

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องช่วยฟัง

HEARING AID DEVICE



T119416

วรุทม์ วัฒนอนันต์
วสันต์ นาคปนคำ
วลิชฐ ศรีขวัณเจริญ

เลขหมู่.....
เลขทะเบียน **119416**
วัน,เดือน,ปี.....-7.S.A. 2554

b. 1236๗๗๑
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องช่วยฟัง

HEARING AID DEVICE

โดย

นาย วรุตม์	วัฒนอนันต์	รหัส 50011406
นาย วสันต์	นาคปนคำ	รหัส 50011420
นาย วลีษฐ	ศรีขวัญเจริญ	รหัส 50011422

อาจารย์ที่ปรึกษา

ผศ.ดร.ชูชาติ ปิณฑวิรุจน์

b. 12๖๖๖๖
i.

ปริญญานิพนธ์นี้สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายงาน ปีการศึกษา 2553

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อเรื่องภาษาไทย เครื่องช่วยฟัง

ชื่อเรื่องภาษาอังกฤษ Hearing Aid Device

ผู้จัดทำ

นายวรุฒม์ วัฒนอนันต์

รหัส 50011406

นายวสันต์ นาคปนคำ

รหัส 50011420

นายวศิษฐ ศรีขวัณเจริญ

รหัส 50011422

ลงชื่อ.....อาจารย์ที่ปรึกษา

(รศ. ดร. ชูชาติ ปิณฑวิรุจน์)

วันที่ 21 / 3 / 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องช่วยฟัง

นาย วรุตม์ วัฒนอนันต์ รหัส 50011406
นาย วสันต์ นาคปนคำ รหัส 50011420
นาย วลีษฐ ศรีขวัญเจริญ รหัส 0011422
รศ.ดร. ชูชาติ ปิณฑวิรุจน์
ปีการศึกษา 2553

บทคัดย่อ

รายงานฉบับนี้ได้อธิบายถึงการออกแบบและสร้างเครื่องช่วยฟังสำหรับผู้พิการทางการได้ยินเฉพาะย่านความถี่ ซึ่งสามารถที่จะช่วยให้ผู้ฟังสามารถรับฟังเสียงในความถี่ที่ผู้พิการนั้นรับฟังไม่ได้ โดยตัวเครื่องช่วยฟัง จะได้ประกอบไปด้วย ไมโครโฟน ภาครองความถี่สัญญาณ ภาคขยาย และ ลำโพง โดยภาครองความถี่จะใช้ตัวกรองความถี่แบบดิจิตอลแบบ FIR โดยเครื่องช่วยฟังนี้จะสามารถที่จะปรับความถี่ที่กรองได้ในช่วง 20 Hz – 2 kHz ผ่านทางการเขียนโปรแกรมลงใน FPGA โดยในการออกแบบ ได้ใช้โปรแกรม MATLAB เพื่อคำนวณหาตัวแปรและสัมประสิทธิ์ FIR จากนั้นจึงนำข้อมูลที่ได้มาประกอบกรเขียนโปรแกรมเพื่อสังเคราะห์ลงบน FPGA

Hearing Aid Device

Mr. Warut Wattana-anan ID.50011406

Mr. Wasan Nakponcom ID.50011420

Mr. Wasit Srikuancharoen ID.50011422

Ass. Prof. Dr. Chuchart Pintavirooj

Education Year 2010

Abstract

This report describes a design and construction of a Hearing Aid Device for patient with disabilities in hearing in some frequency range. The Hearing Aid Device is composed of 4 main parts. They are 1. Microphone 2. Signal Filter 3. Amplifier 4. Speaker. Signal filter is a Digital Filter, FIR. This FIR Bandpass Filter range can be adjusted from 20 Hz – 2 kHz by synthesizing the selected coefficients set that generated from FDATool function in MATLAB.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

คณะผู้ทำโครงการใคร่ขอขอบพระคุณต่อ อาจารย์ ชูชาติ พิณทวีรุจน์ และอาจารย์ มนัส สังวรศิลป์ อาจารย์ที่ปรึกษาโครงการ ผู้ให้คำปรึกษาและแนวทางแก้ปัญหา และข้อเสนอแนะต่างๆ ในการทำโครงการนี้ให้สำเร็จลุล่วงไปด้วยดี

พร้อมกันนี้ใคร่ขอขอบคุณต่อเจ้าหน้าที่ของภาควิชาวิศวกรรมอิเล็กทรอนิกส์ทุกท่าน ที่ให้ความสะดวกในการเบิก และยืมอุปกรณ์ต่างๆ การทำโครงการ

รุ่นพี่ และเพื่อนๆ ทุกคนที่ช่วยแนะนำให้ความช่วยเหลือในการจัดทำโครงการน

นาย วรุตม์ วัฒนอนันต์

นาย วสันต์ นาคปนคำ

นาย วศิษฐ ศรีขวัญเจริญ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อ	ก
Abstract	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
สารบัญตาราง	ณ
บทที่ 1 บทนำ	1
1.1 หลักการและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตโครงการ	1
บทที่ 2 ทฤษฎี	2
2.1 ทฤษฎีหูและการตรวจสอบรรถภาพการได้ยิน	2
2.2 ทฤษฎี Finite Impulse Response (FIR)	11
2.3 ทฤษฎี Analog to Digital Converter, Digital to Analog Converter (ADC, DAC)	23
2.4 ทฤษฎี FPGA และ VHDL	37
2.5 ทฤษฎี SPI Interface	49
2.6 Phase Lock Loop	50
บทที่ 3 การออกแบบและการทำงาน	51
3.1 โครงสร้างของระบบ	51
3.2 อุปกรณ์ที่ใช้ในการออกแบบ	68
3.3 การทำงานของโปรแกรมทดสอบหูอย่างง่าย	71
บทที่ 4 การทดลอง	78
4.1 การทดลอง digital filter จาก โปรแกรม matlab	82
4.2 การทดสอบโปรแกรมทดสอบการได้ยิน	85
4.3 การทดลองการใช้โปรแกรม FPGA ที่ใช้กรองสัญญาณ	89
4.4 การทดลองที่ความถี่คutoffของฟิลเตอร์	89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 ผลการทดลอง	89
5.1 ผลการทดลอง digital filter จากโปรแกรม matlab	89
5.2 ผลการทดสอบโปรแกรมทดสอบการได้ยิน	94
5.3 ผลการทดลองโปรแกรม FPGA ที่เราใช้กรองสัญญาณ	97
5.4 ผลการทดลองความถี่คัทออฟ	101
บทที่ 6 สรุปผลการทดลอง และแนวทางการพัฒนา	103
6.1 สรุปผลการทดลอง	103
6.2 แนวทางการแก้ไขและพัฒนา	103
6.3 ประโยชน์ที่คาดว่าจะได้รับ	104
บรรณานุกรม	105



สารบัญรูป

	หน้า
รูปที่ 2.1 รูปแสดง cochlea (คอคเคลีย)	3
รูปที่ 2.2 รูปแสดงตารางการบกพร่องทางการได้ยิน	10
รูปที่ 2.3 รูปแสดงตารางระดับความบกพร่องทางการได้ยิน	11
รูปที่ 2.4 ผลตอบสนองความถี่ของวงจรกรองความถี่ในอุดมคติ	12
รูปที่ 2.5 ผลตอบสนองต่อสัญญาณอิมพัลส์ของวงจรกรอง FIR และ IIR	13
รูปที่ 2.6 รูปแสดงผลตอบสนองความถี่ของตัวกรองอุดมคติ	13
รูปที่ 2.7 คุณลักษณะเฉพาะของผลตอบสนองความถี่ของตัวกรองแบบ FIR	16
รูปที่ 2.8 ตัวอย่างของไฟล์ FIR_HPF.m	19
รูปที่ 2.9 สัมประสิทธิ์ของวงจรกรองความถี่ต่ำที่ออกแบบ	20
รูปที่ 2.10 ผลตอบสนองอิมพัลส์ของวงจรกรองความถี่สูงผ่าน ที่ N=27	21
รูปที่ 2.11 ผลตอบสนองทางความถี่และเฟสของวงจรกรองความถี่สูงผ่าน ที่ N=27	21
รูปที่ 2.12 การทดสอบการออกแบบวงจรกรองความถี่สูงผ่าน เมื่อ N=201	22
รูปที่ 2.13 Counting Converter	24
รูปที่ 2.14 รูปแสดง Output Voltage Graph	25
รูปที่ 2.15 รูปแสดง flow chart	26
รูปที่ 2.16 รูปแสดง Binary search	27
รูปที่ 2.17 รูปแสดง Dual Slope Converter	27
รูปที่ 2.18 รูปแสดง Output และ Timing ของ Dual Slope Converter	29
รูปที่ 2.19 รูปแสดง Dual Slope A/D Converter Full Circuit	29
รูปที่ 2.20 รูปแสดง Dual Slope A/D Converter – Zero Offset	30
รูปที่ 2.21 รูปแสดง Flash Converter	30
รูปที่ 2.22 ระบบการแปลงสัญญาณดิจิทัลเป็นอนาล็อก	31
รูปที่ 2.23 Transfer Curve ในอุดมคติของ DAC 3 บิต	32
รูปที่ 2.24 คลื่นไซน์ที่สร้างจาก DAC	33
รูปที่ 2.25 Summed Source DAC	34
รูปที่ 2.26 Swiched Voltage R-2R DAC	34
รูปที่ 2.27 Swiched Current R-2R DAC	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.28 Swiched Pole DAC	36
	หน้า
รูปที่ 2.29 รูปแสดงการทำงานของ SPI Interface	49
รูปที่ 2.30 แผนผัง Phase Lock Loop	50
รูปที่ 3.1 รูปแสดงโครงสร้างการออกแบบ	51
รูปที่ 3.2 รูปแสดงหน้าต่างเครื่องมือ FDATool จากโปรแกรม MATLAB ที่ใช้หาสัมประสิทธิ์ FIR	65
รูปที่ 3.3 รูปแสดงวงจรที่เราออกแบบในส่วนวงจรขยาย, ปรับ Offset, ADC, DAC, Low pass filter และส่วนตัดไฟตรง	67
รูปที่ 3.4 รูปแสดง IC เบอร์ MCP3201	68
รูปที่ 3.5 รูปแสดง IC เบอร์ MCP4821	68
รูปที่ 3.6 รูปแสดง IC เบอร์ LM386	69
รูปที่ 3.7 รูปแสดงการนำ LM386 ไปใช้งาน	69
รูปที่ 3.8 รูปแสดงบอร์ดทดลองเอนกประสงค์ FPGA รุ่น Discovery-III XC3S200F4	70
รูปที่ 3.9 รูปแสดงขั้นตอนการเลือก version ของโปรแกรมทดสอบในที่นี้เลือก version ที่ 1	71
รูปที่ 3.10 โปรแกรมที่เราจะใช้งาน	71
รูปที่ 3.11 แถบแสดงการทำงาน	72
รูปที่ 3.12 ปุ่ม Left และ Right	72
รูปที่ 3.13 ปุ่ม New TEST	73
รูปที่ 3.14 ปุ่ม Next	73
รูปที่ 3.15 ปุ่ม Up และ Down	74
รูปที่ 3.16 ปุ่ม Result	74
รูปที่ 3.17 กราฟที่ได้ออกมา	75
รูปที่ 3.18 ปุ่ม Save Result	75
รูปที่ 3.19 เมื่อเรากดปุ่ม Save Result จะได้นหน้าต่างแบบนี้ เราสามารถเลือกเซฟเป็นไฟล์ต่าง	76
รูปที่ 3.20 รูปกราฟที่ได้จากการเซฟผลการทดสอบ	76
รูปที่ 3.21 รูปแสดงขั้นตอนการเลือก version ของโปรแกรมทดสอบในที่นี้เลือก version 2	77
รูปที่ 3.22 รูปแสดงตัวโปรแกรม version ที่ 2	77
รูปที่ 3.23 รูปแสดงแถบใส่ชื่อผู้ที่ทดสอบ	78
รูปที่ 3.24 รูปแสดงรูปแถบความถี่ที่เราจะเลือก	78
รูปที่ 3.25 รูปแสดงแถบที่ใช้เลือกที่จะทดสอบหูข้างใด	79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.26 รูปแสดงแถบที่ใช้เพิ่มความดังของเสียง	79
รูปที่ 3.27 รูปแสดงปุ่มที่กดเพื่อเซฟผลการทดสอบ	80
รูปที่ 3.28 รูปแสดงปุ่ม new ที่ใช้กดเพื่อเริ่มการทดสอบใหม่	80
รูปที่ 3.29 รูปแสดงปุ่ม Home ใช้กดเมื่อต้องการที่จะใช้โปรแกรมทดสอบ version อื่นๆ	81
รูปที่ 4.1 แสดงการวาดวงจรบนโปรแกรม Simulink	82
รูปที่ 4.2 แสดงการเปลี่ยนแปลงค่าตัวแปรต่างๆบนโปรแกรม FDAtool	83
รูปที่ 4.3 แสดงการเปลี่ยนแปลงความถี่ที่อินพุต	84
รูปที่ 4.4 แสดงการเลือกหูฟังที่จะทดสอบ (ทดสอบหูข้างซ้าย)	85
รูปที่ 4.5 แสดงการเลือกความถี่ที่ต้องการทดสอบ (250Hz)	86
รูปที่ 4.6 แสดงการแสดงผลของโปรแกรม	87
รูปที่ 4.7 แสดงทดสอบโปรแกรมทดสอบการได้ยิน	87
รูปที่ 5.1 แสดงผลการทดลองทดลอง digital filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 0Hz, 100Hz	90
รูปที่ 5.2 แสดงผลการทดลองทดลอง digital filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 300Hz, 350Hz	91
รูปที่ 5.3 แสดงผลการทดลองทดลอง digital filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 400Hz, 600Hz	92
รูปที่ 5.4 แสดงผลการทดลองทดลอง Digital Filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 1000Hz, 2000Hz	93
รูปที่ 5.5 แสดงผลการทดลองโปรแกรมทดสอบหู แบบที่ 1 ของผู้ทดสอบจำนวน 5 คน	94
รูปที่ 5.6 รูปแสดงผลการทดสอบหูอย่างง่าย version 1	95
รูปที่ 5.7 รูปแสดงผลการทดสอบที่เราได้ทำการพัฒนาการบันทึกผลให้เป็นกราฟเส้น	95
รูปที่ 5.8 รูปแสดงผลการทดสอบหูอย่างง่าย version 2	96
รูปที่ 5.9 แสดงผลการทดลองเมื่อเราป้อนความถี่ที่ต่ำกว่าช่วงที่เราได้กำหนดไว้	97
รูปที่ 5.10 แสดงผลการทดลองเมื่อป้อนความถี่ในช่วงที่เราได้กำหนดไว้	97
รูปที่ 5.11 แสดงผลการทดลองเมื่อป้อนความถี่ที่เกินจากช่วงที่กำหนดไว้	98
รูปที่ 5.12 รูปแสดงกราฟการเปรียบเทียบระหว่างผลการทดลองจาก FPGA และ Filter ที่เราออกแบบโดยโปรแกรม MATLAB	100
รูปที่ 5.13 รูปแสดงผลการทดลองหาความถี่ที่กรองได้ขนาดแอมพลิจูดสูงสุด (500Hz)	101
รูปที่ 5.14 รูปแสดงผลการทดลองที่ความถี่ต่ำคัทออฟ (400Hz)	102

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.15 รูปแสดงผลการทดลองที่ความถี่สูงคัทออฟ (600Hz)

102

รูปที่ 6.1 รูปแสดงค่ากระแสที่เร้าวัดได้จากวงจร

104



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ตารางแสดงผลตอบสนองอิมพัลส์กรองแบบอุดมคติ	15
ตารางที่ 2.2 พารามิเตอร์ที่สำคัญ ของหน้าต่างแบบต่างๆ	16
ตารางที่ 5.1 ตารางแสดงผลการทดลองDigital Filter	89
ตารางที่ 5.2 ผลการทดสอบ โปรแกรมทดสอบการได้ยิน	94
ตารางที่ 5.3 แสดงผลการทดลองเมื่อป้อนความถี่ตั้งแต่ 100-1000Hz	99
ตารางที่ 5.4 แสดงผลการทดลองที่ความถี่คัทออฟ	101



บทที่ 1

บทนำ

1.1 หลักการและเหตุผล

ปัจจุบันมีคนเรามีโรคภัยมากมายทั้งที่เป็นมาตั้งแต่เกิด เกิดจากอุบัติเหตุ และสาเหตุอีกต่างหากที่ทำให้เราต้องมาเยียวยารักษาโรคเหล่านี้ที่เกิดกับเรา โรคบางโรคอาจเกิดกับเรามาตั้งแต่เกิด หรือหลังจากที่เราเกิดมา เช่น ผู้ที่บกพร่องทางการได้ยิน อาการนี้ทำให้ผู้ที่จำเป็นต้องใช้เครื่องช่วยฟังใน การเพิ่มประสิทธิภาพในการได้ยิน เครื่องช่วยฟังก็มีราคาตั้งแต่ราคาถูก จนถึงราคาแพงเป็นหมื่นบาทก็มี ประสิทธิภาพของเครื่องเหล่านี้ก็จะแตกต่างกันไป ผู้ที่บกพร่องทางการได้ยินนั้น บางคนไม่ได้บกพร่องในทุกระดับความถี่เสียง จึงทำให้ต้องมีการวัดตรวจระดับการได้ยินว่าผู้ที่บกพร่องทางการได้ยินนั้น บกพร่องที่ความถี่ไหน แล้วจึงทำเครื่องช่วยฟังที่ช่วยเพียงความถี่นั้นให้ดีขึ้น วิธีการที่เราจะนำเสนอวันนี้ทั้งการวัดความบกพร่องในการได้ยิน และเครื่องช่วยฟัง เราจะเขียนโปรแกรมเครื่องวัดความบกพร่องทางการได้ยิน โดยใช้โปรแกรม Visual Studio 6.0 ในการเขียน โปรแกรม เราได้นำความถี่ต่างๆมาใส่ไว้ใน โปรแกรมเพื่อทำการวัดตามมาตรฐานการวัด

เครื่องช่วยฟังนั้นเราในนำ FPGA มาใช้ในการทำ Filter กรองเสียงที่เราต้องการ โดยใช้ภาษา VHDL ในการเขียน โปรแกรมขึ้นมา ซึ่งค่าเฉพาะตัวกรองความถี่นั้นเราได้คำนวณมาจากโปรแกรม MATLAB ขั้นตอนในการทำงานนั้น เราจะรับสัญญาณเสียง(Analog)เข้ามาจากนั้นเราจึงแปลงเป็นสัญญาณดิจิตอล เพื่อทำการกรองความถี่ในตัว FPGA จากนั้นเราก็ค่อยแปลงกลับเป็นสัญญาณเสียงอีกครั้ง(Analog)

1.2 วัตถุประสงค์

เพื่อศึกษาการประยุกต์ใช้งานและการใช้งานพื้นฐานการเขียนโปรแกรมลงบน FPGA โดยใช้ภาษา VHDL ในส่วนของโปรแกรมกรองสัญญาณดิจิตอล (FIR) และ โปรแกรมทดสอบหูอย่างง่าย

1.3 ขอบเขตโครงการ

การศึกษางานและประยุกต์ใช้งานของ FPGA ในส่วนของการเขียนโปรแกรมลงบน FPGA นั้นเราได้ใช้ภาษา VHDL ในการเขียนส่วนของโปรแกรมการกรองสัญญาณดิจิตอล (FIR) โดยได้ใช้สัมประสิทธิ์ของวงจรกรองสัญญาณดิจิตอลจากโดยใช้เครื่องมือ FDATool จากโปรแกรม MATLAB ซิมูเลชันค่าสัมประสิทธิ์ออกมา และเราได้ใช้โปรแกรม XILINK ในการเขียนโปรแกรมและทดลอง และในส่วนของโปรแกรมทดสอบหูอย่างง่ายเราได้ใช้โปรแกรม Visual Studio 6.0 ในการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 ทฤษฎีหูและการตรวจสมรรถภาพการได้ยิน

การตรวจสมรรถภาพการได้ยิน

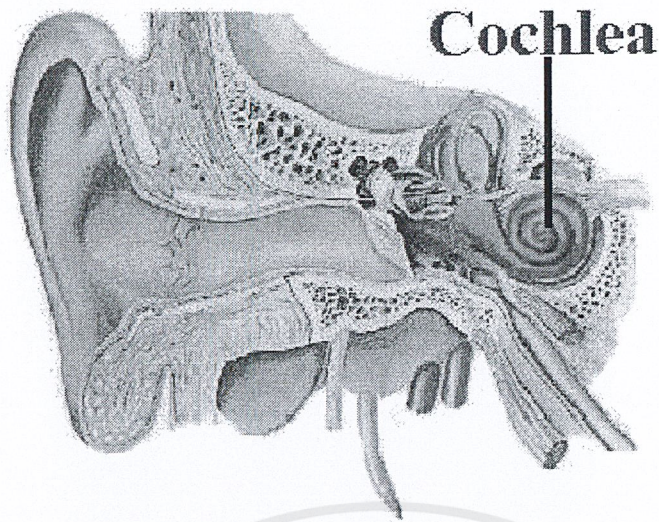
การตรวจสมรรถภาพการได้ยิน หรือ Audiography เป็นการตรวจการได้ยินเสียง ณ ความถี่ต่างๆ ตั้งแต่ระดับความถี่เสียงสนทนา จนถึงเสียงเครื่องจักร ซึ่งเป็นความถี่ที่ไม่ได้ยินกันในชีวิตประจำวัน หรือคนทั่วไปที่ไม่ได้มีหน้าที่เกี่ยวข้องโดยตรงจะไม่ได้มีโอกาสสัมผัส โดยการตรวจจะนำข้อมูลไปสร้างเป็นกราฟ เรียกว่า ออดิโอแกรม (Audiogram) ซึ่งการแปลผลว่ามีสมรรถภาพการได้ยินเป็นอย่างไรนั้นจะดูจากกราฟนี้ การที่เราจะเข้าใจว่าการตรวจสมรรถภาพการได้ยินมีประโยชน์อย่างไรควรทราบถึงกลไกการได้ยินและอันตรายของเสียงดังก่อนดังนี้

กลไกการได้ยินและอันตรายของเสียงดังต่อมนุษย์

คนเราสามารถได้ยินเนื่องจากคลื่นเสียงเคลื่อนที่จากหูชั้นนอก เข้าสู่หูชั้นกลาง แล้วเข้าสู่หูชั้นใน การทำงานของหูในช่วงตั้งแต่ใบหู รูหู กระดูกหูชั้นกลาง จัดเป็นการนำเสียงผ่าน โมเลกุลของอากาศ ซึ่งจะต่อไปยังหูชั้นกลาง ในหูชั้นกลางจะมีกระดูกหู 3 ชิ้นส่งคลื่นเสียงเข้าไปสู่หูชั้นใน ที่กระดูกนี้จะมี กล้ามเนื้ออยู่ซึ่งเมื่อมีเสียงดังมากเกินไปผ่านเข้ามา ร่างกายจะมีกลไกป้องกัน โดยให้กล้ามเนื้อนี้จะหดตัวอัตโนมัติช่วยจะลดระดับเสียงที่จะผ่านเข้าไปสู่หูชั้นในได้ประมาณ 30-40dB

ที่หูชั้นใน จะมีอวัยวะรูปก้นหอยเรียกคอคเคลีย (Cochlea) ภายในกลวงบรรจุของเหลวไว้ พื้นของคอคเคลียจะบุด้วยเซลล์ขน (Hair cell) ซึ่งทำหน้าที่รับความรู้สึกสัมผัสเปลี่ยนแปลงเป็นคลื่นประสาทส่งไปสมองเพื่อแปลความหมายของเสียงที่ได้ยิน การนำคลื่นเสียงจากอากาศมาสู่หูชั้นกลางเรียกว่า Conductive Function และการนำคลื่นเสียงจากการสัมผัสแล้วแปลเป็นกระแสประสาทเพื่อส่งไปรับรู้ที่สมองจะเรียกว่า Sensorineural Function ตัวเซลล์ขนนี้จะมีอายุขัยไม่เท่ากันและมีความจำเพาะเจาะจงต่อความถี่ใด ความถี่หนึ่ง ถ้าเสียงดังมากเกินไปการป้องกันโดยธรรมชาติของร่างกายคือถ้ามีระดับความเข้มของเสียงสูงกว่า 85 dB เมื่อคลื่นเสียงเดินทางมาถึงเซลล์ขนจะทำให้มีการสัมผัสต่อเนื่องเป็นเวลานาน ติดต่อกัน เซลล์ขนจะไม่สามารถปรับสภาพคืนสู่ปกติ และหลุดร่วงไป ก็จะเกิดการขาดช่วงการเดินทางของเสียงที่ไปยังสมอง เกิดการสูญเสียการได้ยินขึ้น และก็จะเป็นเฉพาะความถี่ของเสียงดังนั้นๆ คอคเคลียอยู่ในหูชั้นใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 รูปแสดง cochlea (คอคเคเลีย)

การที่เซลล์ขนถูกทำลาย ทำให้เกิดหูตึงได้ 2 ลักษณะ คือ

- Acoustic trauma คือ การสูญเสียการได้ยินอย่างฉับพลันเมื่อได้ยินเสียงดังมาก เช่น เสียงระเบิด เสียงปืน ฯลฯ
- Noise induced hearing loss คือ การสูญเสียการได้ยินแบบค่อยเป็นค่อยไป เกิดขึ้นในผู้ที่ทำงานอยู่ในที่ที่มีเสียงดังเป็นเวลานานๆ เช่น อุตสาหกรรมสิ่งทอ, อุตสาหกรรมเครื่องเรือน, อุตสาหกรรมถลุงเหล็ก, อุตสาหกรรมเครื่องแก้ว, อุตสาหกรรมเครื่องเหล็ก, โรงเลื่อย, ขับเรือหางยาว, ขับรถสามล้อเครื่อง, ตำรวจจราจร, นักจัดรายการดนตรี จากการศึกษาวิจัยพบว่า ในกลุ่มคนงานที่ทำงานสัมผัสกับเสียงที่ดังกว่า 85 dBA นาน 8 ชั่วโมง/วัน ติดต่อกันนาน 5 ปี มีโอกาสที่จะทำให้สมรรถภาพการได้ยินเสียไป

อันตรายของเสียงต่อสุขภาพทั่วไป

- 1) ทำให้การทำงานของระบบการไหลเวียนโลหิตระบบประสาทและระบบต่อมไร้ท่อทำงานผิดปกติ
- 2) ทำให้สมดุร่างกายเปลี่ยนแปลงโดยทำให้ความดันโลหิตสูงขึ้นกว่าปกติ การเดินของหัวใจผิดปกติ และการหดตัวของเส้นเลือดผิดปกติ

อันตรายของเสียงต่อความปลอดภัยในการทำงาน

- 1) ทำให้พฤติกรรมส่วนบุคคลเปลี่ยนแปลง เช่น เชื่องช้าต่อการตอบสนองสัญญาณต่างๆ และเกิดความวุ่นใจในการทำงานทำให้การทำงานผิดพลาดจนเกิดอุบัติเหตุได้
- 2) ครอบงำการทำงานทำให้ประสิทธิภาพการทำงานลดลงลักษณะของเสียงที่พบว่ามีผลต่อการลดประสิทธิภาพการปฏิบัติงานได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เสียงดังๆหยุดๆเป็นช่วง(Transient noise)
- เสียงที่มีความถี่สูงกว่า2,000Hz.
- เสียงที่ดังต่อเนื่องเป็นระยะเวลานาน(Continuous noise)

3)รบกวนการนอนหลับทำให้เกิดความอ่อนเพลียเมื่อปฏิบัติงานอาจเกิดความผิดพลาดได้ง่าย

4) รบกวนการติดต่อสื่อสาร

ผลกระทบของเสียงดังต่อสุขภาพ

มีหลักฐานชัดเจนเฉพาะในหัวข้อแรก (สูญเสียการได้ยิน)

1).สูญเสียการได้ยิน (Noise Induced Hearing Loss)

สูญเสียการได้ยินอย่างเฉียบพลันจากเสียงที่ดังมาก

สูญเสียการได้ยินแบบถาวรวัดได้หลายวิธีที่ใช้บ่อยคือวัดจากการสูญเสียการได้ยินเฉลี่ยในช่วงความถี่ 1, 2, 3 และ 4kHz ถ้าไม่ได้ยินเสียงที่ 25 dB ให้ถือว่ามีความผิดปกติของการได้ยิน การสูญเสียการได้ยินนี้อาจเกิดขึ้นเร็วมาก แต่โดยทั่วไปถ้าสัมผัสเสียงดัง 90 dB(A) TWA การสูญเสียจะเป็นดังนี้

- ใน 1-2 ปี เสียการได้ยินที่ 4kHz = 10 dB
- ใน 10 ปี เสียการได้ยินระหว่าง 3-6 kHz = 20 dB
- ใน 30 ปี เสียการได้ยิน ระหว่าง 2-6 kHz => 40 dB
- ผู้สัมผัสเสียงดัง 85 dB(A) เป็นเวลา 40 ปี ก็มีโอกาสสูญเสียการได้ยินได้ประมาณ 8%
- มีเสียงดังรบกวนในหู (Tinnitus)
- สูญเสียการได้ยินชั่วคราว Temporary Threshold Shift แม้จะมี TWA ตามตาราง แต่ได้ยินเสียงดังเป็นช่วงๆ การสูญเสียการได้ยินชั่วคราวนี้อาจกลายเป็นสูญเสียถาวรได้

2). ผลกระทบนอกเหนือจากการได้ยิน ได้แก่

- ผลกระทบต่ออารมณ์ (Psychological Stress) หงุดหงิด โมโหง่าย
- กล้ามเนื้อแข็งตึง
- ระบบย่อยอาหารผิดปกติ
- ความดันโลหิตสูง
- หัวใจขาดเลือด

ประเภทของความสูญเสียการได้ยิน

ประเภทของความสูญเสียการได้ยิน แบ่งได้ 5 ประเภท ดังนี้

1). การนำเสียงบกพร่อง (Conductive hearing loss) ความผิดปกติเกิดขึ้นในหูชั้นนอกและชั้นกลาง แต่ ประสาทหูยังดีอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาการ : มีของเหลวออกจากช่องหูอาจจะเป็นเลือดหรือหนอง มีประวัติการอักเสบของช่องหูมาก่อน การพูดคุยมักพูดเสียงเบาทุ้มนุ่มนวล การได้ยินจะดีขึ้นชัดเจนเมื่ออยู่ในที่จืดแต่ไม่ค่อยดีในที่เสียงๆ มักมีปัญหาในการฟังเสียงขณะเคี้ยวอาหาร บางรายมีเสียงรบกวนในหู (tinnitus) เป็นเสียงต่ำๆ การพูดจาชัดเจนออกเสียงได้ตามปกติ ตรวจการได้ยินพบการสูญเสียในช่วงความถี่ต่ำๆ และมักไม่มากกว่า 60 dBHL

สาเหตุ:

1). โรคหรือความผิดปกติที่หูชั้นนอก : หูพิการตั้งแต่กำเนิด สิ่งแปลกปลอมทำให้เกิดการอุดตันในช่องหู จี๋หูอุดตัน (แผ่นช่องหูอักเสบวม จนช่องหูตีบตัน โรคเนื้องอกในช่องหูชั้นนอก ช่องหูพับลง

- โรคหรือความผิดปกติที่แก้วหู : มีรูทะลุที่เยื่อแก้วหู แก้วหูอักเสบ เยื่อแก้วหูหนา
- โรคหรือความผิดปกติในหูชั้นกลาง : มีเลือดออกในหูชั้นกลาง, โรคหูน้ำหนวก (ทั้งชนิดมีน้ำไหลและแห้ง), โรคหูชั้นกลางมีหินปูนจับแข็ง, ภาวะแทรกซ้อนจากการติดเชื้อไวรัส, กระดูก 3 ชิ้นแตกหรือหัก

2). ประสาทรับฟังเสียงบกพร่อง (Sensorinural hearing loss) ความผิดปกติที่เกิดขึ้นในหูชั้นใน (cochlea) หรือที่ประสาทรับฟังเสียง (acoustic nerve)

อาการ : ถ้ามีการสูญเสียของประสาทหูมากกว่า 2 ข้างและเป็นเวลานาน เสียงพูดจะดังมากกว่าปกติ เพราะไม่ได้ยินเสียงตัวเอง มีเสียงรบกวนในหูเป็นเสียงสูงๆ จะฟังเสียงพูดได้ดีเมื่ออยู่ในที่สงบและจะไม่ค่อยเข้าใจคำพูดเมื่ออยู่ในที่จืดแจ่ม มักไม่ค่อยเข้าใจคำพูดแม้ว่าเสียงพูดนั้นดังถึงระดับการได้ยินปกติแล้วก็ตาม มักมีอาการเวียนศีรษะแบบบ้านหมุนร่วมด้วย ถ้าประสาทหูเสียมากกว่า 2 ข้างหรือเป็นมาแต่กำเนิดมักจะพูดไม่ชัดหรือพูดไม่ได้ ไม่มีประวัติของการปวดหู หรือมีของเหลวไหลออกจากหู ตรวจการได้ยินพบการสูญเสียในช่วงความถี่สูงๆ

สาเหตุ :

- ประสาทรับฟังเสียงบกพร่องแต่กำเนิด : ขาดออกซิเจนขณะอยู่ในครรภ์หรือระหว่างคลอด, ติดเชื้อแต่กำเนิดหรือหลังคลอด เช่น ซิฟิลิส หัด หัดเยอรมัน คางทูม สุกใส ไข้หวัดใหญ่ ปอดอักเสบ, การอักเสบของเยื่อหุ้มสมองหรือหูชั้นใน
- ประสาทรับฟังเสียงบกพร่องจากยา : ผู้ป่วยจะมีการสูญเสียการได้ยินของหูทั้ง 2 ข้างพร้อมๆ กัน ยาบางชนิดทำให้มีอาการชั่วคราว เมื่อหยุดยาการได้ยินอาจกลับคืนมาได้ แต่ยาบางชนิดทำให้มีอาการถาวรรักษาไม่หาย เช่น kanamycin, streptomycin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ประสาทรับฟังเสียงบกพร่องจากเสียงดัง (noise induced hearing loss)
- โรคที่เกิดจากความผิดปกติเกี่ยวกับปริมาณของของเหลวในหูชั้นใน (Meniere's disease) ทำให้มีอาการหูอื้อ เวียนศีรษะ บ้านหมุน คลื่นไส้ อาเจียน และมีเสียงรบกวนในหู อาจเป็นหูเดียวหรือสองหูก็ได้ อาการของโรคจะเป็นซ้ำๆ กัน มีอาการเป็นๆ หายๆ
- ประสาทหูพิการจากการจับแฉ่งของกระดูกในหูชั้นใน
- ประสาทหูบกพร่องในวัยชรา (Presbycusis hearing loss) ความผิดปกติเกิดขึ้นจากเซลล์ขนที่อยู่บริเวณฐานของก้นหอยในหูชั้นในมีการเสื่อมไปตามอายุ ทำให้การรับฟังเสียงสูงๆ ได้ไม่ดี มักมีเสียงดังในหูเป็นเสียงสูงๆ ตรวจช่องหูไม่พบสิ่งผิดปกติ มีความผิดปกติของการได้ยินของหูทั้งสองข้าง มักพบในคนที่อายุ 40 ปีขึ้นไป
- ศีรษะถูกกระทบกระเทือน ทำให้ประสาทรับฟังเสียงบกพร่องเล็กน้อยไปจนถึงระดับรุนแรง

3). การรับฟังเสียงบกพร่องแบบผสม (Mixed hearing loss) เป็นภาวะที่เกิดจากความผิดปกติในระบบการนำเสียงร่วมกับประสาทรับฟังเสียงบกพร่อง พบในโรคที่มีความพิการที่หูชั้นนอก ชั้นกลาง และชั้นในร่วมกัน เช่น โรคหูน้ำหนวกเรื้อรังซึ่งอาการลุกลามเข้าไปในหูชั้นใน โรคหินปูนจับแฉ่งที่กระดูกโกลน

4). ความผิดปกติทางจิต (Functional or Psychological hearing loss)

5). ความบกพร่องที่สมองส่วนกลาง (Central Hearing Impairment) สมองไม่สามารถรับและแปลความหมายได้ จึงไม่สามารถเข้าใจความหมายของเสียงที่ได้ยิน เช่น โรคเส้นเลือดในสมองแตก ทำให้ศูนย์การรับฟังไม่สามารถใช้งานได้

การสูญเสียการได้ยินเนื่องจากเสียงดัง (Noise Induced Hearing Loss)

1). ประสาทหูผิดปกติเนื่องจากเสียงดังรบกวน

การสูญเสียความสามารถในการได้ยินชั่วคราว (Temporary thresholds shift : TTS)

เซลล์ประสาทรับฟังการได้ยินมีอาการล้าจากการสัมผัสเสียงดังต่อเนื่องเป็นเวลานานๆ ไม่สามารถ

แปลสัญญาณการสัมผัสเป็นคลื่นประสาทได้ เกิดอาการหูตึงชั่วคราว (Auditory fatigue)

อาการหูตึงนี้มักร่วมกับมีเสียงดังในหู (tinnitus) ในกรณีสงสัยว่าจะสูญเสียความสามารถในการได้

ยินชั่วคราว ควรให้พนักงานพักจากการฟังเสียงที่ต่ำกว่า 70 dBA อย่างน้อย 48 ชั่วโมง

การสูญเสียความสามารถในการได้ยินถาวร (Permanent threshold shift : PTH)

เมื่อผู้ป่วยมีอาการล้าของเซลล์รับเสียงจนไม่สามารถได้ยินเสียงในระดับปกติ หากยังสัมผัสกับเสียง

ดังต่อเนื่องอีกก็จะทำให้เซลล์รับเสียงถูกทำลายอย่างถาวร (Degenerative change of hair cell)

- ในระยะแรกการสูญเสียการได้ยินจะเริ่มเสียที่ช่วงความถี่ของเสียง 3,000 – 6,000 Hz. และจะพบเสมอว่าจะเสียที่ความถี่ของการได้ยินที่ 4,000 Hz. ก่อนความถี่อื่นๆ

- เริ่มมีเสียงดังรบกวนในหู ความไวของหูในการรับเสียงลดลง แต่พอเลิกงานไม่ได้อยู่ในที่ที่มีเสียงดังจะรู้สึกว่าการได้ยินดีขึ้น อาจมีอาการปวดหูหรือเวียนศีรษะร่วมด้วย
- เมื่อทำงานในที่ที่มีเสียงดังเป็นระยะเวลานานๆจะมีการสูญเสียการได้ยินไปที่ละน้อย โดยไม่รู้ตัว จนลุกลามไปถึงช่วงความถี่ของการพูดคุย (500 – 2,000 Hz.) ทำให้การรับฟังเสียงคำพูดไม่เข้าใจ ถ้าผิดปกติมากจะไม่ทราบทิศทางของเสียงที่ได้ยิน
- ตรวจภายในช่องหูไม่พบสิ่งผิดปกติ ตรวจวัดการได้ยินด้วยเครื่องตรวจวัดการได้ยิน จะได้กราฟลักษณะเส้นประสาทหูผิดปกติ (ดังรูป)

2). ประสาทหูผิดปกติเนื่องจากมีเสียงดังมาก ๆ

- หูอื้อทันทีหลังจากได้รับเสียงดัง
- มีเสียงดังในหูตลอดเวลา
- มักฟังคำพูดเข้าใจดี เนื่องจากการได้ยินไม่เสียที่บริเวณความถี่ของการพูดคุย
- เมื่อตรวจวัดการได้ยินพบว่ามึลักษณะความผิดปกติ
- ตรวจภายในช่องหูพบว่า ช่องหูชั้นนอกปกติ แต่อาจมีแก้วหูทะลุร่วมด้วย

การตรวจสมรรถภาพการได้ยิน

กลุ่มคนที่ควรได้รับการตรวจสมรรถภาพการได้ยิน

คนงานใหม่ต้องได้รับการทดสอบการได้ยินก่อนการรับเข้าทำงาน หรือภายใน 6 เดือนแรกที่สัมผัสเสียงเฉลี่ย 8 ชั่วโมงที่ระดับ 85 dBA หรือสูงกว่า อย่างน้อยเป็นประจำทุกปี

