



ใบรับรองวิทยานิพนธ์  
บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

วิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

ปริญญา

วิศวกรรมไฟฟ้า

วิศวกรรมไฟฟ้า

สาขา

ภาควิชา

เรื่อง ระบบเตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสถานะที่มีเสียงรบกวนสูง

Adaptive Auditory Warning System

นามผู้วิจัย นายอภิสิทธิ์ สุนทรพะลิน

ได้พิจารณาเห็นชอบโดย

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

( ผู้ช่วยศาสตราจารย์ปฐมาภรณ์ ศรีผดุงธรรม, Ph.D. )

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

( ผู้ช่วยศาสตราจารย์ชูเกียรติ การะเกตุ, Ph.D. )

หัวหน้าภาควิชา

( รองศาสตราจารย์มงคล รักษาพัชรวงค์, Ph.D. )

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์รับรองแล้ว

( รองศาสตราจารย์กัญจนา ชีระกุล, D.Agr. )

คณบดีบัณฑิตวิทยาลัย

วันที่ ..... เดือน ..... พ.ศ. ....

วิทยานิพนธ์

เรื่อง

ระบบเตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสภาวะที่มีเสียงรบกวนสูง

Adaptive Auditory Warning System

โดย

นายอภิสิทธิ์ สุนทรพะลิน

เสนอ

บัณฑิตวิทยาลัย มหาวิทยาลัยเกษตรศาสตร์

เพื่อความสมบูรณ์แห่งปริญญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า)

พ.ศ. 2553

ลิขสิทธิ์ มหาวิทยาลัยเกษตรศาสตร์

อภิสิทธิ์ สุรินทร์พนิน 2553: ระบบเตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสภาวะที่มีเสียงรบกวนสูง ปรินญาวิศวกรรมศาสตรมหาบัณฑิต (วิศวกรรมไฟฟ้า) สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก:  
ผู้ช่วยศาสตราจารย์ปฐมภรณ์ ศรีผดุงธรรม, Ph.D. 81หน้า

ระบบเตือนภัยแบบตัดแปลงเสียงถูกพัฒนาขึ้นมาในงานวิจัย ซึ่งส่วนประกอบหลักของระบบแบ่งออกเป็น 3 ส่วนคือ ส่วนอินพุตเป็นการแปลงเสียงจากสิ่งแวดล้อมโดยเป็นการแปลงสัญญาณอนาล็อกที่ได้รับจากไมโครโฟนให้เป็นสัญญาณดิจิทัล ส่วนประมวลผลกระบวนการทำงานของระบบเริ่มจากการกดปุ่มสัญญาณเตือนภัย หรือเซนเซอร์ชนิดต่างๆ ตรวจสอบสิ่งผิดปกติที่เกิดขึ้นเช่น ความร้อนสูง หรือปริมาณควันที่มากเกินไป ส่วนประมวลผลจะเริ่มทำการวิเคราะห์ข้อมูลสัญญาณดิจิทัลและทำหน้าที่ควบคุมการทำงานทุกส่วนของระบบ และส่วนเอาต์พุตเป็นการสร้างสัญญาณพัลส์โดยใช้เทคนิค Pulse Width Modulation (PWM) เพื่อไปควบคุมระดับความดังของกริ่งเตือนภัย ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงสามารถรับเสียงจากสิ่งแวดล้อมได้ที่ค่าความดังตั้งแต่ 0- 120 เดซิเบล ปรับระดับค่าความดังของกริ่งเตือนภัยให้มากกว่าค่าความดังของเสียงจากสิ่งแวดล้อมในขณะนั้น 10 dB และสามารถตรวจจับเสียงจากสิ่งแวดล้อมขณะนั้นได้โดยมีรัศมีการตรวจจับ 2 เมตร

---

ลายมือชื่อนิติ

---

ลายมือชื่ออาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

\_\_\_\_ / \_\_\_\_ / \_\_\_\_

Apisit Soontornpalin 2010: Adaptive Auditory Warning System. Master of Engineering (Electrical Engineering), Major Field: Electrical Engineering, Department of Electrical Engineering. Thesis Advisor: Assistant Professor Patamaporn Sripadungtham, Ph.D. 81 pages.

An adaptive auditory warning system was developed in this research to automatically adjust a siren's volume according to the detected background noise. The main components of this system were divided into 3 parts, first was the input which converted audio signals from the environment by converting analog signal into digital signal, the second part used Field-Programmable Gate Array (FPGA) to analyze digital signal data and control systems. The third part was the output part, which created pulse using Pulse Width Modulation (PWM) technique to control the siren level of a siren bell. The adaptive auditory warning system was able to detect background noise between 0-120 dB. Siren sound generated from the system was increased from the detected background noise by 10 dB. The system could detect the background noise within 2 metre radius.

---

Student's signature

---

Thesis Advisor's signature

\_\_\_\_ / \_\_\_\_ / \_\_\_\_

## กิตติกรรมประกาศ

ข้าพเจ้าขอขอบคุณ ผศ. ดร. ปฐมาภรณ์ ศรีผดุงธรรม อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ผศ. ดร. ชูเกียรติ การะเกตุ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม และ ผศ. ดร. คุณิต ชนเพทาย ที่ได้ให้คำปรึกษาแนะนำในการทำงานวิจัย และตรวจแก้ไขข้อบกพร่องในวิทยานิพนธ์ให้สำเร็จลุล่วงไปได้ด้วยดี รวมถึงอาจารย์ทุกท่านที่อบรมสั่งสอนและมอบความรู้แก่ข้าพเจ้าตั้งแต่อดีตจนถึงปัจจุบัน

