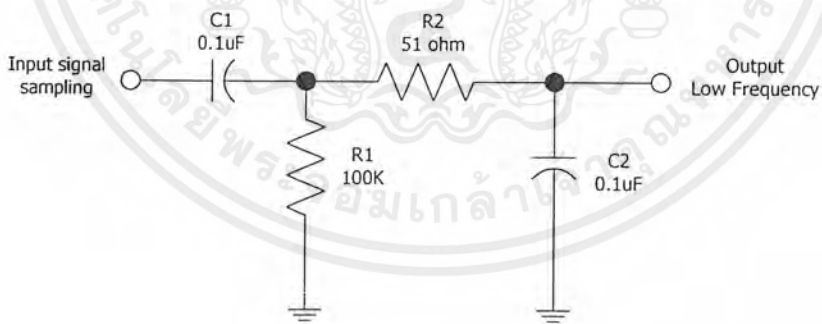
ภาคผนวก ข

วงจรและแผ่นวงจรพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

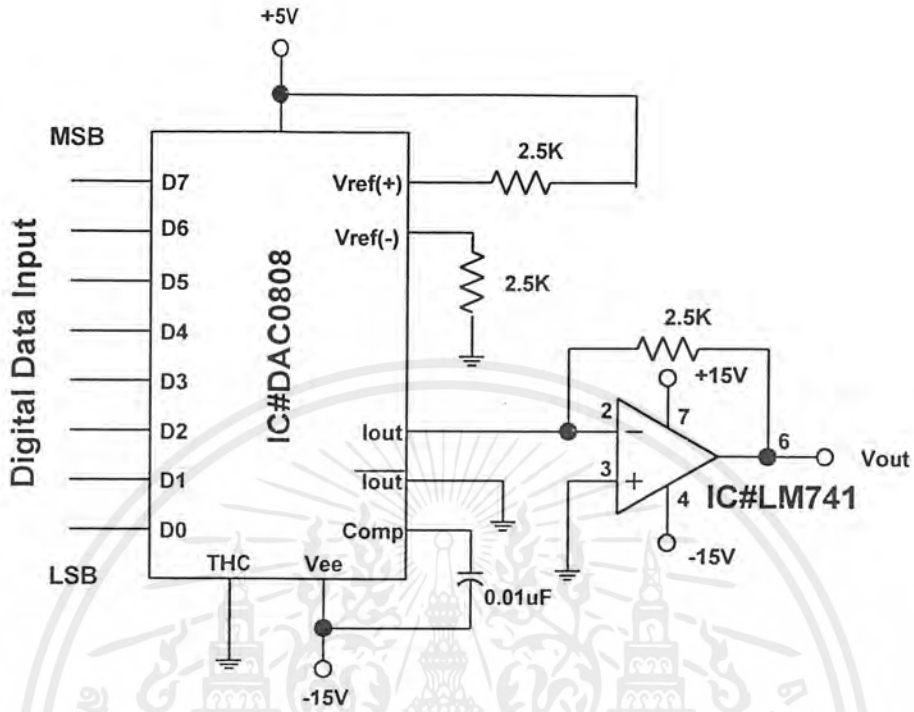


รูปที่ ข.1 วงจรปรีไมโครโฟน

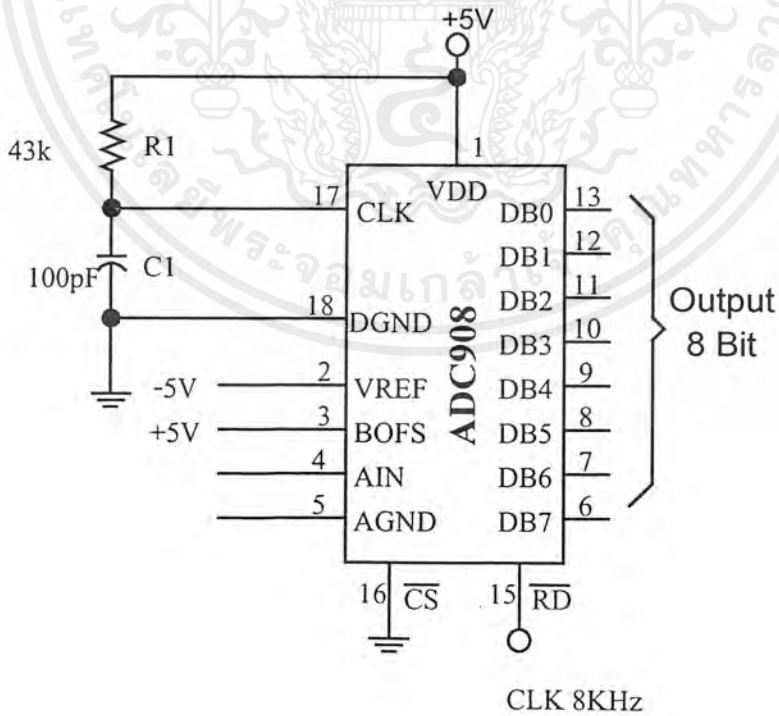


รูปที่ ข.2 วงจรโลว์พาสฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

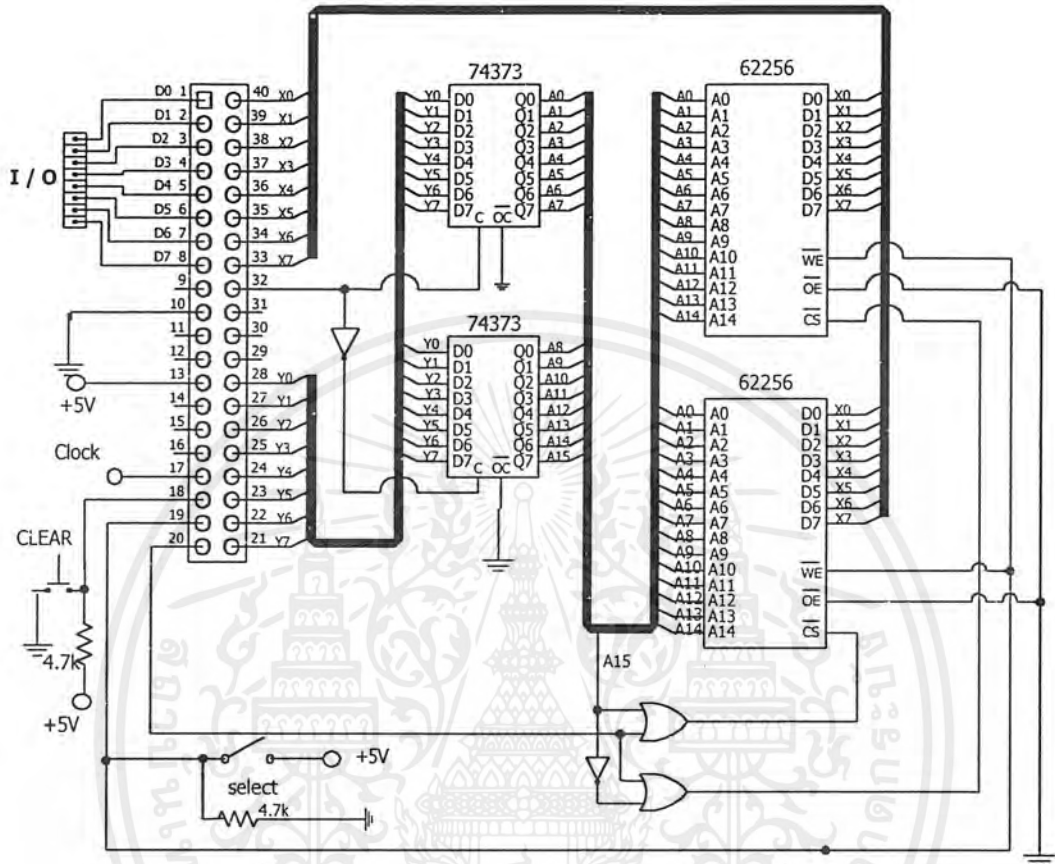


รูปที่ ข.3 วงจรใช้งานจริงของไอซี DAC 0808



รูปที่ ข.4 การต่อขาต่างๆใช้งานของไอซี ADC 908

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.5 วงจรที่ใช้ในการติดต่อระหว่างอุปกรณ์ FPGAs กับอุปกรณ์และหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ค

รายละเอียดข้อมูลและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



XC4000E and XC4000X Series Field Programmable Gate Arrays

May 14, 1999 (Version 1.6)

Product Specification

XC4000E and XC4000X Series Features

Note: Information in this data sheet covers the XC4000E, XC4000EX, and XC4000XL families. A separate data sheet covers the XC4000XLA and XC4000XV families. Electrical Specifications and package/pin information are covered in separate sections for each family to make the information easier to access, review, and print. For access to these sections, see the Xilinx WEBLINUX web site at

<http://www.xilinx.com/partinfo/databook.htm#xc4000>.

- System featured Field-Programmable Gate Arrays
 - Select-RAM™ memory: on-chip ultra-fast RAM with
 - synchronous write option
 - dual-port RAM option
 - Fully PCI compliant (speed grades -2 and faster)
 - Abundant flip-flops
 - Flexible function generators
 - Dedicated high-speed carry logic
 - Wide edge decoders on each edge
 - Hierarchy of interconnect lines
 - Internal 3-state bus capability
 - Eight global low-skew clock or signal distribution networks
- System Performance beyond 80 MHz
- Flexible Array Architecture
- Low Power Segmented Routing Architecture
- Systems-Oriented Features
 - IEEE 1149.1-compatible boundary scan logic support
 - Individually programmable output slew rate
 - Programmable input pull-up or pull-down resistors
 - 12 mA sink current per XC4000E output
- + Configured by Loading Binary File
 - Unlimited re-programmability
- Read Back Capability
 - Program verification
 - Internal node observability
- Backward Compatible with XC4000 Devices
- Development System runs on most common computer platforms
 - Interfaces to popular design environments
 - Fully automatic mapping, placement and routing
 - Interactive design editor for design optimization

Low-Voltage Versions Available

- Low-Voltage Devices Function at 3.0 - 3.6 Volts
- XC4000XL: High Performance Low-Voltage Versions of XC4000EX devices

Additional XC4000X Series Features

- Highest Performance — 3.3 V XC4000XL
- Highest Capacity — Over 180,000 Usable Gates