การตรวจหูและประเมินการได้ยิน

- 1) Otoscopy การตรวจโดยใช้โอโตสโคป ดูสภาพภายในช่องหูชั้นนอก และเงาของช่องหูชั้นกลาง เพื่อตรวจดูสถานะการอักเสบภายใน
- 2) การตรวจการได้ยินโดยใช้ส้อมเสียง (tuning-fork) ใช้เพื่อทดสอบการได้ยินอย่างคร่าวๆ ทราบผลได้อย่างรวดเร็ว มีวิธีการตรวจ 2 วิธี คือ
 - Weber test แยกการนำเสียงพร้อมทั้งประสาทรับฟังเสียงพร้อม ในผู้ที่หูเสีย 1 ข้าง โดยการเคาะส้อมเสียงแล้ววางไว้ที่แนวกลางของศีรษะ
 - Rinne test เพื่อเปรียบเทียบการนำเสียงทางอากาศ(AC) กับ การนำเสียงทางกระดูก(BC) ในหูข้างเดียวกัน โดยวางส้อมเสียงไว้ที่หน้าใบหูและที่หลังใบหูบริเวณกระดูก Mastoid
- 3) การตรวจการได้ยินด้วยเครื่อง Audiometer เป็นการตรวจวัดระดับความดังเสียงต่ำสุด ที่ผู้เข้ารับการตรวจสอบสามารถได้ยินที่ความถี่ต่างๆ

วิธีการตรวจหาระดับการได้ยิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) Routine Audiometry เป็นการทดสอบที่ทำเป็นประจำในคลินิก เพื่อการวินิจฉัยโรค หรือติดตามผลการรักษา

- Puretone Air Conduction (AC) คือการตรวจวัดการได้ยิน โดยการนำเสียงทางอากาศ
- Puretone Bone Conduction (BC) คือการตรวจวัดการได้ยิน โดยการนำเสียงทางกระดูก
- Speech Audiometry คือการวัดการได้ยิน โดยใช้คำพูด

2) Masking Audiometry คือการวัดการได้ยินเสียงโดยวิธีระมัดระวังเป็นพิเศษ ไม่ให้เสียงที่ตรวจในหูข้างหนึ่งข้ามกระโหลกศีรษะมายังหูอีกข้างหนึ่ง โดยใช้เสียงรบกวนปล่อยเข้าไปรบกวนหูด้านที่ติ ขณะที่กำลังตรวจวัดหูอีกข้างหนึ่ง

3) Special Audiometer เป็นการทดสอบพิเศษนอกเหนือไปจากการทดสอบประจำในคลินิกเพื่อหาโรคหรือความผิดปกติของหู

เทคนิควิธีการตรวจการได้ยิน

1) Descending Technique โดยการปล่อยระดับเสียงที่ดัง เพื่อให้ผู้ถูกทดสอบได้ยินก่อนแล้วค่อยๆ ลดความดังลงทีละน้อย ทีละ 10 dBHL จนถึงจุดหนึ่งที่ถูกทดสอบ ไม่ได้ยินเสียง ให้เพิ่มระดับเสียง จากจุดที่ไม่ได้ยิน ทีละ 5 dBHL หากไม่ได้ยินก็ให้เพิ่มอีก 5 dBHL จนเริ่มได้ยิน แล้วลดลงไปอีก 10 dBHL เมื่อแน่ใจว่าผู้ถูกทดสอบได้ยินแน่ชัดที่จุดนั้นๆ ให้ลดลง 10 dBHL อีกครั้ง ถ้าไม่ได้ยิน ให้เพิ่มขึ้น 5 dBHL ทำกลับไปกลับมาจนได้จุดที่ถูกทดสอบได้ยินโดยใช้ระดับเสียงเบาที่สุดที่ถูกตรวจสามารถตอบสนองได้ร้อยละ 50 ถึง 70 ของจำนวนครั้งที่ให้สัญญาณ จุดนั้นคือ hearing threshold

2) Ascending Technique ใช้ในกรณีที่ผู้ถูกทดสอบอายุน้อย หรือหูหนวกมากๆ รวมทั้งผู้ที่ไม่แน่ใจว่าจะแสร้งทำเป็นหูหนวกหรือไม่ วิธีนี้เริ่มจากความตั้งใจที่ผู้ถูกทดสอบ ไม่ได้ยินก่อน แล้วเพิ่มความดังทีละ 10 dBHL จนถึงจุดที่ถูกทดสอบเริ่มได้ยินเสียงเบาที่สุด แล้วลดเสียงลง 5 dBHL ทำกลับไปกลับมาจนได้จุดที่ถูกทดสอบได้ยินเสียงบ้าง ไม่ได้ยินเสียงบ้าง จุดนั้นคือ hearing threshold

3) Combination Technique ใช้วิธีผสมระหว่างวิธีที่ 1 และที่ 2 โดยใช้ระดับเสียงดัง-เบาสลับกันไป การเตรียมผู้ถูกทดสอบ

ก่อนตรวจผู้ถูกทดสอบควรรับฟังเสียงดังเกิน 80 dBA เป็นเวลา 8 - 16 ชั่วโมง เพื่อหลีกเลี่ยงการเกิดหูตึงแบบชั่วคราว (TTS) หากไม่สามารถทำได้ในทางปฏิบัติ ต้องสวมใส่ที่ครอบหู ลดเสียงตลอดเวลาที่สัมผัสเสียงก่อนการทดสอบ

การตรวจการได้ยิน โดยการนำเสียงผ่านทางอากาศ (AC)

- 1) ให้ผู้ถูกทดสอบนั่งในห้องที่มีระดับเสียงในห้องตามมาตรฐานกำหนดไม่เกิน 40 dA ในทุกความถี่
- 2) อธิบายให้ผู้ถูกทดสอบเข้าใจถึงเสียงสัญญาณที่จะได้ยิน และการกดสวิทช์สัญญาณตอบรับ
- 3) ให้ผู้ถูกทดสอบนั่งหันหลังให้ผู้ทำการทดสอบและใช้ head phone สีแดงครอบที่หูขวา สีน้ำเงินครอบที่หูซ้าย
- 4) การสอบถามผู้ถูกทดสอบ และทำการทดสอบในหูข้างที่ติ๊กก่อน เริ่มทดสอบ hearing threshold ที่ความถี่ 1,000 Hz. แล้วหาต่อไปที่ 2,000 3,000 4,000 6,000 และ 8,000 Hz. แล้วกลับมาทดสอบซ้ำที่ 1,000 Hz. ใหม่ แล้วหาต่อไปที่ 500, 250 Hz. ตามลำดับ
- 5) ทำการตรวจการได้ยินของหูอีกข้างตามวิธีข้างต้น

หมายเหตุ

- หาก hearing threshold ที่ความถี่ 2 ความถี่ ต่างกันเกินกว่า 20 dBHL เช่น ที่ความถี่ 1,000 กับ 2,000 ต่างกัน 25 dBHL ก็ควรหา hearing threshold ที่ความถี่ 1,500 Hz. ด้วย
- ถ้าพบว่าการได้ยินของหูทั้งสองข้างต่างกันเกิน 30 dBHL จากการตรวจแบบ AC หรือพบว่าหูข้างเดียวกันมีค่า BC ต่ำกว่า AC เกินกว่าหรือเท่ากับ 15 dBHL ควรใส่เสียงกลบรบกวน (masking : narrow band noise) ในหูข้างที่ติ๊กว่าหรือด้านตรงข้ามกับที่กำลังตรวจอยู่ เพื่อป้องกันการได้ยินเสียงจากหูข้างที่ติ๊กว่า

การบันทึกผลการตรวจ

ให้ใช้เครื่องหมายสำหรับการบันทึกผลการตรวจการได้ยินตามหลักสากลนิยม ดังนี้ การตรวจการได้ยิน โดยการนำเสียงผ่านทางอากาศ ใช้เครื่องหมาย O สีแดง สำหรับหูข้างขวา X สีน้ำเงิน สำหรับหูข้างซ้าย การลากเส้นใช้เส้นทึบ การตรวจการได้ยิน โดยการนำเสียงผ่านทางกระดูก ใช้เครื่องหมาย < สีแดง สำหรับหูข้างขวา > สีน้ำเงิน สำหรับหูข้างซ้าย การลากเส้นใช้เส้นประ

ตรวจไปเพื่ออะไร?

เป็นการตรวจเพื่อเฝ้าระวังว่ามีการสูญเสียสมรรถภาพการได้ยินจากการทำงานหรือไม่ ซึ่งจะเกิดกับบุคลากรที่ต้องสัมผัสหรือทำงานในที่ที่มีเสียงดัง ซึ่งเป็นการกระตุ้นเตือนให้รักษาสุขภาพของความปลอดภัยในการทำงานเสมอ

นอกจากนี้ยังเป็นการตรวจเพื่อค้นหาผู้ที่มีความผิดปกติในการได้ยินในระดับที่เป็นมาก เช่น หูตึงมาก หรือหูตึงรุนแรง เพื่อช่วยในการรักษา ตลอดจนดูแลให้ใช้เครื่องช่วยการได้ยิน เพื่อจะได้มีคุณภาพชีวิตที่ดีขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการตรวจ

อาศัยการแปลผลจากกราฟ โดยจะผลการตรวจจะมี 2 ส่วนคือ

- 1). ระดับการได้ยิน
 - 2). มีความผิดปกติในช่วงคลื่นเสียงความถี่สูงหรือต่ำร่วมด้วยหรือไม่
- โดยผลการตรวจจะแบ่งเป็นระดับดังนี้

ผู้ที่มีหูตึง

ในกรณีที่ เป็นเล็กน้อยถึงปานกลาง อาจทำให้เสียบุคลิกบ้าง ควรใส่เครื่องป้องกันทุกครั้งเข้าสู่ที่ บริเวณที่มีเสียงดัง หลีกเลี่ยงการใช้ยาที่มีพิษต่อระบบประสาท สำหรับผู้ที่มีหูตึงในระดับมากหรือรุนแรง ควรปรึกษาแพทย์ทางหู-คอ-จมุก เพื่อพิจารณาว่ามีความจำเป็นต้องใช้เครื่องช่วยการได้ยินหรือไม่ ผู้ที่มี ระดับการได้ยินปกติแต่มีความผิดปกติของการได้ยินที่ความถี่สูง (หรือความถี่ต่ำ) ร่วมด้วย

หมายความว่า การได้ยินของท่านเป็นปกติดี ท่านสามารถพูดคุย สื่อสารในชีวิตประจำวัน ได้อย่าง ปกติ แต่ระดับการได้ยินนั้นเริ่มมีการสูญเสียที่ความถี่สูง (หรือต่ำ) ซึ่งไม่ใช่เสียงที่คนเราพูดคุยกัน มักเป็น เสียงเครื่องจักร, โลหะ, เสียงนาฬิกา เป็นต้น ส่วนใหญ่มักเกิดจากได้รับเสียงดังๆ เป็นเวลานาน เกณฑ์การ ประเมินและการแบ่งระดับความบกพร่องของการได้ยิน

โรคหูตึงจากการประกอบอาชีพ หมายถึง โรคหูตึงเนื่องจากฟังเสียงดังในการทำงานจนประสาทหู เสื่อม อาจเป็นข้างเดียวหรือสองข้างก็ได้ ผลการตรวจทางห้องปฏิบัติการภาพบันทึกการได้ยิน (audiogram) ต้องมีลักษณะเป็นรูปอักษร V ที่บริเวณ 4,000 เฮิรตซ์ (3,000 - 6,000 Hz) และมีระดับการได้ยินเกิน 25 dBHL การแบ่งระดับความบกพร่องของการได้ยิน

จะพิจารณาจากค่า AC เท่านั้น โดยใช้ค่าเฉลี่ยของระดับการได้ยินที่สำคัญสำหรับการรับฟังเสียงพูด คือ 500, 1000, และ 2,000 Hz. มาคิดคำนวณ

หากค่าเฉลี่ยของการได้ยินในหูทั้ง 2 ข้าง มีค่าแตกต่างกันมากกว่า 25 dBHL ให้บวกอีก 5 dBHL เข้า กับการได้ยินในหูข้างที่ตีกว่านั้น แล้วพิจารณาค่าที่บวกได้ใหม่กับเกณฑ์ประเมิน ตัวอย่างเช่น ค่าเฉลี่ยในหู ขวาเท่ากับ 35 dBHL หูซ้ายเท่ากับ 65 dBHL ต่างกันเกิน 25 dBHL ต้องบวก 5 dBHL เข้ากับค่าเฉลี่ยการได้ ยินของหูขวาเป็น 40 dBHL ความพิการของหูเป็นระดับหูตึงปานกลาง

ระดับความบกพร่องการได้ยิน		
ตารางแบ่งระดับความบกพร่องของการได้ยิน		
ระดับการได้ยิน	ค่าเฉลี่ยความไวของหู ความถี่ 500 - 2000 Hz. ของหูข้างที่ต่ำกว่า (dBHL)	ความสามารถในการเข้าใจคำพูด
หูปกติ	ไม่เกิน 25	ไม่ลำบากในการรับฟังคำพูด
หูดีเล็กน้อย	25 - 40	ไม่ได้ยินเสียงพูดเบา
หูดีปานกลาง	40 - 55	พูดด้วยควมดังปกติแล้วไม่ได้ยิน
หูดีมาก	55 - 70	พูดด้วยดังกว่าตัวก็ยังไม่ได้ยิน
หูดีอย่างแรง	70 - 90	ต้องตะโกนหรือใช้เครื่องขยายเสียงจึงจะได้ยินและได้ยินไม่ชัดด้วย
หูหนวก	มากกว่า 90	ตะโกนหรือขยายเสียงพูดแล้วก็ยังไม่ได้ยินและไม่เข้าใจ

รูปที่ 2.2 รูปแสดงตารางการบกพร่องทางการได้ยิน

ระดับความบกพร่องการได้ยิน		
ระดับการได้ยิน	ระดับความดังที่เบาที่สุดที่จะได้ยิน	ตัวอย่างเสียง
หูปกติ	25 dB	เสียงกระซิบ
หูดีเล็กน้อย	20-40 dB	เสียงพูดในห้องเงียบ
หูดีปานกลาง	41-55 dB	เสียงพูด สทนาเป็นกลุ่ม
หูดีมาก	55-70 dB	เสียงเครื่องดูดฝุ่น
หูดีอย่างแรง	71-90 dB	เสียงในห้องประชุม โรงภาพยนตร์, ระเบิดทิ้ง เครื่องตี
หูหนวก	91-120 หรือมากกว่า dB	เสียงเครื่องบินเจ็ทกำลังขึ้น, เสียงเรือหางยาว, รถจักร

รูปที่ 2.3 รูปแสดงตารางระดับความบกพร่องทางการได้ยิน

2.2 ทฤษฎี Finite Impulse Response (FIR)

การออกแบบและทดสอบวงจรกรองความถี่แบบดิจิทัล (HPF) ที่มีผลตอบสนอง อิมพัลส์จำนวนจำกัด“

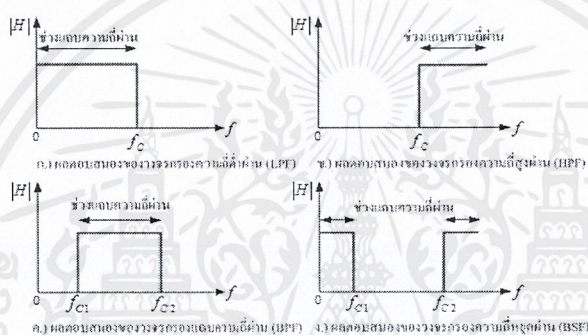
Finite Impulse Response : FIR “ ด้วยโปรแกรม MATLAB

ระบบสื่อสารในเรื่องของวงจรที่จะลืมนำไปได้นั้นก็จะเป็นวงจรกรองความถี่นั่นเอง เพราะถือว่าเป็นวงจรทางอิเล็กทรอนิกส์ที่มีความสำคัญอย่างหนึ่ง ด้วยคุณสมบัติในการทำงาน หรือผลตอบสนองในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานทางด้านความถี่ ด้วยคุณสมบัติหรือผลตอบสนองนี้ จึงได้มีการนำไปประยุกต์ใช้งานในด้านต่างๆ เช่น ทางด้านระบบอิเล็กทรอนิกส์, ระบบสื่อสาร, ระบบควบคุม และทางด้านเครื่องมือแพทย์ โดยในทุกวันนี้ วงจรกรองความถี่นั้นถ้าพิจารณาแล้ว สามารถที่จะแบ่งออกได้เป็นสองรูปแบบด้วยกันครับ นั่นก็คือวงจรกรองความถี่แบบแอนะล็อก(analog filter) กับวงจรกรองความถี่แบบดิจิทัล(digital filter)

ส่วนหน้าที่ในการทำงานของวงจรกรองความถี่นั้นก็สามารถที่จะทำการจำแนกตามความต้องการของผู้ใช้ โดยจะแบ่งตามคุณลักษณะของผลตอบสนองทางความถี่(frequency response) ดังรูปที่ 1 ได้ 4 ชนิดด้วยกันคือ วงจรกรองความถี่ต่ำผ่าน(low-pass filter ; LPF), วงจรกรองความถี่สูงผ่าน(high-pass filter ; HPF), วงจรกรองแถบความถี่ผ่าน(band-pass filter ; BPF) และวงจรกรองแถบความถี่หยุดผ่าน(band-stop filter ; BSF)



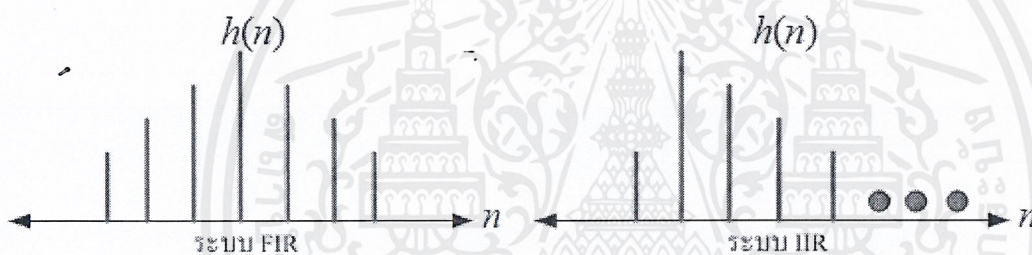
รูปที่ 2.4 ผลตอบสนองความถี่ของวงจรกรองความถี่ในอุดมคติ

จากผลตอบสนองความถี่ของวงจรกรองความถี่ในทางอุดมคติ ในรูปที่ 1 เมื่อให้ คือขนาดของแรงดันทางด้านเอาต์พุต วงจรกรองความถี่ต่ำผ่าน จะยอมให้ความถี่ตั้งแต่ 0 Hz ถึงความถี่ ผ่าน ไปยังขั้วเอาต์พุตของวงจรได้ ส่วนความถี่ที่สูงกว่า ความถี่จะไม่ผ่านไปยังขั้วเอาต์พุตของวงจร สำหรับวงจรกรองความถี่สูงผ่าน จะยอมให้ความถี่สูงกว่าความถี่ ผ่าน ไปยังขั้วเอาต์พุตของวงจรได้ ส่วนความถี่ตั้งแต่ 0Hz ถึงความถี่ จะไม่ผ่านไปยังขั้วเอาต์พุตของวงจร สำหรับวงจรกรองแถบความถี่ผ่าน จะยอมให้ความถี่ตั้งแต่ ถึงความถี่ ผ่าน ไปยังขั้วเอาต์พุตของวงจร ส่วนความถี่ตั้งแต่ 0Hz ถึงความถี่ กับความถี่ที่สูงกว่า จะไม่ผ่านไปยังขั้วเอาต์พุตของวงจร และวงจรกรองแถบความถี่หยุดผ่าน จะไม่ยอมให้ช่วงความถี่ ถึงความถี่ ผ่าน ไปยังขั้วเอาต์พุตของวงจร ส่วนความถี่อื่นๆ วงจรยอมให้ผ่านไปยังขั้วเอาต์พุตได้ วงจรกรองความถี่ที่ใช้งานกันอยู่ทั่วไป มักนิยมใช้วงจรกรองความถี่แบบแอนะล็อก ซึ่งประกอบไปด้วย ตัวความต้านทาน, ตัวเก็บประจุ, ตัวเหนี่ยวนำ และอุปกรณ์กึ่งตัวนำ เช่น ออปแอมป์ ข้อดีคือออกแบบได้ง่าย ราคาถูก แต่มีข้อเสียที่วงจรขาดเสถียรภาพ (stability) ความถี่ที่ต้องการมีความคลาดเคลื่อนสูง แต่ในปัจจุบันได้หันมานิยมใช้วงจรกรองความถี่แบบ

ดิจิตอลกันมาก เพราะมีเสถียรภาพที่ดีกว่า ความถี่ที่ต้องการมีความคลาดเคลื่อนน้อยกว่า แต่มีข้อเสียคือ การออกแบบทำได้ยากกว่า และมีราคาสูงกว่า สำหรับบทความนี้จะขอกล่าวถึงการออกแบบและทดสอบวงจรกรองความถี่แบบดิจิตอล (HPF) โดยใช้โปรแกรมMATLAB

การแบ่งประเภทของวงจรกรองความถี่แบบดิจิตอล

การแบ่งประเภทของวงจรกรองความถี่แบบดิจิตอล จะแบ่งตามผลตอบสนองอิมพัลส์ของระบบดังแสดงในรูปที่ 2 คือผลตอบสนองอิมพัลส์จำนวนจำกัด(Finite Impulse Response : FIR) กับผลตอบสนองอิมพัลส์จำนวนไม่จำกัด(Infinite Impulse Response : IIR) แต่ในที่นี้ขอยกตัวอย่างการออกแบบวงจรกรองความถี่แบบดิจิตอลที่มีผลตอบสนองอิมพัลส์จำนวนจำกัด หรือ FIR เพราะมีลักษณะเด่นคือ มีผลตอบสนองทางเฟสแบบเชิงเส้น (linear phase) เหมาะสำหรับการพัฒนาเครื่องมือวัดทางการแพทย์ และในงานที่เกี่ยวข้อง

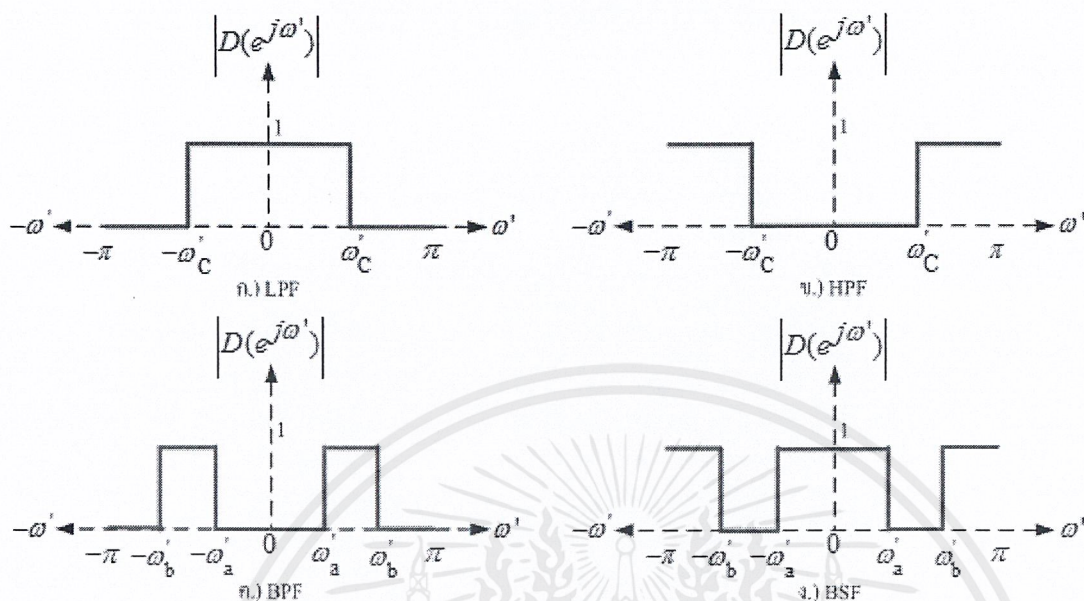


รูปที่ 2.5 ผลตอบสนองต่อสัญญาณอิมพัลส์ของวงจรกรอง FIR และ IIR

การออกแบบวงจรกรองความถี่โดยวิธีหน้าต่าง(window method)

ในการออกแบบวงจรกรองความถี่โดยวิธีหน้าต่างนั้น ซึ่งเป็นวิธีพื้นฐาน โดยมีสิ่งที่สำคัญคือการทำผลตอบสนองอิมพัลส์ของตัวกรองความถี่ต้นแบบ จากผลตอบสนองของความถี่ของตัวกรองความถี่ในอุดม

คติ ดังแสดงในรูปที่ 3



รูปที่ 3 ผลตอบสนองความถี่ของตัวกรองอุดมคติ

รูปที่ 2.6 รูปแสดงผลตอบสนองความถี่ของตัวกรองอุดมคติ

จากรูปที่ 3 จะสมมติให้ แทนเป็นผลตอบสนองต่ออิมพัลส์ และกำหนดให้ แทนผลตอบสนองเชิงความถี่ สำหรับตัวกรองความถี่ในอุดมคติของวงจรความถี่ต่ำผ่าน ดังที่ได้แสดงในรูปที่ 3(ก) เราจะหาผลตอบสนองต่ออิมพัลส์ของตัวกรองความถี่ในอุดมคติได้ โดยใช้เทคนิคของการแปลงฟูรีเยร์แบบเวลาไม่ต่อเนื่องชนิดผกผัน (Inverse Discrete Fourier Transform : IDTFT) ตัวอย่างเช่น วงจรกรองความถี่ต่ำผ่าน ดังนี้

$$\begin{aligned}
 d(n) &= \text{IDTFT}\{D\} \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} D(e^{j\omega'}) e^{j\omega'_n} d\omega' = \frac{1}{2\pi} \int_{-\pi}^{\pi} (1) e^{j\omega'_n} d\omega' \\
 &= \left[\frac{e^{j\omega'_n}}{2\pi j n} \right]_{-\omega'_C}^{\omega'_C} = \frac{e^{j\omega'_n} - e^{-j\omega'_n}}{2\pi j n} \\
 &= \frac{\sin(\omega'_C n)}{\pi n} \quad ; \quad -\infty < n < \infty \quad \dots(1)
 \end{aligned}$$

ในสมการที่ 1 จะพบปัญหาที่ เพราะผลตอบสนองต่ออิมพัลส์ มีค่าเป็นเศษศูนย์ส่วนด้วยศูนย์วิธีแก้ไขปัญหามหา เราสามารถหาค่า ได้โดยใช้ทฤษฎีของโลปีตัล จะได้ว่า

$$d(0) = \frac{\lim_{n \rightarrow 0} \frac{d(\sin(\omega'_C n))}{d(n)}}{\lim_{n \rightarrow 0} \frac{d(\pi n)}{dn}} = \frac{\omega'_C}{\pi} \quad \dots(2)$$

สำหรับวงจรกรองความถี่แบบอื่นๆ เราสามารถหาผลตอบสนองต่ออิมพัลส์ ได้โดยการแปลง IDTFT เพื่อหาผลตอบสนองเชิงความถี่ของตัวกรองนั้นๆ ได้ดังตารางที่ 1 เพื่อใช้ในการออกแบบต่อไป

ตารางที่ 2.1 ตารางแสดงผลตอบสนองอิมพัลส์กรองแบบอุดมคติ

ตารางที่ 1 ผลตอบสนองอิมพัลส์ของตัวกรองอุดมคติแบบต่างๆ

ชนิดของตัวกรอง	$D(n), -\infty < n < \infty$	$D(n), n = 0$
วงจกรองความถี่ต่ำผ่าน	$\frac{\sin(\omega'_C n)}{\pi n}$	$\frac{\omega'_C}{\pi}$
วงจกรองความถี่สูงผ่าน	$\delta(n) - \frac{\sin(\omega'_C n)}{\pi n}$	$1 - \frac{\omega'_C}{\pi}$
วงจกรองแถบความถี่ผ่าน	$\frac{\sin(\omega'_b n) - \sin(\omega'_a n)}{\pi n}$	$\frac{\omega'_b}{\pi} - \frac{\omega'_a}{\pi}$
วงจรแถบความถี่หยุดผ่าน	$\delta(n) - \frac{\sin(\omega'_b n) - \sin(\omega'_a n)}{\pi n}$	$1 - \left(\frac{\omega'_b}{\pi} + \frac{\omega'_a}{\pi} \right)$

จากที่ได้แสดงในตารางข้างต้น เป็นผลตอบสนองต่ออิมพัลส์ ของวงจกรองความถี่แบบต่างๆ โดยวิธีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

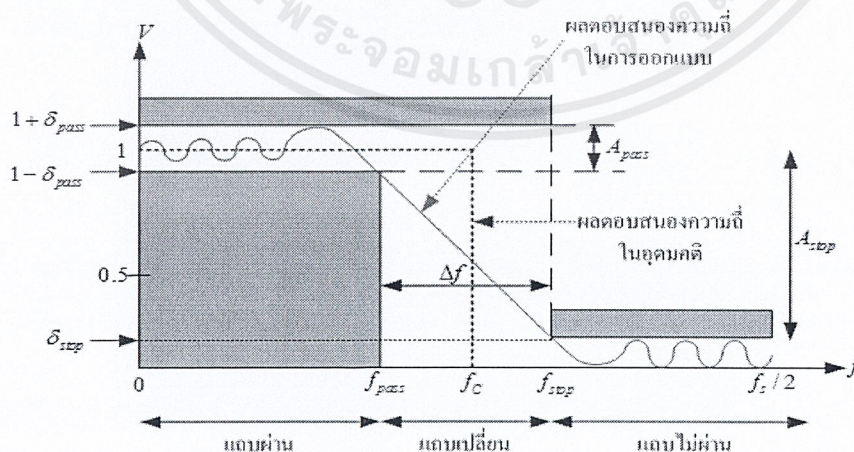
แปลง IDFT ผู้ที่ต้องการออกแบบวงจรกรองความถี่แบบดิจิทัลในที่นี่ ไม่จำเป็นต้องมีความรู้ทางคณิตศาสตร์มากนัก เพราะการออกแบบวงจรกรองความถี่แบบต่างๆ เราจะใช้ผลตอบสนองอิมพัลส์ในตารางที่ 1 ได้โดยตรง ที่ได้แสดงในสมการที่ 1 กับ 2 เป็นเพียงตัวอย่างการหาผลตอบสนองอิมพัลส์ของวงจรกรองความถี่ต่ำผ่าน โดยวิธีการแปลงจาก IDFT ที่มีค่า กับที่ ตามลำดับ ก่อนที่จะทำการออกแบบนั้น จะต้องทำความเข้าใจเพิ่มเติมเกี่ยวกับพารามิเตอร์ที่สำคัญๆตามลักษณะเฉพาะของวิธีหน้าต่าง ต่อการออกแบบวงจรกรองความถี่ เช่น ความถี่ตัด , อัตราการลดทอน , ความพริ้วหรือริบเปิลของแถบผ่าน และ ความกว้างของแถบเปลี่ยน โดยพิจารณาตามรูปที่ 4 ดังนี้ ความถี่ตัด(cut off frequency ;) หมายถึงจุดตัดของความถี่ที่ให้สัญญาณผ่านหรือไม่ผ่านไปยังหัวออกของวงจรกรอง สำหรับวงจรกรองความถี่ของ FIR กำหนดค่าความถี่ที่ขนาดลดลงประมาณ 0.5 นิยามนี้แตกต่างกับแบบ IIR ที่ใช้นิยามเดียวกันกับวงจรกรองแบบแอนะล็อกคือที่ขนาดลดลงประมาณ 0.707 การลดทอนของแถบหยุด(stop-band attenuation ;) คือจำนวนเท่าที่ แถบหยุดลดทอนลง วัดค่าเป็น โดยการลดทอนของแถบหยุดมีความสัมพันธ์กับความพริ้วของแถบหยุด แสดงดังสมการที่ 3

$$A_{stop} = 20 \log \delta_{stop} \dots (3)$$

ความพริ้วหรือริบเปิลของแถบผ่าน(pass-band ripple ; δ_{pass}) หมายถึง ค่าสูงสุดที่ขนาดแถบผ่านแกว่งออกห่างจากค่า 1 ตามรูปที่ 4 เขียนเป็นสมการได้ดังนี้ ♣

$$A_{pass} = 20 \log \frac{1 + \delta_{pass}}{1 - \delta_{pass}} \dots (4)$$

ความกว้างของแถบเปลี่ยน (transition band width ;) คืออัตราการเปลี่ยนแปลงของขนาดเทียบกับช่วงความถี่



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รูปที่ 2.7 คุณลักษณะเฉพาะของผลตอบสนองความถี่ของตัวกรองแบบ FIR

ตารางที่ 2.2 พารามิเตอร์ที่สำคัญ ของหน้าต่างแบบต่างๆ

ตารางที่ 2 พารามิเตอร์ที่สำคัญๆ ของหน้าต่างแบบต่างๆ

หน้าต่าง	δ_{pass}	$A_{stop} = 20 \log \delta_{stop}$	Δf	$W(n), n = 0, 1, \dots, N-1$ $\left(M = \frac{N-1}{2} \right)$
สี่เหลี่ยม (rectangular)	8.9%	21	$2/N$	1
ฮานนิง (Hanning)	0.63%	44	$4/N$	$0.5 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right)$
แฮมมิง (Hamming)	0.22%	53	$4/N$	$0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right)$
แบล็กแมน (Blackman)	0.02%	74	$6/N$	$0.42 - 0.5 \cos\left(\frac{2\pi n}{N-1}\right) - 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$
ไคเซอร์ (Kaiser)	ปรับได้	ปรับได้	$\frac{A - 7.95}{14.36(N-1)}$	$\frac{I_0\left(\alpha\sqrt{1-(n-M)^2/M^2}\right)}{I_0(\alpha)}$

การออกแบบวงจรกรองความถี่ (FIR-HPF)

ตัวอย่างการออกแบบ

ในที่นี้จะทำการออกแบบวงจรกรองความถี่สูงผ่านแบบ FIR โดยใช้วิธีหน้าต่างแบบสี่เหลี่ยม โดยที่ต้องการความถี่ตัดที่ 1 KHz และมีความกว้างของแถบเปลี่ยน (รูปที่ 4) มีค่าเท่ากับ 300 Hz โดยใช้ความถี่สุ่ม (sampling) เท่ากับ 4 KHz

วิธีการคำนวณและการออกแบบ

คำนวณหาค่าความถี่ที่ต้องใช้ในหน่วยของความถี่ดิจิทัล และทำการหาค่าบรรทัดฐาน(normalize) หรือค่าเพื่อใช้ในการหาค่าอันดับ (N) ที่ใช้ ซึ่งมีค่าเท่ากับ

$$\omega_c' = \frac{2\pi f_c}{f_s} = \frac{2\pi \times 1000}{4000} = \pi/2 \text{ rad.}$$

$$\Delta f' = \Delta f / f_s = 300/4000 = 0.075$$

จากตารางที่ 2 ของหน้าต่างแบบสี่เหลี่ยม = $2/N$ จะได้

แต่การออกแบบควรมีค่าถึงคุณสมบัติของความสมมาตร ซึ่งมี 4 ชนิดด้วยกัน คือ

- 1). สมมาตรปกติ N เป็นเลขคี่ ดังนั้นจะใช้ได้กับวงจรกรองความถี่ในทุกรูปแบบ
- 2). สมมาตรปกติ N เป็นเลขคู่ ดังนั้นจะใช้ได้กับวงจรกรองความถี่ต่ำ(LPF) กับวงจรกรองแถบความถี่ผ่าน(BPF)
- 3). สมมาตรตรงข้าม N เป็นเลขคี่ ดังนั้นจะใช้ได้กับวงจรกรองแถบความถี่ผ่าน(BPF)
- 4). สมมาตรตรงข้าม N เป็นเลขคู่ ดังนั้นจะใช้ได้กับวงจรกรองความถี่สูง(HPF) กับวงจรกรองแถบความถี่ผ่าน(BPF) ดังนั้น N ในที่นี้เรากำหนดให้มีค่าเท่ากับ 27 หรือใช้คุณสมบัติของความสมมาตรชนิดที่ 1 ถ้ากำหนดให้มีค่าต่ำกว่าค่าที่คำนวณได้ จะทำให้ความกว้างของแถบเปลี่ยนมีความกว้างมากขึ้น

จากตารางที่ 1 ใช้ผลตอบสนองต่ออิมพัลส์ ของวงจรกรองความถี่สูงผ่านคือ ให้เลื่อนค่า ให้ล่าหลังไป M ตำแหน่ง โดย M มีค่าเท่ากับ

$$M = (N-1)/2 = 13$$

จะได้ผลตอบสนองต่ออิมพัลส์ $d(n-M)$ คือ

$$d(n-M) = \delta(n) - \frac{\sin(\omega_c'(n-M))}{\pi(n-M)} = -\frac{\sin(\omega_c'(n-M))}{\pi(n-M)}$$

ถ้าสืบสุดท้ายเป็นการหาค่าสัมประสิทธิ์ $h(n)$ ของวงจรกรอง ซึ่งมีค่าเท่ากับ

$$h(n) = d(n-M) \bullet w(n) \dots(5)$$

จากตารางที่ 2.2 ของหน้าต่างแบบสี่เหลี่ยม ดังนั้น ค่าสัมประสิทธิ์ของวงจรกรอง หาค่าได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$h(n) = -\frac{\sin(\omega_c'(n-M))}{\pi(n-M)}, n = 0, 1, \dots, N-1$$

$$\therefore h(n) = -\frac{\sin(\omega_c'(n-13))}{\pi(n-13)}, n = 0, 1, \dots, 13$$

คำนวณหาสัมประสิทธิ์ ของวงจร โดยใช้โปรแกรม MATLAB โดยสร้างไฟล์ที่มีนามสกุลเป็น .M โดยสั่งเมนูบาร์ File | New | M-file แล้วพิมพ์ตามตัวอย่าง ดังแสดงในรูปที่ 5

```

MATLAB Editor/Debugger - [C:\MATLAB\bin\FIR_HPF.m]
File Edit View Debug Window Help
Stack:
%FIR-HPF%
fc=1000;
fs=4000;
N=27;
M=(N-1)/2;
wc=2*pi*fc/fs;
%-----%

%TABLE 1 HPF%
n=0:N-1;
d=(-sin(wc*(n-M)))/(n-M)/pi;
d((N+1)/2)=1-wc/pi;
%-----%

%TABLE 2 window method%
h=round(255*n/rmax(n))
w=1;
h=d*w;
%-----%

figure(1)
n=0:N-1;
stem(n,h)
%-----%
%plot frequency response%
figure(2)
freqz(h,1,100,fs)
xlabel('Frequency(Hertz)')
Zoom;
Ready Line 1 1:54 PM
  
```

ตัวอย่างการออกแบบ

จากตารางที่ 1

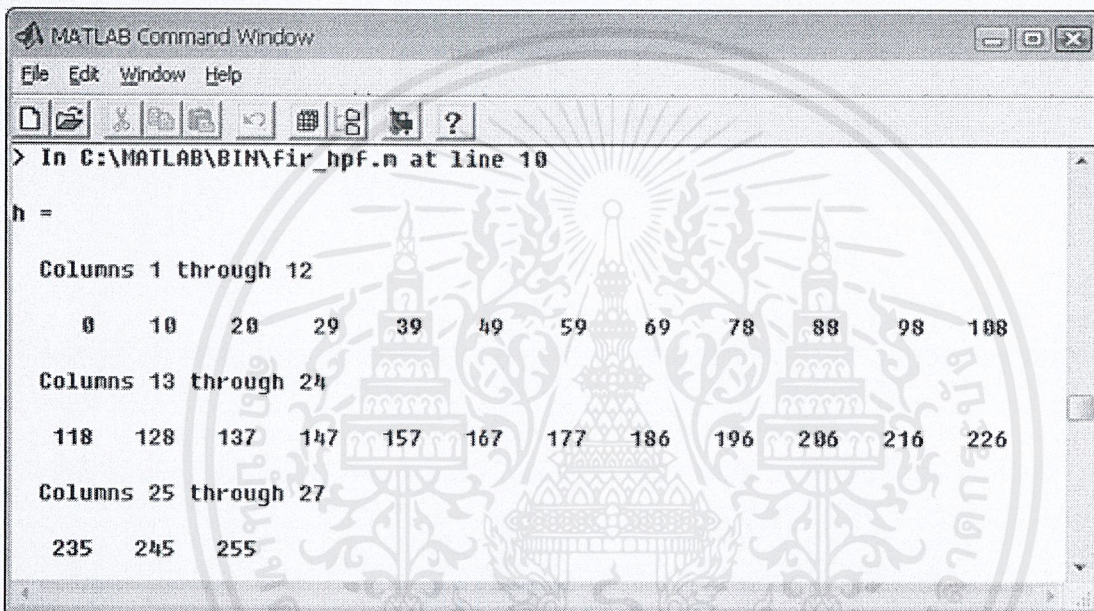
จากตารางที่ 2

การแสดงผลการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 ตัวอย่างของไฟล์ FIR_HPF.m