ข้าพเจ้าขอขอบคุณ โครงการวิจัยทุนอุดหนุนวิจัย มก. เป็นอย่างสูงสำหรับการสนับสนุนเงินทุนในการทำวิจัย บริษัท วิทซ์ ดีไซน์ แอนด์ อิเล็กทรอนิกส์ จำกัด และขอขอบคุณ สารัตน์ อยู่วัฒนะ ที่ให้คำปรึกษาทางด้านเทคนิคบางส่วน ขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเกษตรศาสตร์ สำหรับความอนุเคราะห์ในเรื่องสถานที่ห้องปฏิบัติการวิจัยและพัฒนา รวมทั้งเครื่องมือวัดที่มีประสิทธิภาพหลายชนิดในงานวิจัยนี้

ข้าพเจ้าขอขอบคุณคุณพ่อและคุณแม่ ตลอดจนเพื่อนๆทุกคน ที่ให้การสนับสนุน และให้กำลังใจในการทำวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงไปได้ด้วยดี ข้าพเจ้าหวังว่าวิทยานิพนธ์ฉบับนี้จะนำไปสู่การนำไปใช้ให้เกิดประโยชน์ในระบบเตือนภัยที่มีใช้ในปัจจุบัน หากวิทยานิพนธ์ฉบับนี้มีข้อบกพร่องประการใด ข้าพเจ้ายินดีรับข้อเสนอแนะ และขออภัยมา ณ ที่นี้ด้วย

อภิสิทธิ์ สุทรพะสิน

เมษายน 2553

## สารบัญ

## หน้า

สารบัญ	(1)
สารบัญตาราง	(2)
สารบัญภาพ	(3)
คำนำ	1
วัตถุประสงค์	2
การตรวจเอกสาร	3
อุปกรณ์และวิธีการ	29
อุปกรณ์	29
วิธีการ	30
ผลและวิจารณ์	44
ผล	44
วิจารณ์	49
สรุปและข้อเสนอแนะ	50
สรุป	50
ข้อเสนอแนะ	51
เอกสารและสิ่งอ้างอิง	52
ภาคผนวก	54
ประวัติการศึกษา และการทำงาน	81

## สารบัญตาราง

ตารางที่		หน้า
1	ตัวอย่างแหล่งกำเนิดเสียง ระดับเสียง และผลกระทบที่มีต่อมนุษย์	4
2	แสดงระดับเสียงที่เกิดจากแหล่งกำเนิดเสียงชนิดต่างๆ	5
3	ผลกระทบที่เกิดขึ้นกับหูของมนุษย์ที่ค่าความดังระดับต่างๆ	6
4	แสดงความสัมพันธ์ระหว่าง แรงดันไฟฟ้าเอาต์พุต (V) ของ IC LM3915 กับค่าความดังของเสียง (dB)	15
5	ตัวอย่างความสามารถของชิป FPGA	17
6	แสดงความสามารถ ประสิทธิภาพด้านต่างๆ ระหว่าง ไมโครคอนโทรลเลอร์ FPGA และ ASIC (Application Specific IC)	20
7	สรุปข้อดีและข้อด้อยของอัลกอริทึมของนักวิจัยท่านต่างๆ ในการแยกองค์ประกอบเสียงพูดชั่วคราว	28
8	แสดงความสัมพันธ์ระหว่าง Duty Cycle ของระดับแรงดันไฟฟ้า กับค่าความดังของกริ่งเตือนภัย (dB)	42
9	เปรียบเทียบค่า $V_{OUTTyp}$ จาก Datasheet กับค่า $V_{OUTTyp}$ ที่วัดได้จากดิจิทัลมัลติมิเตอร์	46

## สารบัญภาพ

ภาพที่		หน้า
1	ส่วนประกอบของระบบสัญญาณเตือนอัคคีภัยโดยทั่วไป	8
2	ส่วนประกอบระบบอิเล็กทรอนิกส์เตือนภัย	11
3	วงจร Counting Converter	12
4	กราฟเอาต์พุตของวงจร Counter เทียบกับแกนเวลา	13
5	โครงสร้างภายในของ IC LM3915 ซึ่งแสดงให้เห็นว่ามีวงจร Parallel Comparator ในการแปลงสัญญาณอนาล็อกเป็นดิจิทัลประกอบอยู่ด้วย	14
6	การต่อวงจรสำหรับใช้งาน IC LM3915 จำนวน 2 ตัว แบบแคสเคด	16
7	การต่อวงจรสำหรับใช้งาน IC LM3915 จำนวน 2 ตัว โดยประยุกต์ใช้ Op-Amp ในการขยายสัญญาณอินพุต	16
8	ทฤษฎีขั้นตอนการออกแบบ FPGA มีทั้งหมด 4 ขั้นตอน	18
9	สัญญาณ PWM แบบ NPWM	21
10	สัญญาณ PWM แบบ UPWM	22
11	โครงสร้างของการสร้างสัญญาณ PWM ทั้งสองแบบ	22
12	โครงสร้างของการกำเนิดสัญญาณ PWM โดยทั่วไป	23
13	สัญญาณแต่ละภาคตาม โครงสร้างของการกำเนิดสัญญาณ PWM	24
14	ระบบเตือนภัยแบบตัดแปลงเสียง (Adaptive Auditory Warning System)	30
15	ส่วนอินพุตของระบบเตือนภัยแบบตัดแปลงเสียง	31
16	วงจรภาคเพาเวอร์ซัพพลายของระบบ	32
17	วงจรกรองความถี่หูดผ่าน (BSF)	33
18	การต่อวงจรสำหรับการใช้งาน IC LM3915	34
19	วงจร RDIV นำมาต่อกับ IC เบอร์ LM3915	34
20	แสดงการต่อ IC LM 3915 แสดงการต่อ IC LM 3915 โดยทฤษฎี Parallel ออปแอมป์	36
21	วงจรส่วนประมวลผล และเฟลชเมมโมรี่	37