- 5 V tolerant I/Os on XC4000XL
- 0.35 μm SRAM process for XC4000XL
- Additional Routing Over XC4000E
 - almost twice the routing capacity for high-density designs
- Buffered Interconnect for Maximum Speed Blocks
- Improved VersaRing™ I/O Interconnect for Better Fixed Pinout Flexibility
- 12 mA Sink Current Per XC4000X Output
- Flexible New High-Speed Clock Network
 - Eight additional Early Buffers for shorter clock delays
 - Virtually unlimited number of clock signals
- Optional Multiplexer or 2-input Function Generator on Device Outputs
- Four Additional Address Bits in Master Parallel Configuration Mode
- XC4000XV Family offers the highest density with 0.25 μm 2.5 V technology

Introduction

XC4000 Series high-performance, high-capacity Field Programmable Gate Arrays (FPGAs) provide the benefits of custom CMOS VLSI, while avoiding the initial cost, long development cycle, and inherent risk of a conventional masked gate array.

The result of thirteen years of FPGA design experience and feedback from thousands of customers, these FPGAs combine architectural versatility, on-chip Select-RAM memory with edge-triggered and dual-port modes, increased speed, abundant routing resources, and new, sophisticated software to achieve fully automated implementation of complex, high-density, high-performance designs.

The XC4000E and XC4000X Series currently have 20 members, as shown in Table 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1: XC4000E and XC4000X Series Field Programmable Gate Arrays

Device	Logic Cells	Max Logic Gates (No RAM)	Max. RAM Bits (No Logic)	Typical Gate Range (Logic and RAM)*	CLB Matrix	Total CLBs	Number of Flip-Flops	Max. User I/O
XC4002XL	152	1,600	2,048	1,000 - 3,000	8 x 8	64	256	64
XC4003E	238	3,000	3,200	2,000 - 5,000	10 x 10	100	360	80
XC4005E/XL	466	5,000	6,272	3,000 - 9,000	14 x 14	196	616	112
XC4006E	608	6,000	8,192	4,000 - 12,000	16 x 16	256	768	128
XC4008E	770	8,000	10,368	6,000 - 15,000	18 x 18	324	936	144
XC4010E/XL	950	10,000	12,800	7,000 - 20,000	20 x 20	400	1,120	160
XC4013E/XL	1368	13,000	18,432	10,000 - 30,000	24 x 24	576	1,536	192
XC4020E/XL	1862	20,000	25,088	13,000 - 40,000	28 x 28	784	2,016	224
XC4025E	2432	25,000	32,768	15,000 - 45,000	32 x 32	1,024	2,560	256
XC4028EX/XL	2432	28,000	32,768	18,000 - 50,000	32 x 32	1,024	2,560	256
XC4036EX/XL	3078	36,000	41,472	22,000 - 65,000	36 x 36	1,296	3,168	288
XC4044XL	3800	44,000	51,200	27,000 - 80,000	40 x 40	1,600	3,840	320
XC4052XL	4598	52,000	61,952	33,000 - 100,000	44 x 44	1,936	4,576	352
XC4062XL	5472	62,000	73,728	40,000 - 130,000	48 x 48	2,304	5,376	384
XC4085XL	7448	85,000	100,352	55,000 - 180,000	56 x 56	3,136	7,168	448

* Max values of Typical Gate Range include 20-30% of CLBs used as RAM.