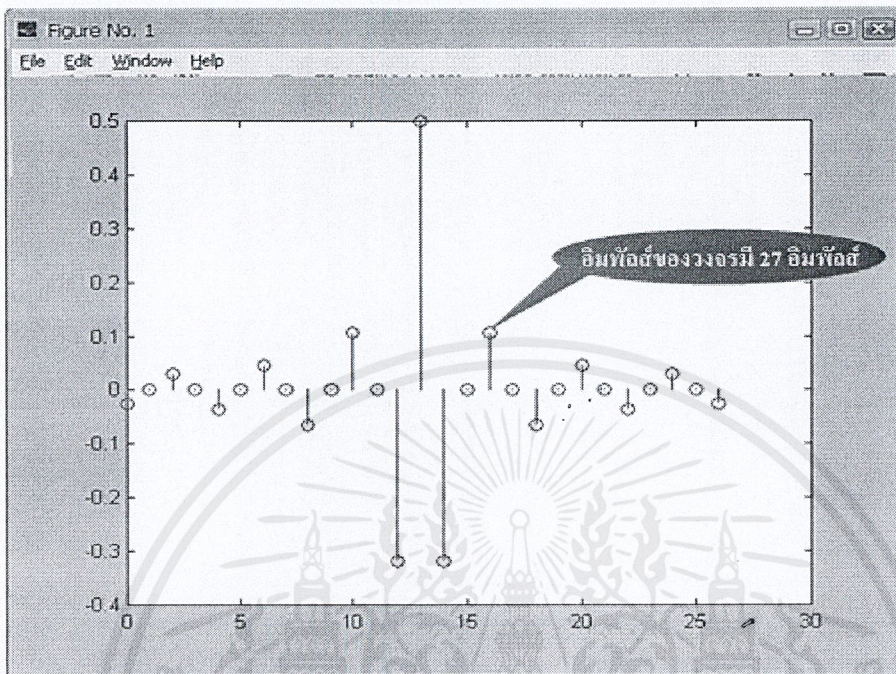
จากรูปที่ 2.8 ได้บอกถึงส่วนต่างๆ ของการเขียนโปรแกรมแบบ m-file ไว้แล้ว ซึ่งในส่วนของเครื่องหมาย หมายถึงส่วนที่ใช้อธิบายหรือแบ่งส่วนของโปรแกรม สำหรับ คือค่าความถี่ตัด , คือความถี่สุ่ม และ คือจำนวนของอันดับ ถ้าต้องการปรับคุณสมบัติของวงจรกรองความถี่สูงผ่าน สามารถปรับค่าดังกล่าวได้ตามความต้องการหรือแล้วแต่ที่เราจะออกแบบ ส่วนวงจรกรองความถี่แบบอื่นๆ ก็สามารถออกแบบได้ครับ โดยจะต้องทำการแก้ไขบรรทัดของผลตอบสนองอิมพัลส์ ตามตารางที่ 2.1 กับพารามิเตอร์ของวิธีหน้าต่าง ตามตารางที่ 2.2



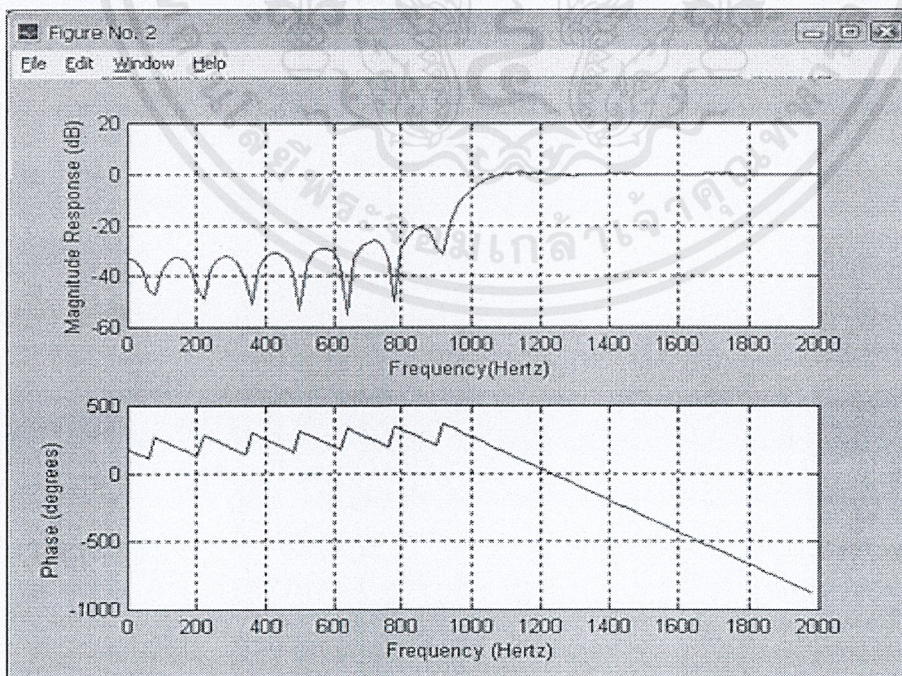
รูปที่ 2.9 สัมประสิทธิ์ของวงจรกรองความถี่ต่ำที่ออกแบบ

นอกจากการคำนวณหาค่าสัมประสิทธิ์แล้ว ไฟล์ FIR_HPF.m ยังมีคำสั่งที่ใช้สำหรับวาดผลตอบสนองอิมพัลส์ กับผลตอบสนองความถี่รวมถึงผลตอบสนองทางเฟส(phase)ของวงจรที่ได้ออกแบบ คือคำสั่ง stem กับคำสั่ง freqz ตามลำดับ สำหรับคำสั่ง stem กับ freqz มีฟังก์ชันอยู่ใน โปรแกรม MATLAB แล้ว สามารถเรียกใช้ได้โดย ในรูปที่ 6 เป็นตัวอย่างของการคำนวณหาสัมประสิทธิ์ $h(n)$ โดยพิมพ์ชื่อ FIR_HPF ลงในหน้าต่าง MATLAB Command Window โปรแกรมก็จะทำการคำนวณหาสัมประสิทธิ์ แล้ววาดกราฟแสดงผลตอบสนองอิมพัลส์ กับผลตอบสนองทางความถี่ ดังรูปที่ 7 กับรูปที่ 8 ตามลำดับ สำหรับค่าสัมประสิทธิ์ที่ได้มาแล้วนี้ จะนำไปสร้างเป็นวงจรกรองความถี่แบบดิจิทัล ในที่นี้จะยกตัวอย่างการสร้าง

วงจรกรองความถี่ที่สร้างจากชิพวงจรรวม FPGA (Field Programmable Gate Array)



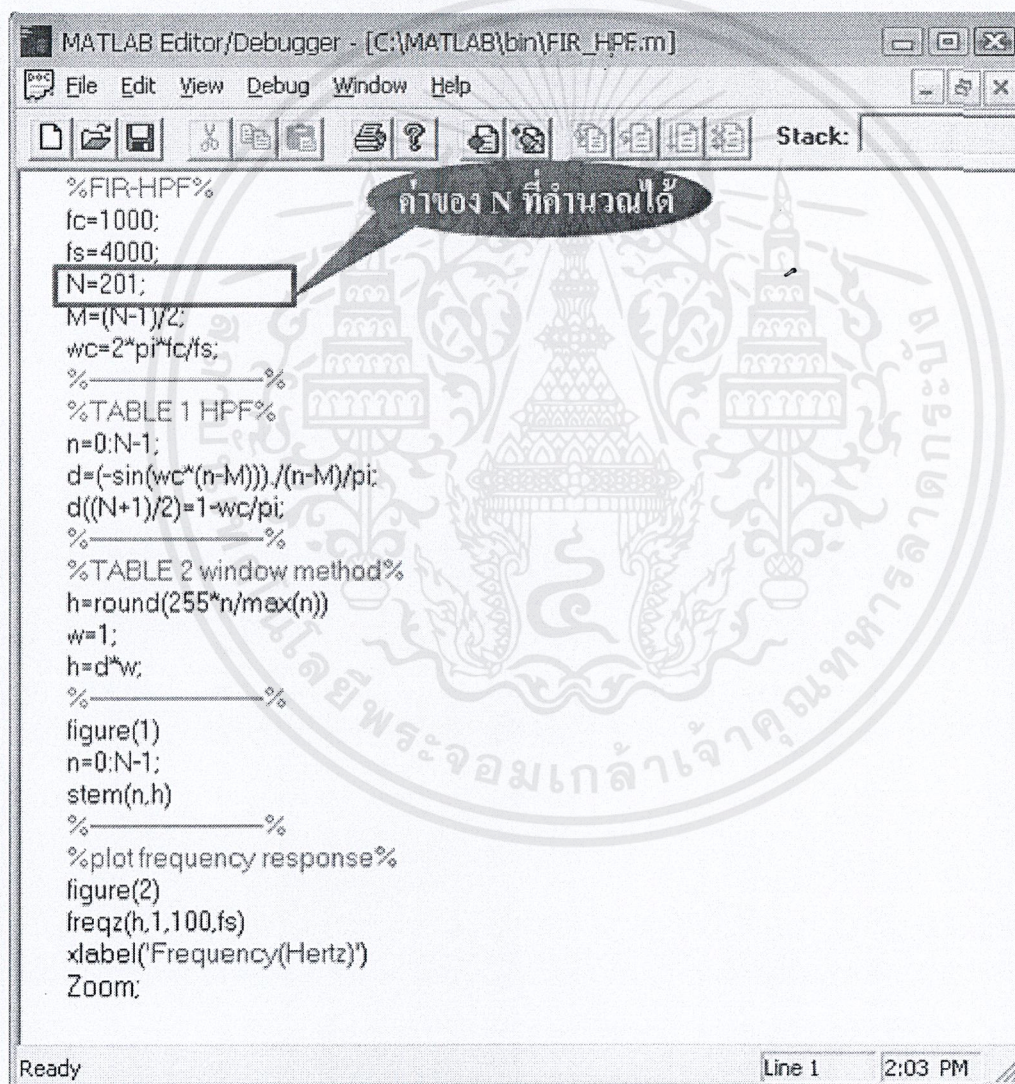
รูปที่ 2.10 ผลตอบสนองอิมพัลส์ของวงจรกรองความถี่สูงผ่าน ที่ $N=27$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.11 ผลตอบสนองทางความถี่และเฟสของวงจรกรองความถี่สูงผ่าน ที่ $N=27$

ถ้าเราต้องการให้ความกว้างของแถบเปลี่ยน (ดูรูปที่ 4) โดยให้มีค่าเท่ากับ 40 Hz โดยมีค่าความถี่ตัดและค่าความถี่ส้อมงเดิมเหมือนในตัวอย่างข้างต้น ดังนั้นเราก็สามารถคำนวณหาค่า ได้เท่ากับ 200 แต่การเลือกชนิดความสมมาตร ถ้าเป็นสมมาตรชนิดที่ 1 จะกำหนดให้ มีค่าเท่ากับ 201 (สมมาตรปกติ เป็นเลขคี่) จากนั้นก็จะใช้โปรแกรม MATLAB โดยสร้างไฟล์ที่มีนามสกุลเป็น .M โดยสั่งเมนูบาร์ File | New | M-file แล้วพิมพ์ตามตัวอย่าง ดังแสดงในรูปที่ 2.12



```

MATLAB Editor/Debugger - [C:\MATLAB\bin\FIR_HPF.m]
File Edit View Debug Window Help
Stack:
%FIR-HPF%
fc=1000;
fs=4000;
N=201;
M=(N-1)/2;
wc=2*pi*fc/fs;
%-----%
%TABLE 1 HPF%
n=0:N-1;
d=(-sin(wc*(n-M)))/(n-M)/pi;
d((N+1)/2)=1-wc/pi;
%-----%
%TABLE 2 window method%
h=round(255*n/max(n))
w=1;
h=d*w;
%-----%
figure(1)
n=0:N-1;
stem(n,h)
%-----%
%plot frequency response%
figure(2)
freqz(h,1,100,fs)
xlabel('Frequency(Hertz)')
Zoom:
Ready Line 1 2:03 PM

```

รูปที่ 2.12 การทดสอบการออกแบบวงจรกรองความถี่สูงผ่าน เมื่อ $N=201$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเมื่อทำการคำนวณจากโปรแกรม MATLAB เราก็จะเห็นถึงผลตอบสนองอิมพัลส์ของวงจรกรองความถี่สูงผ่าน เมื่อ $N=201$ และลักษณะของผลตอบสนองความถี่และทางเฟสของวงจรกรองความถี่ต่ำผ่าน

2.3 ทฤษฎี Analog to Digital Converter, Digital to Analog Converter (ADC, DAC)

2.3.1 Digital to Analog Converter

สัญญาณที่ใช้ในอุปกรณ์อิเล็กทรอนิกส์ มี 2 ชนิด คือ สัญญาณอนาล็อก และสัญญาณดิจิทัล สัญญาณอนาล็อก จะใช้ใน อุปกรณ์ต่างๆ ไป และใช้ในการควบคุมแบบเก่า

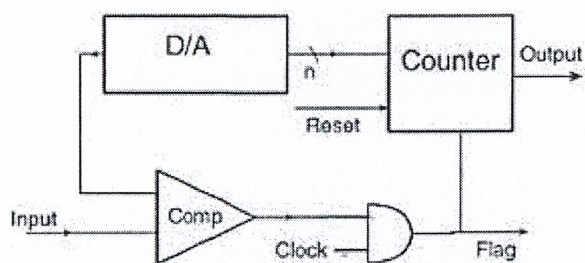
ในปัจจุบันมีไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์ เข้ามาช่วยในการควบคุมอุปกรณ์ต่างๆ มากมาย ซึ่งทำให้การควบคุมนั้นทำได้ง่าย และรวดเร็วยิ่งขึ้น แต่ในการควบคุมนั้น เราจำเป็นต้องใช้สัญญาณดิจิทัลในการติดต่อกับไมโครโปรเซสเซอร์ หรือไมโครคอนโทรลเลอร์ แต่ในความเป็นจริงนั้น เราใช้สัญญาณอนาล็อกในการควบคุม ดังนั้นเราจึงจำเป็นต้องมีการเปลี่ยนสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล แล้วจึงนำสัญญาณนั้นเข้ามาสู่ไมโครโปรเซสเซอร์ หรือไมโครคอนโทรลเลอร์ เพื่อใช้ควบคุมระบบต่อไป

แม้ว่าสัญญาณอนาล็อกนั้นมีความแน่นอน และแม่นยำสูง แต่สัญญาณอนาล็อกนั้นก็ควบคุมได้ยาก เนื่องจากในสภาพแวดล้อม มีสัญญาณรบกวนอยู่มาก และการที่จะทำให้ การควบคุมแบบอนาล็อก มีความสามารถควบคุม เท่ากับการควบคุมแบบดิจิทัลนั้น ทำได้ยาก เนื่องจากวงจรควบคุมแบบ อนาล็อก จะต้องมีความซับซ้อนสูง

อย่างไรก็ตาม สัญญาณดิจิทัลก็ไม่สามารถทดแทนความละเอียดของสัญญาณอนาล็อกได้อย่างสมบูรณ์ แต่ทำให้การควบคุมนั้นทำได้ง่าย และสะดวกยิ่งขึ้น

Counting Converter

Counting Converter เป็นวิธีที่ง่ายที่สุดของการแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล โดยใช้ อัลกอริทึม การนับค่าเพิ่มขึ้นเรื่อยๆ แล้วนำผลที่ได้จากการนับ ไปเปรียบเทียบกับค่าที่ต้องการที่ตั้งไว้ ลักษณะการทำงานเป็นดังรูป



Analog To digital Converter

รูปที่ 2.13 Counting Converter

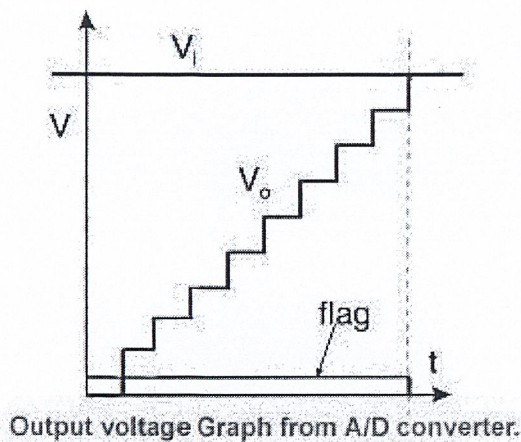
จากวงจร Counter เป็นอุปกรณ์นับค่าที่เพิ่มขึ้นทีละหนึ่ง แล้วส่งค่าที่ได้ให้ D/A มีขา Reset รับสัญญาณ Reset เมื่อต้องการให้เริ่มนับใหม่

D/A เมื่อรับค่าที่นับเพิ่มขึ้นทีละหนึ่งจากตัวนับ ก็แปลงค่าให้เป็นสัญญาณ อนุภาคที่มีค่าความต่างศักย์ค่าๆ หนึ่ง แล้วส่งต่อเข้าไปที่อุปกรณ์ตัวเปรียบเทียบ(Comparator)

Comparator จะเป็นอุปกรณ์ตัวเปรียบเทียบค่าความต่างศักย์ ของอินพุต และค่าจากที่ตัวนับ ถ้าหากทั้งสองสัญญาณมีค่าเท่ากันส่งค่าความต่างศักย์ 0 โวลต์ออกมา(ลอจิก 0) ถ้าไม่เท่ากันก็จะส่งความต่างศักย์ที่ไม่ใช่ 0 โวลต์ออกมา(ลอจิก 1)

ซึ่งค่าความต่างศักย์ที่ออกมา จะนำมาเข้าลอจิกเกต "และ" กับ สัญญาณนาฬิกา จะได้ค่าลอจิกออกมา ถ้าผลลัพธ์ออกมาเป็นสัญญาณนาฬิกาแสดงว่ายังไม่ได้ผลลัพธ์เท่าที่ต้องการ สัญญาณนาฬิกาจะไปทำให้ตัวนับนับเพิ่มขึ้นต่อไป และเมื่อได้ค่าผลลัพธ์ดิจิทัลที่ต้องการแล้ว ค่าที่ได้จาก ตัวเปรียบเทียบจะให้ความต่างศักย์เป็น 0 (ลอจิก 0) ซึ่งเมื่อนำมาเข้าลอจิกเกต "และ" กับสัญญาณนาฬิกาแล้ว ก็จะให้ลอจิก 0 ซึ่งทำให้ตัวนับไม่นับเพิ่มอีก ก็จะได้ค่าดิจิทัลจากตัวนับที่ต้องการ

จากคำอธิบายข้างต้นจะได้กราฟของ V_O ดังนี้



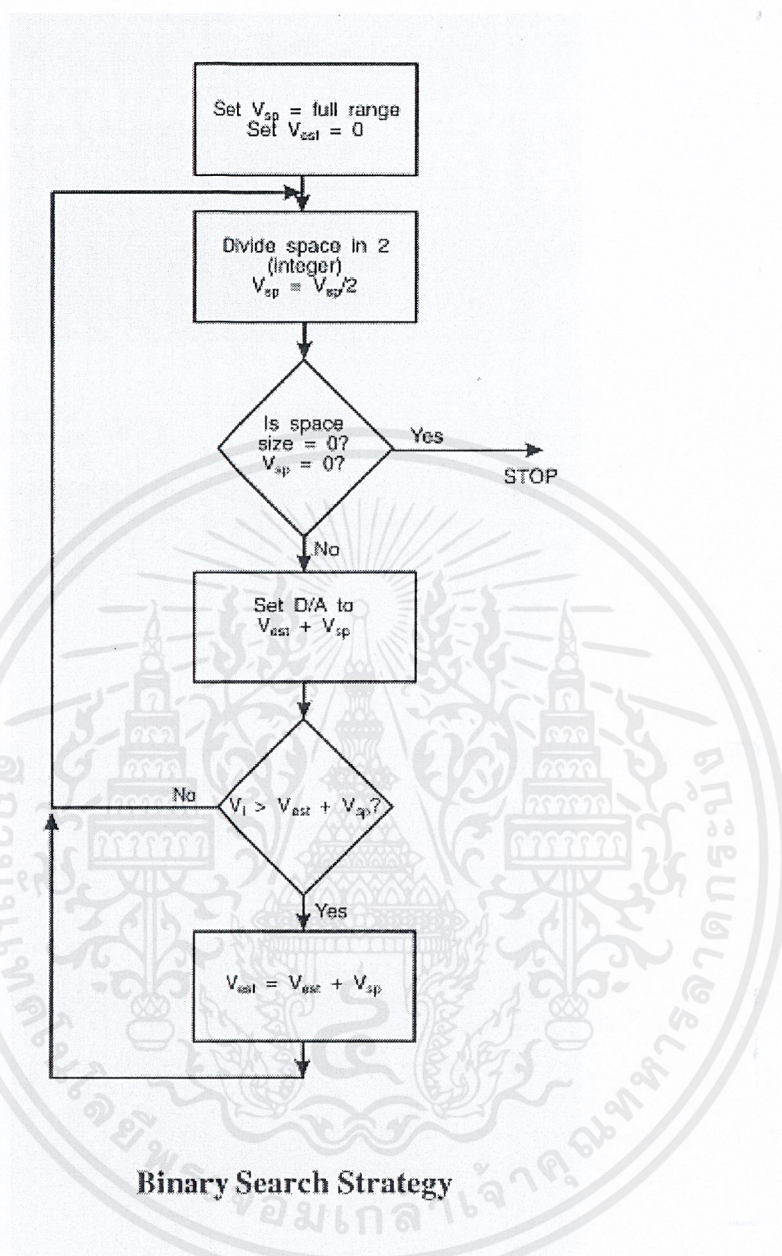
รูปที่ 2.14 รูปแสดง Output Voltage Graph

ข้อเสียของวิธีนี้ คือ การนับต้องเริ่มนับที่ 0 เสมอ และนับเพิ่มขึ้นเรื่อยๆ ทำให้ช้า เอาที่พุดที่ได้จะมี delay จึงไม่ค่อยนิยมใช้เท่าที่ควร จึงได้เปลี่ยนตัวนับเป็นแบบนับลงได้ด้วย ซึ่งจะอ้างอิงระดับจากระดับเก่า ทำให้ไม่จำเป็นต้องนับ 0 ใหม่ เมื่อมีการเปลี่ยนอินพุตใหม่ แต่ให้อ้างอิงกับผลลัพธ์เดิม ทำให้ได้ผลลัพธ์เร็วขึ้น

Successive Approximation

ใช้หลักการของ "binary search" ในการหาคำตอบ โดยนำค่าผลลัพธ์มาเปรียบเทียบกับค่ากึ่งกลางของช่วง เพื่อให้ทราบว่า คำนั้นๆ มากกว่า หรือน้อยกว่า โดยจะปรับช่วงให้แคบลงมาเรื่อยๆ แล้วเปรียบเทียบผลลัพธ์กับค่ากึ่งกลางของช่วงไปเรื่อยๆ จนได้ผลลัพธ์ที่ต้องการ เช่น เลขที่เป็นคำตอบคือ 3 จากช่วงของคำตอบที่ 0-7 ครั้งแรกเอาค่า $(0+7)/2 = 4$ มาเปรียบเทียบ ได้ผลว่า คำตอบที่ต้องการอยู่ในช่วงที่น้อยกว่า 4 ครั้งที่ 2 ก็เลือกค่า $(0+4)/2 = 2$ มาเปรียบเทียบ ได้ผลว่าคำตอบที่ต้องการอยู่ในช่วงที่มากกว่า 2 แต่น้อยกว่า 4 ครั้งที่ 3 ก็เลือกค่า $(2+4)/2 = 3$ มาเปรียบเทียบ ได้ผลว่าคำตอบที่ต้องการ

จากหลักการที่กล่าวมาอาจเขียน flow chart ได้ดังนี้

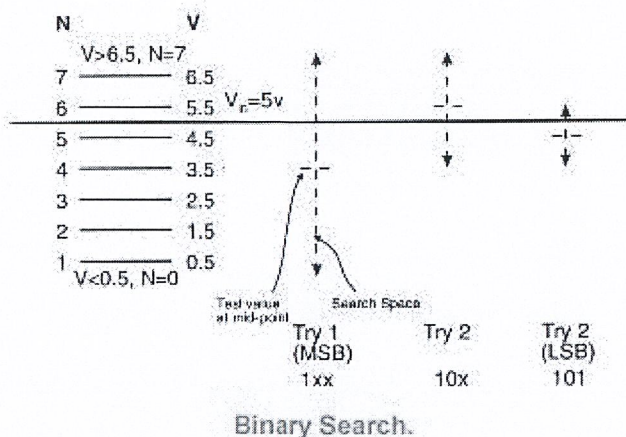


รูปที่ 2.15 รูปแสดง flow chart

ข้อดีของวิธีนี้ คือ เวลาที่ใช้ในการหาคำตอบ n รอบแน่นอน (สำหรับ n bit converter ซึ่งอ้างอิงได้ 2^n ระดับ และระดับ V_{in} ที่คงที่) ซึ่งใช้เวลาน้อยกว่าแบบ "Counting Algorithm"

แต่มีข้อเสีย คือ ถ้า V_{in} เปลี่ยนทันทีทันใด ขณะที่กำลังทำ binary search อยู่ นั้น คำตอบที่ได้จะผิดพลาด ตัวอย่างเช่น เปลี่ยน V_{in} จาก 5 Volt เป็น 2 Volt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 รูปแสดง Binary search

ช่วงของ V_{in} คือ 1-7 ใช้ $n=3$ (เพราะว่า $2^3=8$)

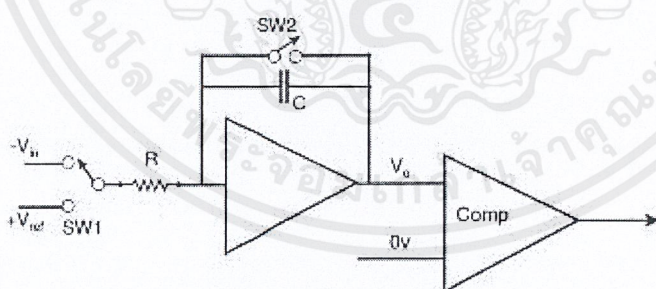
ครั้งแรก ใช้ 4 เปรียบเทียบกับ V_{in} (ซึ่งเท่ากับ 5 โวลต์) พบว่า อยู่ในช่วง lower ได้ 1xx

ครั้งที่ 2 ใช้ 2 เปรียบเทียบกับ V_{in} (ซึ่งเท่ากับ 5 โวลต์) พบว่า อยู่ในช่วง upper ได้ 10x

ครั้งที่ 3 ใช้ 3 เปรียบเทียบกับ V_{in} (ซึ่งเท่ากับ 5 โวลต์) พบว่า ผลลัพธ์ที่ได้จะผิดพลาด ได้ 100

Dual-Slope ADC

ใช้หลักการของวงจร Integrater ทำงานร่วมกับตัว Comparater ดังรูป



Dual Slope A/D converter .

รูปที่ 2.17 รูปแสดง Dual Slope Converter

Input Voltage มี 2 ตัว คือ ค่าความต่างศักย์อนาล็อกที่ต้องการแปลงเป็นดิจิทัล ($-V_{in}$) และความต่างศักย์ที่คงที่ค่าหนึ่ง (V_{ref}) และมีสวิตช์ SW1 ซึ่งทำหน้าที่เลือกค่าสัญญาณ จากวงจรตอนเริ่มต้นสวิตช์ SW2 ทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่คายประจุของตัวเก็บประจุ C แล้วจึงเปิด SW2 ออก เมื่อสวิตช์ SW1 สับมาที่ $-V_{in}$ จากวงจร Intergrater จะพิสูจน์สมการได้ดังนี้

$$I = C \frac{dV_0}{dt}$$

$$-V_{in} + iR - V_0 + V_0 = 0$$

$$V_{in} = RC \frac{dV_0}{dt}$$

$$-V_{in} + RC \frac{dV_0}{dt} = 0$$

$$\int dV_0 = \int \frac{V_{in}}{RC} dt$$

$$V_0 = \frac{V_{in}(t)}{RC}$$

$$\frac{V_{in}}{RC}$$

slope มีค่าเท่ากับ

ค่า t ที่ใช้มีค่าคงที่ t_m

เมื่อ t เพิ่มจากศูนย์ถึง t_m

ให้ SW1 สับไปที่ V_{ref}

$$V_0 = \frac{V_{ref}(t)}{RC}$$

จะได้สมการ

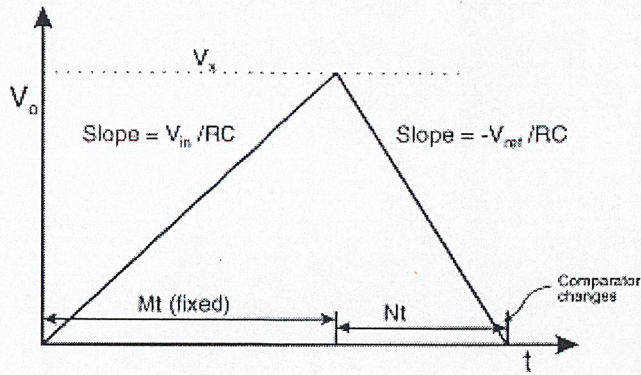
$$\frac{V_{ref}}{RC}$$

slope มีค่า

สมมติ ช่วงเวลาตั้งแต่ความต่างศักย์ที่ t_m จนความต่างศักย์เป็น 0 มีค่าเท่ากับ t_n

ได้ดังแสดงในกราฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Dual Slope A/D Converter Output and Timing

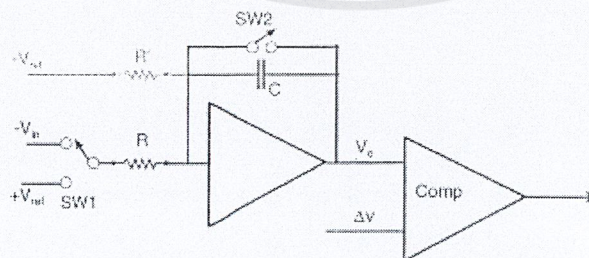
รูปที่ 2.18 รูปแสดง Output และ Timing ของ Dual Slope Converter

จากหลักของสามเหลี่ยมคล้าย จะได้สมการ
$$V_{in} = V_{ref} \frac{t_n}{t_m}$$

เนื่องจาก V_{ref} และ t_n มีค่าคงที่ สัญญาณอนาล็อกขึ้นกับค่า t_n เพราะการควบคุมการเปลี่ยนสัญญาณดิจิทัลที่ขึ้นกับค่า t_n การแปลงเป็นสัญญาณดิจิทัลจะทำโดยจับคู่ค่า t_n กับเอาต์พุตค่าๆ หนึ่ง ตามความเหมาะสมสำหรับ V_{ref} นั้นๆ เหมือนการเทียบค่าในตาราง

ความเร็วของการแปลงสัญญาณแบบนี้ ขึ้นอยู่กับ V_{in} และ Slope ของวงจรถ่าย integrater

โดยธรรมชาติแล้ว ลักษณะของตัวเปรียบเทียบเองนั้น จะไม่เป็นอุดมคติ คือจะมีผลต่างของความต่างศักย์อยู่ V โวลต์ แม้ว่า จะต่ออินพุตทั้งสองลงกราวด์แล้วก็ตาม ซึ่งถ้า V_{ref} ที่ใช้อยู่นั้นมีค่าน้อยกว่าค่าผลต่างของความต่างศักย์ที่เกิดจากตัวเปรียบเทียบ ความชันก็จะน้อย ทำให้เวลา t_m ใช้เวลานานมาก กว่าที่จะฟันค่าความต่างศักย์ที่เกิดจากตัวเปรียบเทียบ เราจึงต้องนำค่าความต่างศักย์มาเพิ่มให้กับ V_{ref} เพื่อหาผลลัพธ์ ดังรูป



Dual Slope A/D Converter - Full Circuit

รูปที่ 2.19 รูปแสดง Dual Slope A/D Converter Full Circuit

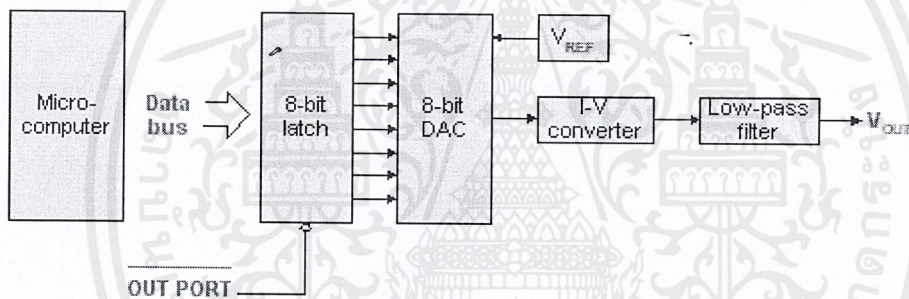
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าความต่างศักย์จะเพิ่มขึ้นเรื่อยๆ จากค่าความต้านทานที่ต่อเพิ่มขึ้น ความต่างศักย์ที่ได้นั้น เมื่อนำไปเปรียบเทียบกับ V_{in} แล้วมากกว่าก็จะปล่อยลอจิกออกมา ถ้ามากกว่าก็จะให้ลอจิก 1 ถ้าน้อยกว่าหรือเท่ากันก็จะให้ลอจิก 0 วิธี Flash Converter นี้จะเร็วที่สุด แต่ใช้อุปกรณ์ทาง Hardware มากกว่าแบบอื่นๆ อุปกรณ์ทางไฟฟ้า/อิเล็กทรอนิกส์ โดยทั่วไปที่เป็นอะนาล็อก สามารถควบคุมการทำงาน โดยการให้อินพุตเป็นระดับแรงดันที่แตกต่างกัน ตัวอย่างเช่น มอเตอร์กระแส ตรง ซึ่งควบคุมความเร็วโดยเปลี่ยนระดับแรงดัน (หรือกระแส) ของขดลวดสนาม

2.3.2 Digital to Analog Converter

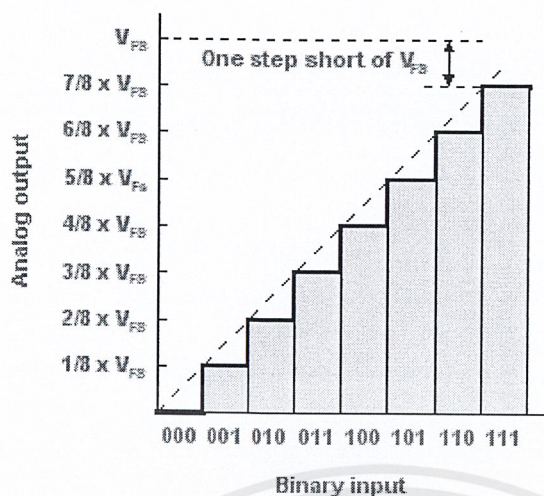
เมื่อนำระบบดิจิทัล หรือไมโครคอนโทรลเลอร์ มาใช้ควบคุม อุปกรณ์ทางอะนาล็อกเหล่านี้ จึงต้องมีวงจรที่สามารถแปลงสัญญาณทางดิจิทัลเป็นระดับแรงดันต่อเนื่อง แบบอะนาล็อก ตั้งแต่ศูนย์โวลต์จนถึงระดับสูงสุดที่กำหนดไว้ เรียกว่าวงจร Digital to Analog Converter (DAC)

คุณสมบัติของ Digital to Analog Converter



รูปที่ 2.22 ระบบการแปลงสัญญาณดิจิทัลเป็นอะนาล็อก

รูปที่ 1 แสดงถึงส่วนประกอบหลักของระบบ DAC โดยทั่วไป ไมโครคอมพิวเตอร์จะมีเอาต์พุตเป็นค่าไบนารี วงจรเลขรับค่าไบนารีเข้ามาเพื่อส่งไปยัง DAC ในวงจรจะใช้แหล่งกำเนิดแรงดันหรือ กระแสคงที่ เพื่ออ้างอิงในการแปลงข้อมูล ไบนารีเป็นระดับกระแส ต่อมาจะมีวงจรแปลง จากกระแสเป็นระดับแรงดัน (current-to-voltage converter) ซึ่ง ปกติจะใช้อุปกรณ์ที่ง่ายที่สุด สัญญาณอะนาล็อกที่ได้จะผ่าน วงจร low-pass filter เพื่อกำจัดสัญญาณความถี่สูงที่ แผลงอยู่ในสัญญาณที่ถูกสร้างขึ้นมา



รูปที่ 2.23 Transfer Curve ในอุดมคติของ DAC 3 บิต

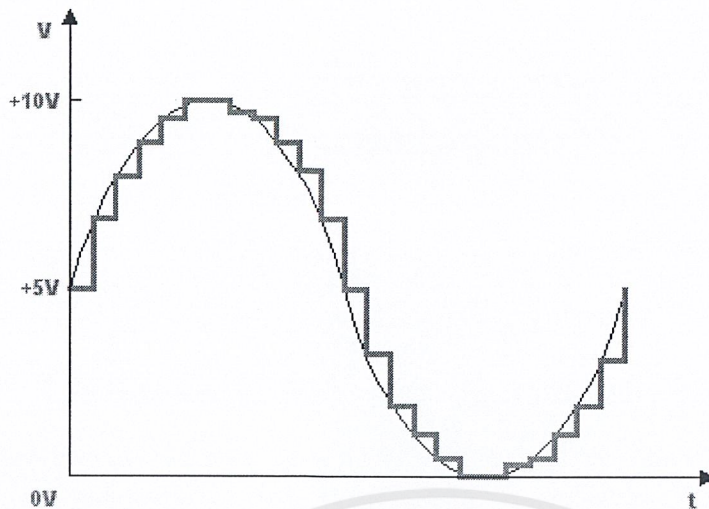
รูปที่ 2 เป็นกราฟแสดงถึงความสัมพันธ์ระหว่างเอาต์พุตที่เป็นอนาล็อกกับอินพุตที่เป็นดิจิทัลขนาด 3 บิต เรียกว่า transfer curve สังเกตว่าเมื่ออินพุต ไบนารีเพิ่มขึ้น เอาต์พุตอนาล็อกจะเพิ่ม ในลักษณะขั้นบันได ขนาดของแต่ละขั้นจะ หาได้จาก

$$\text{stepsize} = V_{FS}/2^n$$

เมื่อให้ V_{FS} คือระดับแรงดันเอาต์พุตสูงสุด

n คือจำนวนบิตของอินพุต

เนื่องจากเอาต์พุตของ DAC จะเพิ่มเป็นขั้นๆ รูปคลื่นสัญญาณ ที่ได้จาก DAC จึงมีลักษณะไม่เรียบ ดัง ตัวอย่าง ในรูปที่ 3 ซึ่งแสดงถึงสัญญาณไซน์ ที่สร้างจาก DAC