## สารบัญญภาพ (ต่อ)

ภาพที่		หน้า
22	วงจรแสดงค่าความดังของเสียงภาคอินพุต และภาคเอาต์พุต	38
23	กราฟความสัมพันธ์ระหว่างความดังอินพุตและความดังเอาต์พุตโดยไม่สนใจความถี่	39
24	Flow Chart หลักการทำงานของส่วนประมวลผลของระบบ	40
25	วงจรทรานซิสเตอร์คาร์ลิงตัน วงจรขับแรงดันมอสเฟต และ กริ่งเตือนภัย	41
26	PWM 50 % Duty cycle ค่าความดังที่ได้ 100 dB	41
27	กริ่งเตือนภัย และเครื่องวัดเสียง	43
28	ไมโครโฟนสำหรับใช้รับสัญญาณเสียงจากสิ่งแวดล้อม	44
29	ผลตอบสนองต่อความถี่ของวงจร BSF ที่ความถี่ 3.1 kHz	45
30	บอร์ดคอนโทรลของระบบเตือนภัยแบบตัดแปลงเสียง	47
31	สัญญาณ PWM ที่ Duty Cycle 10% เทียบได้ค่าความดัง 70 dB	48
32	สัญญาณ PWM ที่ Duty Cycle 30% เทียบได้ค่าความดัง 90 dB	48
33	สัญญาณ PWM ที่ Duty Cycle 70% เทียบได้ค่าความดัง 105 dB	48

## ระบบเตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสถานะที่มีเสียงรบกวนสูง

### Adaptive Auditory Warning System

#### คำนำ

เสียงเป็นพลังงานที่เกิดจากการสั่นสะเทือนของโมเลกุลอากาศเกิดเป็นคลื่นเสียงและอาศัยอากาศเป็นตัวกลางเคลื่อนที่ไปสู่อวัยวะรับเสียงคือหู ปกติหูของมนุษย์ได้ยินความถี่ของเสียงตั้งแต่ระดับ 20 Hz – 20k Hz และองค์การอนามัยโลกได้กำหนดว่าเสียงที่เป็นอันตรายได้แก่เสียงที่มีความดังอยู่ในระดับเกิน 85 เดซิเบลที่ทุกความถี่ (มโนส, 2530) เป็นมลพิษทางเสียง ความหมายของมลพิษทางเสียง คือ สภาพแวดล้อมที่มีเสียงดังเกินค่ามาตรฐานที่กรมควบคุมมลพิษกำหนดซึ่งก่อให้เกิดความรำคาญ สร้างความรบกวน ทำให้เกิดความเครียดทั้งทางร่างกายและจิตใจ ทำให้ตกใจ และอาจถึงขั้นเป็นอันตรายต่อสุขภาพอนามัยได้ เช่น เสียงที่ดังมาก หรือเสียงที่ดังต่อเนื่องยาวนานไม่จบสิ้น ประเภทของแหล่งกำเนิดมลพิษทางเสียง ส่วนใหญ่เกิดจากการกระทำของมนุษย์แบ่งได้เป็น 2 ประเภท คือ ประเภทเคลื่อนที่ เช่น เสียงจากพาหนะเครื่องยนต์ทางบก ทางน้ำ ทางอากาศ เสียงจากเครื่องจักรหนักที่ใช้ในการก่อสร้าง และประเภทอยู่กับที่ ได้แก่ เสียงฟ้าผ่า ฟ้าร้อง อุโมงค์รถ สวนสนุก และโรงงานอุตสาหกรรม เป็นต้น

ในงานวิจัยระบบเตือนภัยแบบตัดแปลงเสียง (Adaptive Auditory Warning System) เป็นการควบคุมแหล่งกำเนิดมลพิษทางเสียงประเภทอยู่กับที่ คือควบคุมเสียงของสัญญาณเตือนภัยในโรงงานอุตสาหกรรม เมื่อมีการกดปุ่มสัญญาณเตือนภัย หรือเซนเซอร์ชนิดต่างๆ ตรวจจับความผิดปกติที่เกิดขึ้นภายในโรงงานได้ เช่น ปริมาณควันมากกว่าปกติ หรือความร้อนสูง ระบบเตือนภัยแบบตัดแปลงเสียง จะทำการปรับเปลี่ยนเสียงสัญญาณเตือนภัยให้ดังกว่าเสียงของสภาวะแวดล้อม เช่น เสียงของเครื่องจักรหนัก เสียงของมนุษย์เวลาตื่นตกใจ หรือเสียงอื่นๆ ที่ดังไม่เท่ากันในแต่ละช่วงเวลาเป็นต้น ด้วยแนวคิดที่สัญญาณเตือนภัยไม่จำเป็นต้องมีความดังที่ระดับเดียวกันตลอดเวลา แต่ควรให้ระบบเตือนภัยสามารถตรวจจับและปรับระดับความดังของสัญญาณเตือนภัยให้มากกว่าเสียงของสภาวะแวดล้อมในขณะนั้นได้ตลอดเวลา เพราะเสียงของสัญญาณเตือนภัยเองอาจทำให้เกิดมลพิษทางเสียงแก่ผู้ปฏิบัติงานในโรงงานได้เช่นกัน เพื่อลดมลพิษทางเสียงที่มีผลต่อสภาพร่างกายและจิตใจของพนักงานในโรงงานให้เกิดขึ้นน้อยที่สุดและให้เป็นไปตามมาตรฐานของกรมควบคุมมลพิษ

## วัตถุประสงค์

โครงการวิจัยนี้มีวัตถุประสงค์เพื่อศึกษาค้นคว้า ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสถานะที่มีเสียงรบกวนสูง โดยมุ่งศึกษาและพัฒนาระบบอิเล็กทรอนิกส์สำหรับใช้ในโรงงาน เพื่อนำมาประยุกต์ใช้ในงานด้านวิศวกรรมไฟฟ้าและอิเล็กทรอนิกส์ โดยสามารถแจกแจงวัตถุประสงค์หลักได้ดังนี้