Note: All functionality in low-voltage families is the same as in the corresponding 5-Volt family, except where numerical references are made to timing or power.

Description

XC4000 Series devices are implemented with a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs). They have generous routing resources to accommodate the most complex interconnect patterns.

The devices are customized by loading configuration data into internal memory cells. The FPGA can either actively read its configuration data from an external serial or byte-parallel PROM (master modes), or the configuration data can be written into the FPGA from an external device (slave and peripheral modes).

XC4000 Series FPGAs are supported by powerful and sophisticated software, covering every aspect of design from schematic or behavioral entry, floor planning, simulation, automatic block placement and routing of interconnects, to the creation, downloading, and readback of the configuration bit stream.

Because Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs

where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening design and development cycles, and also offer a cost-effective solution for production rates well beyond 5,000 systems per month. For lowest high-volume unit cost, a design can first be implemented in the XC4000E or XC4000X, then migrated to one of Xilinx' compatible HardWire mask-programmed devices.

Taking Advantage of Re-configuration

FPGA devices can be re-configured to change logic function while resident in the system. This capability gives the system designer a new degree of freedom not available with any other type of logic.

Hardware can be changed as easily as software. Design updates or modifications are easy, and can be made to products already in the field. An FPGA can even be re-configured dynamically to perform different functions at different times.

Re-configurable logic can be used to implement system self-diagnostics, create systems capable of being re-configured for different environments or operations, or implement multi-purpose hardware for a given application. As an added benefit, using re-configurable FPGA devices simplifies hardware design and debugging and shortens product time-to-market.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

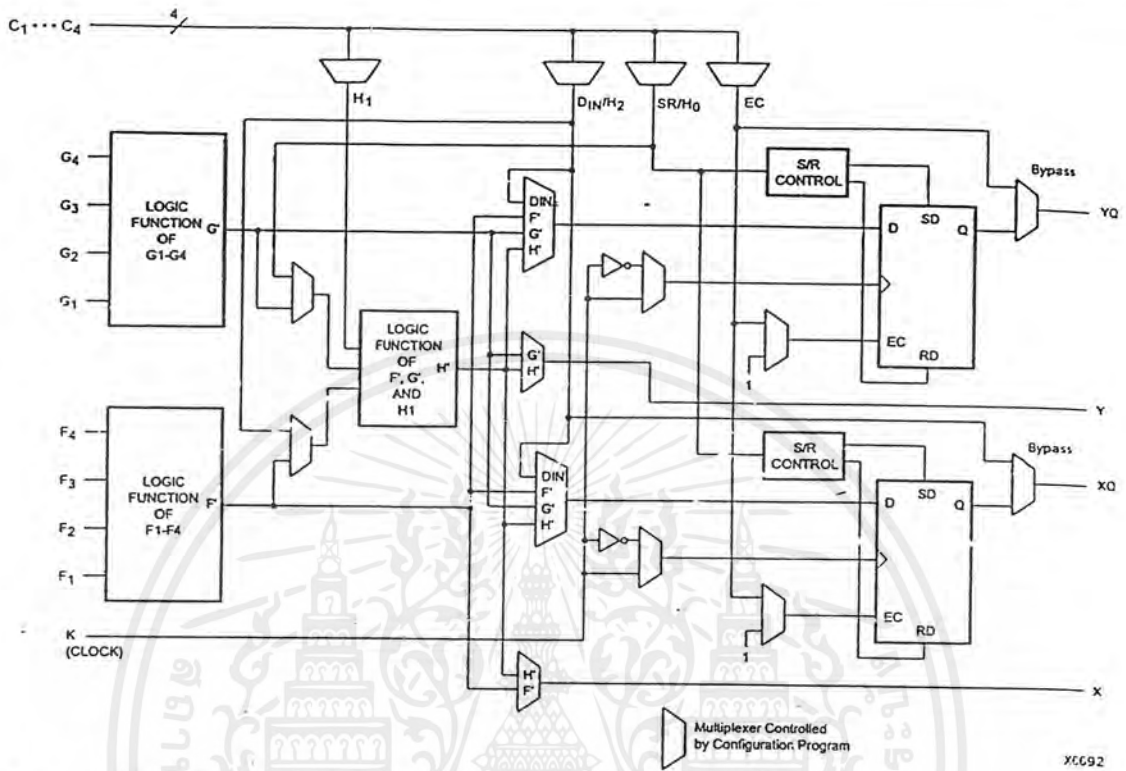


Figure 1: Simplified Block Diagram of XC4000 Series CLB (RAM and Carry Logic functions not shown)

Flip-Flops

The CLB can pass the combinational output(s) to the interconnect network, but can also store the combinational results or other incoming data in one or two flip-flops, and connect their outputs to the interconnect network as well.

The two edge-triggered D-type flip-flops have common clock (K) and clock enable (EC) inputs. Either or both clock inputs can also be permanently enabled. Storage element functionality is described in Table 2.

Latches (XC4000X only)

The CLB storage elements can also be configured as latches. The two latches have common clock (K) and clock enable (EC) inputs. Storage element functionality is described in Table 2.

Clock Input

Each flip-flop can be triggered on either the rising or falling clock edge. The clock pin is shared by both storage elements. However, the clock is individually invertible for each storage element. Any inverter placed on the clock input is automatically absorbed into the CLB.

Clock Enable

The clock enable signal (EC) is active High. The EC pin is shared by both storage elements. If left unconnected for either, the clock enable for that storage element defaults to the active state. EC is not invertible within the CLB.