รูปที่ 2.24 คลื่นไซน์ที่สร้างจาก DAC

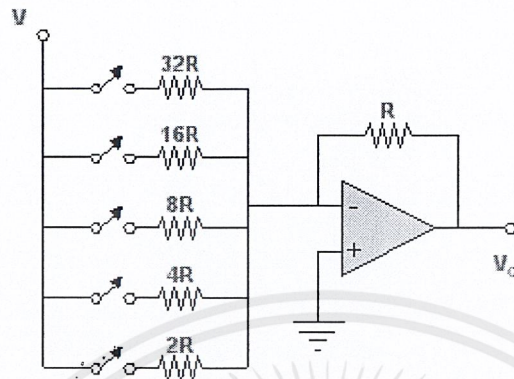
ถ้าเพิ่มจำนวนบิต ความละเอียดของ DAC จะเพิ่มขึ้น เช่น เมื่อ ใช้ DAC 12 บิต และ $V_{FS} = 5.0 \text{ V}$ ความละเอียดคือ $5.0 \text{ V} / 4096 = 1.22 \text{ mV}$ ซึ่งจะ ละเอียดกว่า DAC 8 บิตถึง 16 เท่า

ความถูกต้องของ DAC ขึ้นอยู่กับหลายส่วน

1. Quantization error DAC บิต $V_{FS} = 5.0 \text{ V}$ เอาต์พุตจะมีความละเอียด 19.53 mV ถ้าต้องการเอาต์พุต 4.00 V DAC จะให้เอาต์พุตได้ใกล้เคียง ที่สุดคือ 4.04 V ($19.53 \text{ mV} \times 205$) ผิดพลาด 4 mV โดยทั่วไปค่าผิดพลาดจะเท่ากับ $\pm 0.5 \text{ LSB}$ (least significant bit) ตัวอย่างเช่น DAC 8 บิต ความผิดพลาดจะเป็น 1 ใน 512 หรือ $\pm 0.195 \%$
2. Offset and gain errors เมื่ออินพุตไบนารีเท่ากับ 0 แต่เอาต์พุตของ DAC ไม่เป็น 0 เรียกว่า offset error และอาจเกิดร่วมกับ gain error ความผิดพลาดเหล่านี้ จะทำให้ transfer curve ในรูปที่ 2 โค้งขึ้น หรือลง ขึ้นอยู่กับความไม่สมมูลภายใน DAC อย่างไรก็ดี ตาม offset error และ gain error จะแก้ไขได้โดยใช้ความต้านทานปรับค่าได้ต่อไว้ภายนอก
3. Nonlinearity คือค่าความคลาดเคลื่อนสูงสุดของ transfer curve เทียบกับเส้นตรงจากจุดศูนย์และจุดสูงสุด ซึ่งจะขึ้นอยู่กับความผิดพลาดของส่วนประกอบภายใน DAC ใน data sheet ของ DAC จะระบุเป็นเปอร์เซ็นต์เทียบกับค่าสูงสุด หรือ ระบุเป็นเศษส่วนของ LSB (โดยทั่วไปคือ $\pm 0.5 \text{ LSB}$)
4. Settling time คือช่วงเวลานับแต่ให้อินพุตจนกระทั่ง DAC ให้ เอาต์พุต วัดเมื่อเอาต์พุตที่ได้ผิดพลาดจากค่าจริง น้อยกว่า 0.5 LSB ค่าเวลานี้อาจน้อยกว่า 100 ns สำหรับ DAC ความเร็วสูง และอาจมากกว่า 100 us สำหรับ DAC ราคาถูก

วงจร Digital to Analog Converter

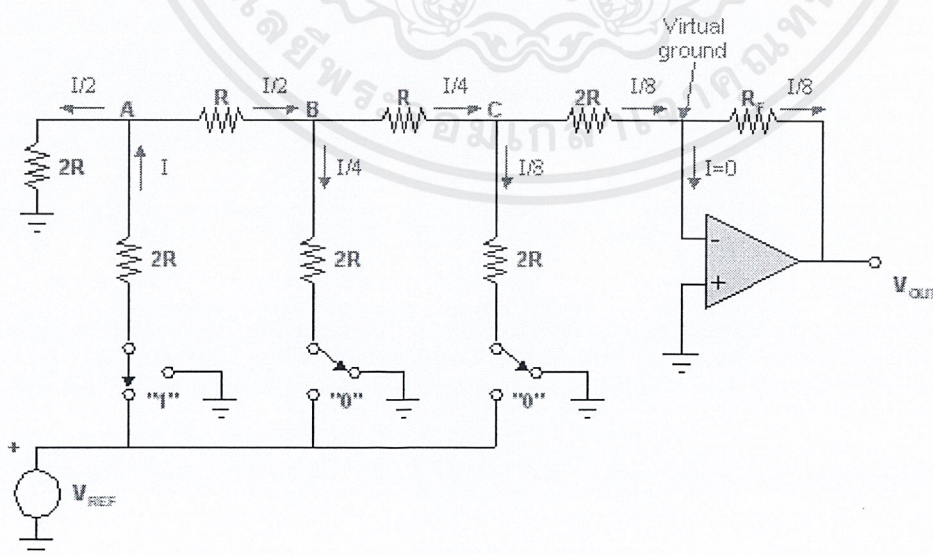
Summed Source DAC



รูปที่ 2.25 Summed Source DAC

เป็นวงจรอย่างง่ายในการแปลงสัญญาณดิจิทัลเป็นอะนาล็อก จาก รูปที่ 4 จะเห็นว่าเป็นวงจร Summing Amp มีความต้านทานค่า $2R$, $4R$ และ $8R$ เพื่อให้กระแสที่ผ่านความต้านทานแต่ละตัวมีค่า ลดลงเป็น 2 เท่า ความต้านทานตัวล่างสุด ($2R$) จะ เป็น MSB ส่วนตัวบนสุดจะเป็น LSB ข้อเสียของการใช้วงจรลักษณะ นี้ ในทางปฏิบัติ ค่าความต้านทานที่ต่างกันเป็น 2 เท่า คือ $2R$, $4R$, $8R$, ... จะ ไม่สามารถหาได้ง่าย จึงมีการ ปรับปรุงเป็น วงจร R-2R

Switched Voltage R-2R DAC

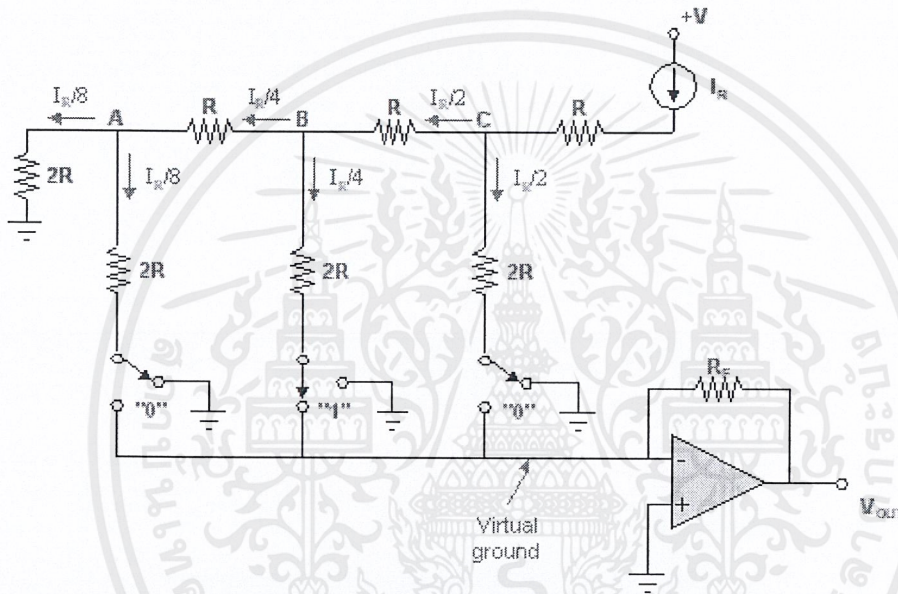


รูปที่ 2.26 Switched Voltage R-2R DAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.26 เป็น DAC 3 บิต ใช้โอปแอมป์และความต้านทาน เพียง 2 ค่าคือ R และ 2R สังเกตว่าอินพุตดิจิทัล จะมาจากสวิตช์ทั้ง 3 ซึ่ง อาจต่อกับกราวด์ (ลอจิก 0) หรือต่อกับ V_{REF} (ลอจิก 1) ตัวอย่าง นี้อินพุตเป็น 001 พิจารณากระแส I เมื่อผ่านจุด A จะถูกแบ่งเป็นสองส่วน เท่าๆ กัน เหลือ $I/2$ เมื่อผ่านจุด B และ C จะถูกแบ่ง อีกครั้ง เหลือ $I/4$ และ $I/8$ ตามลำดับ ดังนั้นกระแสที่ป้อนให้กับออปแอมป์จะ เหลือ $I/8$ เมื่อพิจารณาที่สวิตช์ ตัวอื่นๆ ก็จะมีลักษณะคล้ายกัน ดังนั้นกระแสที่ผ่านออปแอมป์เมื่อปิดสวิตช์อื่นนับจากซ้าย มาขวา จะมี ขนาด $I/8$, $I/4$ และ $I/2$ ตามลำดับ สวิตช์ซ้ายสุด จะเป็น LSB ส่วนขวาสุดจะเป็น MSB

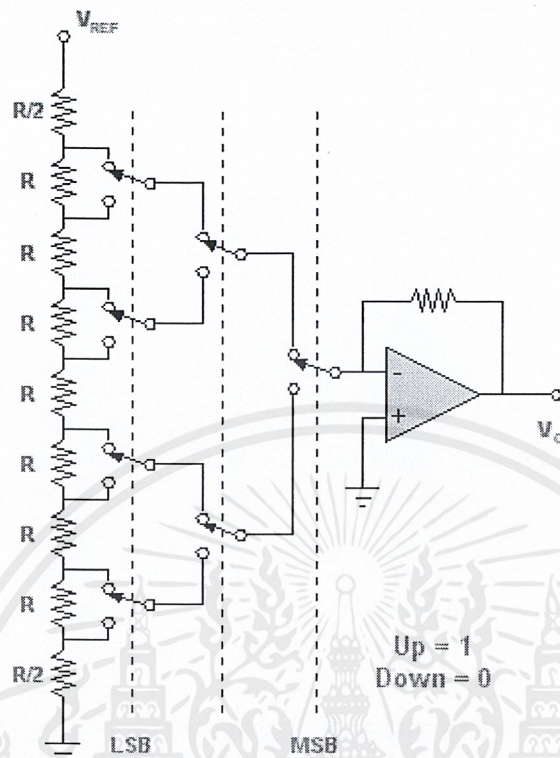
Switched Current R-2R DAC



รูปที่ 2.27 Switched Current R-2R DAC

วงจรนี้เปลี่ยนจากการใช้แรงดันอ้างอิง (V_{REF}) มาเป็นกระแสอ้างอิง (I_R) กระแสที่ ผ่านสวิตช์แต่ละตัวจาก ขวามาซ้ายจะเป็น $I_R/2$, $I_R/4$ และ $I_R/8$ ตามลำดับ วงจร ลักษณะนี้จะมีความเร็วสูงกว่าวงจร Switched Voltage เนื่องจาก คาปาซิแตนซ์ที่รอยต่อ (junction capacitance) ของความต้าน ทานแต่ละตัวจะไม่ถูกชาร์จและ ดิสชาร์จเหมือนวง จร Switched Voltage

Switched Pole DAC



รูปที่ 2.28 Switched Pole DAC

จะมีการใช้ความต้านทานต่ออนุกรมกัน หลายตัว เนื่องจากวงจรนี้ต้องการความต้านทานค่าเท่าๆ กัน ดังนั้นจึงเป็นที่นิยมสำหรับ ผู้ผลิต Integrated Circuit สังเกตว่า จะมีความต้านทานที่ ปลายทั้งสองของอนุกรมเพื่อปรับ offset ของเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ทฤษฎี FPGA และ VHDL

2.4.1 FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ ออกแบบ ลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามที่ผู้ออกแบบต้องการ ในการทำ FPGA เมื่อ เทียบกับการทำASICs (Application Specific Integrated Circuits) แล้วนั้นก็มีทั้งข้อดีและข้อเสีย คือ การทำ FPGA จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในอุปกรณ์ FPGA จะมีจำนวน gate ให้ใช้จำนวนจำกัด และการทำ FPGA ก็เหมาะกับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิตในปริมาณต่ำ ส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียน code อธิบายฮาร์ดแวร์จนกระทั่ง download นั้นน้อยกว่าการทำ ASIC มาก และการตรวจสอบหรือแก้ไขdesign ก็ทำได้สะดวก การทำ FPGA ในปัจจุบันมีประสิทธิภาพ มากขึ้น และสะดวกขึ้น ทั้งนี้ก็เนื่องจากทางบริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบภายในหรือปรับปรุง โครงสร้างสถาปัตยกรรมภายในและยังได้เพิ่ม ประสิทธิภาพของซอฟต์แวร์ ที่ใช้ทำ PPR (Partitioning, Placement and Routing) สำหรับอุปกรณ์นั้นๆด้วย ASIC ความก้าวหน้าของอุตสาหกรรมอิเล็กทรอนิกส์ปัจจุบันทำให้เกิดการพัฒนาความสามารถของอุปกรณ์ ต่างๆ มากมายซึ่งทำให้เกิดการลดค่าใช้จ่าย การสิ้นเปลืองพลังงานและขนาด ในขณะเดียวกันก็มีการเพิ่ม ประสิทธิภาพและ ระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจาก เทคโนโลยีไมโคร โพรเซสเซอร์ และหน่วยความจำปัจจุบัน ทุกๆ ครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างวงจรรวมและไอซีมาตรฐานมากขึ้น ในการพัฒนาเพิ่มความหนาแน่น และจำนวน ฟังก์ชันลอจิก ที่เหมาะสม นักออกแบบอุปกรณ์ทางด้าน ดิจิตอลได้พิจารณาถึงการผลิตให้ขนาดมากๆ และการผลิตวงจรรวม (ASIC: Application Specific Integrated Circuit) ซึ่งวงจรรวม จะแบ่งตามการสร้างออกเป็น 2 กลุ่ม คือ Field programmable และ Mask programmable Programmable Logic สามารถแบ่งได้เป็นหลายชนิด โดยแบ่งตามเทคโนโลยีที่ใช้ผลิต ความ ซับซ้อนของอุปกรณ์ หรือ โครงสร้างภายในอุปกรณ์เอง ซึ่งสามารถแบ่งได้ดังนี้

- Simple Programmable Logic Device (SPLD)
- Complex Programmable Logic Device (CPLD)
- Field Programmable Gate Array (FPGA)
- Field Programmable Interconnects (FPIC)

ถ้าเราพิจารณาถึงข้อดี ในเชิงการออกแบบและการผลิต ไอซีตระกูล mask programmable นั้น อาจจะทำให้ ได้ ASIC ที่มีประสิทธิภาพสูงกว่า แต่ต้องมีการส่งวงจรไปทำหน้ากาก (mask set) และเอกสารยัง ต่างประเทศ ทำให้ มีค่าใช้จ่ายเริ่มต้น (fixed cost) ที่สูงมาก ซึ่งเหมาะแก่การใช้งานที่ต้องการปริมาณ (volume) สูงเพื่อเฉลี่ยค่า fixed

cost เท่านั้น สำหรับการพัฒนา ASIC ใน volume น้อยถึงปานกลางนั้น ควรใช้ ASIC ชนิดโปรแกรมได้ (field programmable) ชนิดใดชนิดหนึ่งซึ่งสามารถหาซื้อได้ในราคาข้อมเยากว่า โดยผู้ใช้งานสามารถ ออกแบบ และสร้างวงจรที่ต้องการใช้ลงในตัวอุปกรณ์ได้เอง โดยไม่ต้องไปโรงงานเพื่อผลิต จึงเป็น ทางเลือกที่น่าสนใจ โดยเฉพาะอย่างยิ่ง เทคโนโลยีที่ก้าวหน้าไปมาก ทำให้ประสิทธิภาพของไอซีแบบ โปรแกรมได้ สามารถตอบสนองความต้องการของการใช้งานที่มีความซับซ้อนสูงได้ -Simple Programmable Logic Device (SPLD) SPLD เป็นอุปกรณ์ไอซีที่โปรแกรมได้ที่มีโครงสร้างแบบง่ายมีขนาดเล็กโดยมีจำนวนมาโครเซลล์ไม่เกิน 30 มาโครเซลล์ CPLD (Complex Programmable Logic Device) หรือมีชื่อเรียกอย่างอื่นเช่น

- EPLD (Erasable Programmable Logic Device) ผู้ผลิตที่สำคัญเช่น Xilinx
- EEPLD (Electrically Erasable programmable logic Device) ตัวอย่างไอซีได้ ตระกูล MAX (MultipleArray Matrix) ของบริษัท Altera CPLD มีโครงสร้างคล้ายกับ SPLD
- CPLD ถูกผลิตขึ้นโดยเทคโนโลยีแบบ EPROM, EEPROM หรือ FLASH CPLD แบบ EPROM มักจะสามารถโปรแกรมได้เพียงครั้งเดียว (OTP) นอกเสียว่าจะมีช่อง (window) สำหรับใช้ลบข้อมูลด้วยแสงอุลตราไวโอเล็ต (UV) ในการ โปรแกรม CPLD จะต้องใช้ เครื่องโปรแกรมไอซีเฉพาะที่เรียกทั่วไป ว่า IC programmer ใน CPLD รุ่นใหม่ ๆ มักจะเป็นแบบ EEPROM หรือFLASH เนื่องจากสามารถโปรแกรม ได้โดยตรงในภาคสนามได้ (in-System Programmable)

FPIC (Field Programmable Interconnect device)

FPIC จริงๆ แล้วไม่ใช่ไอซีทางด้านลอจิก แต่จะเป็นอุปกรณ์ที่สามารถโปรแกรมส่วนที่เชื่อมต่อ (wiring) เช่นการเชื่อมต่อตัวให้ไอซีหนึ่งตัวสามารถเลือกได้ว่าจะให้เชื่อมต่อเข้ากับไอซีตัวไหนได้ บริษัทที่ผลิต อุปกรณ์ประเภทนี้ได้แก่ ไอซีDigital Crosspoint Switch ของบริษัท I-Cube Digital crosspoint device ของ บริษัท Lattice หรือ อุปกรณ์Hardware emulator ของบริษัท Aptix

FPGA (Field Programmable gate Array)

FPGA มีโครงสร้างที่แตกต่างไปจาก SPLD และ CPLD และมักจะมีขนาดความหนาแน่นของเกตมากกว่า โครงสร้างของ FPGA ประกอบด้วย แผงของบล็อกลอจิก ล้อมรอบด้วยส่วน I/Oที่สามารถโปรแกรมได้ และมีการเชื่อมต่อแบบ โปรแกรมได้ เช่นกัน FPGA มีขนาดตั้งแต่ระดับ1000 เกต จนถึงหลายล้านเกต การแบ่งหมวดของ FPGA นั้นสามารถแบ่งแยกตามคุณลักษณะได้ดังนี้

- โครงสร้างสถาปัตยกรรม
- เทคโนโลยีการ โปรแกรม
- I/O
- Interconnect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FPGA สามารถแบ่งตามโครงสร้างสถาปัตยกรรมภายในอุปกรณ์ดังนี้

- LCA (Logic Cell Array)
- PASIC (Programmable ASIC)
- FLEX, APEX
- ACT (Actel)

การแบ่งโครงสร้างสถาปัตยกรรมของ FPGA แบ่งได้เป็น 2 ลักษณะคือ

- Coarse-grained โครงสร้างสถาปัตยกรรมแบบ Coarse-grained จะประกอบด้วยลอจิกบล็อกขนาดใหญ่ เช่นมักประกอบด้วย LUT และ Flip-Flops 2 อัน หรือมากกว่าตัวอย่าง FPGA ได้แก่ Xilinx ตระกูล XC4K, Spartan, Virtex หรือ Altera ตระกูล FLEX, APEX
- Fine-grained ประกอบด้วย ลอจิกบล็อกแบบง่าย จำนวนมาก ลอจิกบล็อกประกอบด้วย ลอจิกฟังก์ชัน หรือ 2 อินพุต หรือมัลติเพล็กซ์เซอร์แบบ 4-1 และฟลิปฟล็อป ตัวอย่าง FPGA แบบ fine-grained เช่น ตระกูล ACT ของบริษัท ACTEL

เทคโนโลยีการโปรแกรม (Programming Technology)

เทคนิคการโปรแกรมไอซี FPGA มีดังนี้

- SRAM
- Fuse/Anti-fuse
- EPROM/EEPROM
- FLASH

ข้อแตกต่างอื่นๆของโครงสร้างสถาปัตยกรรมของ FPGA คือเทคโนโลยีที่ใช้ในการผลิตตัวชิป FPGA ซึ่งได้แก่

SRAM และ anti-fuse FPGA

FPGA ชนิด SRAM

นั้นสามารถทำการโปรแกรมซ้ำๆ ได้หลายครั้ง แต่ในการใช้งานในภาคสนามจำเป็นต้องใช้ไอซีสำหรับเก็บข้อมูลคอนฟิกูเรชันของวงจร การโหลดข้อมูลจากตัวเก็บบิต สามารถเกิดขึ้นได้โดยอัตโนมัติเมื่อเปิดสวิตช์ หรืออาจโหลด โปรแกรมผ่านไมโครโปรเซสเซอร์ก็ได้ เช่น โหลดโปรแกรมด้วย PC ผ่านสายคาวอร์โหลดข้อดี FPGA นอกจากจะมีขนาดของเกตสูงแล้ว ยังสามารถโปรแกรมซ้ำได้บ่อยตามที่ต้องการ FPGA ชนิด Fuse หรือ Anti-fuse จะสามารถทำการโปรแกรมได้เพียงครั้งเดียว (OTP) และไม่สามารถแก้ไขหรือโปรแกรมซ้ำได้อีก แต่ข้อมูลการโปรแกรมไม่สูญหายเมื่ วงจรถูกตัดแหล่งจ่ายไฟ การโปรแกรมไอซี FPGA ชนิดนี้จะต้องใช้เครื่องโปรแกรมไอซี

FPGA ชนิด EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM (Static RAM) ทำให้สามารถโปรแกรมได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกทในระดับปานกลางถึงสูงมาก (ประมาณ 10,000 - 1,000,000 เกท) ซึ่งข้อดีของ SRAM คือใช้เวลาในการโปรแกรมน้อย (ระดับ nsc) การโปรแกรมทำได้ง่ายเทียบกับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีความสลับซับซ้อน ส่วนข้อเสียคือไม่สามารถเก็บโปรแกรมในสถานะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิปในขณะเริ่มต้นใช้งาน

การออกแบบด้วย FPGA

การออกแบบวงจรดิจิทัลด้วย FPGA โดยทั่วไปมีองค์ประกอบ 3 ส่วน

-ซอฟต์แวร์ Design Entry

-โดยใช้ Schematic Design Entry ใช้ไลบรารีของ FPGA

-ใช้ภาษา HDL เช่น VHDL, Verilog, การออกแบบโดยใช้ภาษาขั้นสูงนั้นการทดสอบ

ของวงจรมันยังไม่ขึ้นกับเทคโนโลยีเป้าหมาย (Technology independent) ผู้ออกแบบไม่จำเป็นต้องกังวลถึงค่าความหน่วงทางเวลาของอุปกรณ์ที่มากับเทคโนโลยีนั้นการทดสอบความถูกต้องเป็นในลักษณะการตรวจสอบระดับฟังก์ชันการทำงานโดยใช้ซอฟต์แวร์สำหรับจำลองการทำงาน (Simulation)

Design Implementation

ขั้นตอนนี้ ต่อเนื่องจากขั้นตอนที่ 1 ซึ่งเกี่ยวข้องกับการแปลงแบบที่ได้ออกแบบจาก Schematic หรือ HDL ให้เป็นลอจิก ซึ่งอาจใช้ซอฟต์แวร์สำหรับสังเคราะห์วงจร (Logic synthesis) แล้วทำการแบ่งลอจิกเป็นส่วนๆ (Partitioning) และวางตำแหน่ง (Placement) ของลอจิกทำการเชื่อมต่อสายสัญญาณ (routing) สุดท้ายเป็นการสร้างไฟล์สำหรับโปรแกรมลงชิป (bit file)

Device Programming

การโปรแกรมอุปกรณ์หรือชิป FPGA นั้น มีเทคนิคหรือวิธีใหญ่ๆ 3 ลักษณะ ทั้งนี้ตัวชิปจะต้องสนับสนุนการทำงานในโหมดของการโปรแกรมเหล่านี้ด้วย

- การโปรแกรมโดยผ่านสายคาวิน์โหลด หรือผ่าน JTAG หรือผ่าน ISP

- การโปรแกรมโดยใช้ ตัวเก็บข้อมูลฟลैชบิต

- การโปรแกรมโดยใช้เครื่องโปรแกรมไอซี

ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว

ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิป เพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึง ภาษา Assembly ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจร หรือ HDL (Hardware Description Language) เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูง ทำได้รวดเร็ว และไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นตัวซอฟต์แวร์จะทำ Synthesis and Optimize ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตรฐานเดียวกันสามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสายควาน์โพลดทางพอร์ตของ คอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพขณะที่อยู่ในระบบได้ โดยไม่จำเป็นต้องถอดมาโปรแกรมข้างนอก และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสียค่าใช้จ่ายเพิ่มแต่อย่างใด

เครื่องมือสำหรับการออกแบบ FPGA

จะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้นทำได้สะดวกกว่า ASIC มากเพราะใช้นเวลาน้อยกว่ามากด้วย ส่วนสำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่เขียน โค้ดอธิบายฮาร์ดแวร์จนกระทั่งควาน์โพลดลงในอุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็น ซอฟต์แวร์ที่ทำงานต่อเนื่องกันได้ สำหรับซอฟต์แวร์ที่ใช้ทำการ จำลองการทำงานของวงจรมัน ต้องสามารถใช้งานต่อเนื่องกับซอฟต์แวร์ที่ใช้ทั้งระบบ เพราะโมเดลที่ได้จาก การทำขั้นตอนต่างๆ (ด้วยซอฟต์แวร์ ต่างๆ ต้องเอามาจำลองการทำงาน ได้ และในการจำลองการทำงานของ วงจรควรใช้ซอฟต์แวร์ตัวเดียวกันตลอดทั้งระบบ เพื่อจะได้เปรียบเทียบผลได้ง่าย ในอดีตซอฟต์แวร์ส่วนใหญ่ จะใช้งานอยู่บนคอมพิวเตอร์สมรรถนะสูงอย่างเวิร์คสเตชัน (Workstation) ในปัจจุบันมีการพัฒนาซอฟต์แวร์ ที่ใช้บนพีซี(PC)มากขึ้นซึ่งสามารถลดค่าใช้จ่ายในด้านอุปกรณ์คอมพิวเตอร์

2.42 VHDL

VHDL ย่อมาจาก VHSIC (Very High Speed Integrated Circuits) Hardware Description ซึ่งเป็นภาษา บรรยายฮาร์ดแวร์ประเภทหนึ่ง เป็นภาษาโครงสร้างประกอบด้วยหลายๆ ส่วนและเป็นภาษาที่ใช้ ออกแบบวงจรดิจิทัลได้ตั้งแต่ระบบดิจิทัล (System) ที่ซับซ้อน ไปจนถึง โลจิกเกต (Gate) ซึ่งง่ายต่อการ เรียนรู้และเป็นภาษาที่สามารถอ่านแล้วเข้าใจได้ และที่สำคัญของภาษานี้คือ ภาษานี้มีความเป็น Concurrent ซึ่งเป็นหัวใจสำคัญของการเขียนออกแบบอุปกรณ์อิเล็กทรอนิกส์

ประโยชน์ของภาษา VHDL มีดังนี้คือ

- เป็นภาษามาตรฐานสากลโดยรับรองจากสถาบัน IEEE ทำให้มีโปรแกรมและเครื่องมือต่างๆ และบริษัทที่สนับสนุนการทำงานมากมาย นอกจากนี้วงจรที่ออกแบบโดยภาษา VHDL ก็จะใช้งานได้นานเนื่องจากมีความเข้ากันได้ของภาษากับวงจรที่ได้รับการออกแบบใหม่
- เป็นภาษาที่สามารถใช้ได้หลายระบบ การออกแบบโดยใช้ภาษา VHDL สามารถนำไปจำลองการทำงานหรือสังเคราะห์ด้วยซอฟต์แวร์ตัวใดก็ได้ที่รองรับภาษา VHDL จึงทำให้การออกแบบด้วยภาษา

VHDL เป็นการออกแบบที่ไม่ยึดติดกับซอฟต์แวร์ที่ใช้ในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เป็นภาษาที่สามารถจำลองรูปแบบการทำงานของวงจร ผู้ออกแบบวงจรสามารถออกแบบวงจรโดยใช้ภาษา VHDL ได้หลายระดับ ตั้งแต่ระดับมาโครบล็อก (Macro block) จนถึงระดับเกต (Gate) และสามารถออกแบบวงจรที่มีความซับซ้อนสูงและมีขนาดใหญ่มากได้

- เป็นภาษาที่สามารถนำกลับมาใช้ใหม่ได้ วงจรที่ออกแบบโดยภาษา VHDL สามารถนำกลับมาใช้ใหม่ได้ง่าย เนื่องจากสามารถแก้ไขเปลี่ยนแปลงวงจรได้ง่าย คล้ายคลึงกับโปรแกรมของซอฟต์แวร์

- เป็นภาษาที่สามารถนำไปใช้เป็นเอกสารประกอบได้ เป็นภาษาในรูปแบบบรรยายพฤติกรรม ทำให้สามารถอธิบายการทำงานของวงจรภายในการออกแบบได้ทันทีที่ลักษณะการใช้งานภาษา VHDL การใช้งานภาษา VHDL อาจจำแนกออกเป็น 5 ประเภทดังนี้คือ

- Document Language : เป็นภาษาที่ใช้อธิบายรายละเอียดการทำงานของวงจรที่ออกแบบ
- Design Language : เป็นภาษาที่ใช้สำหรับออกแบบวงจรที่มีความซับซ้อนสูง
- Verification Language : เป็นภาษาที่ใช้ตรวจสอบความถูกต้องของวงจรที่ออกแบบ
- Test Language : เป็นภาษาที่ใช้สำหรับทดสอบการทำงานของวงจรที่ออกแบบ
- Synthesis Language : เป็นภาษาที่ใช้สำหรับสังเคราะห์วงจร (Synthesis) จริง แต่รูปแบบดังกล่าวนี้

ไม่ได้ครอบคลุมทุกรูปแบบทั้งหมดของการเขียนในภาษา VHDL แต่มีเพียงบางรูปแบบเท่านั้นที่สามารถเขียนสามารถนำไปสังเคราะห์เป็นวงจรได้ หรือเรียกว่าเป็น รูปแบบการเขียนสำหรับการสังเคราะห์ (VHDL for Synthesis)

วิธีการออกแบบระบบอิเล็กทรอนิกส์

การออกแบบระบบอิเล็กทรอนิกส์ จะเริ่มต้นจากการกำหนดหน้าที่การทำงานของระบบอิเล็กทรอนิกส์ที่ต้องการ ซึ่งโดยส่วนใหญ่แล้วระบบอิเล็กทรอนิกส์มักจะประกอบด้วย 2 ส่วนคือ ส่วนที่เป็นระบบฮาร์ดแวร์และส่วนที่เป็นซอฟต์แวร์ โดยทั่วไปแล้วการออกแบบทั้งสองส่วนดังกล่าวนี้จะต้องทำไปพร้อมๆ กัน หลังจากนั้นจะเป็นขั้นตอนในการกำหนดหน้าที่การทำงานของระบบฮาร์ดแวร์และระบบซอฟต์แวร์ โดยสำหรับการออกแบบทางด้านฮาร์ดแวร์ สามารถเลือกได้ว่าต้องการออกแบบเป็น ASIC (Application-Specific Integrated Circuits), FPGA (Field Programmable Gate Arrays), PLD (Programmable Logic Devices) หรือจากไอซีมาตรฐานที่มีขายอยู่ทั่วไปตามท้องตลาด ส่วนการออกแบบระบบซอฟต์แวร์ ผู้ออกแบบสามารถเลือกใช้ภาษาโปรแกรมต่างๆ นำไปออกแบบ เช่น ภาษา C/C++ ภาษา Assembly ส่วนสำหรับในการออกแบบ Firmware และ Application โดยในระหว่างการออกแบบนั้น ผู้ออกแบบทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ จะต้องมีการทดสอบการทำงานร่วมกันเพื่อทำให้สามารถแก้ไขความผิดพลาดในการออกแบบที่อาจเกิดขึ้นได้

กระบวนการออกแบบลักษณะ Top-Down design

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบวนการออกแบบลักษณะ Top-Down design เป็นกระบวนการที่ใช้ในการออกแบบระบบดิจิทัลในปัจจุบัน โดยมีขั้นตอนต่างๆ ดังนี้ ในขั้นตอนแรกคือ การวางข้อกำหนดคุณลักษณะของระบบ (Requirements) ทั้งด้านสมรรถนะและด้าน ภายนอก และจากข้อกำหนดต่างๆ ในการกำหนดด้านคุณลักษณะของระบบเราก็จะสามารถนำมาออกแบบเชิงหน้าที่ระดับสูง (High-Level function design) ในเบื้องต้นได้ ซึ่งต้องอาศัยการจำลองการทำงาน (Simulation) เพื่อดูว่าระบบที่ออกแบบไว้ทำงานสอดคล้องกับข้อกำหนดต่างๆ หรือไม่ เพื่อแก้ไขให้สอดคล้องในที่สุดเมื่อเสร็จสิ้นการออกแบบเชิงหน้าที่ (Function design) แล้ว ขั้นตอนต่อไปคือ การใส่รายละเอียดลงไปหน่วยความจำ (Memory) หน่วยคำนวณและหน่วยตรรกะ (ALU) และเครื่องสถานะ (State machine) เราเรียกรูปแบบในระดับนี้ว่า Register Transfer Level หรือ RTL หลังจากการออกแบบในระดับ RTL ก็จะเป็นการออกแบบในระดับตรรกะ (Logic design) ซึ่งเป็นการเพิ่มเติมรายละเอียดให้กับหน่วยต่างๆ ที่เป็นส่วนประกอบของงานออกแบบระดับใน RTL เช่นการกำหนดในรายละเอียดว่า รีจิสเตอร์ตัวหนึ่งประกอบไปด้วยฟลิปฟล็อปและลอจิกเกตแบบใดจำนวนเท่าใด เป็นต้นการออกแบบในระดับ RTL และระดับ โลจิก (Logic) ต้องมีการจำลองการทำงานควบคู่ไปด้วยเช่นกัน เหมือนการออกแบบระดับ Functional ในบางระบบนอกจากการจำลองเชิงตรรกะ (Logic simulation) แล้ว อาจมีการจำลองเพื่อหาข้อบกพร่อง (Fault simulation) ของวงจรหรือระบบด้วย Fault simulation สามารถจำลองผลกระทบที่อาจเกิดจากข้อบกพร่องในกระบวนการผลิตรวมถึงข้อบกพร่องที่เกิดจากการเหนี่ยวนำของสิ่งแวดล้อม ตัวอย่างเช่น กรณีที่มีผู้นำชิปประมวลผลภาพไปใช้บนดาวเทียม รังสีต่างๆ ในอวกาศอาจเหนี่ยวนำให้วงจรบนชิปเปลี่ยนสถานะโดยไม่ต้องทำให้เกิดความผิดพลาด เช่น Signal-bit error ได้ ซึ่งถ้าอัตราความผิดพลาดบิต (bit error rate) สูงเกินไป เราอาจปรับแก้งานออกแบบของเราให้ทนทานต่อข้อบกพร่องด้วยเทคนิคต่างๆ ได้

ขั้นตอนสองขั้นสุดท้าย คือการออกแบบในระดับวงจร (Circuit design) และการออกแบบในระดับกายภาพ (Physical design) การออกแบบในระดับวงจรคือการแปลงลอจิกเกตและฟลิปฟล็อปต่างให้กลายเป็นทรานซิสเตอร์ ซึ่งในขั้นตอนนี้จะมีการวิเคราะห์วงจรและจำลองเชิงเวลา (Timing simulation) ควบคู่กันไปด้วยเพื่อตรวจสอบว่าวงจรและระบบที่ออกแบบไว้ ยังทำงานสอดคล้องกับข้อกำหนดต่างๆ ที่ตั้งไว้ในตอนแรกหรือไม่ การออกแบบในระดับกายภาพ หรือระดับล่างสุด คือการออกแบบผังภูมิ (Layout) ของวงจรรวมที่ออกแบบไว้ โดยการแปลงทรานซิสเตอร์ต่างๆ ให้เป็นรูปเหลี่ยมและลายเส้นที่มีสีสันทนวัสดุชั้นต่างๆ ตามกระบวนการผลิตชิปวงจรรวม ภายหลังจากการออกแบบจะมีการตรวจสอบความถูกต้องตามกฎการออกแบบ การสกัดพารามิเตอร์ต่างๆ และการจำลองในระดับวงจร เช่น Timing simulation อีกครั้ง โดยการนำค่าพารามิเตอร์ที่สกัดได้มาประกอบการคำนวณ ในขั้นนี้เราสามารถประเมินคุณสมบัติทางกายภาพที่ถูกต้องได้ เช่น พื้นที่ชิปและกำลังที่สูญเสียเป็นความร้อน

การแบ่งระดับขั้นการออกแบบที่เราเรียกแต่ละระดับว่าระดับของการกำหนดสาระสำคัญ (Level of abstraction) การกำหนดสาระสำคัญระดับสูงจะประกอบไปด้วยส่วนประกอบที่ซับซ้อนและทรงพลังเป็นน้อยเช่น วงจรบวก (Adder) และหน่วยความจำ (Memory) ตรงข้ามกับการกำหนดสาระสำคัญในระดับล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่จะประกอบไปด้วยส่วนประกอบจำนวนมากที่ไม่ซับซ้อนและไม่ทรงพลัง เช่น ประตูสัญญาณ (Gate) และทรานซิสเตอร์โดยทั่วไปกระบวนการออกแบบจะเป็นไปลักษณะจากบนลงล่างหากมีการค้นพบความผิดพลาดที่ระดับล่าง การแก้ไขในระดับนี้จะเป็นไปได้ยากและแพงกว่ากรณีที่ตรวจพบข้อผิดพลาดแต่เนิ่นๆ ในระดับสูง เพราะมีรายละเอียดที่ต้องแก้ไขมาก และยังทำให้เสียเวลาอีกด้วย

การเขียนภาษา VHDL เบื้องต้น

ภาษา VHDL เป็นภาษาลักษณะเป็นแบบ Case insensitive หมายถึงไม่มีความแตกต่างกันในการเขียน อักษรพิมพ์เล็กหรือพิมพ์ใหญ่ ไม่ว่าจะเขียนแบบใดก็มีความหมายเหมือนกันหลักการเขียนภาษา VHDL เบื้องต้นในการออกแบบวงจรดิจิทัล โดยเริ่มจากเขียนโค้ดในแต่ละบรรทัด จะต้องจบด้วยเครื่องหมายอัฒภาค (;) และการกำหนดค่า (Assignment) ให้ออบเจกต์ (Object) อาทิเช่น สัญญาณ (Signal) จะใช้เครื่องหมายน้อยกว่าหรือเท่ากับ (\leq) หรือมีชื่อเรียกว่า Signal Assignment โดยทางขวามือของเครื่องหมายจะเป็นค่าอินพุต (Input) และทางซ้ายมือจะเป็นค่าเอาต์พุต (Output) การประกาศสัญญาณหลายๆ ค่าให้ใช้เครื่องหมายจุลภาค (,) คั่นระหว่างสัญญาณแต่ละสัญญาณ ถ้าในกรณีที่ไม่มีกำหนดค่าเริ่มต้น ระบบจำลองการทำงานจะนำค่าที่อยู่ทางซ้ายมือของกลุ่มค่า ของแต่ละประเภทมาเป็นค่าเริ่มต้นเสมอ (Default initial) การเขียนคำอธิบายวงจร (Comment) จะมีความหมายในบรรทัดนั้นไม่ว่าจะเขียนจะอะไรไป Compiler จะไม่ทำการตรวจสอบไวยากรณ์ของการเขียนในบรรทัดนั้น โดยในภาษา VHDL จะใช้เครื่องหมาย (--) โดยเขียนไว้หน้าบรรทัดนั้นๆ ที่ไม่ต้องการให้ Compiler สนใจหรือต้องการเขียนอธิบายโค้ดที่เขียนขึ้นในแต่ละบรรทัด ทำให้โค้ดอ่านเข้าใจยิ่งขึ้น และง่ายต่อการแก้ไขปรับปรุง การเขียนคำอธิบายในภาษา VHDL จะใช้เครื่องหมาย (--) ไว้ที่หน้าบรรทัดที่เราต้องการใช้เป็น คำอธิบายเท่านั้น โดยการเขียนเครื่องหมายดังกล่าว จะใช้ได้บรรทัดต่อบรรทัดเท่านั้น ไม่สามารถทำเป็นกลุ่มบรรทัด(Block of line) ได้ การทำงานแบบขนาน (Concurrent) ภาษา VHDL เป็นประเภท Concurrent language ที่มีคุณสมบัติ Concurrency หมายถึงการเขียน โค้ดในแต่ละบรรทัดจะไม่มีลำดับความสำคัญของตำแหน่งของแต่ละบรรทัด (Order independent) ซึ่งในภาษาอื่นๆ จะไม่มีคุณสมบัตินี้ จึงไม่สามารถนำไปออกแบบพฤติกรรมการทำงานวงจรฮาร์ดแวร์ จริงได้