1. เพื่อประยุกต์ใช้ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงสำหรับใช้ในโรงงานอุตสาหกรรมเพื่อลดมลพิษทางเสียงที่อาจเกิดจากตัวเครื่องเตือนภัยแบบเดิมที่ไม่สามารถปรับระดับความดังของเสียงตามสถานะแวดล้อมได้
2. เพื่อทดสอบระบบเตือนภัยแบบตัดแปลงเสียง ส่วนอินพุต ส่วนเอาต์พุต และส่วนประมวลผล โดยส่วนอินพุตทำการทดสอบความสัมพันธ์ระหว่างค่าความดัง (dB) กับค่าแรงดันไฟฟ้า (V) ส่วนเอาต์พุตประยุกต์ใช้เทคนิค PWM (Pulse Width Modulation) ทำการปรับระดับความดังของกริ่งเตือนภัย และส่วนประมวลผล ใช้ FPGA (Field-Programmable Gate Array) ในการควบคุมการทำงานส่วนต่างๆ ของระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสถานะที่มีเสียงรบกวนสูง
3. เพื่อเป็นระบบต้นแบบสำหรับการพัฒนา และประยุกต์ใช้กับระบบอิเล็กทรอนิกส์อื่นๆ เพื่อควบคุมมลพิษทางเสียงที่อาจเกิดขึ้นจากการใช้เสียงเตือนภัย หรือใช้เสียงในการสื่อสารในลักษณะต่างๆ ให้เกิดขึ้นน้อยที่สุด

## การตรวจเอกสาร

### ทฤษฎีเบื้องต้นที่เกี่ยวข้อง

ในหัวข้อนี้จะกล่าวถึงหลักการพื้นฐานที่เกี่ยวข้องในการที่จะทำการสร้างระบบเตือนภัยแบบตัดแปลงเสียงให้เหมาะสมในสภาวะที่มีเสียงรบกวนสูง โดยการนำระบบอิเล็กทรอนิกส์มาประยุกต์ใช้เป็นหลัก ซึ่งมีทฤษฎีต่างๆที่เกี่ยวข้องมาก โดยเริ่มตั้งแต่ทฤษฎีเสียงและปัญหามลพิษทางเสียงกับสิ่งแวดล้อม, ระบบเตือนภัยแบบปกติ, ส่วนประกอบทางด้านระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียง, ทฤษฎีการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล, ทฤษฎีเบื้องต้นเกี่ยวกับ FPGA (Field-Programmable Gate Array) และการประยุกต์ใช้เทคนิค PWM (Pulse Width Modulation) เพื่อทำการปรับระดับความดังของกริ่งเตือนภัย

#### 1. ทฤษฎีเสียงและปัญหามลพิษทางเสียงกับสิ่งแวดล้อม

เสียงเป็นพลังงานที่เกิดจากการสั่นสะเทือนของโมเลกุลของอากาศ เกิดเป็นคลื่นเสียง และอาศัยอากาศเป็นตัวกลางเคลื่อนที่ไปสู่ตัวรับเสียง คือ หู เสียงที่จัดเป็นมลพิษหรือทำลายโสตประสาทของมนุษย์สามารถแบ่งได้เป็น 3 กลุ่มคือ เสียงดัง เสียงอันน่ารำคาญ และเสียงที่เกิดขึ้นอย่างกะทันหัน เช่นเสียงเครื่องบิน ไอพ่นที่บินผ่าน เป็นต้น

โดยทั่วไปแล้วเสียงแบ่งออกเป็น 2 ประเภท คือ เสียงสบายอารมณ์ (Sound) หมายถึงเสียงที่ฟังแล้วสบายใจ สามารถปฏิบัติงานได้ดี และเสียงอึกทึก (Noise) หมายถึงเสียงที่คนไม่ต้องการ เนื่องจากไม่มีความไพเราะนุ่มนวล กระจ่างหู ซึ่งเสียงอึกทึกจะมีลักษณะดังนี้ คือ เสียงพุ่มที่เกิดขึ้นเป็นเวลานานๆ เช่น เสียงเครื่องยนต์ เสียงแหลมเกิดขึ้นเป็นเวลานาน เสียงที่เกิดจากการกระทบใน ช่วงเวลาสั้นๆ เช่น เสียงค้อน เสียงที่เกิดจากการกระทบที่ดังมากเป็นจังหวะหรือเป็นครั้งคราว เช่น เสียงเครื่องเจาะคอนกรีต และเสียงที่เกิดดังขึ้นเป็นพักๆ เช่น เสียงการจราจร (มนัส, 2530)

สำหรับลักษณะของเสียงพิจารณาได้ 2 ประการ คือ ความถี่ของเสียง (Frequency of sound) วัดเป็นรอบ/วินาที หรือ Hertz (Hz) เสียงที่มีความถี่มากจะมีความถี่สูง ปกติมนุษย์สามารถได้ยินเสียงที่มีขนาดความถี่ 20 - 20,000 Hz (มนัส, 2530) ความดังของเสียง (Intensity) มีหน่วยวัดเป็น (decibel, dB) องค์การอนามัยโลกกำหนดไว้ว่า เสียงที่เป็นอันตรายหมายถึงเสียงที่ดังเกิน 85

dBa ที่ทุกๆ ความถี่ถ้าสัมผัสนานเกินไป ซึ่งเดซิเบลเอ (dBA) คือ หน่วยการวัดระดับเสียงเพื่อศึกษาผลกระทบที่เกิดขึ้นกับมนุษย์ และเสียงที่มีความดังขนาด 160 dB จะทำให้แก้วหูทะลุเช่นเสียงระเบิด และเสียงบางประเภทเช่น Ultrasonic Sound ที่มีความถี่สูง 15,000 Hz หากผ่านลงในน้ำจะทำให้แบคทีเรียสลายตัวและสัตว์น้ำตาย ซึ่งตัวอย่างแหล่งกำเนิดเสียง และค่าความดังของเสียงมีผลกระทบต่อร่างกายมนุษย์ แสดงดังตารางที่ 1