Table 2: CLB Storage Element Functionality (active rising edge is shown)

Mode	K	EC	SR	D	Q
Power-Up or GSR	X	X	X	X	SR
Flip-Flop	X	X	1	X	SR
	$\overline{\text{—}}$	1*	0*	D	D
Latch	0	X	0*	X	Q
	1	1*	0*	X	Q
Both	0	1*	0*	D	D
	X	0	0*	X	Q

Legend:
 X Don't care
 $\overline{\text{—}}$ Rising edge
 SR Set or Reset value. Reset is default.
 0* Input is Low or unconnected (default value)
 1* Input is High or unconnected (default value)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 16: Pin Descriptions

Pin Name	I/O During Config.	I/O After Config.	Pin Description
Permanently Dedicated Pins			
VCC	I	I	Eight or more (depending on package) connections to the nominal +5 V supply voltage (+3.3 V for low-voltage devices). All must be connected, and each must be decoupled with a 0.01 - 0.1 μ F capacitor to Ground.
GND	I	I	Eight or more (depending on package type) connections to Ground. All must be connected.
CCLK	I or O	I	During configuration, Configuration Clock (CCLK) is an output in Master modes or Asynchronous Peripheral mode, but is an input in Slave mode and Synchronous Peripheral mode. After configuration, CCLK has a weak pull-up resistor and can be selected as the Readback Clock. There is no CCLK High or Low time restriction on XC4000 Series devices, except during Readback. See "Violating the Maximum High and Low Time Specification for the Readback Clock" on page 56 for an explanation of this exception.
DONE	I/O	O	DONE is a bidirectional signal with an optional internal pull-up resistor. As an output, it indicates the completion of the configuration process. As an input, a Low level on DONE can be configured to delay the global logic initialization and the enabling of outputs. The optional pull-up resistor is selected as an option in the XACStep program that creates the configuration bitstream. The resistor is included by default.
PROGRAM	I	I	PROGRAM is an active Low input that forces the FPGA to clear its configuration memory. It is used to initiate a configuration cycle. When PROGRAM goes High, the FPGA finishes the current clear cycle and executes another complete clear cycle, before it goes into a WAIT state and releases NIT. The PROGRAM pin has a permanent weak pull-up, so it need not be externally pulled up to Vcc.
User I/O Pins That Can Have Special Functions			
RDY/BUSY	O	I/O	During Peripheral mode configuration, this pin indicates when it is appropriate to write another byte of data into the FPGA. The same status is also available on D7 in Asynchronous Peripheral mode, if a read operation is performed when the device is selected. After configuration, RDY/BUSY is a user-programmable I/O pin. RDY/BUSY is pulled High with a high-impedance pull-up prior to NIT going High.
RCLK	O	I/O	During Master Parallel configuration, each change on the A0-A17 outputs (A0 - A21 for XC4000X) is preceded by a rising edge on RCLK, a redundant output signal. RCLK is useful for clocked PROMs. It is rarely used during configuration. After configuration, RCLK is a user-programmable I/O pin.
M0, M1, M2	I	I (M0), O (M1), I (M2)	As Mode inputs, these pins are sampled after NIT goes High to determine the configuration mode to be used. After configuration, M0 and M2 can be used as inputs, and M1 can be used as a 3-state output. These three pins have no associated input or output registers. During configuration, these pins have weak pull-up resistors. For the most popular configuration mode, Slave Serial, the mode pins can thus be left unconnected. The three mode inputs can be individually configured with or without weak pull-up or pull-down resistors. A pull-down resistor value of 4.7 k Ω is recommended. These pins can only be used as inputs or outputs when called out by special schematic definitions. To use these pins, place the library components MD0, MD1, and MD2 instead of the usual pad symbols. Input or output buffers must still be used.
TDO	O	O	If boundary scan is used, this pin is the Test Data Output. If boundary scan is not used, this pin is a 3-state output without a register, after configuration is completed. This pin can be user output only when called out by special schematic definitions. To use this pin, place the library component TDO instead of the usual pad symbol. An output buffer must still be used.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 16: Pin Descriptions (Continued)

Pin Name	I/O During Config.	I/O After Config.	Pin Description
TDI, TCK, TMS	I	I/O or I (JTAG)	If boundary scan is used, these pins are Test Data In, Test Clock, and Test Mode Select inputs respectively. They come directly from the pads, bypassing the IOBs. These pins can also be used as inputs to the CLB logic after configuration is completed. If the BSCAN symbol is not placed in the design, all boundary scan functions are inhibited once configuration is completed, and these pins become user-programmable I/O. In this case, they must be called out by special schematic definitions. To use these pins, place the library components TDI, TCK, and TMS instead of the usual pad symbols. Input or output buffers must still be used.
HDC	O	I/O	High During Configuration (HDC) is driven High until the I/O go active. It is available as a control output indicating that configuration is not yet completed. After configuration, HDC is a user-programmable I/O pin.
$\overline{\text{LDC}}$	O	I/O	Low During Configuration (LDC) is driven Low until the I/O go active. It is available as a control output indicating that configuration is not yet completed. After configuration, $\overline{\text{LDC}}$ is a user-programmable I/O pin.
$\overline{\text{INIT}}$	I/O	I/O	Before and during configuration, $\overline{\text{INIT}}$ is a bidirectional signal. A 1 k Ω - 10 k Ω external pull-up resistor is recommended. As an active-Low open-drain output, $\overline{\text{INIT}}$ is held Low during the power stabilization and internal clearing of the configuration memory. As an active-Low input, it can be used to hold the FPGA in the internal WAIT state before the start of configuration. Master mode devices stay in a WAIT state an additional 30 to 30 μ s after $\overline{\text{INIT}}$ has gone High. During configuration, a Low on this output indicates that a configuration data error has occurred. After the I/O go active, $\overline{\text{INIT}}$ is a user-programmable I/O pin.
PGCK1 - PGCK4 (XC4000E only)	Weak Pull-up	I or I/O	Four Primary Global inputs each drive a dedicated internal global net with short delay and minimal skew. If not used to drive a global buffer, any of these pins is a user-programmable I/O. The PGCK1-PGCK4 pins drive the four Primary Global Buffers. Any input pad symbol connected directly to the input of a BUFGRP symbol is automatically placed on one of these pins.
SGCK1 - SGCK4 (XC4000E only)	Weak Pull-up	I or I/O	Four Secondary Global inputs each drive a dedicated internal global net with short delay and minimal skew. These internal global nets can also be driven from internal logic. If not used to drive a global net, any of these pins is a user-programmable I/O pin. The SGCK1-SGCK4 pins provide the shortest path to the four Secondary Global Buffers. Any input pad symbol connected directly to the input of a BUFGRS symbol is automatically placed on one of these pins.
GCK1 - GCK8 (XC4000X only)	Weak Pull-up	I or I/O	Eight inputs can each drive a Global Low-Skew buffer. In addition, each can drive a Global Early buffer. Each pair of global buffers can also be driven from internal logic, but must share an input signal. If not used to drive a global buffer, any of these pins is a user-programmable I/O. Any input pad symbol connected directly to the input of a BUFGLS or BUFGE symbol is automatically placed on one of these pins.
FCLK1 - FCLK4 (XC4000XLA and XC4000XV only)	Weak Pull-up	I or I/O	Four inputs can each drive a Fast Clock (FCLK) buffer which can deliver a clock signal to any IOB clock input in the octant of the die served by the Fast Clock buffer. Two Fast Clock buffers serve the two IOB octants on the left side of the die and the other two Fast Clock buffers serve the two IOB octants on the right side of the die. On each side of the die, one Fast Clock buffer serves the upper octant and the other serves the lower octant. If not used to drive a Fast Clock buffer, any of these pins is a user-programmable I/O.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 16: Pin Descriptions (Continued)