ออบเจกต์ในภาษา VHDL แบ่งออกเป็น 3 ประเภทคือ

- ค่าคงที่ (CONSTANT) เป็นออบเจกต์ที่เมื่อกำหนดค่าเริ่มต้นให้แล้วจะคงค่านั้นไว้ตลอด ไม่สามารถดัดแปลง หรือแก้ไขได้ สามารถประกาศใช้ได้ในส่วนที่เป็นส่วนประกาศต่างๆ ของโมเดล
- สัญญาณ (SIGNAL) เป็นออบเจกต์ที่สามารถกำหนดค่าที่สัมพันธ์กับเวลานั้น หมายความว่า สัญญาณสามารถรับค่าได้เพียงค่าเดียวเท่านั้น ในขณะเวลาหนึ่ง สัญญาณจะรับค่าๆ หนึ่งได้จากการจับสัญญาณ ซึ่งตัวจับนี้อาจเก็บค่าในอนาคตสำหรับสัญญาณไว้ด้วยสัญญาณสามารถประกาศใช้ได้

ในส่วนที่เป็นเนื้อหาของ Concurrent body เท่านั้น ดังนั้นสัญญาณจึงสามารถนำไปใช้ได้ตลอดในโครงสร้างของโมเดลหรือเรียกว่า Global object

- ตัวแปร (VARIABLE) เป็นออบเจกต์ที่สามารถกำหนดค่าใดๆ ให้ได้ และสามารถที่จะเปลี่ยนแปลงค่าได้ตลอดเวลาการทำงาน แต่จะเก็บค่าได้เพียงค่าเดียวเท่านั้นในขณะเวลาหนึ่ง เนื่องจาก

ตัวแปรสามารถประกาศใช้ได้ส่วนที่เป็นส่วนของ Sequential body เท่านั้นอันได้แก่ส่วนประกาศของ PROCESS, FUNCTION หรือ PROCEDURE ดังนั้น ตัวแปร จึงสามารถนำไปใช้ได้เฉพาะในขอบเขตที่ถูกประกาศใช้เท่านั้นหรือเรียกว่าเป็น Local object

การที่จะใช้งานออบเจกต์ต่างๆ ตามที่ได้กล่าวมาข้างต้น นั้นจะต้องมีการประกาศออบเจกต์ใช้ก่อนเสมอถึงจะสามารถนำไปใช้งาน

องค์ประกอบของภาษา VHDL

การใช้ภาษา VHDL ในการออกแบบวงจรดิจิทัลไม่ว่าจะเป็นวงจรพื้นฐานหรือวงจรที่ซับซ้อน ทุกวงจรที่ออกแบบ จะต้องเขียนในรูปแบบของ VHDL Component ซึ่งเป็นวงจรพื้นฐานของการออกแบบ โดยหน่วยการออกแบบ (Design unit) ในภาษา VHDL จะแบ่งออกเป็นประเภทต่างๆ ดังนี้

- Entity
- Architecture
- Package
- Configuration

Entity

เป็นการออกแบบที่ใช้สำหรับติดต่อระหว่างอุปกรณ์ภายนอกกับวงจรที่จะเขียนขึ้น รวมทั้งการส่งผ่าน ค่าพารามิเตอร์บางอย่างระหว่างวงจรรกับอุปกรณ์ภายนอก ประเภทของพอร์ตที่สามารถประกาศใช้ใน Entity มี 4 ประเภทดังนี้

- In (Input)

เป็นพอร์ตทิศทางเดียว ที่นำค่าสัญญาณจากอุปกรณ์ภายนอกเข้ามาใช้ภายในวงจร สามารถนำมาป้อนให้กับสัญญาณอื่นหรืออ่านค่าอื่นได้แต่ไม่สามารถถูกเขียนจากภายในวงจรได้

- Out (Output)

เป็นพอร์ตทิศทางเดียว ที่นำค่าสัญญาณจากวงจร ส่งออกไปยังอุปกรณ์ภายนอกสามารถเขียนจากภายในวงจรได้แต่ไม่สามารถอ่านจากภายในวงจรได้

- Inout (Bidirectional)

เป็นพอร์ต 2 ทิศทาง ที่สามารถส่งถูกเขียนและอ่านได้จากภายในวงจร

- Buffer (Output with internal Feedback)

เป็นพอร์ตเอาท์พุทประเภทหนึ่งที่สามารถอ่านค่ากลับ (Feedback) เข้ามาภายในวงจรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Architecture

เป็นหน่วยการออกแบบส่วนที่ใช้เขียนบรรยายพฤติกรรมการทำงานของวงจรที่ต้องการออกแบบ โดยมีความสัมพันธ์กับสิ่งที่กำหนดใน Entity ถ้าผู้ออกแบบต้องการออกแบบวงจรดิจิทัลใดๆ จะต้องออกแบบหน่วยการออกแบบวงจรพื้นฐานก็คือ ส่วน Entity และส่วนของ Architecture ที่มีความสัมพันธ์กัน ซึ่งมีด้วยกันหลายรูปแบบสำหรับการออกแบบวงจรเดียวกัน และความเข้าใจของผู้ออกแบบเอง ขึ้นอยู่กับผู้ออกแบบที่จะนำรูปใดมาใช้งาน

Package

แพ็คเกจ (Package) จะเป็นหน่วยการออกแบบที่ใช้เก็บข้อมูลต่างๆ ตลอดจน โปรแกรมย่อย (Subprogram) ที่เป็นประโยชน์ในการเขียนรูปแบบการบรรยายวงจรดิจิทัล โดยข้อมูลใน Package สามารถถูกเรียกใช้ได้โดย Entity, Architecture หรือด้วย Package อื่นๆ ด้วยคำสั่ง USE statement นอกจากนั้นสิ่งที่ยึดนิยามทำกันมากคือรูปแบบมาตรฐานต่างๆ อาทิ เช่น Standard component ต่างๆ จะถูกเก็บไว้ใน Package ที่ทุกคนเรียกใช้งานได้โดยสิ่งที่สามารถสร้างไว้ใช้ใน Package ได้แก่

- Subprogram
- Type
- Constant
- Signal
- Aliases
- Attributes
- Component
- Disconnection Specification

การเรียกใช้งาน Package จะใช้คำสั่ง Use statement โดยปกติการเขียน Package จะแบ่งออกเป็น 2 ส่วนคือ

- Package declaration
- Package body

Package declaration

เป็นส่วนที่สำคัญที่สุดของ Package เพราะจะเป็นส่วนที่กำหนดชื่อ (Identifier) ของสิ่งที่ประกาศ อยู่ใน Package สำหรับนำไปใช้ภายนอกตัวของ Package ถ้าสิ่งใดที่ถูกประกาศอยู่ในส่วนของ Package body แต่ไม่ถูกประกาศในส่วน of Package declaration ก็จะไม่สามารถนำไปใช้งานภายนอกได้ ซึ่งเปรียบเทียบกับการประกาศพอร์ตใน Entity ที่มีหน้าที่ติดต่ออุปกรณ์ภายนอก ดังนั้นโดยทั่วไปแล้ว Package สามารถเขียนได้โดยไม่ต้องมีส่วนของ Package body และยังสามารถถูกนำไปใช้ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายนอก เช่น ใช้สำหรับประกาศชนิดของข้อมูล (Type) หรือสัญญาณ (Signal) ในทางกลับกันกับ Package body ที่ไม่จำเป็นต้องมี Package declaration แต่ Package นั้นจะไม่สามารถถูกนำไปใช้จากภายนอกได้

Package body

โครงสร้างที่ประกอบด้วยคำสั่งต่างๆ ในรูปแบบของคำสั่งลำดับ (Sequential statement) ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ที่ได้ประกาศไว้ใน Package declaration แล้ว

Configuration

ดังที่กล่าวมาแล้วว่ารูปแบบการเขียนส่วน Entity สามารถมีหลายๆ Architecture และภายใน Architecture ยังมีอุปกรณ์ที่เรียกว่า Component ที่อ้างถึง Entity อื่นๆ อีก ดังนั้นในการจำลองการทำงาน (Simulate) จึงเกิดปัญหาว่าจะนำเอา Architecture อันไหน ไปจำลองการทำงาน ฉะนั้นเราจึงต้องเขียนส่วนของ Configuration ในการกำหนดว่าจะนำ Architecture อันไหน ไปใช้งาน Configuration ส่วนใหญ่ไม่สามารถนำไปใช้กับ Synthesis tools ได้จะใช้ได้เฉพาะขั้นตอนการ Simulation เท่านั้น เพราะฉะนั้นในขั้นตอนการ Synthesis Entity หนึ่งๆ ต้องมีเพียง หนึ่ง Architecture เท่านั้น แต่ในบางครั้งถ้าเราไม่ต้องการเขียน Configuration ก็ได้ถ้าเราออกแบบให้หนึ่ง Entity มีเพียงหนึ่ง Architecture เท่านั้น ในกรณีที่ไม่จำเป็นต้องเขียน Configuration บอก Simulator เนื่องจาก ไม่มีทางเลือกอื่นใด ให้ซอฟต์แวร์ Simulator เลือกวิธีนี้เราเรียกว่า Default configuration หรือถ้าบางครั้งเราออกแบบให้หนึ่ง Entity มีหลาย Architecture แต่เราก็สามารถไม่ต้องเขียน Configuration ได้เช่นกัน แต่ก็เกิดคำถามตามมาว่าเมื่อจำลองการทำงาน Simulator จะเลือกเอา Architecture อันไหนมาจำลองการทำงาน คำตอบ ในกรณีไม่เขียน Configuration นั่นคือ Last compileล่าสุดที่มีการ Compile ใน Architecture อันไหน Simulator ก็จะเอา Architecture อันนั้นไปจำลองการทำงาน ดังนั้น ควรระวังในการเขียน โค้ดรูปแบบดังกล่าวด้วยการเขียนแบบลำดับชั้น (Hierarchical Model)

ในการออกแบบวงจรดิจิทัล บางครั้งต้องมีการแบ่งวงจรออกเป็นบล็อกย่อยๆตามหน้าที่การทำงาน เนื่องจากระบบที่ต้องการออกแบบมีความซับซ้อนสูงไม่สามารถที่จะออกแบบได้เพียงบล็อกเดียว ทำให้จำเป็นต้อง แบ่งวงจรออกเป็นบล็อกย่อยๆ ซึ่งในภาษา VHDL ก็มีความสามารถในการเขียนลำดับชั้น (Hierarchical) ได้ โดย เขียนบรรยายวงจรตั้งแต่ระดับบนสุด (Top model) ที่มีแต่ความสัมพันธ์ของการเชื่อมต่อในแต่ละบล็อกย่อย โดยยังไม่มียรายละเอียดของวงจรในแต่ละบล็อก และในระดับล่างลงไปถึงจะมีรายละเอียดของวงจร

การประกาศคอมโปเนนต์ (Component Declaration)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

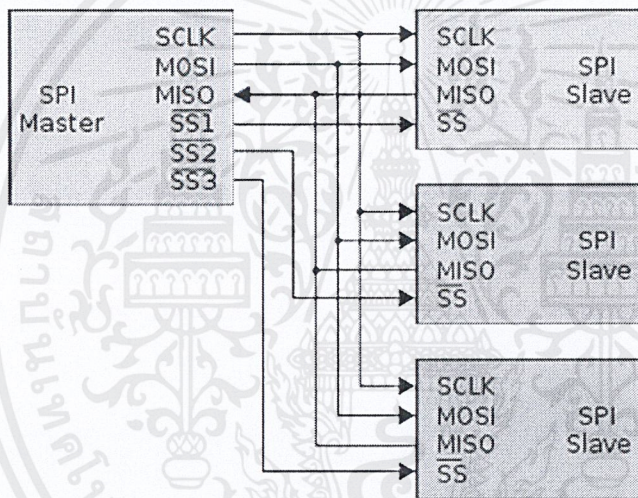
เป็นการประกาศคอมโปเนนต์ที่ต้องนำมาเชื่อมต่อกันภายในวงจรส่วนบนสุด (Top model) จะเป็นการเขียนในลักษณะลำดับชั้น (Hierarchical) โดยการประกาศคอมโปเนนต์นั้นจะต้องทำการออกแบบคอมโปเนนต์ดังกล่าวไว้ก่อนเรียบร้อยแล้ว เพียงแต่เราเรียกคอมโปเนนต์ดังกล่าวมาใช้งาน โดยการออกแบบคอมโปเนนต์ ก็คล้ายกับการออกแบบวงจรทั่วไป คือสร้างส่วนของ Entity และ Architecture ขึ้นมาก่อนแล้วเก็บไว้ในไลบรารี จากนั้นถ้าวงจรที่ออกแบบขึ้นมาใหม่ต้องการเรียกใช้วงจรที่เคยออกแบบไว้แล้วสามารถทำได้โดยการประกาศส่วนของโมเดลที่มีอยู่แล้วในรูปของคอมโปเนนต์ (Component Declaration) ซึ่งตำแหน่งของการประกาศคอมโปเนนต์สามารถประกาศในบริเวณระหว่างคำสั่ง ARCHITECTURE และ BEGIN การประกาศส่วนของคอมโปเนนต์นั้นถ้านำไปเปรียบเทียบกับ การออกแบบวงจรบน PCB (Printed Circuit Board) จะเปรียบเสมือนว่าเป็นซ็อกเก็ต ว่างๆ ไว้สำหรับเสียบไอซีลงไป โดยซ็อกเก็ตมีหน้าที่เพียงเชื่อมต่อสัญญาณ เท่านั้น ไม่มีหน้าที่สร้างฟังก์ชันการทำงาน การประกาศส่วนของอุปกรณ์ก็เช่นกัน จะไม่มีการแสดงถึงฟังก์ชันการทำงาน แต่มีหน้าที่เพียงเชื่อมต่อสัญญาณต่างๆ เข้ากับพอร์ตของโมดูลย่อย (Component) นั้นๆ

การเรียกใช้งานคอมโปเนนต์

เป็นการเรียกใช้ Component ที่มีอยู่แล้ว โดยการเชื่อมต่อสัญญาณต่างๆ ของโมดูลย่อยที่ประกาศไว้ในส่วนของ Component declaration เข้ากับพอร์ตหรือสัญญาณภายในของวงจรระดับบนสุด (Top model) ตำแหน่งการเขียน Component instantiation เป็นรูปแบบหนึ่งของ Concurrent statement ดังนั้นสามารถเขียนภายใต้ BEGIN ของ ARCHITECTURE ได้ คำสั่ง GENERIC MAP เป็นคำสั่งเสริม (Optional) ถ้าวงจรมีการใช้คำสั่ง GENERIC มาก่อนหน้านี้แล้ว ถ้าเรานำเอาบล็อกของวงจรดังกล่าวมาใช้งาน เวลาเขียน Component instantiation จะต้องมีคำสั่ง GENERIC MAP เสมอ แต่ถ้าบล็อกของวงจรที่เราใช้งานไม่มีการใช้คำสั่ง GENERIC เวลาเขียน Component instantiation ก็ไม่ต้องมี ส่วน GENERIC MAP ในการเชื่อมต่อสัญญาณต่างๆ เข้ากับพอร์ตของคอมโปเนนต์ สามารถทำได้ 2 รูปแบบ คือ เชื่อมต่อสัญญาณแบบ Port association list คือ พิจารณาจากตำแหน่งของพอร์ต และตำแหน่งในสัญญาณ ที่ตรงกันจะมีความหมายว่าต่อถึงกันอยู่ ไม่มีการอ้างชื่อพอร์ตเพียงแต่วางตำแหน่งให้ตรงกันก็พอ และอีกรูปแบบหนึ่งการเขียนคำสั่ง PORT MAP ในรูปแบบ Name association list จะหมายถึงจะไม่พิจารณาตำแหน่ง แต่ต้องอ้างอิงด้วยชื่อในการเชื่อมต่อ

2.5 ทฤษฎี SPI Interface

Serial Peripheral Interface (SPI) ระบบ SPI เป็น Synchronous Serial Data Link ซึ่งจัดการในโหมด Full Duplex มีการติดต่อกับอุปกรณ์ในโหมด Master/Slave ระบบบัสของ SPI จะใช้สายสัญญาณจำนวน 4 เส้น (ไม่นับสาย GND) คือ SCK (Serial Clock), สัญญาณ SCK เป็นสัญญาณนาฬิกาที่สร้างให้กับบัสเมื่อทำงานในโหมดมาสเตอร์หรือเป็นขารับสัญญาณนาฬิกาเมื่อทำงานเป็น Slave SSEL (Slave Select), สายสัญญาณสำหรับเลือกอุปกรณ์สเลฟ MISO (Master In Slave Output) MISO เป็นขารับข้อมูลสำหรับอุปกรณ์มาสเตอร์ที่ส่งมาจากอุปกรณ์สเลฟ และ MOSI (Master Out Slave In) MOSI เป็นขาส่งข้อมูลจากอุปกรณ์มาสเตอร์ไปยังอุปกรณ์สเลฟ โดยความถี่สูงสุดที่ทำงานได้นั้นจะขึ้นอยู่กับความถี่สูงสุดของอุปกรณ์สเลฟ



รูปที่ 2.29 รูปแสดงการทำงานของ SPI Interface

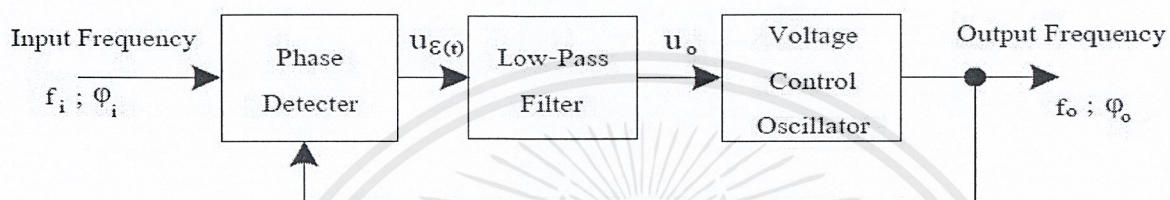
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 Phase Lock Loop

หลักการ

PLL เป็นระบบควบคุมความถี่ โดยใช้วิธีเปรียบเทียบเฟส (Phase) ของความถี่ทางด้านเอาต์พุต กับเฟสของความถี่อ้างอิง (Reference Frequency) ซึ่งถูกป้อนเข้าทางด้านอินพุตของระบบ

ในรูปข้างล่างนี้ เป็นแผนผังของ PLL เบื้องต้น



รูปที่ 2.30 แผนผัง Phase Lock Loop

จากแผนผัง พบว่า PLL เบื้องต้น จะประกอบด้วย 3 ส่วนใหญ่ๆ คือ

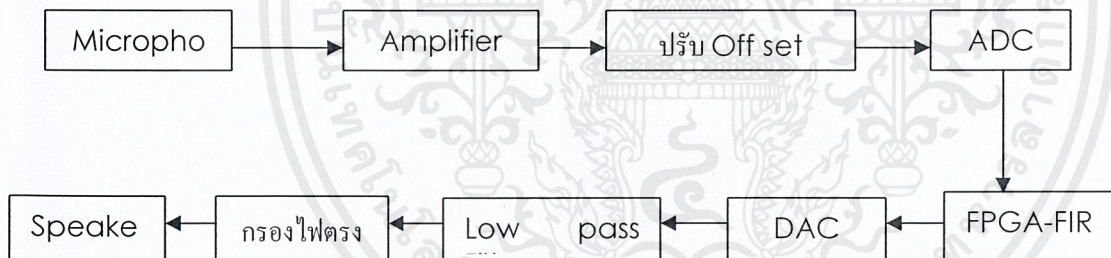
1. ภาคตรวจจับเฟส หรือตัวเปรียบเทียบสัญญาณ (Phase Detector หรือ Comparator) มีหน้าที่เปรียบเทียบความแตกต่างของเฟสระหว่าง i และ o ให้กำเนิดแรงดันคลาดเคลื่อน (Error Voltage) ; u_E ออกมาทางเอาต์พุต
2. ภาคกรองความถี่ต่ำผ่าน หรือภาคกรองความถี่ลูป (Low – Pass Filter หรือ Loop Filter) ทำหน้าที่กำจัดส่วนประกอบทางไฟลต์ลับที่ปะปนมากับแรงดันคลาดเคลื่อน และปล่อยให้ส่วนประกอบทางไฟตรงของแรงดันคลาดเคลื่อน ผ่านไปยังเอาต์พุต
3. ภาคกำเนิดสัญญาณควบคุมด้วยแรงดัน (Voltage Controlled Oscillator, VCO) ความถี่ o ของ VCO จะเปลี่ยนไปตามแรงดัน; o u ทางอินพุต ดังนั้นเมื่อ o u เปลี่ยนไปก็จะมีผลทำให้ o f และ o ϕ เปลี่ยนแปลงตามไปด้วย

บทที่ 3

การออกแบบและการทำงาน

3.1 โครงสร้างของระบบ

โครงสร้างของระบบส่วนหลักแบ่งออกเป็น 1. ส่วนรับเสียงเข้ามา (Microphone) 2. ส่วนของวงจรขยายสัญญาณ 3. ส่วนของการปรับ Offset 4. ส่วนแปลงสัญญาณจากสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC) 5. ส่วนของวงจรกรองสัญญาณ (FPGA-FIR) 6. ส่วนของการแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก 7. วงจรกรองความถี่ต่ำผ่าน 8. กรองไฟตรง โดยเมื่อเรารับเสียงเข้ามาทางไมโครโฟนแล้วเราก็นำเสียงที่ได้ไปเข้าวงจรขยาย เข้าวงจรขยายเสร็จเราจึงไปเปลี่ยนจาสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลเพื่อเข้าวงจรกรองความถี่ (FIR) จากนั้นนำสัญญาณที่กรองแล้วซึ่งเป็นสัญญาณดิจิทัลนำมาแปลงกลับเป็นสัญญาณอนาล็อก แล้วผ่านวงจรกรองความถี่ต่ำผ่านเพื่อกรองสัญญาณรบกวนออกไป แล้วผ่าน c เพื่อกรองไปตรงออก



รูปที่ 3.1 รูปแสดงโครงสร้างการออกแบบ

3.1.1 ส่วนรับเสียงเข้ามา (Microphone)

ส่วนนี้เราจะใช้อุปกรณ์ที่เรียกว่า Small Talk ซึ่งมีทั้งไมค์ และหูฟังไปในตัวด้วย

3.1.2 ส่วนของการขยายสัญญาณที่ได้รับเข้ามา (Amplifier)

เนื่องจากสัญญาณที่ได้รับมาจากไมโครโฟนมีขนาดเล็กมากจึงต้องนำสัญญาณที่ได้มาเข้าวงจรขยาย (LM386) เพื่อให้ได้ขนาดที่เหมาะสมกับการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC)

3.1.3 ส่วนของการปรับ Off set

เนื่องจาก ADC ที่เราใช้งานรับสัญญาณอนาล็อกอินพุตที่ 0-3 โวลต์ แต่สัญญาณที่ได้รับมาจะมีทั้งค่าบวกและค่าลบ จึงต้องยกระดับไฟสถับนี้ขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 แปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC)

เมื่อได้สัญญาณที่พร้อมจะเข้าสู่การแปลงแล้วก็นำมาเข้าการแปลงสัญญาณอนาล็อกเป็นดิจิทัล โดยใช้ ไอซี MCP 3201 ขนาด 12 bit

3.1.5 ส่วนของวงจรกรองสัญญาณ (FIR)

ส่วนนี้เราได้ใช้ภาษา VHDL ในการเขียนโปรแกรมตัวกรองสัญญาณขึ้นมา แล้วนำมาทดสอบการใช้งานบนโปรแกรม XILINK ว่าโปรแกรมนั้นใช้ได้จริงหรือไม่ และสัมประสิทธิ์ของวงจรกรองความถี่นั้นได้มาจากการจำลองตัวกรองสัญญาณในโปรแกรม MATLAB โดยใช้ ฟังก์ชัน FDATool ส่วนโปรแกรมที่เขียนขึ้นมามีดังต่อไปนี้

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity SPI_INTERFACE_MERGE is
port( osc_clk : in std_logic;

trap_in : out signed(11 downto 8);
trap_out : out signed(11 downto 8);

-----ADC PART-----
cs_adc : buffer std_logic;
adc_clk : in std_logic; --Into module.
-- spi_clk_in : in std_logic; --Generated from DCM.
spi_clk_out : out std_logic; --Toward ADC.
-- data_out : out signed(11 downto 0);
d_in : in std_logic;

-----DAC PART-----
dac_clk : in std_logic; --Into Module.
sck : out std_logic; --Toward DAC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

data_out : out std_logic;
cs_dac : out std_logic
-- clk : in std_logic;
-- data_in : in signed(11 downto 0)
);
end SPI_INTERFACE_MERGE;

```

architecture Behavioral of SPI_INTERFACE_MERGE is

```

component SPI_INTERFACE_OUT_TEST

```

```

port(sck : out std_logic;
data_out : out std_logic;
cs : out std_logic;
clk : in std_logic;
data_in : in signed(11 downto 0));
end component;

```

```

component SPI_INTERFACE_IN

```

```

port( cs : buffer std_logic;
clk : in std_logic;
-- spi_clk_in : in std_logic; --Generated from DCM.
spi_clk_out : out std_logic; --Toward ADC.
data_out : out signed(11 downto 0);
d_in : in std_logic);
end component;

```

```

component BANDPASS

```

```

generic( tap : natural := 84;
adc_bit : natural := 12;
coeff_resolution : natural := 14);
port( clk : in std_logic;

```

```

trap_in : out signed(11 downto 8);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

trap_out : out signed(11 downto 8);

filter_in : in signed(adc_bit-1 downto 0);
output : out signed(adc_bit-1 downto 0);
end component;

signal data_in_buffer : signed(11 downto 0);
signal data_in_buffer2 : signed(11 downto 0);
begin
ADC_SUB_CIRCUIT : SPI_INTERFACE_IN port map (cs => cs_adc,clk => adc_clk,
    spi_clk_out => spi_clk_out,
    data_out => data_in_buffer,
    d_in => d_in);
DAC_SUB_CIRCUIT : SPI_INTERFACE_OUT_TEST port map (sck => sck,data_out => data_out,
    cs => cs_dac,clk => dac_clk,data_in => data_in_buffer2);
FIR_BANDPASS_CIRCUIT : BANDPASS generic map ( tap => 51, adc_bit => 12, coeff_resolution =>
14)
    port map ( clk => osc_clk, filter_in => data_in_buffer, output => data_in_buffer2,
        trap_in => trap_in, trap_out => trap_out);

end Behavioral;
--*****SPI INTERFACE OUT*****
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.std_logic_signed.all;
use ieee.std_logic_arith.all;

entity SPI_INTERFACE_OUT_TEST is
port(sck : out std_logic;
    data_out : out std_logic;
    cs : out std_logic;
    clk : in std_logic;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    data_in : in signed(11 downto 0));
end SPI_INTERFACE_OUT_TEST;

```

architecture Behavioral of SPI_INTERFACE_OUT_TEST is

```

    signal sig_data_in : std_logic_vector(11 downto 0);

```

```

    signal sck_counter : integer := 0; --SCK clock counter

```

```

    signal sig_sck : std_logic := '0'; --pre SCK clock

```

```

    signal state_counter : integer := 0;

```

```

begin

```

```

--*****Frequency Generator*****--

```

```

    process(clk)

```

```

    begin

```

```

        if rising_edge(clk) then

```

```

            sck_counter <= sck_counter + 1;

```

```

            if sck_counter = 2 then

```

```

                sig_sck <= not sig_sck;

```

```

                sck_counter <= 0;

```

```

            end if;

```

```

        end if;

```

```

    end process;

```

```

    sck <= sig_sck;

```

```

-----
    process(state_counter,sig_sck)

```

```

    begin

```

```

        if falling_edge(sig_sck) then

```

```

            case state_counter is

```

```

                --*****Settling & Transferation & Fetching Period*****--

```

```

                when 0 =>

```

```

                    sig_data_in <= std_logic_vector(data_in);

```

```

                    cs <= '1';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

state_counter <= 1;
--*****Configuration Period*****--
when 1 =>
cs <= '0';
state_counter <= 2;
data_out <= '0'; --Select DACa to work
when 2 =>
state_counter <= 3; --Null Bit
when 3 =>
state_counter <= 4;
data_out <= '1'; --Gain = 1x
when 4 =>
state_counter <= 5;
data_out <= '1'; --Device in active
-----
--*****Pushing Data Period*****--
when 5 =>
state_counter <= 6;
data_out <= sig_data_in(11);
when 6 =>
state_counter <= 7;
data_out <= sig_data_in(10);
when 7 =>
state_counter <= 8;
data_out <= sig_data_in(9);
when 8 =>
state_counter <= 9;
data_out <= sig_data_in(8);
when 9 =>
state_counter <= 10;
data_out <= sig_data_in(7);
when 10 =>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

state_counter <= 11;
data_out <= sig_data_in(6);
when 11 =>
state_counter <= 12;
data_out <= sig_data_in(5);
when 12 =>
state_counter <= 13;
data_out <= sig_data_in(4);
when 13 =>
state_counter <= 14;
data_out <= sig_data_in(3);
when 14 =>
state_counter <= 15;
data_out <= sig_data_in(2);
when 15 =>
state_counter <= 16;
data_out <= sig_data_in(1);
when 16 =>
state_counter <= 0;
data_out <= sig_data_in(0);
-----
-- --*****Transferation Data Period*****--
-- when 17 =>
-- state_counter <= 1;
-- cs <= '1';
-- -----

when others => --for safety reason
state_counter <= 1;
end case;
end if;
end process;
end Behavioral;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----
```

```
---*****ADC 12 bits. SPI interface*****---
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity SPI_INTERFACE_IN is
port( cs : buffer std_logic;
      clk : in std_logic;
      -- spi_clk_in : in std_logic; --Generated from DCM.
      spi_clk_out : out std_logic; --Toward ADC.
      data_out : out signed(11 downto 0);
      d_in : in std_logic);
end SPI_INTERFACE_IN;

architecture Behavioral of SPI_INTERFACE_IN is
signal d_0 : std_logic;
signal d_1 : std_logic;
signal d_2 : std_logic;
signal d_3 : std_logic;
signal d_4 : std_logic;
signal d_5 : std_logic;
signal d_6 : std_logic;
signal d_7 : std_logic;
signal d_8 : std_logic;
signal d_9 : std_logic;
signal d_10 : std_logic;
signal d_11 : std_logic;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

signal pre_state_counter : integer := 0;
-- signal state_counter : integer := 0;

signal done_time : integer := 0;
signal pre_out : signed(11 downto 0);

```

```

signal counter : integer := 0;
signal sig_spi : std_logic := '0';

```

```
begin
```

```
spi_clk_out <= sig_spi;
```

```
--*****Frequency Generator (800 kHz.)*****--
```

```
process(clk)
```

```
begin
```

```
if rising_edge(clk) then
```

```
counter <= counter + 1;
```

```
if counter = 625 then
```

```
counter <= 0;
```

```
sig_spi <= not sig_spi;
```

```
end if;
```

```
end if;
```

```
end process;
```

```
-----
```

```
process(pre_state_counter,sig_spi)
```

```
begin
```

```
if rising_edge(sig_spi) then
```

```
case pre_state_counter is
```

```
--**Hold CS High when first power on**--
```

```
when 0 =>
```

```
cs <= '1';
```

```
done_time <= done_time + 1;
```

```
if done_time = 2 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cs <= '0';
done_time <= 0;
pre_state_counter <= 1;
end if;
--*****Sample Period*****--
when 1 =>
pre_state_counter <= 2;
when 2 =>
pre_state_counter <= 3;
-----
--*****Null bit period*****--
when 3 =>
pre_state_counter <= 4;
-----
--*****Data periode*****--
when 4 =>
d_11 <= d_in;
pre_state_counter <= 5;
when 5 =>
d_10 <= d_in;
pre_state_counter <= 6;
when 6 =>
d_9 <= d_in;
pre_state_counter <= 7;
when 7 =>
d_8 <= d_in;
pre_state_counter <= 8;
when 8 =>
d_7 <= d_in;
pre_state_counter <= 9;
when 9 =>
d_6 <= d_in;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pre_state_counter <= 10;
when 10 =>
d_5 <= d_in;
pre_state_counter <= 11;
when 11 =>
d_4 <= d_in;
pre_state_counter <= 12;
when 12 =>
d_3 <= d_in;
pre_state_counter <= 13;
when 13 =>
d_2 <= d_in;
pre_state_counter <= 14;
when 14 =>
d_1 <= d_in;
pre_state_counter <= 15;
when 15 =>
d_0 <= d_in;
cs <= '1'; --Hold Tcsh
pre_state_counter <= 16;
-----
when 16 =>
cs <= '0'; --Release CS begin new conversion
pre_state_counter <= 1;
pre_out <= conv_signed((d_11 & d_10 & d_9 & d_8 & d_7 & d_6 & d_5 & d_4 & d_3 & d_2 & d_1
& d_0),12);
-----
when others =>
pre_state_counter <= 0; --If some error occur go back to first state.
end case;
end if;
end process;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
data_out <= pre_out;
```

```
end Behavioral;
```

```
__*****_
```

```
-----FIR for bandpass filter (Kaiser Window)-----
```

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
use ieee.std_logic_arith.all;
```

```
use ieee.std_logic_signed.all;
```

```
entity bandpass is
```

```
generic( tap : natural := 51;
```

```
adc_bit : natural := 12;
```

```
coeff_resolution : natural := 14);
```

```
port( clk : in std_logic;
```

```
trap_in : out signed(11 downto 8);
```

```
trap_out : out signed(11 downto 8);
```

```
filter_in : in signed(adc_bit-1 downto 0);
```

```
output : out signed(adc_bit-1 downto 0));
```

```
end bandpass;
```

```
architecture behave of bandpass is
```

```
signal counter : integer := 0;
```

```
signal samp_sig : std_logic := '0';
```

```
signal samp : std_logic := '0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

type coeff_table is array (tap-1 downto 0) of real;

constant coeff : coeff_table :=(