ตารางที่ 1 ตัวอย่างแหล่งกำเนิดเสียง ระดับเสียง และผลกระทบที่มีต่อมนุษย์

แหล่งกำเนิดเสียง	dbA	ผลกระทบเมื่อรับฟังเป็นเวลานาน
เครื่องบินไอพ่นขณะวิ่งขึ้น (ระยะใกล้)	150	เยื่อหูขาด
บริเวณขนถ่ายผู้โดยสารสนามบิน	140	
เสียงสัญญาณเตือนภัย	130	
เครื่องบินไอพ่นขณะวิ่ง (200 ฟุต)	120	ระดับสูงสุดที่มนุษย์สามารถทนได้
วงดนตรีร็อก, โรงงานถลุงเหล็ก	110	
จักรยานยนต์, เครื่องพิมพ์ขนาดใหญ่	100	ปวดหูรุนแรง (8 ชั่วโมง)
ถนนในกรุงเทพฯ ๗ ชั่วโมงเร่งรีบ	90	ปวดหู (8 ชั่วโมง)
รถสินค้า, โรงงานทอผ้า	80	
ถนนซูเปอร์ไฮเวย์, เครื่องดูดฝุ่น	70	รบกวนโสตประสาท
การพูดคุยในภัตตาคาร	60	เริ่มรำคาญ
ชนบท, การพูดคุยในห้องรับแขก	50	เงียบ
ห้องสมุด	40	
ชนบทตอนกลางคืน	30	
เสียงกระซิบ, ใบไม้ร่วง	20	เงียบสงบ
เสียงลมหายใจ	10	
เสียงที่มนุษย์สามารถได้ยิน	0	

ที่มา: มนัส (2530)

แหล่งกำเนิดเสียงที่เกิดจากการกระทำของมนุษย์ แบ่งตามลักษณะของแหล่งกำเนิดเสียงได้ 2 ประเภท คือ แหล่งกำเนิดของเสียงประเภทอยู่กับที่ เป็นเสียงที่เกิดจากสถานประกอบการ โรงงานอุตสาหกรรม อาคารและสิ่งปลูกสร้าง หรือโรงมหรสพ และแหล่งกำเนิดของเสียงที่เคลื่อนที่ได้ เป็นเสียงที่เกิดจากยานพาหนะที่ใช้เครื่องยนต์เกี่ยวกับการคมนาคมขนส่ง (Transportation) เช่น รถยนต์ รถไฟ เรือยนต์ เครื่องบิน รถมอเตอร์ไซด์ หรือเครื่องจักร เช่น เครื่องเจาะ เครื่องขุด เป็นต้น เสียงที่เกิดจากการกระทำของมนุษย์ และมีผลต่อสุขภาพร่างกายหรือจิตใจมนุษย์แบ่งตามแหล่งกำเนิดของเสียงได้เป็น 4 ลักษณะคือ การคมนาคมขนส่ง (Transportation), สิ่งก่อสร้าง อาคารสถานที่ (Construction), อาคารบ้านเรือนที่เกี่ยวข้องกับกิจกรรมบ้าน (House hold appliances and internal building noise) และ โรงงานอุตสาหกรรม (Industry) ซึ่งความความสัมพันธ์ระหว่างแหล่งกำเนิดเสียงและค่าความดังของเสียงนั้นแสดงดังตารางที่ 2

ตารางที่ 2 ระดับเสียงที่เกิดจากแหล่งกำเนิดเสียงชนิดต่างๆ

แหล่งกำเนิดเสียง	dBA	หมายเหตุ
รถบรรทุกทุกสปีด	96.1	วัดที่ระยะ 4.6 เมตรมาตรฐาน
สามล้อเครื่อง	91.8	วัดที่ระยะ 4.6 เมตรมาตรฐาน
จักรยานยนต์	87.8	วัดที่ระยะ 4.6 เมตรมาตรฐาน
เรือยนต์	85-96	วัดที่ระยะ 1 เมตรความถี่ 125-4,000 เฮิรตซ์
โรงงานทอผ้า	3 – 98	ขึ้นอยู่กับลักษณะของแผนก
โรงงานซ่อมเครื่องบิน	71-113	ความถี่ 125-4,000 เฮิรตซ์
โรงงานผลิตท่อน้ำ	97	
โรงงานองค์การแก้ว	84-97	ความถี่ 125-4,000 เฮิรตซ์

ที่มา: มนัส (2530)

มลพิษทางเสียง หมายถึง สภาพแวดล้อมที่มีเสียงดังเกินค่ามาตรฐานที่กรมควบคุมมลพิษกำหนด อันก่อให้เกิดความรำคาญ สร้างความรบกวน ทำให้เกิดความเครียดทั้งทางร่างกายและจิตใจ ทำให้ตกใจ และอาจถึงขั้นเป็นอันตรายต่อสุขภาพอนามัยได้ เช่น เสียงที่ดังมาก หรือเสียงที่ดังต่อเนื่องยาวนานไม่จบสิ้น (วีระเดช, 2540) แหล่งที่ก่อให้เกิดเสียงรบกวนอันเป็นมลพิษทางเสียง ส่วนใหญ่เกิดจากการกระทำของมนุษย์ แหล่งกำเนิดของมลพิษทางเสียงแบ่งได้เป็น 2 ประเภท คือ ประเภทเคลื่อนที่ และประเภทอยู่กับที่