Pin Name	I/O During Config.	I/O After Config.	Pin Description
$\overline{CS0}$, CS1, \overline{WS} , \overline{RS}	I	I/O	These four inputs are used in Asynchronous Peripheral mode. The chip is selected when $\overline{CS0}$ is Low and CS1 is High. While the chip is selected, a Low on Write Strobe (\overline{WS}) loads the data present on the D0 - D7 inputs into the internal data buffer. A Low on Read Strobe (\overline{RS}) changes D7 into a status output — High if Ready, Low if Busy — and drives D0 - D6 High. In Express mode, CS1 is used as a serial-enable signal for daisy-chaining. \overline{WS} and \overline{RS} should be mutually exclusive, but if both are Low simultaneously, the Write Strobe overrides. After configuration, these are user-programmable I/O pins.
A0 - A17	O	I/O	During Master Parallel configuration, these 18 output pins address the configuration EPROM. After configuration, they are user-programmable I/O pins.
A18 - A21 (XC4003XL to XC4085XL)	O	I/O	During Master Parallel configuration with an XC4000X master, these 4 output pins add 4 more bits to address the configuration EPROM. After configuration, they are user-programmable I/O pins. (See Master Parallel Configuration section for additional details.)
D0 - D7	I	I/O	During Master Parallel and Peripheral configuration, these eight input pins receive configuration data. After configuration, they are user-programmable I/O pins.
DIN	I	I/O	During Slave Serial or Master Serial configuration, DIN is the serial configuration data input receiving data on the rising edge of CCLK. During Parallel configuration, DIN is the D0 input. After configuration, DIN is a user-programmable I/O pin.
DOUT	O	I/O	During configuration in any mode but Express mode, DOUT is the serial configuration data output that can drive the DIN of daisy-chained slave FPGAs. DOUT data changes on the falling edge of CCLK, one-and-a-half CCLK periods after it was received at the DIN input. In Express mode for XC4000E and XC4000X only, DOUT is the status output that can drive the CS1 of daisy-chained FPGAs, to enable and disable downstream devices. After configuration, DOUT is a user-programmable I/O pin.
Unrestricted User-Programmable I/O Pins			
I/O	Weak Pull-up	I/O	These pins can be configured to be input and/or output after configuration is completed. Before configuration is completed, these pins have an internal high-value pull-up resistor (25 k Ω - 100 k Ω) that defines the logic level as High.

Boundary Scan

The 'bed of nails' has been the traditional method of testing electronic assemblies. This approach has become less appropriate, due to closer pin spacing and more sophisticated assembly methods like surface-mount technology and multi-layer boards. The IEEE Boundary Scan Standard 1149.1 was developed to facilitate board-level testing of electronic assemblies. Design and test engineers can imbed a standard test logic structure in their device to achieve high fault coverage for I/O and internal logic. This structure is easily implemented with a four-pin interface on any boundary scan-compatible IC. IEEE 1149.1-compatible devices may be serial daisy-chained together, connected in parallel, or a combination of the two.

The XC4000 Series implements IEEE 1149.1-compatible BYPASS, PRELOAD/SAMPLE and EXTEST boundary scan instructions. When the boundary scan configuration option is selected, three normal user I/O pins become dedicated inputs for these functions. Another user output pin becomes the dedicated boundary scan output. The details

of how to enable this circuitry are covered later in this section.

By exercising these input signals, the user can serially load commands and data into these devices to control the driving of their outputs and to examine their inputs. This method is an improvement over bed-of-nails testing. It avoids the need to over-drive device outputs, and it reduces the user interface to four pins. An optional fifth pin, a reset for the control logic, is described in the standard but is not implemented in Xilinx devices.

The dedicated on-chip logic implementing the IEEE 1149.1 functions includes a 16-state machine, an instruction register and a number of data registers. The functional details can be found in the IEEE 1149.1 specification and are also discussed in the Xilinx application note XAPP 017: *Boundary Scan in XC4000 Devices*.

Figure 40 on page 43 shows a simplified block diagram of the XC4000E Input/Output Block with boundary scan implemented. XC4000X boundary scan logic is identical.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XC4005E/XL Device Pinout Tables

The following table may contain pinout information for unsupported device/package combinations. Please see the availability charts elsewhere in the XC4000 Series data sheet for availability information.