```

    -0.01083012402564, -0.01332712035711, -0.007630739519343, -1.496427457416e-018,
    0.002905066918429, -3.3907946898e-018, -0.003236367683309, 6.081192478288e-018,
    0.01057201568454, 0.02063393926443, 0.01877946723007, -1.385578761212e-017,
    -0.02745418170286, -0.04500617679373, -0.03613552324226, 1.746072603218e-017,
    0.0443381657561, 0.06803780125321, 0.0515864185953, -3.752992180704e-017,
    -0.05744871901905, -0.08449257609722, -0.06156947219718, -2.509780446386e-017,
    0.06369560307819, 0.09045946500768, 0.06369560307819, -2.509780446386e-017,
    -0.06156947219718, -0.08449257609722, -0.05744871901905, -3.752992180704e-017,
    0.0515864185953, 0.06803780125321, 0.0443381657561, 1.746072603218e-017,
    -0.03613552324226, -0.04500617679373, -0.02745418170286, -1.385578761212e-017,
    0.01877946723007, 0.02063393926443, 0.01057201568454, 6.081192478288e-018,
    -0.003236367683309, -3.3907946898e-018, 0.002905066918429, -1.496427457416e-018,
    -0.007630739519343, -0.01332712035711, -0.01083012402564

```

);

type mul_raw_table is array (tap-1 downto 0) of signed((adc_bit+coeff_resolution) downto 0); --create a temp multiplication result

signal mul_raw : mul_raw_table;

type sum_table is array (tap downto 0) of signed(adc_bit+coeff_resolution downto 0);

signal add_result : sum_table;

type input_table is array (tap-1 downto 0) of signed(adc_bit downto 0);

signal input_pipeline : input_table;

constant offset : signed(12 downto 0) := "001111111111"; --Assume that input shifted by 1.65V. (Max at 3.3V)

constant offset2 : signed(12 downto 0) := "001111111111"; --Assume that we able to shift output by 1.65V.

signal pre_out : signed(adc_bit downto 0);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
--*****Frequency Generator (50 kHz.)*****--

process(clk)
begin
if clk'event and clk = '1' then
if counter = 10000 then
samp_sig <= not samp_sig;
counter <= 0;
else
counter <= counter + 1;
end if;
end if;
end process;
samp <= samp_sig;
-----
process(samp)
begin
if samp'event and samp = '1' then
input_pipeline(0) <= ('0' & filter_in) - offset;
input_pipeline(tap-1 downto 1) <= input_pipeline(tap-2 downto 0);
end if;
end process;

-- trap_in <= input_pipeline(0)(11 downto 8);
-- trap_out <= pre_out(adc_bit-1 downto 8);
-- trap_in <= add_result(tap)(26 downto 23);
-- trap_out <= add_result(tap)(22 downto 19);
trap_in <= filter_in(11 downto 8);
trap_out <= filter_in(7 downto 4);

multiplication : for i in tap-1 downto 0 generate
mul_raw(i) <= input_pipeline(i) * conv_signed(integer(coeff(i)*16384.0),coeff_resolution);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end generate;

summation : for i in 0 to tap-1 generate

add_result(i+1) <= mul_raw(i) + add_result(i);

end generate;

pre_out <= add_result(tap)((adc_bit+coeff_resolution) downto coeff_resolution) + offset2;

output <= pre_out(adc_bit-1 downto 0);

-- output <= offset2(11 downto 0);

-- output <= add_result(tap)(26 downto 15);

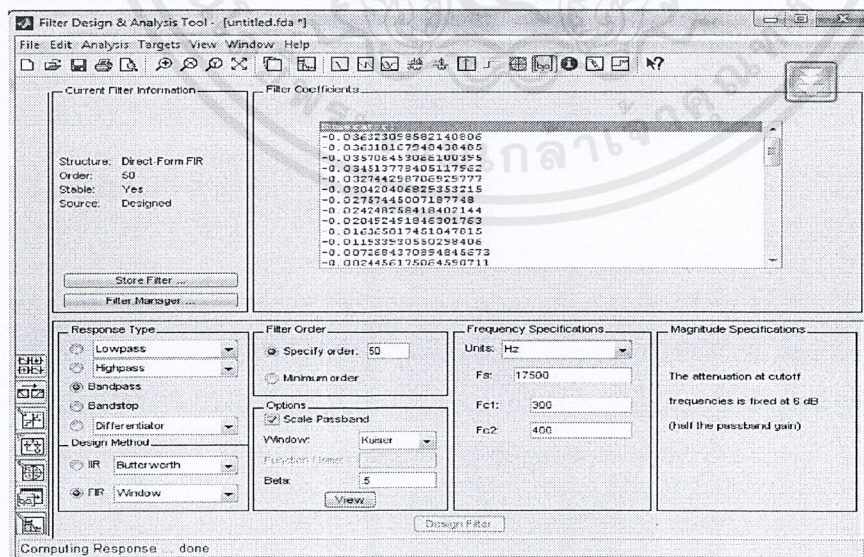
-- output <= filter_in;

end behave;

--

*****_

ในส่วนของสัมประสิทธิ์ที่เราได้มาจาก เครื่องมือ FDATool ในโปรแกรม MATLAB โดยหน้าต่างของที่เราออกแบบเป็นดังนี้



รูปที่ 3.2 รูปแสดงหน้าต่างเครื่องมือ FDATool จากโปรแกรม MATLAB ที่ใช้หาสัมประสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 การแปลงสัญญาณดิจิทัลเป็นอนาล็อก (DAC)

เมื่อเราทำการกรองสัญญาณผ่านโปรแกรม FIR แล้วเรานำสัญญาณดิจิทัลที่ได้นำกลับมาแปลงเป็นสัญญาณอนาล็อกอีกครั้ง โดยที่นี้ใช้ ไอซีเบอร์ MCP 4821 ขนาด 12 bit

3.1.7 ส่วนของการตัดไฟตรง

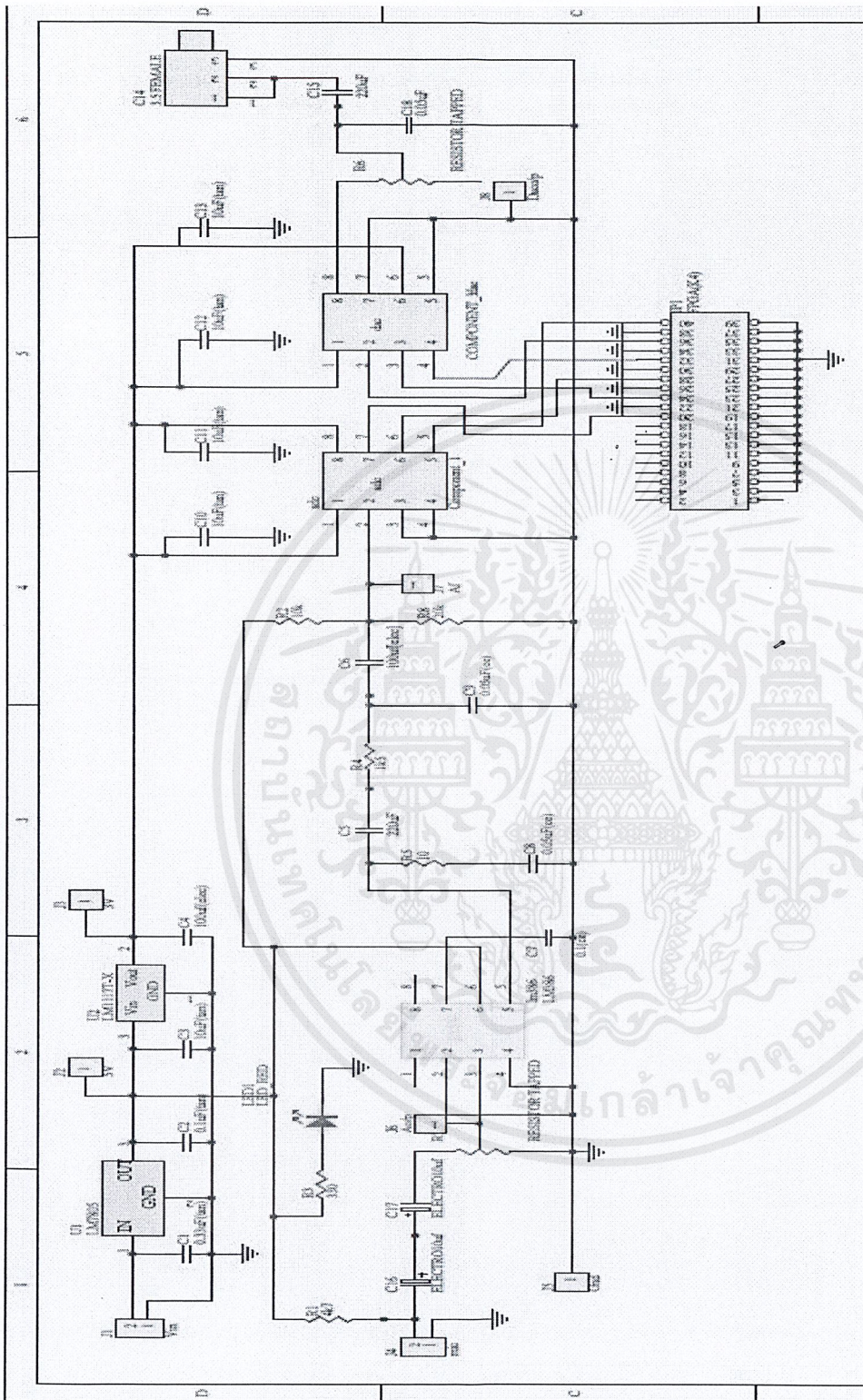
ใช้ตัวเก็บประจุต่ออนุกรมกับสายสัญญาณเพื่อเป็นการกรองส่วนสัญญาณไฟตรงที่ได้ทำการบวกรวมเอาไว้จากตัว FPGA เพื่อกันส่วน Speaker ไม่ให้เสียหาย

3.1.8 Speaker

เมื่อเสดกระบวนการทั้งหมดแล้วเอาพูทที่ได้ก็จะนำไปออกลำโพง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



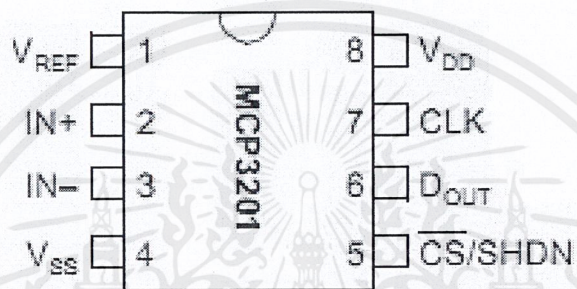
รูปที่ 3.3 รูปแสดงวงจรที่เรากออกแบบในส่วนวงจรขยาย, ปรับ Offset, ADC, DAC, Low pass filter และ ส่วนตัดไฟตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อุปกรณ์ที่ใช้ในการออกแบบ

3.2.1 ADC (MCP3201)

- SPI Interface
- ขนาด 12 bit
- ที่ไฟเลี้ยงขนาด 5 โวลต์ มีSampling Rate 100ksp/s
- กระแสสูงสุด 400uA ที่แรงดัน 5 โวลต์

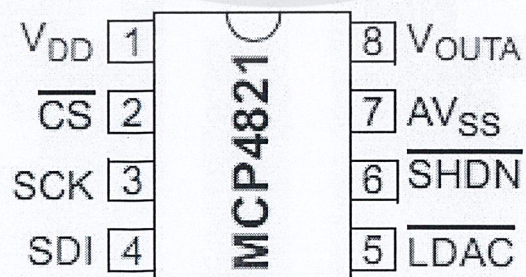


รูปที่ 3.4 รูปแสดง IC เบอร์ MCP3201

3.2.2 DAC (DAC4821)

- SPI Interface
- ขนาด 12 bit
- ทำงานที่แรงดัน 2.7-5.5 โวลต์

8-Pin PDIP, SOIC, MSOP



รูปที่ 3.5 รูปแสดง IC เบอร์ MCP4821

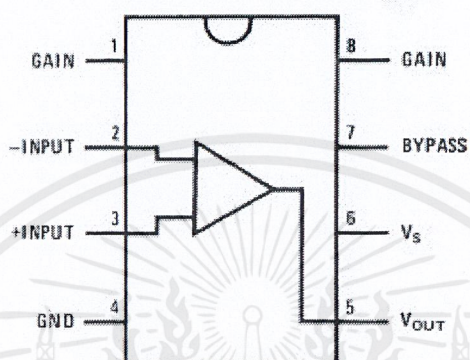
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ออปแอมป์ (LM386)

-Low Voltage Audio Amplifier

-Voltage Gain 20 ถึง 200

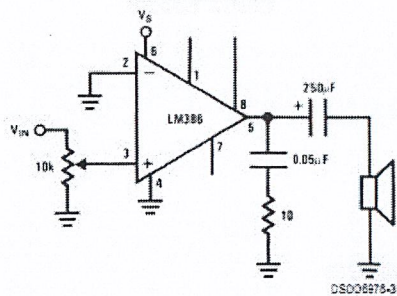
Small Outline,
Molded Mini Small Outline,
and Dual-In-Line Packages



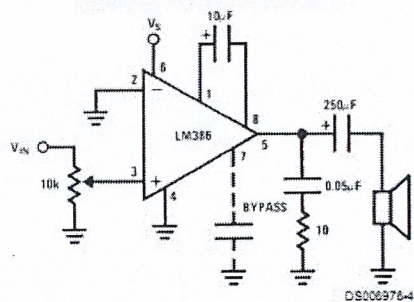
รูปที่ 3.6 รูปแสดง IC เบอร์ LM386

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

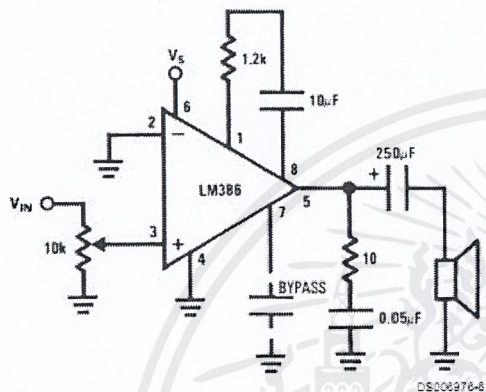
Amplifier with Gain = 20
Minimum Parts



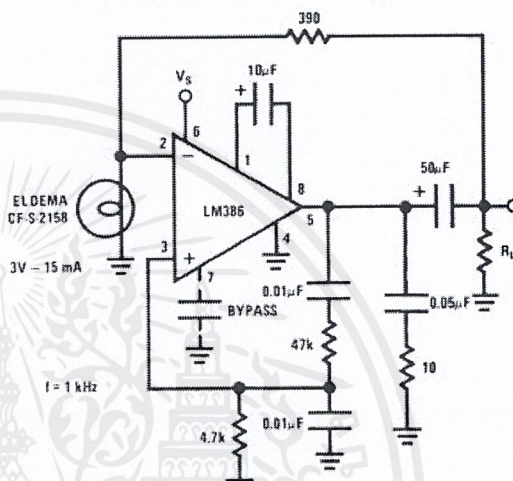
Amplifier with Gain = 200



Amplifier with Gain = 50



Low Distortion Power Wienbridge Oscillator

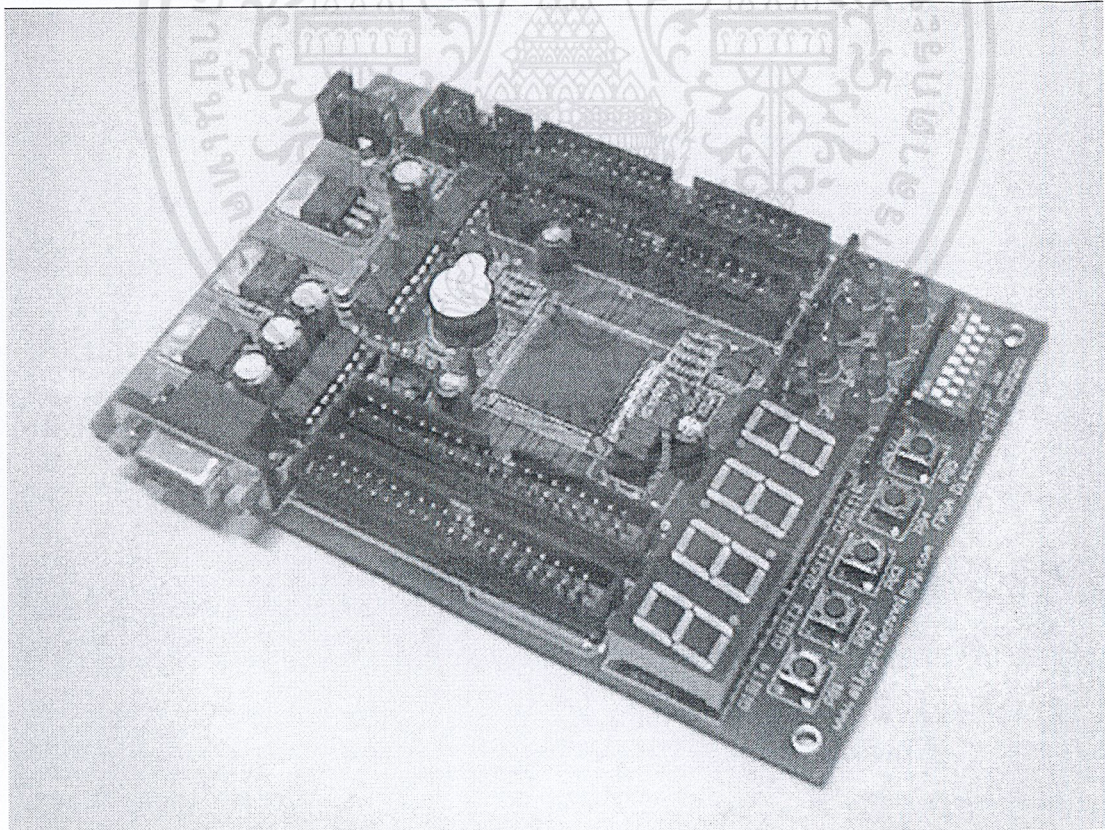


รูปที่ 3.7 รูปแสดงการนำ LM386 ไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 FPGA(Discovery-III XC3S200F4)

- 7-Segment จำนวนสี่หลัก (ใช้ร่วมกับ Expansion ports และสามารถถอดออกได้)
- LED จำนวน 8 ดวง
- Buzzer จำนวน 1 ตัว
- DIP Switch 8 บิต
- Push Button Switch จำนวน 5 ตัว
- Expansion ports (80 บิต 3.3V. I/O)
- RS-232C PORT 1 Port
- I2C Socket สำหรับ EEPROM
- 25 MHz Oscillator
- ความจุวงจร 40000 เกต
- 18Kb Block RAM จำนวน 16 ชุด (288Kb)
- 18×18 Hardware Multiplier จำนวน 16 ชุด
- Digital Clock Manager (DCM) จำนวน 4 ชุด



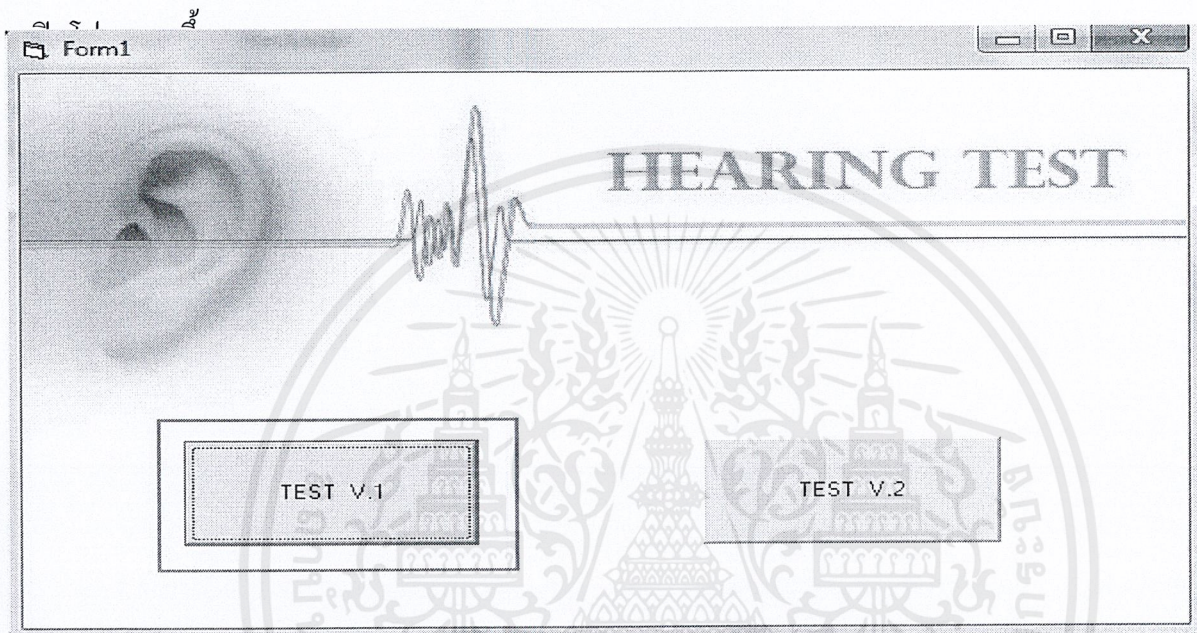
รูปที่ 3.8 รูปแสดงบอร์ดทดลองเอนกประสงค์ FPGA รุ่น **Discovery-III XC3S200F4**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

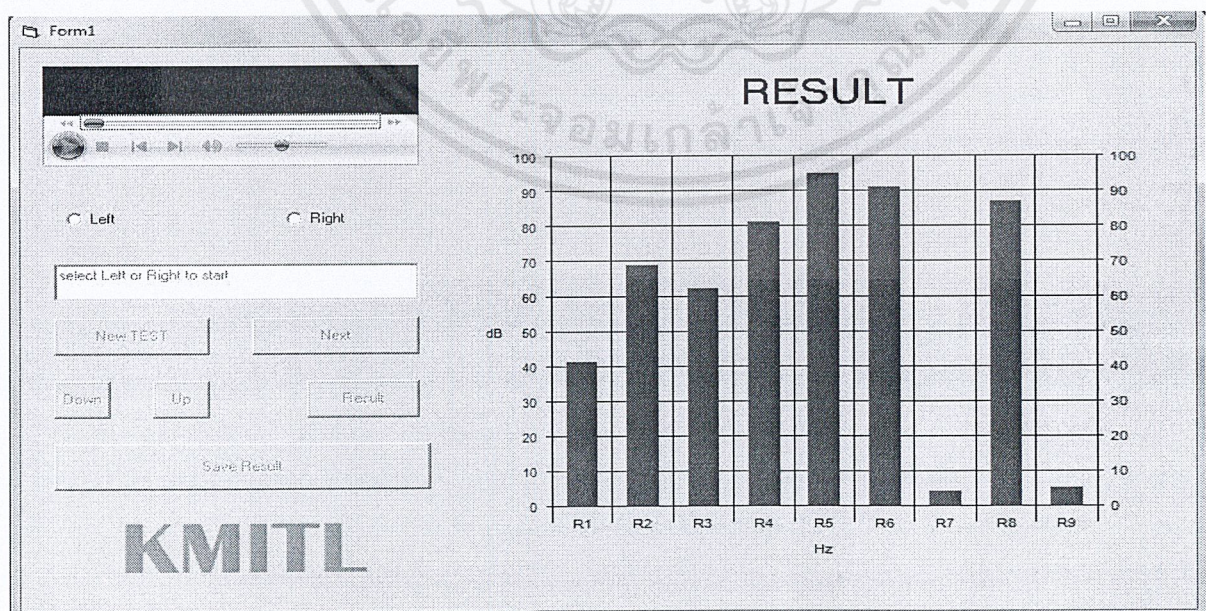
3.3 การทำงานของโปรแกรมทดสอบหูอย่างง่าย

3.3.1 โปรแกรมทดสอบหูอย่างง่าย version 1

โปรแกรมนี้ได้ออกแบบเพื่อทดสอบการได้ยินของคนเราว่าบกพร่องมากน้อยประการใด โดยได้ทำการออกแบบโปรแกรมแบบคร่าวๆ โดยที่โปรแกรมมีการทำงานดังนี้

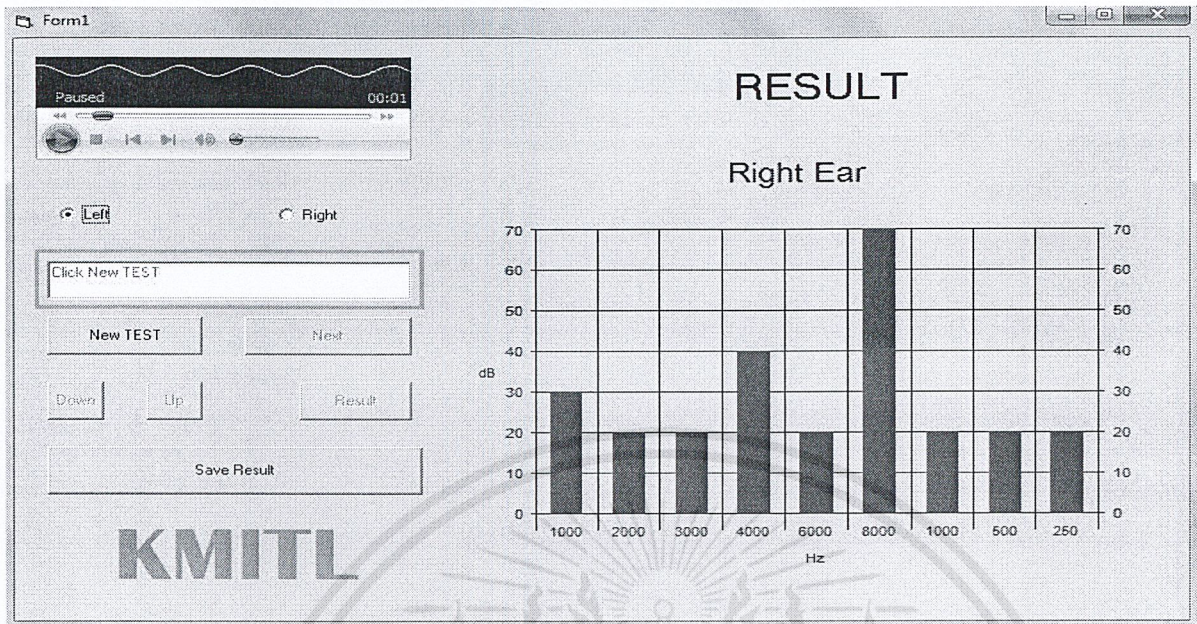


รูปที่ 3.9 รูปแสดงขั้นตอนการเลือก version ของโปรแกรมทดสอบในที่นี้เลือก version ที่ 1



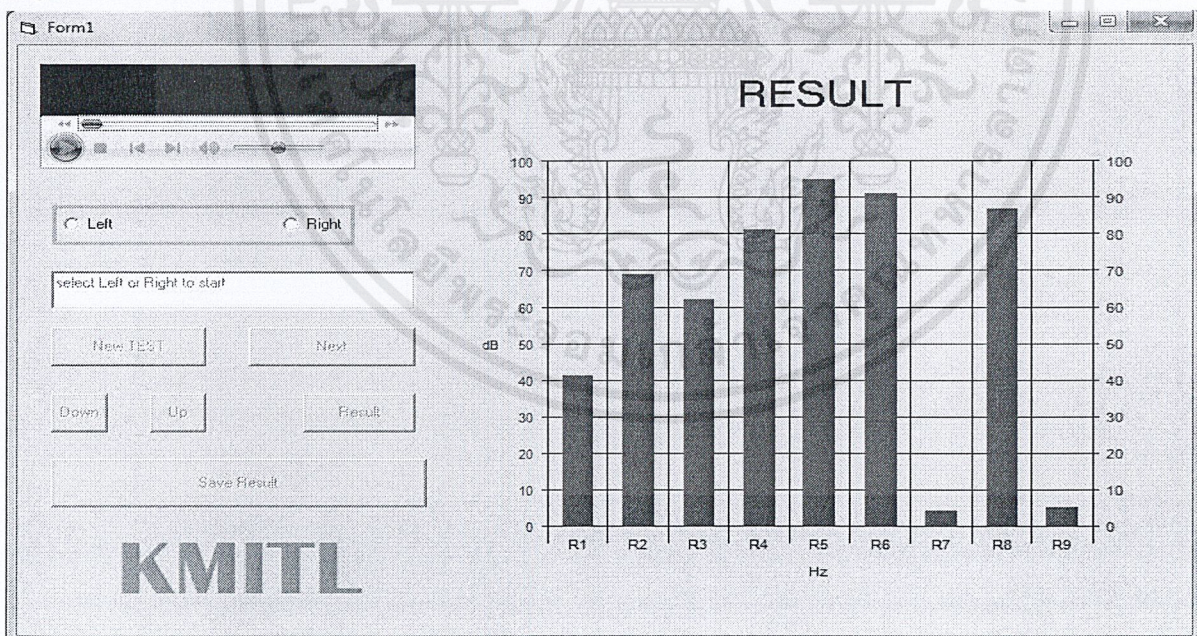
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 โปรแกรมที่เราจะใช้งาน



รูปที่ 3.11 แถบแสดงการทำงาน

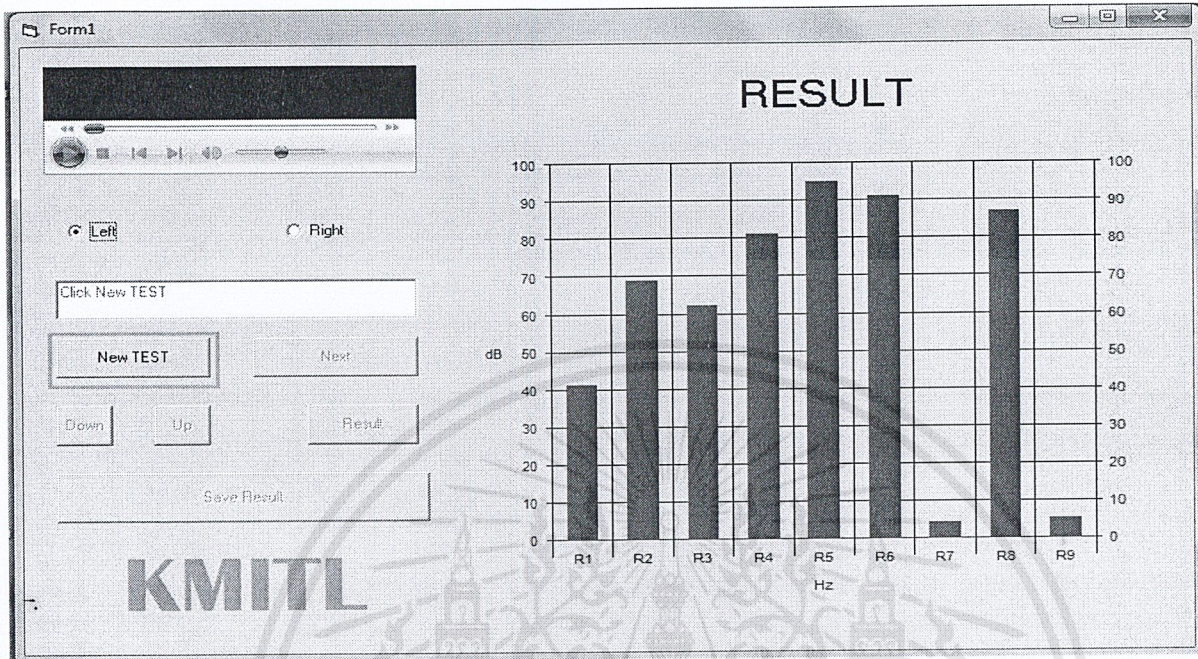
2.เมื่อเราจะเริ่มการทดสอบให้เลือกทดสอบหูซ้ายหรือหูขวาข้างใดข้างหนึ่งก่อน โดยกดที่ปุ่ม Left หรือ Right



รูปที่ 3.12 ปุ่ม Left และ Right

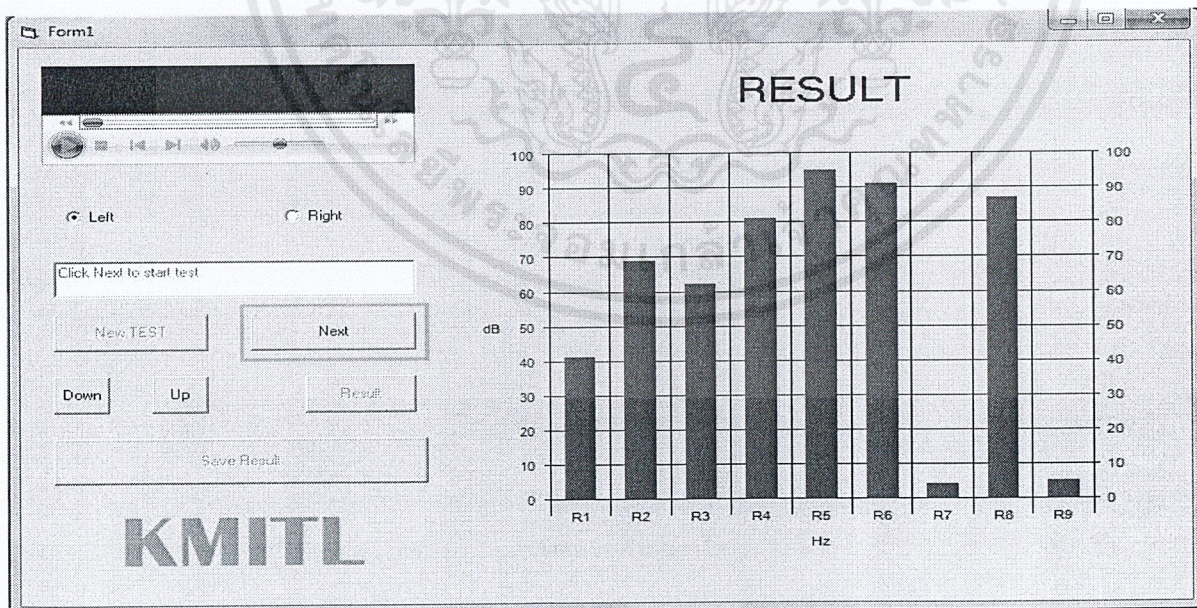
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้นกดปุ่ม New TEST เพื่อเริ่มการทดสอบ



รูปที่ 3.13 ปุ่ม New TEST

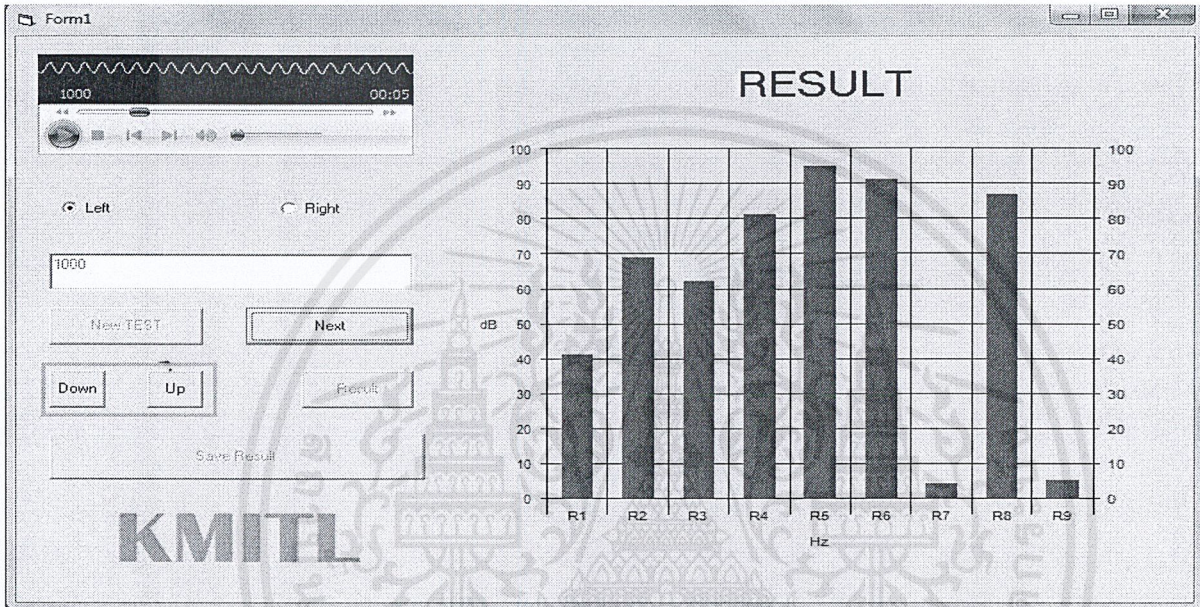
4. กดปุ่ม Next เพื่อเริ่มที่ความถี่แรกที่เราจะทดสอบ



รูปที่ 3.14 ปุ่ม Next

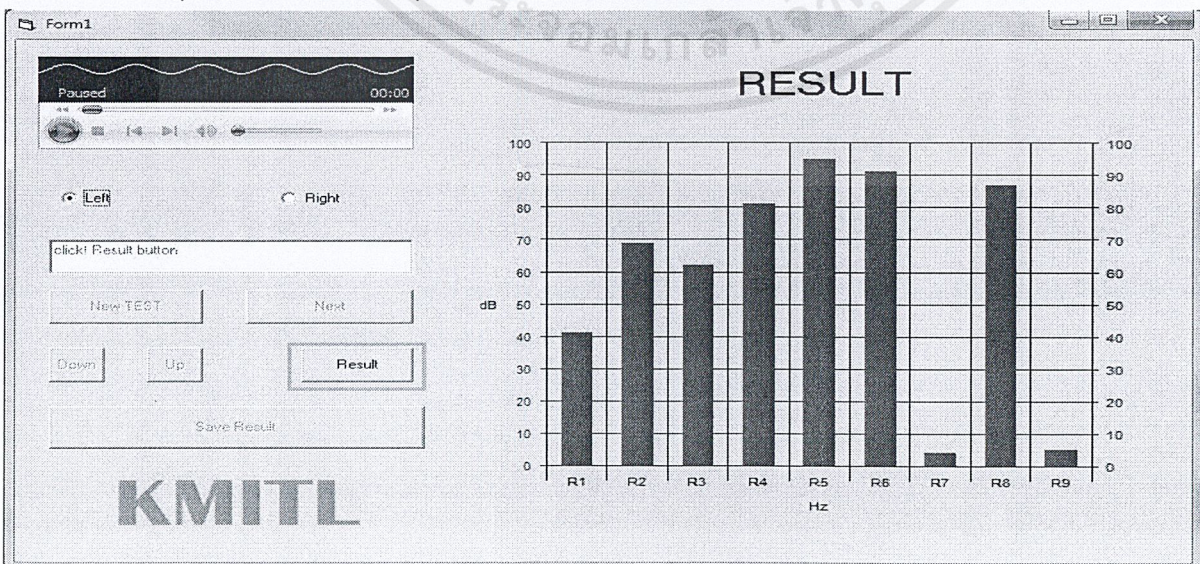
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้น ให้กดปุ่ม Up เพื่อเพิ่มระดับเสียงและกดปุ่ม Down เพื่อลดระดับเสียง ในการทดสอบนั้นเพิ่มระดับเสียงที่ดังก่อน แล้วค่อยลดระดับเสียงลงทีละ 10 เดซิเบล จนผู้ทดสอบไม่ได้ยินแล้วจึงค่อยๆเพิ่มขึ้นทีละ 5 เดซิเบล จนผู้ทดสอบได้ยินจากนั้นจึงลดลงอีกทีละ 10 เดซิเบล ทำซ้ำไปซ้ำมาจนแน่ใจว่าผู้ทดสอบแน่ใจว่าได้ยินที่จุดนั้น



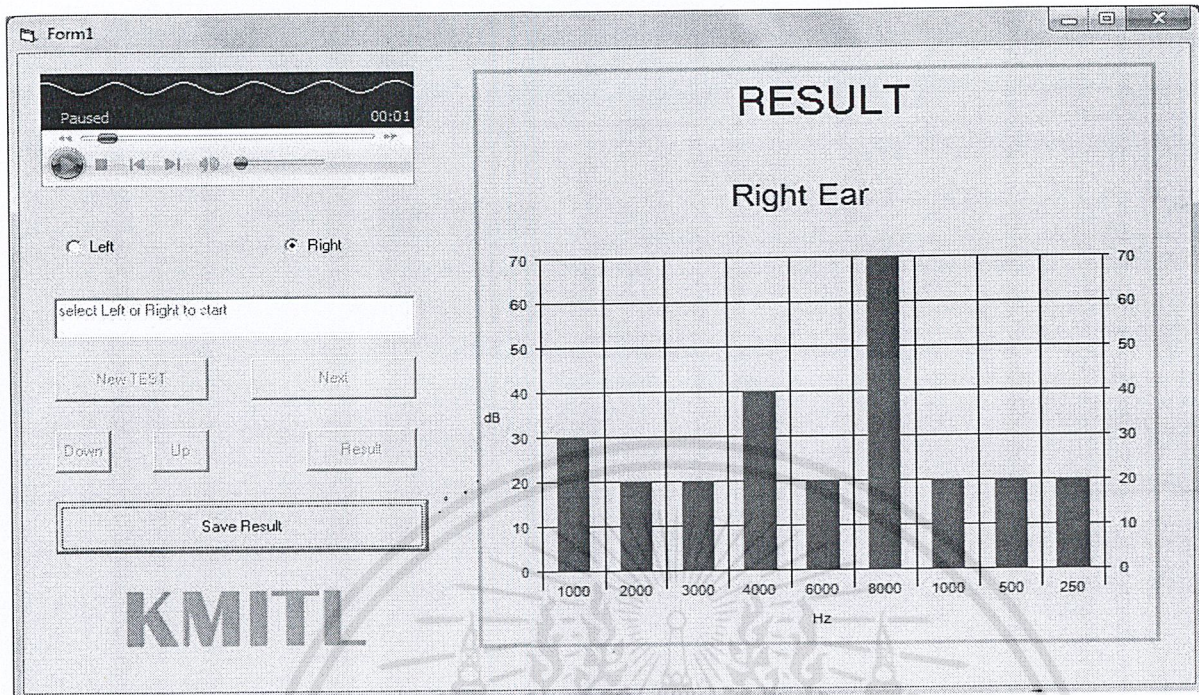
รูปที่ 3.15 ปุ่ม Up และ Down

6. จากนั้นกดปุ่ม Next อีก 1 ครั้ง เพื่อเลื่อนไปความถี่ต่อไปที่จะทดสอบ แล้วทำเช่นเดียวกับขั้นตอนที่ 5 เมื่อทำงานครบ ทุกความถี่แล้ว ให้กดปุ่ม Result เพื่อให้ได้ผลการทดสอบออกมาเป็นกราฟแท่งทางด้านขวา



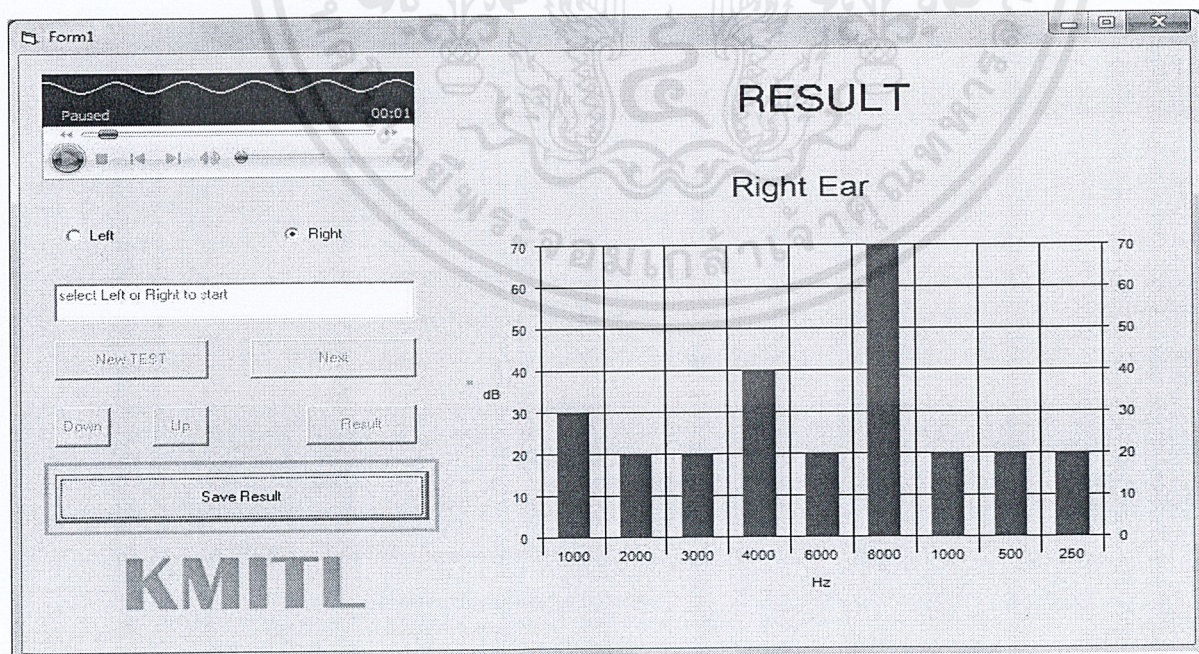
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.16 ปุ่ม Result



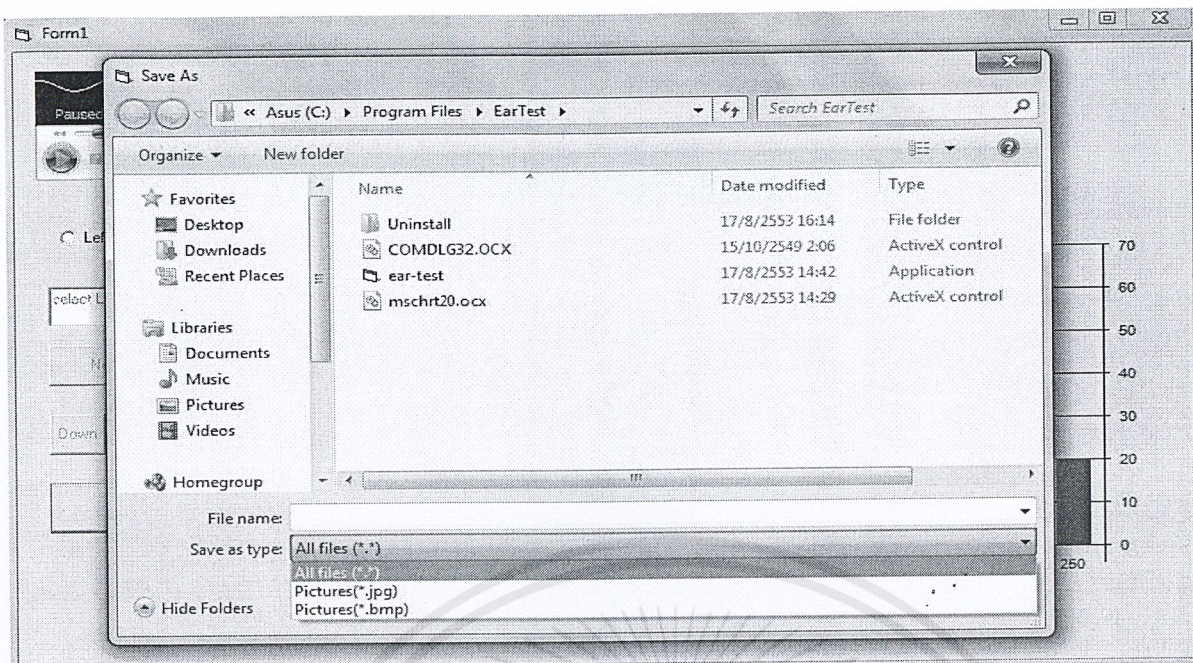
รูปที่ 3.17 กราฟที่ได้ออกมา

7. เราสามารถ Save ผลการทดสอบได้โดยกดที่ปุ่ม Save result เมื่อ save ผลแล้ว ให้ทำการทดสอบกับหูอีกข้างหนึ่งด้วยเทคนิคเดียวกัน

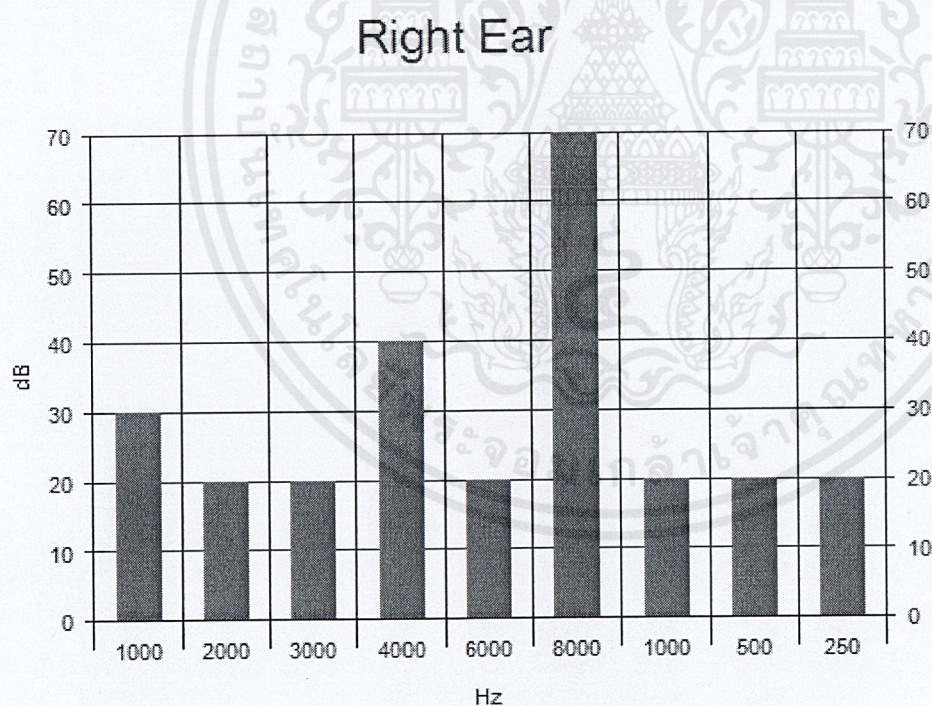