ระดับเสียงที่ปลอดภัยสำหรับการได้ยิน คือ เสียงที่มีความดังไม่เกิน 85 เดซิเบล แต่เมื่อสัมผัสวันละ 8 ชั่วโมง จะได้รับอันตราย ซึ่งอันตรายที่เกิดจากมลพิษของเสียงมักจะไม่เห็นผลโดยทันที แต่ถ้าได้สัมผัสวันละหลายๆ ชั่วโมงเป็นเวลานานๆ ก็อาจจะก่อให้เกิดอันตรายได้ องค์การพิทักษ์สิ่งแวดล้อมอเมริกา (EPA) ได้สรุปว่าผู้ที่ได้ยินเสียงตลอด 24 ชั่วโมง เฉลี่ยเกิน 70 เดซิเบล จะกลายเป็นคนหูตึงภายในเวลา 40 ปี ผลกระทบของค่าความดังของเสียงแต่ละระดับที่มีต่อหูของมนุษย์แสดงดังตารางที่ 3

เสียงที่ดังมากเกินไปเป็นปัญหาสำคัญ และจะเพิ่มอันตรายมากขึ้นทุกที ถ้าสัมผัสนานเกินไป หรือได้ยินติดต่อกันเป็นเวลานานกว่าวันละ 2 ชั่วโมงครึ่ง จะทำให้ประสาทหูเสื่อมได้ และจากการสำรวจเรื่องเสียงจากแหล่งต่างๆ พบว่า เสียงยานพาหนะตามท้องถนนในช่วงกลางวันในกรุงเทพมหานครมีความดังถึง 95 เดซิเบล ในเวลากลางคืน ดังถึง 97 เดซิเบล (วีระเดช, 2540)

ตารางที่ 3 ผลกระทบที่เกิดขึ้นกับหูของมนุษย์ที่ค่าความดังระดับต่างๆ

ผลกระทบที่เกิดขึ้นกับหูของมนุษย์	ค่าความดัง
หูหนวกทันที	เกิดขึ้นจากที่อยู่ในบริเวณที่มีเสียงดังเกิน 120 เดซิเบล เอ
หูอื้อชั่วคราว	เกิดขึ้นเมื่ออยู่ในบริเวณที่มีเสียงระดับ 80 เดซิเบล เอ ขึ้นไป ไปในเวลาไม่นานนัก
หูอื้อถาวร	เกิดขึ้นเมื่ออยู่ในบริเวณระดับความดังมากเป็นระยะ เวลานาน

ที่มา: วีระเดช (2540)

เนื่องจากมลพิษทางเสียงเป็นภัยใกล้ตัวของประชาชนและผู้ปฏิบัติงาน จึงมีกฎหมาย บังคับใช้สำหรับควบคุมมลพิษทางอากาศและเสียง

มาตรา 68 ให้รัฐมนตรีโดยคำแนะนำของคณะกรรมการควบคุมมลพิษมีอำนาจ ประกาศในราชกิจจานุเบกษากำหนดประเภทของแหล่งกำเนิดมลพิษที่จะต้องควบคุม การปล่อย อากาศเสีย รั่วสี หรือมลพิษอื่นใดที่อยู่ในสภาพเป็นควัน ไอ ก๊าซ เขม่าฝุ่น ละออง ฝ้าถ่าน หรือ มลพิษอากาศ ในรูปแบบไดออกไซด์บรรยากาศไม่เกินมาตรฐานควบคุมมลพิษจากแหล่งกำเนิดที่ กำหนดตามมาตรา 55 หรือมาตรฐานที่ส่วนราชการใดกำหนดโดยอาศัยอำนาจตามกฎหมายอื่นและ มาตรฐานนั้นยังผลใช้บังคับตามมาตรา 56 หรือมาตรฐานที่ผู้ว่าราชการจังหวัดกำหนดเป็นพิเศษ สำหรับเขตควบคุมมลพิษตามมาตรา 58

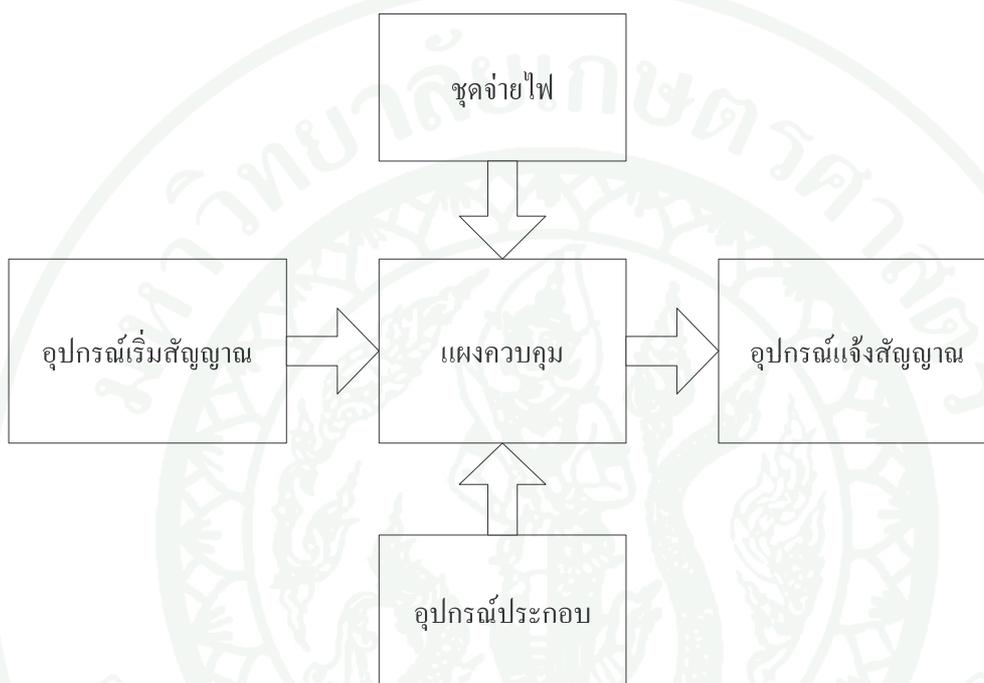
เจ้าของหรือผู้ครอบครองแหล่งกำเนิดมลพิษที่กำหนดตามวรรคหนึ่งมีหน้าที่ต้องติดตั้ง หรือจัดให้มีระบบบำบัดอากาศเสีย อุปกรณ์หรือเครื่องมืออื่นใดสำหรับการควบคุม กำจัด ลด หรือ ขจัดมลพิษ ซึ่งอาจมีผลกระทบต่อคุณภาพอากาศตามที่เจ้าพนักงานควบคุมมลพิษกำหนด เว้นแต่จะ ได้มีระบบ อุปกรณ์หรือเครื่องมือดังกล่าว ที่เจ้าพนักงานควบคุมมลพิษได้ทำการตรวจสอบและ ทดลองแล้วเห็นว่ายังใช้การได้อยู่ทั้งนี้เจ้าพนักงานควบคุมมลพิษจะกำหนดให้มีผู้ควบคุมการ ดำเนินงานระบบบำบัดอากาศเสียอุปกรณ์หรือเครื่องมือดังกล่าวด้วยก็ได้