XC4005E/XL Pad Name	PC 84	PQ 100	VQ 100††	TQ 144	PG 156†	PQ 160	PQ 208	Bndry Scan
VCC	P2	P92	P89	P128	H3	P142	P183	-
I/O (A8)	P3	P93	P90	P129	H1	P143	P184	44
I/O (A9)	P4	P94	P91	P130	G1	P144	P185	47
I/O (A19) ††	-	P95	P92	P131	G2	P145	P186	50
I/O (A18) ††	-	P96	P93	P132	G3	P146	P187	53
I/O (A10)	P5	P97	P94	P133	F1	P147	P190	56
I/O (A11)	P6	P98	P95	P134	F2	P148	P191	59
I/O	-	-	-	P135	E1	P149	P192	62
I/O	-	-	-	P136	E2	P150	P193	65
GND	-	-	-	P137	F3	P151	P194	-
I/O (A12)	P7	P99	P96	P138	E3	P154	P199	68
I/O (A13)	P8	P100	P97	P139	C1	P155	P200	71
I/O	-	-	-	P140	C2	P156	P201	74
I/O	-	-	-	P141	D3	P157	P202	77
I/O (A14)	P9	P1	P98	P142	B1	P158	P203	80
I/O, SGCK1 †, GCK8 †† (A15)	P10	P2	P99	P143	B2	P159	P204	83
VCC	P11	P3	P100	P144	C3	P160	P205	-
GND	P12	P4	P1	P1	C4	P1	P2	-
I/O, PGCK1†, GCK1†† (A16)	P13	P5	P2	P2	B3	P2	P4	86
I/O (A17)	P14	P6	P3	P3	A1	P3	P5	89
I/O	-	-	-	P4	A2	P4	P6	92
I/O	-	-	-	P5	C5	P5	P7	95
I/O, TDI	P15	P7	P4	P6	B4	P6	P8	98
I/O, TCK	P16	P8	P5	P7	A3	P7	P9	101
GND	-	-	-	P8	C6	P10	P14	-
I/O	-	-	-	P9	B5	P11	P15	104
I/O	-	-	-	P10	B6	P12	P16	107
I/O, TMS	P17	P9	P6	P11	A5	P13	P17	110
I/O	P18	P10	P7	P12	C7	P14	P18	113
I/O	-	-	-	P13	B7	P15	P21	116
I/O	-	P11	P8	P14	A6	P16	P22	119
I/O	P19	P12	P9	P15	A7	P17	P23	122
I/O	P20	P13	P10	P16	A8	P18	P24	125
GND	P21	P14	P11	P17	C8	P19	P25	-
VCC	P22	P15	P12	P18	B8	P20	P26	-
I/O	P23	P16	P13	P19	C9	P21	P27	128
I/O	P24	P17	P14	P20	B9	P22	P28	131
I/O	-	P18	P15	P21	A9	P23	P29	134
I/O	-	-	-	P22	B10	P24	P30	137
I/O	P25	P19	P16	P23	C10	P25	P33	140
I/O	P26	P20	P17	P24	A10	P26	P34	143
I/O	-	-	-	P25	A11	P27	P35	146
I/O	-	-	-	P26	B11	P28	P36	149
GND	-	-	-	P27	C11	P29	P37	-
I/O	P27	P21	P18	P28	B12	P32	P42	152
I/O	-	P22	P19	P29	A13	P33	P43	155
I/O	-	-	-	P30	A14	P34	P44	158
I/O	-	-	-	P31	C12	P35	P45	161
I/O	P28	P23	P20	P32	B13	P36	P46	164
I/O, SGCK2 †, GCK2 ††	P29	P24	P21	P33	B14	P37	P47	167
O (M1)	P30	P25	P22	P34	A15	P38	P48	170
GND	P31	P26	P23	P35	C13	P39	P49	-
I (M0)	P32	P27	P24	P36	A16	P40	P50	173
VCC	P33	P28	P25	P37	C14	P41	P55	-
I (M2)	P34	P29	P26	P38	B15	P42	P56	174
I/O, PGCK2 †, GCK3 ††	P35	P30	P27	P39	B16	P43	P57	175
I/O (HDC)	P36	P31	P28	P40	D14	P44	P58	178
I/O	-	-	-	P41	C15	P45	P59	181
I/O	-	-	-	P42	D15	P46	P60	184
I/O	-	P32	P29	P43	E14	P47	P61	187
I/O (LDC)	P37	P33	P30	P44	C16	P48	P62	190

XC4005E/XL Pad Name	PC 84	PQ 100	VQ 100††	TQ 144	PG 156†	PQ 160	PQ 208	Bndry Scan
GND	-	-	-	P45	F14	P51	P67	-
I/O	-	-	-	P46	F15	P52	P68	193
I/O	-	-	-	P47	E16	P53	P69	196
I/O	P38	P34	P31	P48	F16	P54	P70	199
I/O	P39	P35	P32	P49	G14	P55	P71	202
I/O	-	P36	P33	P50	G15	P56	P74	205
I/O	-	P37	P34	P51	G16	P57	P75	208
I/O	P40	P38	P35	P52	H16	P58	P76	211
I/O (INIT)	P41	P39	P36	P53	H15	P59	P77	214
VCC	P42	P40	P37	P54	H14	P60	P78	-
GND	P43	P41	P38	P55	J14	P61	P79	-
I/O	P44	P42	P39	P56	J15	P62	P80	217
I/O	P45	P43	P40	P57	J16	P63	P81	220
I/O	-	P44	P41	P58	K16	P64	P82	223
I/O	-	P45	P42	P59	K15	P65	P83	226
I/O	P46	P46	P43	P60	K14	P66	P86	229
I/O	P47	P47	P44	P61	L16	P67	P87	232
I/O	-	-	-	P62	M16	P68	P88	235
I/O	-	-	-	P63	L15	P69	P89	238
GND	-	-	-	P64	L14	P70	P90	-
I/O	P48	P48	P45	P65	P16	P73	P95	241
I/O	P49	P49	P46	P66	M14	P74	P96	244
I/O	-	-	-	P67	N15	P75	P97	247
I/O	-	-	-	P68	P15	P76	P98	250
I/O	P50	P50	P47	P69	N14	P77	P99	253
I/O, SGCK3 †, GCK4 ††	P51	P51	P48	P70	R16	P78	P100	256
GND	P52	P52	P49	P71	P14	P79	P101	-
DONE	P53	P53	P50	P72	R15	P80	P103	-
VCC	P54	P54	P51	P73	P13	P81	P106	-
PROGRAM	P55	P55	P52	P74	R14	P82	P108	-
I/O (D7)	P56	P56	P53	P75	T16	P83	P109	259
I/O, PGCK3†, GCK5††	P57	P57	P54	P76	T15	P84	P110	262
I/O	-	-	-	P77	R13	P85	P111	265
I/O	-	-	-	P78	P12	P86	P112	268
I/O (D6)	P58	P58	P55	P79	T14	P87	P113	271
I/O	-	P59	P56	P80	T13	P88	P114	274
GND	-	-	-	P81	P11	P91	P119	-
I/O	-	-	-	P82	R11	P92	P120	277
I/O	-	-	-	P83	T11	P93	P121	280
I/O (D5)	P59	P60	P57	P84	T10	P94	P122	283
I/O (CS0)	P60	P61	P58	P85	P10	P95	P123	286
I/O	-	P62	P59	P86	R10	P96	P126	289
I/O	-	P63	P60	P87	T9	P97	P127	292
I/O (D4)	P61	P64	P61	P88	R9	P98	P128	295
I/O	P62	P65	P62	P89	P9	P99	P129	298
VCC	P63	P66	P63	P90	R6	P100	P130	-
GND	P64	P67	P64	P91	P8	P101	P131	-
I/O (D3)	P65	P68	P65	P92	T8	P102	P132	301
I/O (RS)	P66	P69	P66	P93	T7	P103	P133	304
I/O	-	P70	P67	P94	T6	P104	P134	307
I/O	-	-	-	P95	R7	P105	P135	310
I/O (D2)	P67	P71	P68	P96	P7	P106	P138	313
I/O	P68	P72	P69	P97	T5	P107	P139	316
I/O	-	-	-	P98	R5	P108	P140	319
I/O	-	-	-	P99	T4	P109	P141	322
GND	-	-	-	P100	P6	P110	P142	-
I/O (D1)	P69	P73	P70	P101	T3	P113	P147	325
I/O (RCLK, RDY/BUSY)	P70	P74	P71	P102	P5	P114	P148	328
I/O	-	-	-	P103	R4	P115	P149	331
I/O	-	-	-	P104	R3	P116	P150	334
I/O (D0, DIN)	P71	P75	P72	P105	P4	P117	P151	337