รูปที่ 3.18 ปุ่ม Save Result

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 เมื่อเรากดปุ่ม Save Result จะได้นหน้าต่างแบบนี้ เราสามารถเลือกเซฟเป็นไฟล์ต่างๆได้



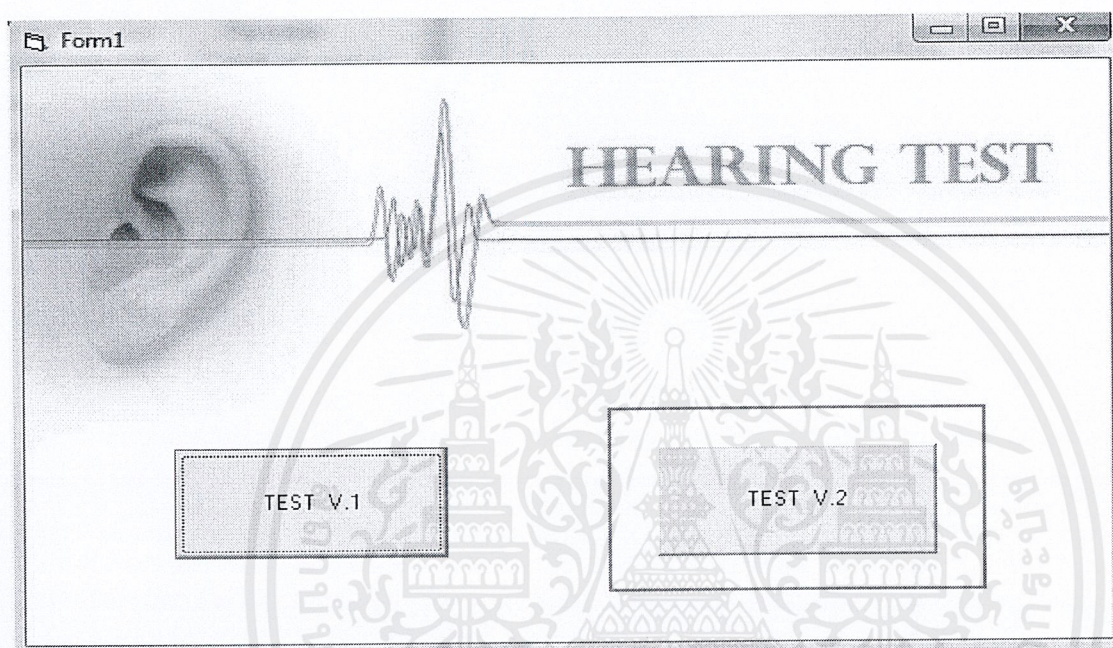
รูปที่ 3.20 รูปกราฟที่ได้จากการเซฟผลการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

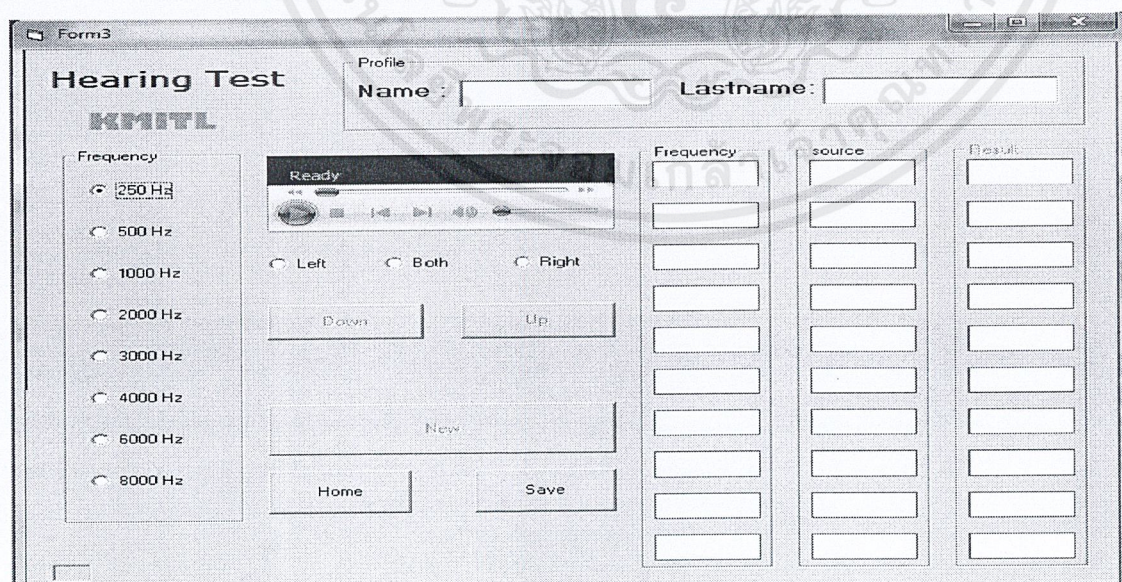
3.3.2 โปรแกรมทดสอบหูอย่างง่าย version 2

ส่วนของโปรแกรมที่ 2 เป็นแกรมที่สามารถใช้ทดสอบว่าผู้ที่ทดสอบนั้น มีการตอบสนองทางการได้ยินของหูถูกต้องหรือไม่ (ข้างซ้ายหรือข้างขวา) ขั้นตอนการใช้งานมีดังนี้

1.เปิดโปรแกรมขึ้นมา



รูปที่ 3.21 รูปแสดงขั้นตอนการเลือก version ของโปรแกรมทดสอบในที่นี้เลือก version 2



รูปที่ 3.22 รูปแสดงตัวโปรแกรม version ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ใส่ชื่อผู้ทดสอบ

The screenshot shows a software window titled 'Form3' with the heading 'Hearing Test KMITL'. At the top right, there is a 'Profile' section with two input fields: 'Name : []' and 'Lastname: []'. Below this, on the left, is a 'Frequency' list with radio buttons for 250 Hz (selected), 500 Hz, 1000 Hz, 2000 Hz, 3000 Hz, 4000 Hz, 6000 Hz, and 8000 Hz. In the center is a media player interface with a 'Ready' status, a progress bar, and playback controls. Below the media player are radio buttons for 'Left', 'Both', and 'Right', and buttons for 'Down', 'Up', 'New', 'Home', and 'Save'. On the right side, there are three columns of input fields labeled 'Frequency', 'source', and 'Result', each with 10 empty rows for data entry.

รูปที่ 3.23 รูปแสดงแถบใส่ชื่อผู้ทดสอบ

3. เลือกความถี่ ที่ช่องความถี่ด้านซ้าย

This screenshot is identical to the one above, but it highlights the 'Frequency' list on the left side of the interface. The radio button next to '250 Hz' is selected, while the others are unselected. The rest of the interface, including the name fields, media player, and data entry columns, remains the same.

รูปที่ 3.24 รูปแสดงรูปแถบความถี่ที่เราจะเลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.เลือกว่าจะทดสอบผู้ฟังข้างใด

รูปที่ 3.25 รูปแสดงแถบที่ใช้เลือกว่าจะทดสอบหูข้างใด

5.จากนั้นค่อยๆเพิ่มความถี่ จากแถบในโปรแกรม เมื่อผู้ทดสอบได้ยินหูข้างขวาให้กด ปุ่ม Z บนคีย์บอร์ด และถ้าเป็นหูข้างซ้ายให้กดปุ่ม / บนคีย์บอร์ด แต่ถ้าได้ยินทั้งสองข้างให้กด ปุ่ม B บนคีย์บอร์ด

รูปที่ 3.26 รูปแสดงแถบที่ใช้เพิ่มความดังของเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อทดสอบจนครบทุกความถี่แล้ว ถ้าหากต้องการเซฟผลการทดสอบให้กดปุ่ม Save เมื่อเซฟผลการทดสอบ (ผลการทดสอบจะออกมาเป็น ไฟล์ภาพ) แล้วเราจะทดสอบคนต่อไปให้กดปุ่ม New หรือถ้าเราต้องการที่จะทดสอบในแกรม version 1 ให้กดปุ่ม Home เพื่อเลือก version ที่เราจะใช้ในการทดสอบ

The screenshot shows the 'Hearing Test' software interface. At the top, there is a 'Profile' section with 'Name' and 'Lastname' input fields. Below this is a 'Frequency' selection panel with radio buttons for 250 Hz (selected), 500 Hz, 1000 Hz, 2000 Hz, 3000 Hz, 4000 Hz, 6000 Hz, and 8000 Hz. A central audio player shows 'Ready' and a volume slider. Below the audio player are radio buttons for 'Left', 'Both', and 'Right', and 'Down' and 'Up' buttons. At the bottom, there are 'Home', 'New', and 'Save' buttons. The 'Save' button is highlighted with a red box. To the right, there is a table with columns for 'Frequency', 'source', and 'Result', each with 10 empty rows.

รูปที่ 3.27 รูปแสดงปุ่มที่กดเพื่อเซฟผลการทดสอบ

This screenshot is identical to the previous one, showing the 'Hearing Test' software interface. In this version, the 'New' button is highlighted with a red box, indicating the action to start a new test.

รูปที่ 3.28 รูปแสดงปุ่ม new ที่ใช้กดเพื่อเริ่มการทดสอบใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.29 รูปแสดงปุ่ม Home ใช้กดเมื่อต้องการที่จะใช้โปรแกรมทดสอบ version อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง

4.1 การทดลอง digital filter จากโปรแกรม matlab

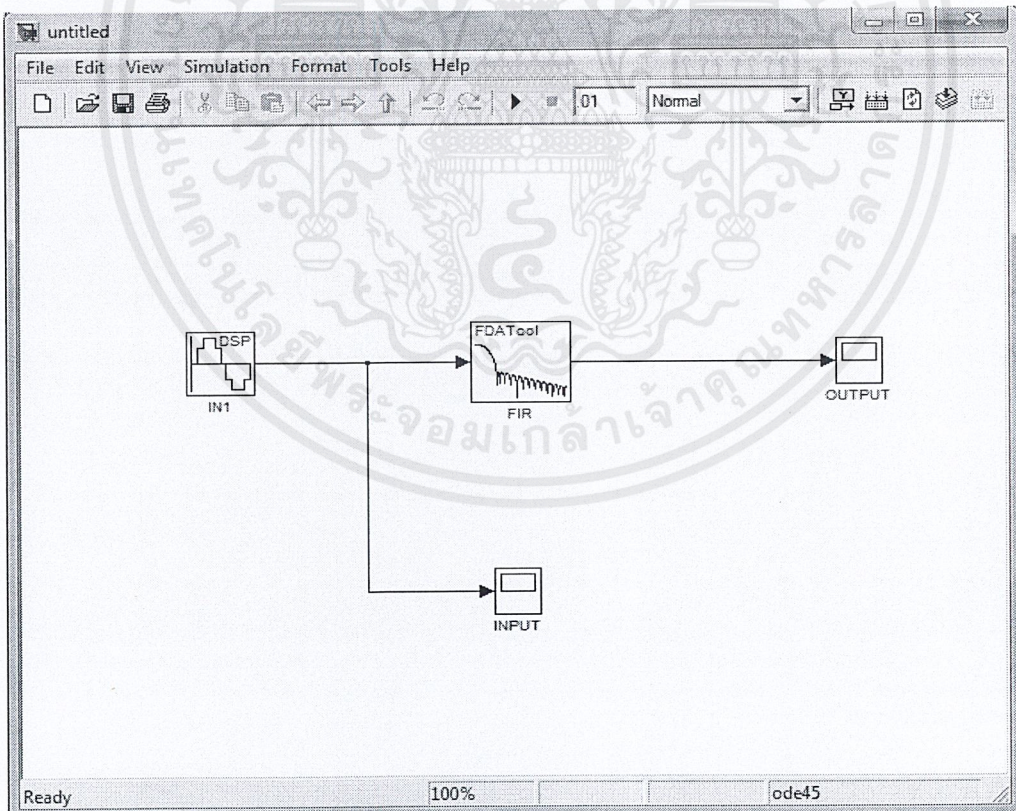
การทดลองเราได้ใช้โปรแกรม matlab เป็นอุปกรณ์ในการทดสอบ โดยใช้ฟังก์ชัน simulink เราได้ป้อน Input เป็นสัญญาณ sine wave ที่ความถี่ต่างๆแล้วดู Output ที่ผ่าน digital filter

วัตถุประสงค์

เพื่อศึกษาการทำงานของดิจิทัลฟิลเตอร์ที่ออกแบบไว้สามารถใช้เป็นตัวกรองสัญญาณได้

ขั้นตอนการทดลอง

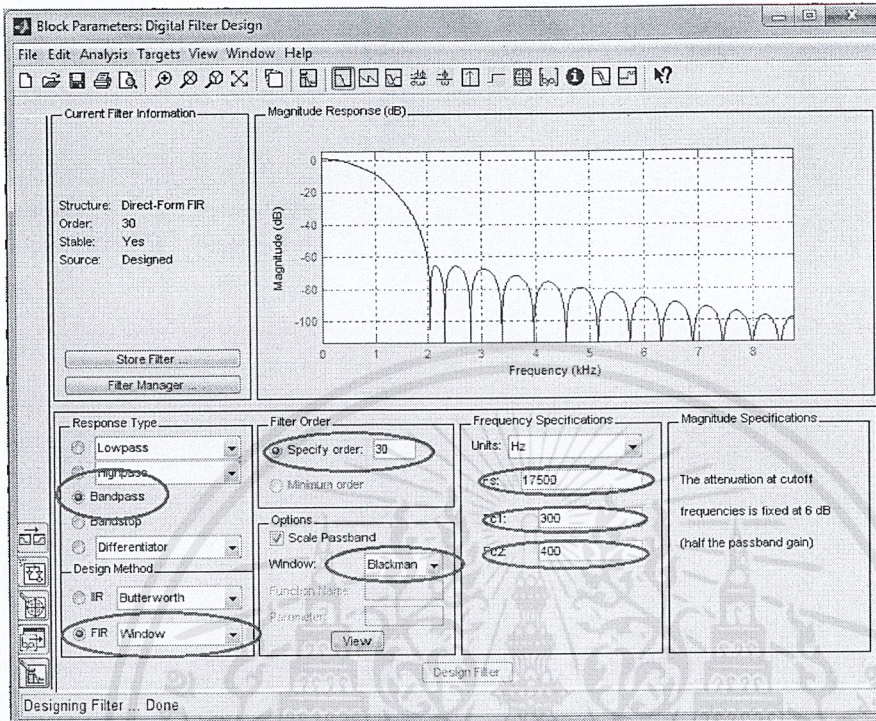
1. เปิดโปรแกรม Matlab แล้วพิมพ์ simulink ที่หน้าต่าง command เพื่อรันโปรแกรม simulink ขึ้นมาใช้งาน
2. วาดวงจรดังในรูป



รูปที่ 4.1 แสดงการวาดวงจรบนโปรแกรม Simulink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คลิกที่FIR เพื่อทำการออกแบบดิจิทัลฟิลเตอร์ ใ้ค่าตามรูป



รูปที่ 4.2 แสดงการเปลี่ยนแปลงค่าตัวแปรต่างๆบน โปรแกรม FDAtool

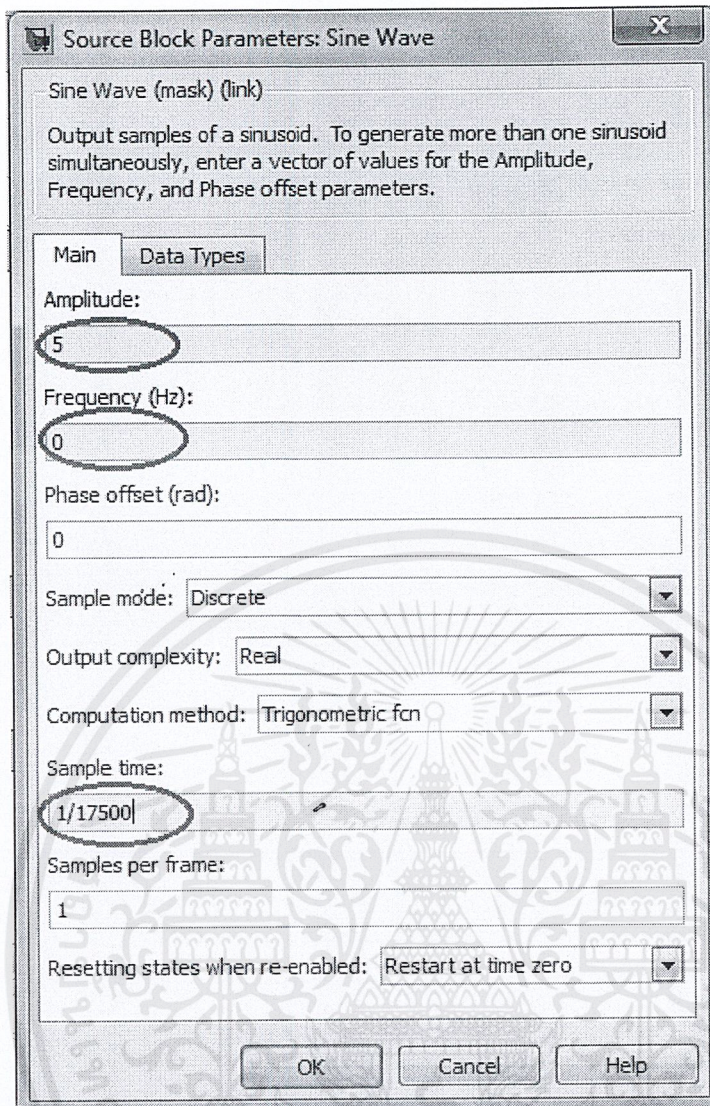
- เลือก Response type เป็น Band Pass
- Design method เลือก FIR Windows
- Filter Order กำหนด Specify order = 30
- Options Window เลือกเป็น Blackman
- Frequency Specifications กำหนด

$$F_s = 17500$$

$$F_{c1} = 300$$

$$F_{c2} = 400$$

คลิกที่IN1 ใ้ค่าตามรูป



รูปที่ 4.3 แสดงการเปลี่ยนแปลงความถี่ที่อินพุต

Amplitude = 5

Frequency (Hz) = 0

Sample time = 1/17500

4. ทำการsimulateคู่สัญญาณที่อินพุตและเอาท์พุต
5. บันทึกผล
6. ทำการทดลองซ้ำ4-6 โดยเปลี่ยนค่า Frequency (Hz) = 0 เป็น 100, 200, 300, 350, 400, 500, 600, 1000, 2000, 4000 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบโปรแกรมทดสอบการได้ยิน

นำโปรแกรมทดสอบการได้ยินทั้งสองแบบไปทดสอบใช้งานจริงกับกลุ่มผู้ทดสอบสิบคน
วัตถุประสงค์

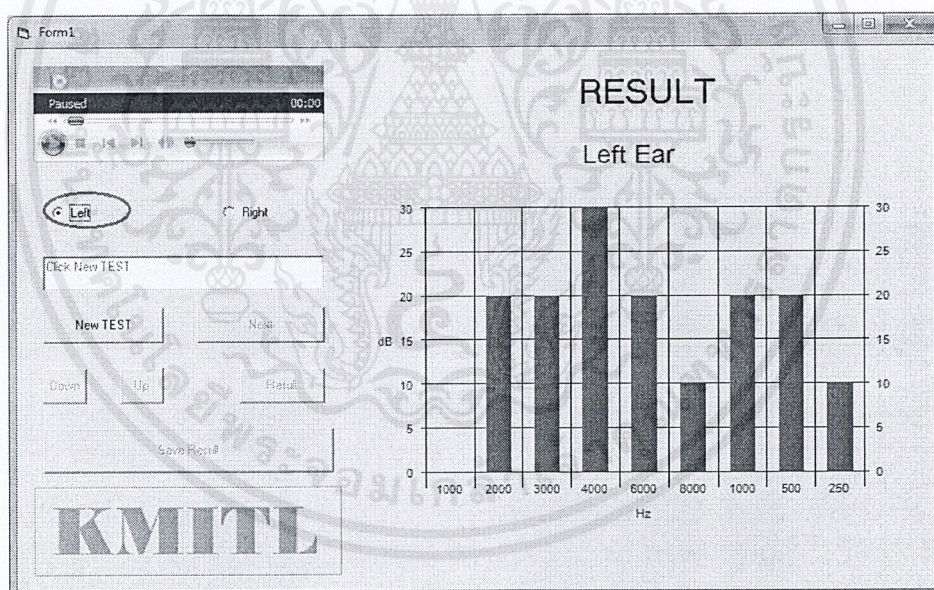
1. เพื่อทดสอบโปรแกรมทดสอบการได้ยินว่าสามารถใช้งาน ได้จริง

อุปกรณ์

1. โปรแกรมทดสอบหูฟัง
2. หูฟัง
3. คีย์บอร์ด

ขั้นตอนการทดสอบ

1. เปิดโปรแกรมทดสอบการได้ยิน
2. เลือกการทดสอบเป็นรูปแบบที่หนึ่ง
 - 2.1 ทำการเลือกหูข้างที่จะทดสอบ(ในรูปเลือกทดสอบหูข้างซ้าย)



รูปที่ 4.4 แสดงการเลือกหูข้างที่จะทดสอบ(ทดสอบหูข้างซ้าย)

2.2 ให้ผู้ทดสอบใส่หูฟัง

2.3 คลิก New Test คลิก Next

2.4 คลิก Up เพื่อเพิ่มระดับความดังเสียงไปจนกว่าผู้ทดสอบจะได้ยิน

2.5 เมื่อผู้ทดสอบ ได้ยินแล้วให้ทำการคลิก Nex เพื่อทดสอบหูที่ความถี่ถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เมื่อทำการทดสอบจนเสร็จแล้ว ให้คลิกเลือกResult เพื่อดูผลการทดสอบสามารถบันทึกผล
การทดสอบได้โดยคลิกSave Result

3. เลือกการทดสอบเป็นรูปแบบที่สอง

3.1 ใส่ชื่อ-สกุลผู้ทดสอบ

3.2 ให้ผู้ทดสอบใส่หูฟัง

3.3 เลือกความถี่ที่ต้องการจะทดสอบ(ในรูปแบบเลือกเป็น 250Hz)

รูปที่ 4.5 แสดงการเลือกความถี่ที่ต้องการทดสอบ (250Hz)

3.4 เลือกช่องทางที่จะให้เสียงออก(ซ้าย ขวา ทั้งสอง)

3.5 เมื่อผู้ทดสอบได้ยินเสียงที่หูข้างไหนให้กดคีย์บอร์ดเพื่อบอกว่าได้ยินข้างไหน

ซ้าย กด Z

ขวา กด /

ทั้งสองข้าง กด b

3.6 เมื่อผู้ทดสอบกดเพื่อเลือกข้างที่ได้ยินแล้วจะปรากฏผลขึ้นมาที่ช่องResult

รูปที่ 4.6 แสดงการแสดงผลของโปรแกรม

3.7 เมื่อทดสอบเสร็จแล้วสามารถบันทึกผลได้โดยการคลิกSave



รูปที่ 4.7 แสดงทดสอบโปรแกรมทดสอบการได้ยิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการใช้โปรแกรม FPGA ที่ใช้กรองสัญญาณ

เราได้นำสัญญาณ sin ขนาดความถี่ต่างๆกันป้อนเข้าสู่ อุปกรณ์แล้วลองวัดเอาพุทที่ได้ว่าเป็นอย่างไร โดยป้อนในช่วงที่ไม่อยู่ในช่วงความถี่ band pass และป้อนในช่วง stop band แล้วดูผลที่ออกมา วัตถุประสงค์

1. เพื่อทดสอบว่าโปรแกรมที่เราเขียนขึ้นมาสามารถใช้งานได้จริง

อุปกรณ์

- 1.บอร์ด FPGA
- 2.Power Supply

ขั้นตอนการทดลอง

- 1.นำบอร์ด FPGA ที่เราทำการเขียนโปรแกรมลงไปบนบอร์ด มาต่อกับวงจรที่เราเตรียมไว้
- 2.ป้อนคลื่น sin ที่ความถี่ต่างๆ
- 3.ถ่ายภาพบันทึกผลการทดลอง
- 4.ทำค่าคำนวณค่า Gain ที่ความถี่ขึ้นต่างๆ

4.4 การทดลองที่ความถี่คัทออฟของฟิลเตอร์

จากฟิลเตอร์ที่เราออกแบบบนโปรแกรม Matlab ทำให้ได้ความถี่คัทออฟที่ 8400 Hz และความถี่

สูงคัทออฟที่ 13200 Hz

วัตถุประสงค์

เพื่อให้ทราบว่าจะวงจรที่เราออกแบบมีความถูกต้องตามที่เรากำหนดหรือไม่

อุปกรณ์

- 1.บอร์ด FPGA
- 2.Power Supply

ขั้นตอนการทดลอง

- 1.นำบอร์ด FPGA ที่เราทำการเขียนโปรแกรมลงไปบนบอร์ด มาต่อกับวงจรที่เราเตรียมไว้
- 2.ป้อนความถี่ในช่วง Band Pass เพื่อหาความถี่ที่ให้เอาพุทมีขนาดสูงสุด
- 3.บันทึกผลที่ได้
- 4.ป้อนความถี่คัทออฟที่เราได้จากโปรแกรม Matlab (8400 Hz)
- 5.บันทึกผลที่ได้
- 6.ป้อนความถี่สูงคัทออฟ (13200 Hz)
- 7.บันทึกผลที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

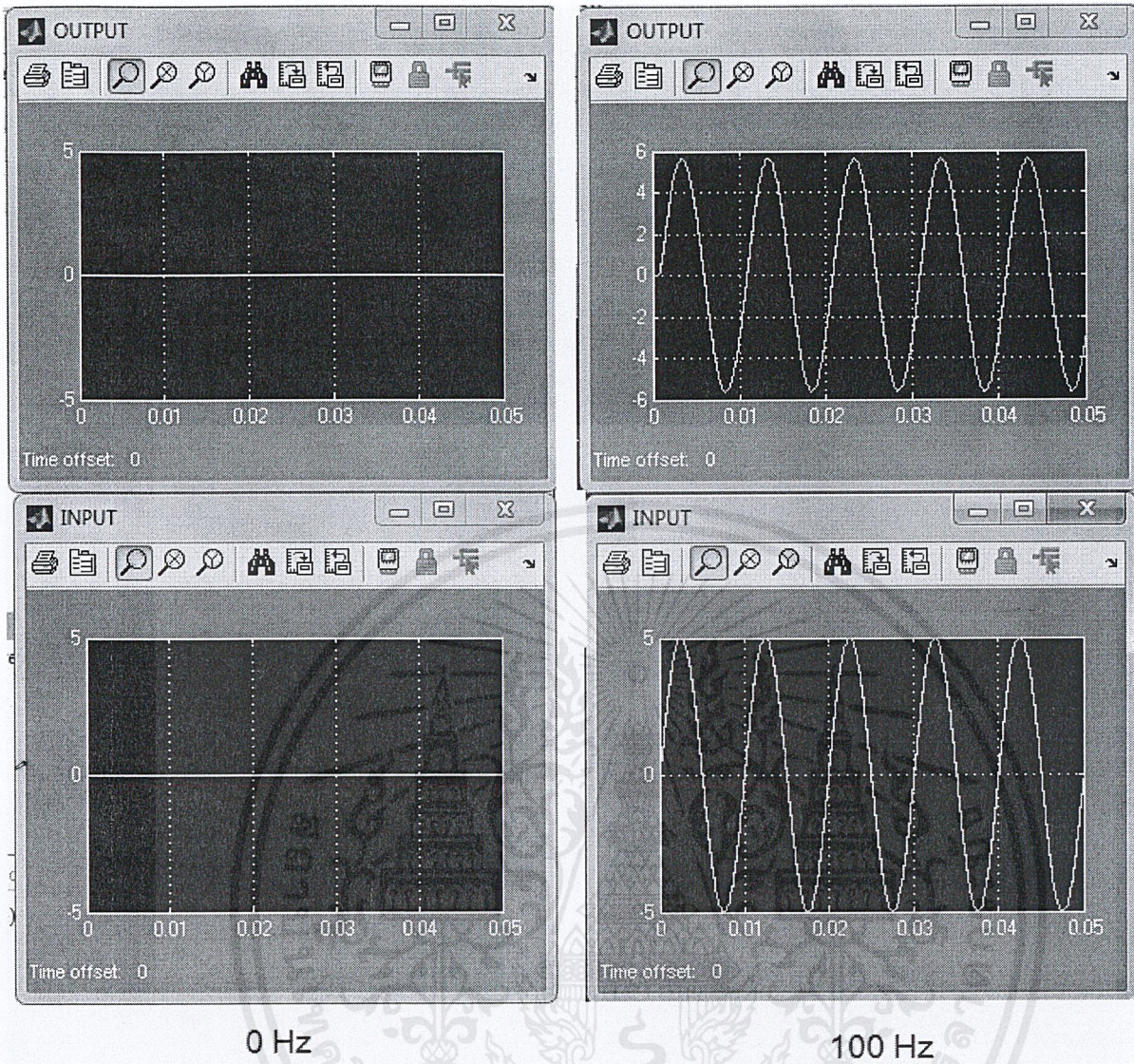