ให้นำความในวรรคหนึ่งและวรรคสองมาใช้บังคับกับแหล่งกำเนิดที่ปล่อยหรือ ก่อให้เกิดเสียงหรือความสั่นสะเทือนเกินกว่าระดับมาตรฐานควบคุมมลพิษจากแหล่งกำเนิดตาม มาตรา 55 หรือมาตรฐานที่ส่วนราชการใดกำหนดโดยอาศัยอำนาจตามกฎหมายอื่นและมาตรฐาน นั้นยังมีผลใช้บังคับตามมาตรา 56 หรือมาตรฐานที่ผู้ว่าราชการจังหวัดกำหนดเป็นพิเศษสำหรับเขต ควบคุมมลพิษตามมาตรา 58 ด้วย โดยอนุโลม (กรมควบคุมมลพิษ, 2536)

## 2. ระบบเตือนภัยแบบปกติ

ระบบสัญญาณเตือนแบบปกติที่ใช้กันอย่างแพร่หลายโดยส่วนมากมักเป็นระบบ สัญญาณเตือนอัคคีภัยเกิดมาจากความพยายามในการแก้ปัญหาเรื่องชีวิตและทรัพย์สิน อัคคีภัย ก่อให้เกิดความสูญเสียมากสาเหตุส่วนใหญ่มาจากขณะที่เริ่มเกิดเพลิงไหม้จะไม่มีคนอยู่หรือไม่มีผู้ พบเห็นในขณะนั้น ซึ่งกว่าจะรู้ตัวเพลิงก็ลุกลามจนเกินกำลังที่คนไม่ก็คนหรืออุปกรณ์ดับเพลิง

ขนาดเล็กที่มีอยู่ภายในอาคารจะทำการสกดไฟได้ (อภิรักษ์, 2541) ดังนั้นจึงจำเป็นต้องมีระบบสัญญาณเตือนอัคคีภัยติดตั้งไว้ในอาคารเพื่อให้สามารถรับรู้เหตุการณ์ล่วงหน้าก่อนที่ไฟจะลุกลามจนควบคุมไม่ได้ ส่วนประกอบที่สำคัญของระบบเตือนอัคคีภัยมี 5 ส่วนใหญ่ๆ ซึ่งทำงานเชื่อมโยงกันดังแสดงในรูปภาพที่ 1



ภาพที่ 1 ส่วนประกอบของระบบสัญญาณเตือนอัคคีภัยโดยทั่วไป

จากภาพที่ 1 ส่วนประกอบของระบบสัญญาณเตือนอัคคีภัย และระบบเตือนภัยแบบปกติโดยทั่วไปมีส่วนประกอบหลักแบ่งออกเป็น 5 ส่วน ดังนี้

## 2.1 ชุดจ่ายไฟ (Power Supply)

ชุดจ่ายไฟ เป็นอุปกรณ์แปลงกำลังไฟฟ้าของแหล่งจ่ายไฟมาเป็นกำลังไฟฟ้ากระแสตรงที่ใช้ปฏิบัติงานของระบบและจะต้องมีระบบไฟฟ้าสำรองเพื่อให้ระบบทำงานได้ในขณะที่ไฟปกติดับ

## 2.2 แผงควบคุม (Fire Alarm Control Panel)

แผงควบคุมเป็นส่วนควบคุมและตรวจสอบการทำงานของอุปกรณ์ต่างๆ ในระบบทั้งหมดจะประกอบด้วยวงจรควบคุมคอยรับสัญญาณจากอุปกรณ์เริ่มสัญญาณ, วงจรทดสอบการทำงาน, วงจรป้องกันระบบ, วงจรสัญญาณแจ้งการทำงานในสภาวะปกติ และภาวะขัดข้อง เช่น สายไฟจากอุปกรณ์ตรวจจับขาด, แบตเตอรี่ต่ำ หรือไฟจ่ายตู้แผงควบคุมโดนตัดขาด เป็นต้น ตู้ควบคุม (FCP) จะมีสัญญาณไฟและเสียงแสดงสถานะต่างๆ บนหน้าตู้ เช่น Fire Lamp : จะติดเมื่อเกิดเพลิงไหม้, Main Sound Buzzer : จะมีเสียงดังขณะแจ้งเหตุ, Zone Lamp : จะติดค้างแสดงโซนที่เกิด Alarm, Trouble Lamp : แจ้งเมื่อมีเหตุขัดข้อง, Control Switch : สำหรับการควบคุม เช่น เปิดปิดเสียงที่ตู้ และกระดิ่ง, ทดสอบ Battery, Reset ระบบหลังเหตุการณ์เป็นปกติ