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XC4005E/XL Pad Name	PC 84	PQ 100	VQ 100††	TQ 144	PG 156†	PQ 160	PQ 208	Bndry Scan
I/O, SGCK4 †, GCK6 †† (DOOUT)	P72	P76	P73	P106	T2	P118	P152	340
CCLK	P73	P77	P74	P107	R2	P119	P153	-
VCC	P74	P78	P75	P108	P3	P120	P154	-
O, TDO	P75	P79	P76	P109	T1	P121	P159	0
GND	P76	P80	P77	P110	N3	P122	P160	-
I/O (A0, WS)	P77	P81	P78	P111	R1	P123	P161	2
I/O, PGCK4 †, GCK7 †† (A1)	P78	P82	P79	P112	P2	P124	P162	5
I/O	-	-	-	P113	N2	P125	P163	8
I/O	-	-	-	P114	M3	P126	P164	11
I/O (CS1, A2)	P79	P83	P80	P115	P1	P127	P165	14
I/O (A3)	P80	P84	P81	P116	N1	P128	P166	17
GND	-	-	-	P118	L3	P131	P171	-
I/O	-	-	-	P119	L2	P132	P172	20
I/O	-	-	-	P120	L1	P133	P173	23
I/O (A4)	P81	P85	P82	P121	K3	P134	P174	26
I/O (A5)	P82	P86	P83	P122	K2	P135	P175	29
I/O (A21) ††	-	P87	P84	P123	K1	P137	P178	32
I/O (A20) ††	-	P88	P85	P124	J1	P138	P179	35
I/O (A6)	P83	P89	P86	P125	J2	P139	P180	38
I/O (A7)	P84	P90	P87	P126	J3	P140	P181	41
GND	P1	P91	P88	P127	H2	P141	P182	-

6/10/97

† = E only

†† = XL only

Additional XC4005E/XL Package Pins

TQ144

Not Connected Pins							
P117	-	-	-	-	-	-	-

5/5/97

PG156

Not Connected Pins					
A4	A12	D1	D2	D16	E15
M1	M2	M15	N16	R5	R12
T12	-	-	-	-	-

5/5/97

PQ160

Not Connected Pins					
P8	P9	P30	P31	P49	P50
P71	P72	P89	P90	P111	P112
P129	P130	P136	P152	P153	-

6/16/97

PQ208

Not Connected Pins					
P1	P3	P10	P11	P12	P13
P19	P20	P31	P32	P36	P39
P40	P41	P51	P52	P53	P54
P63	P64	P65	P66	P72	P73
P84	P85	P91	P92	P93	P94
P102	P104	P105	P107	P115	P116
P117	P118	P124	P125	P136	P137
P143	P144	P145	P146	P155	P156
P157	P158	P167	P168	P169	P170
P176	P177	P188	P189	P195	P196
P197	P198	P206	P207	P208	-

6/5/97

XC4006E Device Pinout Tables

XC4006E Pad Name	PC 84	TQ 144	PG 156	PQ 160	PQ 208	Bndry Scan
VCC	P2	P128	H3	P142	P183	-
I/O (A8)	P3	P129	H1	P143	P184	50
I/O (A9)	P4	P130	G1	P144	P185	53
I/O	-	P131	G2	P145	P186	56
I/O	-	P132	G3	P146	P187	59
I/O (A10)	P5	P133	F1	P147	P190	62
I/O (A11)	P6	P134	F2	P148	P191	65
I/O	-	P135	E1	P149	P192	68
I/O	-	P136	E2	P150	P193	71
GND	-	P137	F3	P151	P194	-
I/O	-	-	D1	P152	P197	74
I/O	-	-	D2	P153	P198	77
I/O (A12)	P7	P138	E3	P154	P199	80
I/O (A13)	P8	P139	C1	P155	P200	83
I/O	-	P140	C2	P156	P201	86
I/O	-	P141	D3	P157	P202	89
I/O (A14)	P9	P142	B1	P158	P203	92
I/O, SGCK1 (A15)	P10	P143	B2	P159	P204	95
VCC	P11	P144	C3	P160	P205	-
GND	P12	P1	C4	P1	P2	-
I/O, PGCK1 (A16)	P13	P2	B3	P2	P4	98
I/O (A17)	P14	P3	A1	P3	P5	101
I/O	-	P4	A2	P4	P6	104
I/O	-	P5	C5	P5	P7	107
I/O, TDI	P15	P6	B4	P6	P8	110
I/O, TCK	P16	P7	A3	P7	P9	113
I/O	-	-	A4	P8	P10	116
I/O	-	-	-	P9	P11	119
GND	-	P8	C6	P10	P14	-