ผลการทดลอง

5.1 ผลการทดลอง digital filter จากโปรแกรม matlab

ตารางที่ 5.1 ตารางแสดงผลการทดลอง Digital Filter ที่เราได้ป้อนอินพุตที่มีขนาดของแอมพลิจูดเท่ากับ 5 โวลต์ และความถี่ขนาด ตั้งแต่ 0-2000 Hz

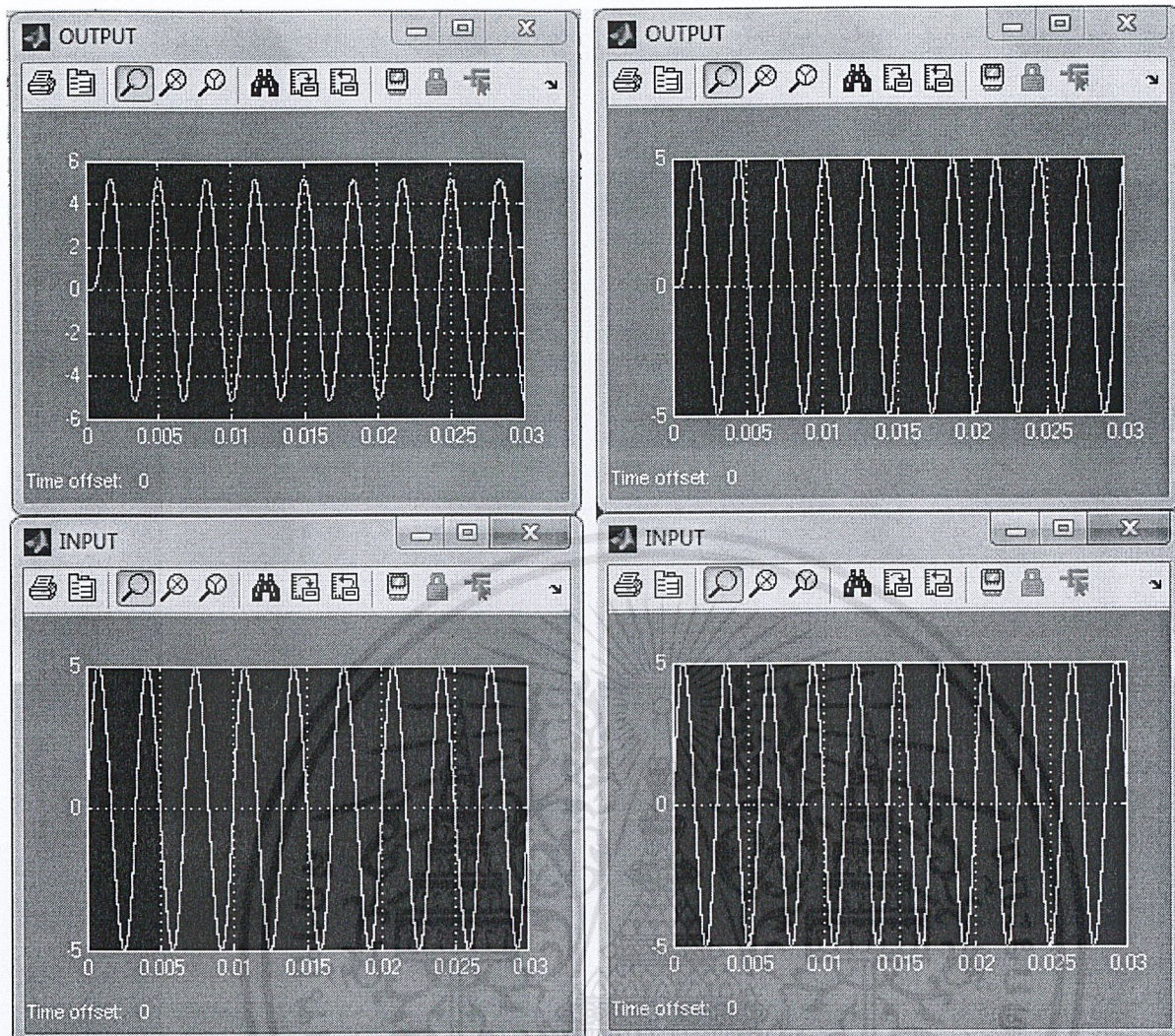
สัญญาณทางเข้า		สัญญาณทางออก
ความถี่ (Hz)	แอมพลิจูด (volt)	แอมพลิจูด (volt)
0	5.00	0
100	5.00	5.60
300	5.00	5.20
350	5.00	5.00
400	5.00	5.00
600	5.00	4.00
1000	5.00	2.00
2000	5.00	0.00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงผลการทดลองทดลอง digital filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 0Hz, 100Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

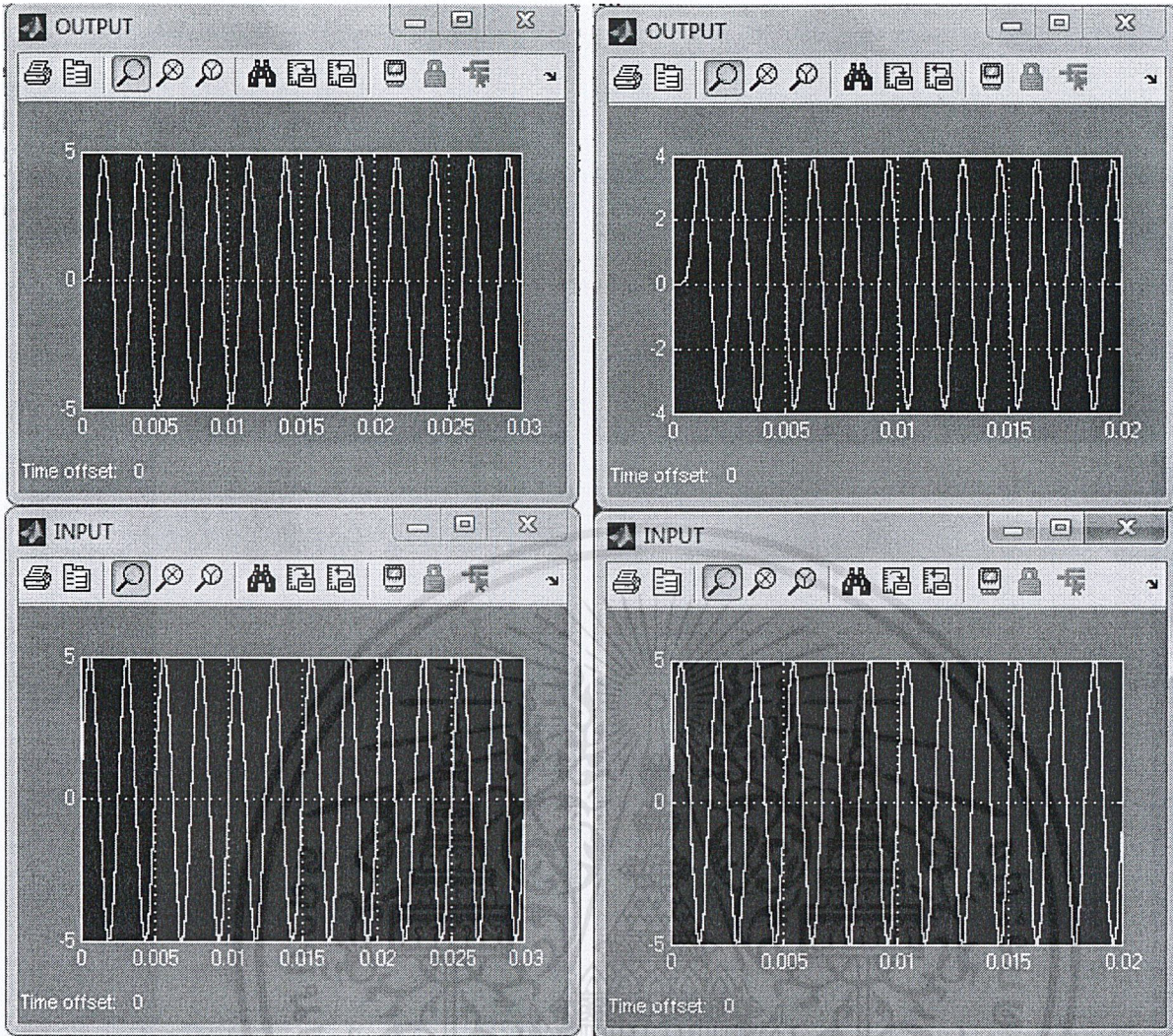


300 Hz

350 Hz

รูปที่ 5.2 แสดงผลการทดลองทดลอง digital filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 300Hz,
350Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

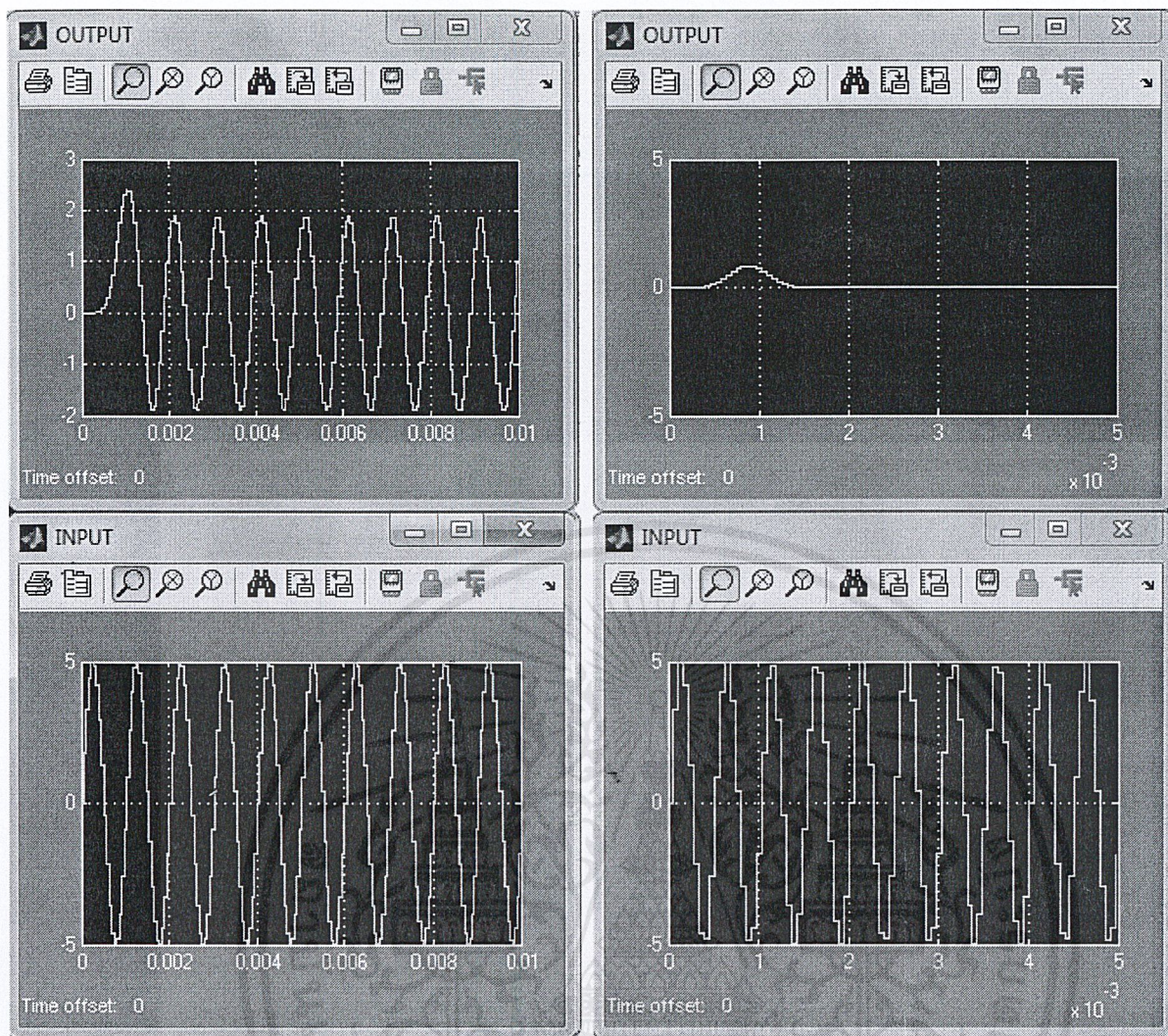


400 Hz

600 Hz

รูปที่ 5.3 แสดงผลการทดลองทดลอง digital filter จากโปรแกรม MATLAB ที่ความถี่อินพุต 400Hz,600Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1000 Hz

2000 Hz

รูปที่ 5.4 แสดงผลการทดลองทดลอง Digital Filter จาก โปรแกรม MATLAB ที่ความถี่อินพุต 1000Hz, 2000Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ผลการทดสอบโปรแกรมทดสอบการได้ยิน

ตารางที่ 5.2 ผลการทดสอบโปรแกรมทดสอบการได้ยินทดสอบโดยผู้ทดสอบทั้งหมด 5 คน

ผู้ทดสอบคนที่ 1 เพศชาย อายุ 22 ปี

ผู้ทดสอบคนที่ 2 เพศชาย อายุ 22 ปี

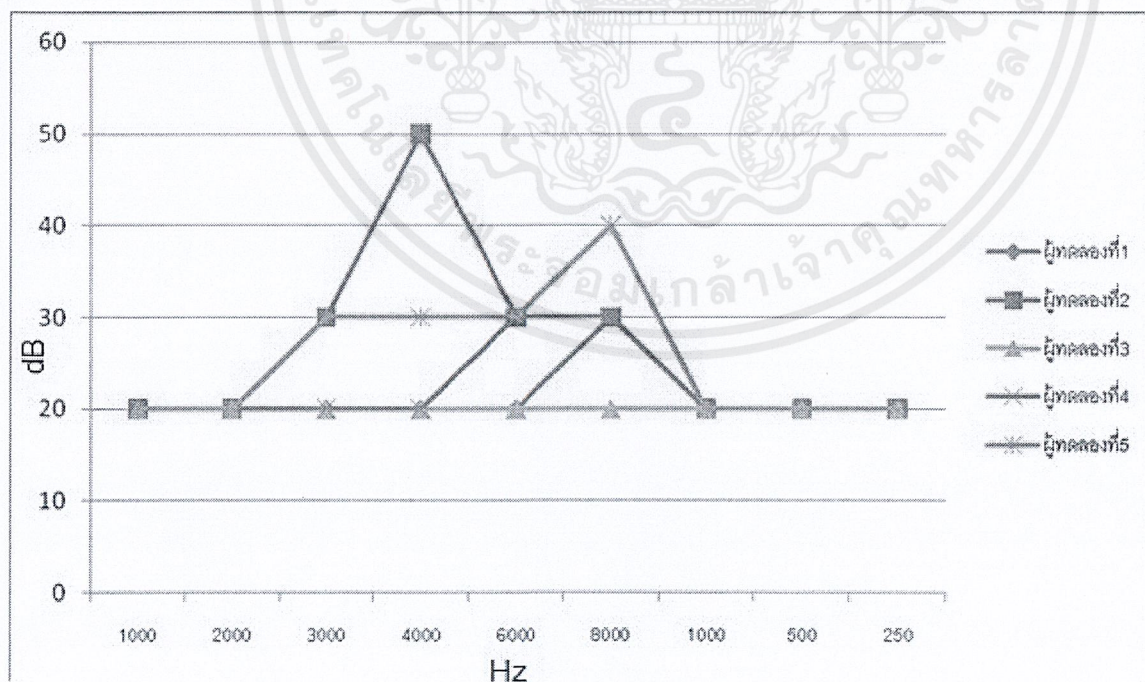
ผู้ทดสอบคนที่ 3 เพศหญิง อายุ 22 ปี

ผู้ทดสอบคนที่ 4 เพศหญิง อายุ 21 ปี

ผู้ทดสอบคนที่ 5 เพศชาย อายุ 23 ปี

ผู้ทดลองที่	ความถี่(Hz)									
	1000	2000	3000	4000	6000	8000	1000	500	250	
1	20dB	20dB	20dB	20dB	20dB	30dB	20dB	20dB	20dB	
2	20dB	20dB	30dB	50dB	30dB	30dB	20dB	20dB	20dB	
3	20dB	20dB	20dB	20dB	20dB	20dB	20dB	20dB	20dB	
4	20dB	20dB	20dB	20dB	30dB	40dB	20dB	20dB	20dB	
5	20dB	20dB	30dB	30dB	30dB	40dB	20dB	20dB	20dB	

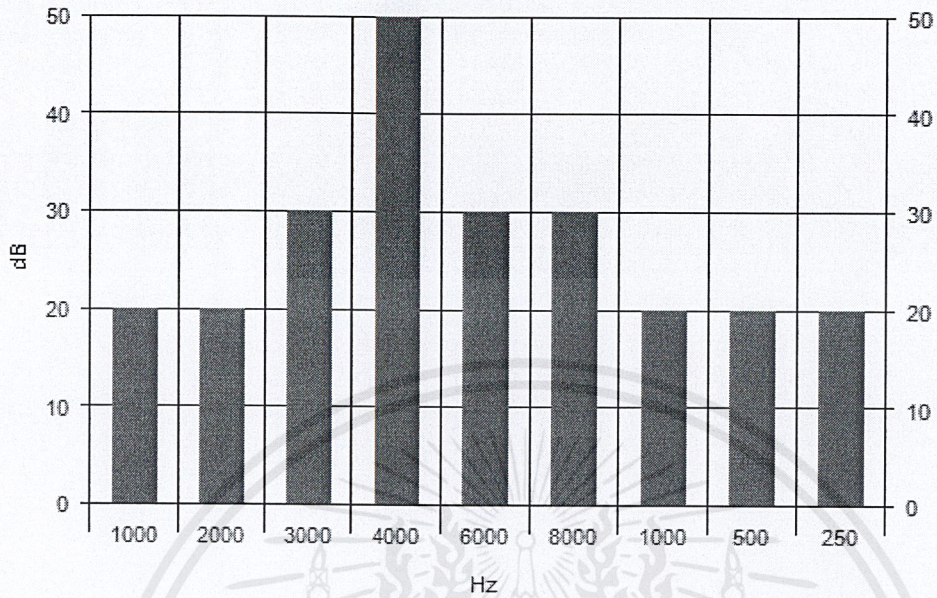
จากผลการทดสอบในตารางที่ นำมาสร้างเป็นรูปภาพดังแสดงในรูปที่



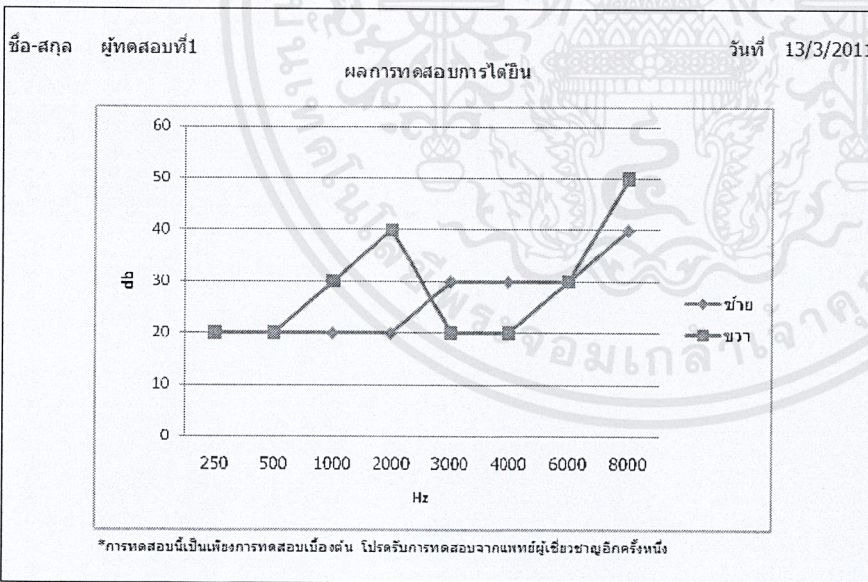
รูปที่ 5.5 แสดงผลการทดลองโปรแกรมทดสอบหู แบบที่ 1 ของผู้ทดสอบจำนวน 5 คน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Left Ear



รูปที่ 5.6 รูปแสดงผลการทดสอบหูอย่างง่าย version 1



ความถี่(Hz)	ระดับที่ได้ยิน(dB)	
	ชาย	ขวา
250	20	20
500	20	20
1000	20	30
2000	20	40
3000	30	20
4000	30	20
6000	30	30
8000	40	50

รูปที่ 5.7 รูปแสดงผลการทดสอบที่เราได้ทำการพัฒนาการบันทึกผลให้เป็นกราฟเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hearing Test

KNITL

Profile

Name : Lastname:

Frequency	source	Result
<input type="radio"/> 250 Hz	Left	Pass
<input type="radio"/> 500 Hz	Both	Pass
<input type="radio"/> 1000 Hz	Right	Pass
<input type="radio"/> 2000 Hz	Left	Pass
<input type="radio"/> 3000 Hz	Both	Pass
<input type="radio"/> 4000 Hz	Right	Pass
<input type="radio"/> 6000 Hz	Left	Pass
<input checked="" type="radio"/> 8000 Hz	Both	Pass
	Right	Pass
	Left	

Paused 00:00

Left Both Right

Down Up

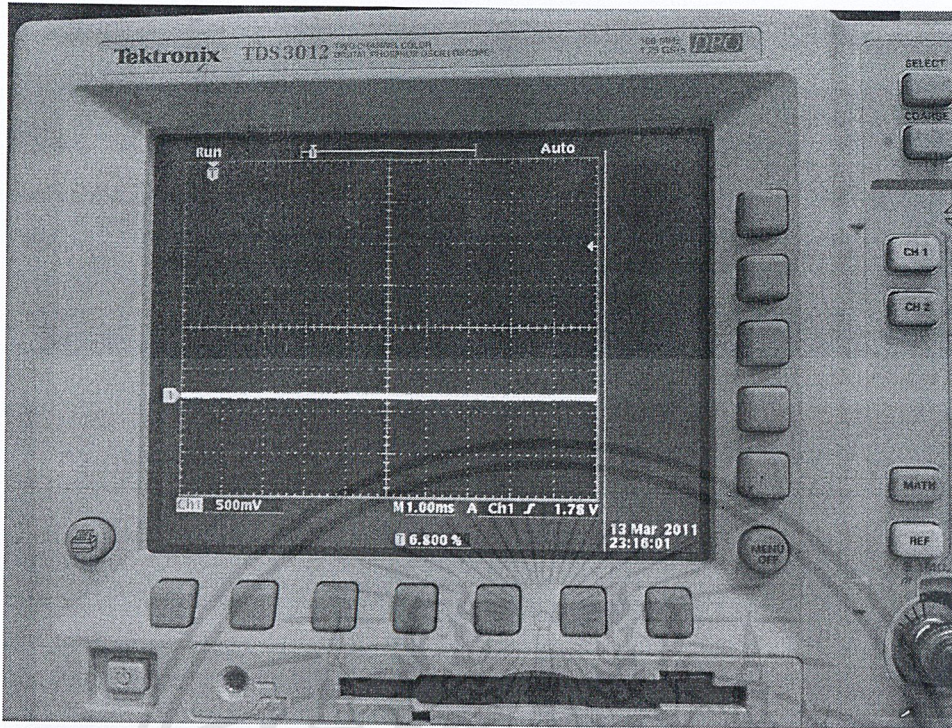
New

Home Save

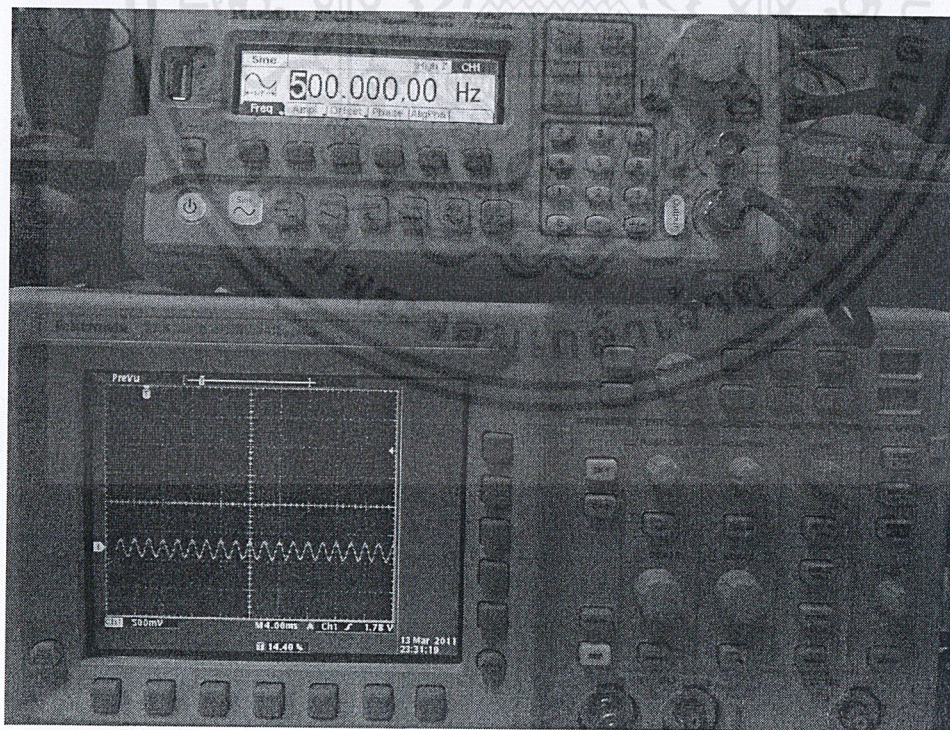
รูปที่ 5.8 รูปแสดงผลการทดสอบหูอย่างง่าย version 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ผลการทดลองโปรแกรม FPGA ที่เราใช้กรองสัญญาณ

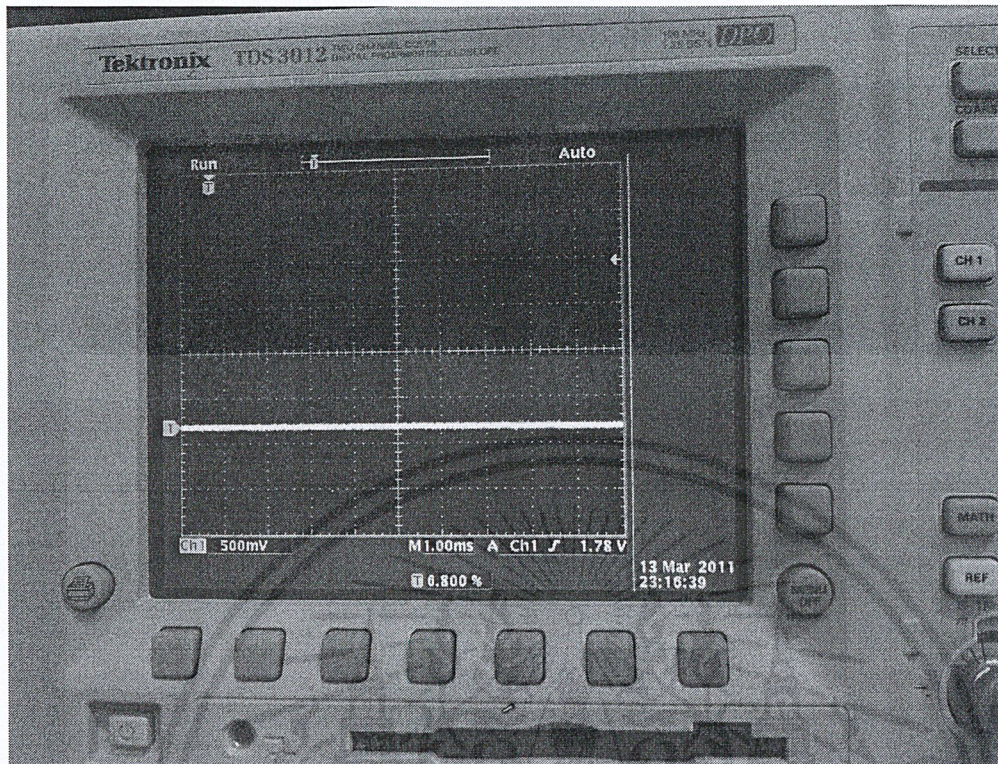


รูปที่ 5.9 แสดงผลการทดลองเมื่อเราป้อนความถี่ที่ต่ำกว่าช่วงที่เราได้กำหนดไว้



รูปที่ 5.10 แสดงผลการทดลองเมื่อป้อนความถี่ในช่วงที่เราได้กำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.11 แสดงผลการทดลองเมื่อป้อนความถี่ที่เกินจากช่วงที่กำหนดไว้

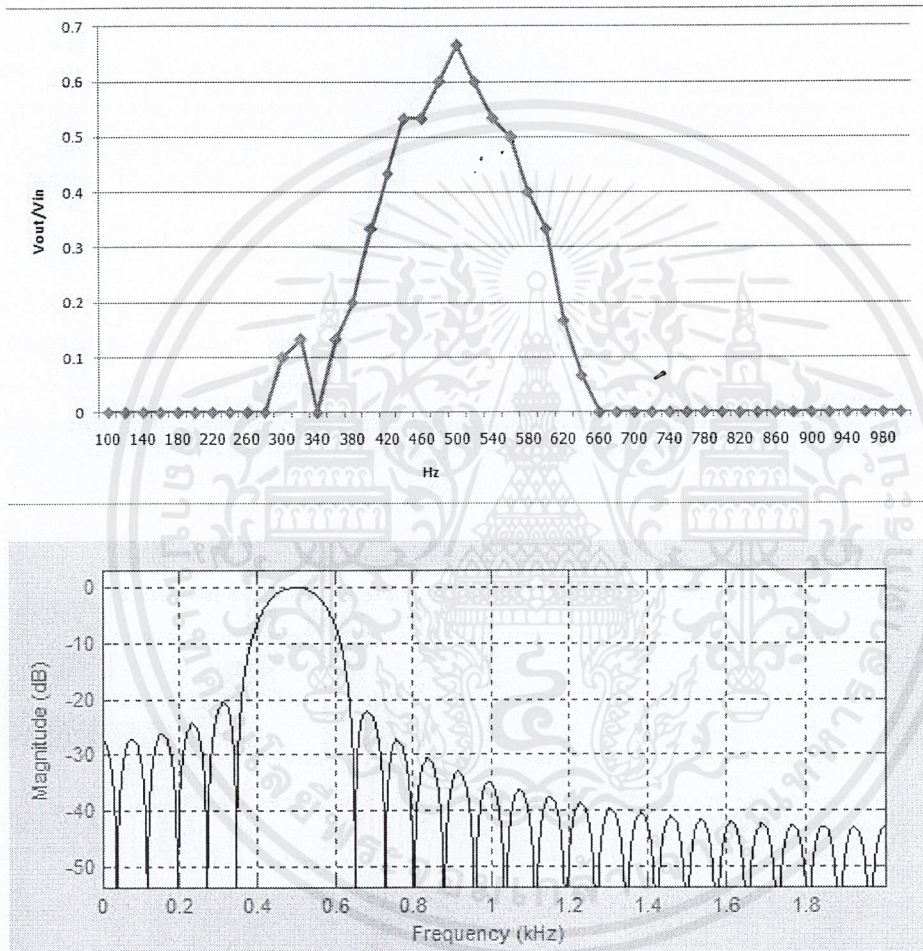
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3 แสดงผลการทดลองเมื่อป้อนความถี่ตั้งแต่ 100-1000Hz โดยเพิ่มทีละ100 จากนั้นบันทึกผล
ขนาดแอมพลิจูดของเอาพุทและอินพุท

ความถี่ (Hz)	Input(mV)	Output(mV)	Gain (Output/Input)	Gain MATLAB	Error (%)
100	600	0	0	0.000707	100
200	600	0	0	0.000184	100
220	600	0	0		
240	600	0	0		
260	600	0	0		
280	600	0	0		
300	600	60	0.1	0.006476	1444.16
320	600	80	0.13		
340	600	0	0		
360	600	80	0.2		
380	600	120	0.33		
400	600	200	0.43	0.225061	77.73
420	600	260	0.53		
440	600	320	0.6		
460	600	320	0.67		
480	600	360	0.6		
500	600	400	0.53	1.001152	47.06
520	600	360	0.5		
540	600	320	0.4		
560	600	300	0.33		
580	600	240	0.16		
600	600	200	0.06	0.189758	68.38
620	600	100	0		
640	600	40	0		
660	600	0	0		
680	600	0	0		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

700	600	0	0	0.002402	100
1800	600	0	0	8.68E-06	100
900	600	0	0	0.00027	100
1000	600	0	0	0.000317	100
				เฉลี่ย	223.733



รูปที่ 5.12 รูปแสดงกราฟการเปรียบเทียบระหว่างผลการทดลองจาก FPGA และ Filter ที่เราออกแบบโดยโปรแกรม MATLAB

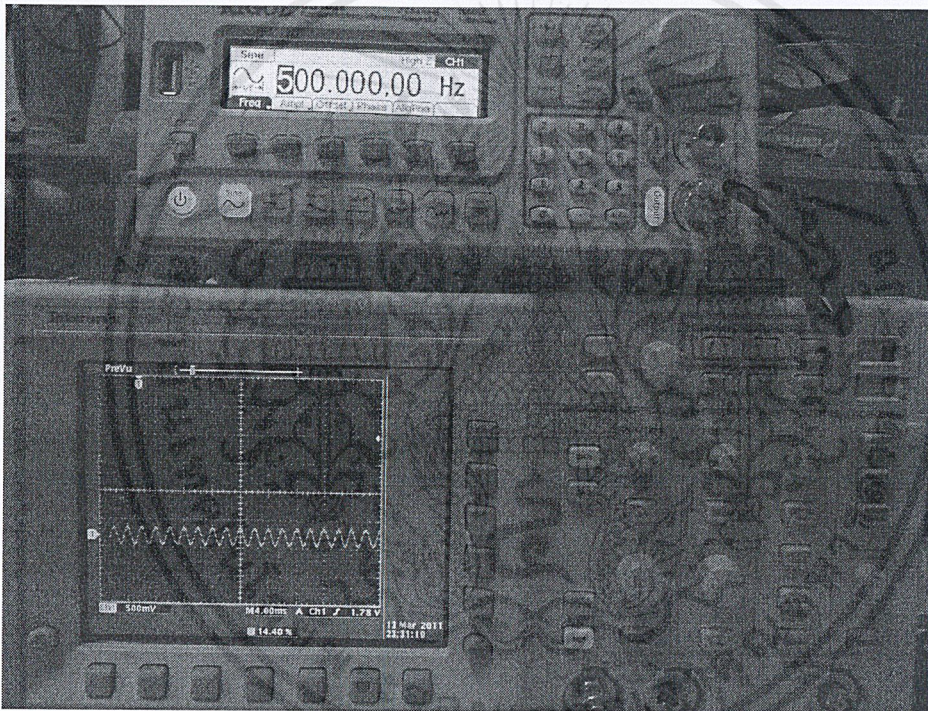
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 ผลการทดลองความถี่คัทออฟ

ตารางที่ 5.4 แสดงผลการทดลองที่ความถี่คัทออฟ โดยใช้ความถี่คัทออฟที่เราได้จากการจำลอง Filter

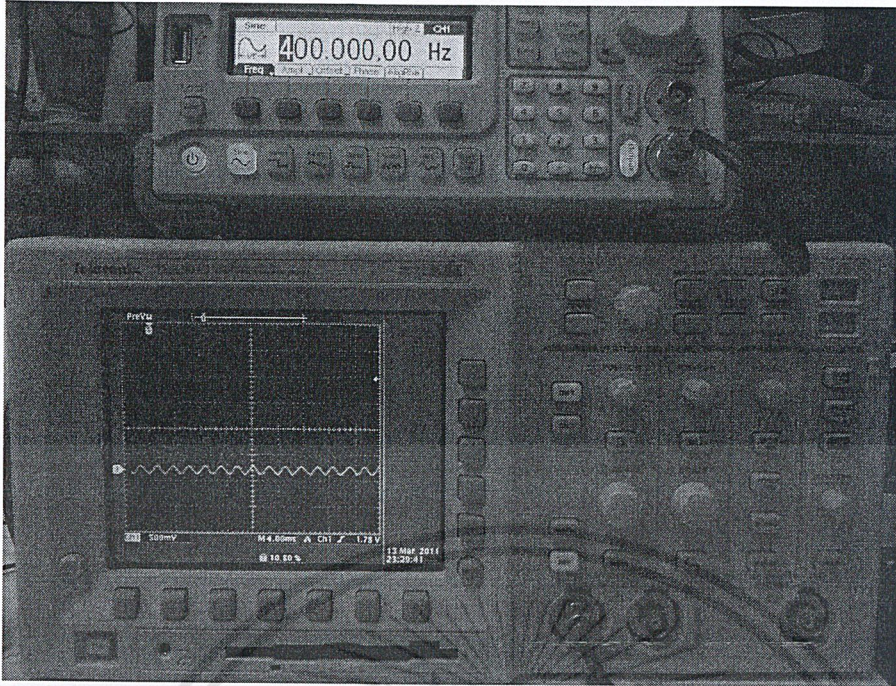
จากเครื่องมือ FDATOLL ในโปรแกรม MATLAB ทำการทดลองโดยป้อนความถี่ในช่วงแบนด์พาสเพื่อหาความถี่ที่ให้เอาพุทขนาดสูงสุด จากนั้นป้อนความถี่ต่ำคัทออฟและความถี่สูงคัทออฟแล้วบันทึกผลที่ได้

	ช่วงที่กรองแล้วได้ขนาด ที่ใหญ่ที่สุด(400Hz)	ความถี่ต่ำคัทออฟ (500Hz)	ความถี่สูงคัทออฟ (600Hz)
ขนาด(mV)	400	200	200

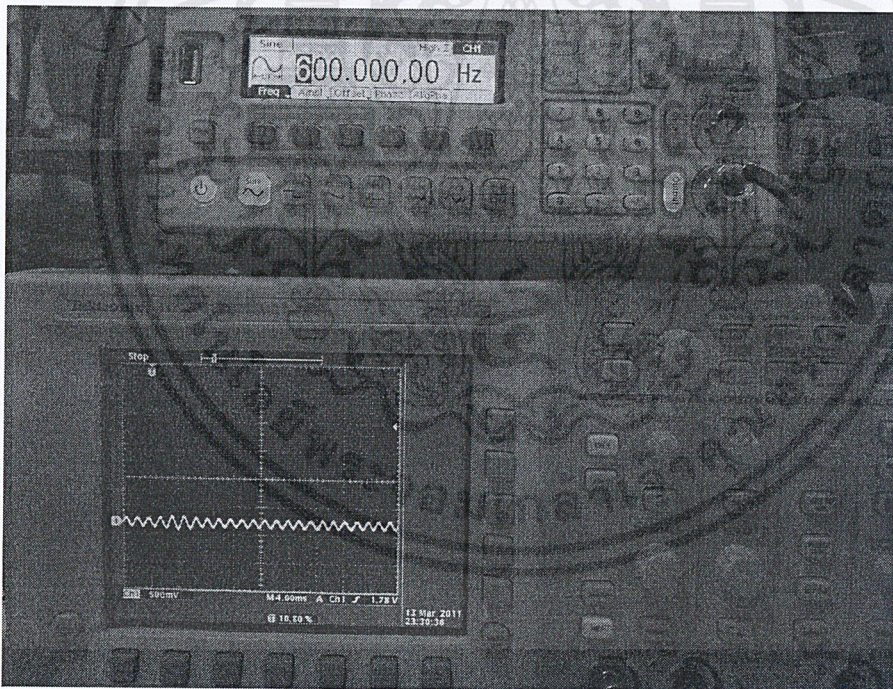


รูปที่ 5.13 รูปแสดงผลการทดลองหาความถี่ที่กรองได้ขนาดแอมพลิจูดสูงสุด (500Hz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.14 รูปแสดงผลการทดลองที่ความถี่ต่ำคัทออฟ (400Hz)



รูปที่ 5.15 รูปแสดงผลการทดลองที่ความถี่สูงคัทออฟ (600Hz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุปผลการทดลองและแนวทางการพัฒนา

6.1 สรุปผลการทดลอง

การทดลองกรองความถี่ดิจิทัลแบบ FIR Band Pass 400 Hz – 500 Hz พบว่า เมื่อส่งสัญญาณความถี่ที่อยู่ในช่วง band pass เข้าไปในตัวกรองความถี่ดิจิทัลบนโปรแกรม MATLAB สัญญาณสามารถผ่านออกมาได้หมดโดยมีการลดทอนลงเล็กน้อยอันเนื่องมาจากขีดจำกัดของ DAC และเมื่อส่งสัญญาณความถี่ที่อยู่ในช่วง Stop Band เข้าไปในตัวกรองความถี่ดิจิทัลพบว่าสัญญาณที่ออกมาน้อยมากเมื่อเทียบกับสัญญาณที่เข้าไปกล่าวคือใกล้เคียง 0 โวลต์ จากผลการทดลองนี้ทำให้เราทราบว่าวงจรกรองที่เราออกแบบสามารถใช้ได้จริง และในส่วนของโปรแกรมวงจรกรองสัญญาณที่เราเขียนขึ้นโดยใช้ภาษา VHDL เมื่อนำไปทดลองผลบนโปรแกรม XILINX ผลที่ออกมาโปรแกรมนั้นสามารถกรองความถี่ได้เป็นอย่างดี

โปรแกรมทดสอบการได้ยินของหูซึ่งเขียนบน Visual basic6 สามารถใช้ตรวจวัดความผิดปกติของการได้ยินที่ความถี่ต่างๆและประสาทสัมผัสในการได้ยินเสียงแบบเบื้องต้นได้ดี แต่หากต้องการผลที่แน่ชัดควรไปพบแพทย์ผู้เชี่ยวชาญเฉพาะทางเพื่อทำการทดสอบด้วยอุปกรณ์ที่ได้มาตรฐานอีกที

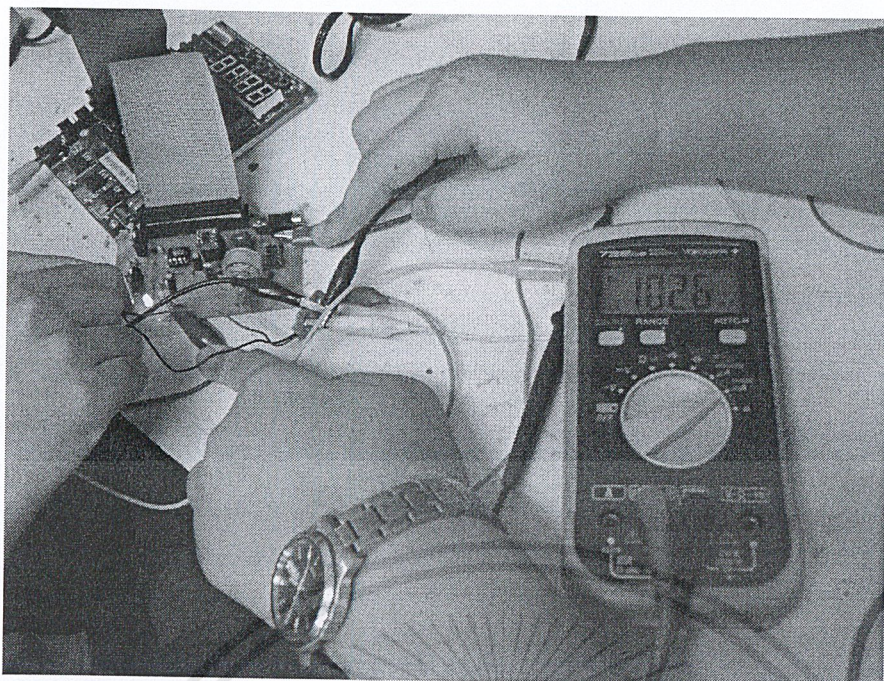
6.2 แนวทางการแก้ไขและพัฒนา

ในส่วนของ FPGA เนื่องจากผลการทดลองแสดงให้เห็นว่ามีการใช้กระแสค่อนข้างสูง สำหรับอุปกรณ์พกพาจึงควรมีการแก้ไขและพัฒนา ด้วยแนวทางเบื้องต้นดังต่อไปนี้

1. ใช้สัมประสิทธิ์ที่เป็นเลขฐานสองทั้งหมด
2. ใช้ประโยชน์จากความสมมาตรของทรานเฟอร์ฟังก์ชัน (Transfer function) โดยการกำหนดสัมประสิทธิ์ที่จะนำมาใช้คำนวณเพียงครั้งเดียว ของสัมประสิทธิ์ที่ Generate ได้ จากฟังก์ชัน FDATAOL ของ MATLAB
3. ในส่วนของวงจรคูณให้ทำการคูณสองครั้ง โดยแบ่งทีละครึ่ง โดยใช้สัญญาณเวลาในการควบคุมที่มีความถี่มากกว่า Sampling Rate ของ ADC สองเท่า

ขั้นตอนดังกล่าวนี้จะสามารถช่วยลดทรัพยากรที่ต้องใช้ใน FPGA ได้เป็นอย่างดี

ในส่วนของวงจรมอกเหนือจาก FPGA ควรมีการออกแบบให้ป้องกัน Noise ได้ดีกว่านี้ โดยอาจจะใช้อุปกรณ์ที่ต้องการความถี่ควบคุมต่ำกว่าที่ใช้อยู่



รูปที่ 6.1 รูปแสดงค่ากระแสที่เราวัดได้จากวงจร

6.3 ประโยชน์ที่คาดว่าจะได้รับ

1. ความรู้ความเข้าใจในภาษา VHDL
2. เข้าใจถึงคุณสมบัติและขีดความสามารถของ FPGA
3. ความรู้ความเข้าใจในการเขียน โปรแกรม Visual Studio 6.0
4. ความรู้ความเข้าใจในทฤษฎี FIR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ตำราโสต นาสิก ลาริงซ์วิทยา (หน้า 2), โดย ชัย อยู่สวัสดิ์, 2540, กรุงเทพมหานคร : โฮลิสติก แพบลิชซิ่ง.
2. PHY 406F -Microprocessor Interfacing Techniques by JamesR.Drummon -September 1996
3. <http://www.eartone.co.th>
4. <http://www.i-medipro.com/index.php?lay=show&ac=article&Id=70931&Ntype=4>
5. <http://wara.com/article-765.html>
6. <http://www.te.kmutnb.ac.th/~msn/223361han07.pdf>