XC4006E Pad Name	PC 84	TQ 144	PG 156	PQ 160	PQ 208	Bndry Scan	
I/O	-	P9	B5	P11	P15	122	
I/O	-	P10	B5	P12	P16	125	
I/O, TMS	P17	P11	A5	P13	P17	128	
I/O	P18	P12	C7	P14	P18	131	
I/O	-	P13	B7	P15	P21	134	
I/O	-	-	P14	A6	P16	P22	137
I/O	P19	P15	A7	P17	P23	140	
I/O	P20	P16	A8	P18	P24	143	
GND	P21	P17	C8	P19	P25	-	
VCC	P22	P18	B8	P20	P26	-	
I/O	P23	P19	C9	P21	P27	146	
I/O	P24	P20	B9	P22	P28	149	
I/O	-	P21	A9	P23	P29	152	
I/O	-	P22	B10	P24	P30	155	
I/O	P25	P23	C10	P25	P33	158	
I/O	P26	P24	A10	P26	P34	161	
I/O	-	P25	A11	P27	P35	164	
I/O	-	P26	B11	P28	P36	167	
GND	-	P27	C11	P29	P37	-	
I/O	-	-	A12	P30	P40	170	
I/O	-	-	-	P31	P41	173	
I/O	P27	P28	B12	P32	P42	176	
I/O	-	P29	A13	P33	P43	179	
I/O	-	P30	A14	P34	P44	182	
I/O	-	P31	C12	P35	P45	185	
I/O	P28	P32	B13	P36	P46	188	
I/O, SGCK2	P29	P33	B14	P37	P47	191	
O (M1)	P30	P34	A15	P38	P48	194	
GND	P31	P35	C13	P39	P49	-	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ไกรวุฒิ โรจน์ประเสริฐสุด. เข้าใจ/สร้าง/เล่น ไมโครโปรเซสเซอร์2. พิมพ์ครั้งที่ 1. กรุงเทพฯ:
ซีเอ็ดยูเคชั่น. 2539

บริษัท ซีเอ็ดยูเคชั่นจำกัด. คู่มือ/เทียบเบอร์ ไอซี TTL. กรุงเทพฯ:เม็คทรายพรีนติ้ง. 2538

มนัส สัจวรศิลป์. คู่มือการใช้งาน MATLAB. พิมพ์ครั้งที่ 1. กรุงเทพฯ:สำนักพิมพ์
อินโฟเพรส. 2543

ชินภัทร นันทจิวารักษ์. “การลดขนาดข้อมูลเสียงภาพโดยใช้ เวฟเล็ตทรานฟอร์ม.”

วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย,
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.2542

ณรงค์ คงสมเพ็ชร ธนพล แก้วคำแจ้ง และสุรเกียรติ์ โสภณพุทธพร. “การสร้างวงจรกรองเชิงเลข

แบบบัตเตอร์เวิร์ธอันดับที่ 6” ปรียญานิพนธ์ครุศาสตร์อุตสาหกรรมบัณฑิต

สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์อุตสาหกรรม

คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร

ลาดกระบัง.2542

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญยานิพนธ์	นายประจักษ์ วันดี
วันเดือนปีเกิด	19 ธันวาคม พ.ศ. 2521
สถานที่เกิด	จังหวัดแพร่
ภูมิลำเนาเดิม	137/2 หมู่ 6 ถนน ซ่อแฮ ต.ซ่อแฮ อ.เมือง จ.แพร่ 54000
ที่อยู่ปัจจุบัน	275 หมู่ 10 ซอยคลองชัย ถนนคลองกรุง แขวงลำปลาทิว เขตลาดกระบัง กรุงเทพมหานคร 10520
โทรศัพท์	(02) 3274144
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนบ้านโน “อินทราชฎ์บำรุง”
มัธยมศึกษาตอนต้น	โรงเรียนเมืองแพร่
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคแพร่
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	วิทยาลัยเทคนิคแพร่
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	ทุนงบประมาณ
คติพจน์	เมื่อคิดได้ตรงอย่างถึถ้วนแล้ว จงคิดว่าสิ่งที่ทำเป็นสิ่งที่ถูกต้องและดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญาณพันธ	นายอัสวิน มีเงิน
วันเดือนปีเกิด	25 กุมภาพันธ์ พ.ศ. 2522
สถานที่เกิด	จังหวัดเชียงใหม่
ภูมิลำเนาเดิม	270 หมู่ 4 ถนน รอบเวียง ต.เวียง อ.ฝาง จ.เชียงใหม่ 50110
ที่อยู่ปัจจุบัน	ห้อง 202 หอพักกมล แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพมหานคร 10520
โทรศัพท์	(02) 3267716 # 202
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนรังษีวิทยา
มัธยมศึกษาตอนต้น	โรงเรียนรังษีวิทยา
ประกาศนียบัตรวิชาชีพ (ปวช.)	วิทยาลัยเทคนิคเชียงใหม่
ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคพายัพ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	-
ทุนการศึกษา	-
คติพจน์	หนทางยังอีก ไกล แต่จะสั้นเข้าไปเมื่อก้าวเดิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง



ชื่อผู้ทำปฏิญานិพนธ์	นายอาคม ช่างเทียน
วันเดือนปีเกิด	13 ธันวาคม พ.ศ. 2520
สถานที่เกิด	จังหวัดสงขลา
ภูมิลำเนาเดิม	199/141 หมู่บ้านสวนนนทรี ถนนเชียงใหม่-แม่โจ้ ต.หนองจ่อม อ.สันทราย จ.เชียงใหม่ 50210
ที่อยู่ปัจจุบัน	275 หมู่ 10 ซอยฉลองชัย ถนนฉลองกรุง แขวงลำปลาทิว เขตลาดกระบัง กรุงเทพมหานคร 10520
โทรศัพท์	(02) 3274144 ,(053)854113
ประวัติการศึกษา	
ประถมศึกษา	โรงเรียนศรีธนาวิทยา
มัธยมศึกษาตอนต้น	โรงเรียนเทพดินทร์ (ศรีธนาเดิม)
มัธยมศึกษาตอนปลาย (ม.4)	โรงเรียนนวมินทร์ทราฐศึกษาพิทย เชียงใหม่
ประกาศนียบัตรวิชาชีพชั้นสูง (4ปี)	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคพายัพ
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับ	นักศึกษาดีเด่นในระดับปวส.
ทุนการศึกษา	-
คติพจน์	สิ่งที่เข้าคิดถูกเสมอและสิ่งที่เข้าทำย่อมถูกเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้