## 2.3 อุปกรณ์เริ่มสัญญาณ (Initiating Devices)

อุปกรณ์เริ่มสัญญาณแบ่งออกเป็น 2 ชนิด คือ อุปกรณ์เริ่มสัญญาณจากบุคคล (Manual Station) ได้แก่ สถานีแจ้งสัญญาณเตือนอัคคีภัยแบบใช้มือกด (Manual Push Station) และอุปกรณ์เริ่มสัญญาณโดยอัตโนมัติเป็นอุปกรณ์อัตโนมัติที่มีปฏิกิริยาไวต่อสภาวะ ตามระยะต่างๆ ของการเกิดเพลิงไหม้ ได้แก่ อุปกรณ์ตรวจจับควัน (Smoke Detector) อุปกรณ์ตรวจจับความร้อน (Heat Detector) อุปกรณ์ตรวจจับเปลวไฟ (Flame Detector) อุปกรณ์ตรวจจับแก๊ส (Gas Detector)

## 2.4 อุปกรณ์แจ้งสัญญาณด้วยเสียงและแสง (Audible & Visual Signalling Alarm Devices)

หลังจากอุปกรณ์เริ่มสัญญาณทำงานโดยส่งสัญญาณมายังตู้ควบคุม (FCP) แล้ว FCP จึงส่งสัญญาณออกมาโดยผ่านอุปกรณ์ ได้แก่ กริ่งเตือนภัย, ไฟสัญญาณ เป็นต้น เพื่อให้ผู้อยู่อาศัย, ผู้รับผิดชอบหรือเจ้าหน้าที่ดับเพลิงได้ทราบว่าเกิดเหตุเพลิงไหม้เกิดขึ้น

## 2.5 อุปกรณ์ประกอบ (Auxiliary Devices)

เป็นอุปกรณ์ที่ทำงานเชื่อมโยงกับระบบอื่นที่เกี่ยวข้องกับการควบคุมป้องกันและดับเพลิงโดยจะถ่ายทอดสัญญาณระหว่างระบบเตือนอัคคีภัยกับระบบอื่น เช่น ส่งสัญญาณ

กระตุ้นการทำงาน ของระบบบังคับลิฟท์ลงชั้นล่าง, การปิดพัดลมในระบบปรับอากาศ, เปิดพัดลมในระบบระบายอากาศ, การควบคุมเปิดประตูทางออก, เปิดประตูหนีไฟ, ปิดประตูกันควันไฟ, ควบคุมระบบกระจายเสียง และการประกาศแจ้งข่าว, เปิดระบบดับเพลิง หรือรับสัญญาณของระบบอื่นมากระตุ้นการทำงาน ของระบบสัญญาณเตือนอัคคีภัย เช่น จากระบบพ่นน้ำป้อนดับเพลิงระบบดับเพลิงด้วยสารเคมีชนิดอัตโนมัติ เป็นต้น

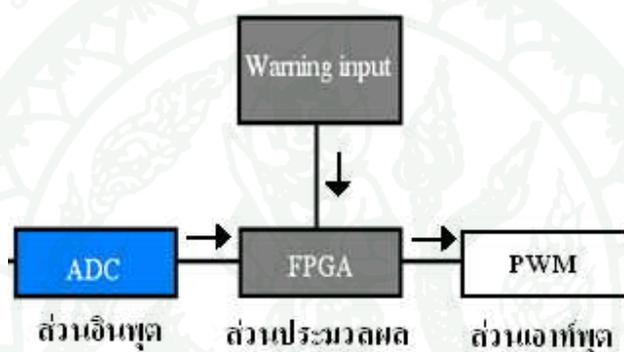
การออกแบบระบบสัญญาณเตือนอัคคีภัยมีปัจจัยที่ต้องพิจารณาในการออกแบบหลัก 4 อย่างคือ ความสูงของเพดาน, สภาพแวดล้อม, ระดับความสำคัญกับความเสียง และงบประมาณที่ตั้งไว้ ส่วนการจัดแบ่งโซนทำให้สามารถค้นหาจุดเกิดเหตุได้เร็วขึ้น หมายถึงความสามารถในการรับเหตุก็จะมากขึ้นด้วย ดังนั้นการจัดโซนจึงเป็นความสำคัญในการออกแบบระบบ Fire Alarm กรณีเกิดเหตุเริ่มต้นจะทำให้กริ่งเตือนภัยดังเฉพาะโซนนั้นๆ ถ้าคุมสถานการณ์ไม่ได้ก็จะสั่งให้กริ่งเตือนภัยโซนอื่นๆ ดังตาม (ประกอบ, 2551) แนวทางการแบ่งโซนมีดังนี้ ต้องจัดโซนอย่างน้อย 1 โซนต่อ 1 ชั้น, แบ่งตามความเกี่ยวข้องของพื้นที่ ซึ่งเป็นที่เข้าใจสำหรับคนในอาคารนั้น เช่น โซน Office หรือ โซน Workshop, ถ้าเป็นพื้นที่ราบบริเวณกว้างจะแบ่งประมาณ 600 ตารางเมตร ต่อ 1 โซน เพื่อสามารถมองเห็นหรือค้นพบจุดเกิดเหตุโดยเร็ว, คนที่อยู่ในโซนใดๆ ต้องสามารถได้ยินเสียงกริ่งเตือนภัยในโซนนั้นๆ ได้อย่างชัดเจน

### 3. ส่วนประกอบทางด้านระบบอิเล็กทรอนิกส์เตือนภัยแบบดัดแปลงเสียง

ระดับความดังของเสียงสัญญาณเตือนภัยในปัจจุบันนับว่าเป็นแหล่งกำเนิดมลพิษทางเสียงชนิดหนึ่งเนื่องจากมีความดังประมาณ 120 dB ตลอดเวลา ดังนั้นในการแก้ปัญหามลพิษทางเสียงที่เกิดจากเสียงของตัวกริ่งเตือนภัยเอง ระบบอิเล็กทรอนิกส์ได้ถูกนำมาใช้ในการปรับระดับเสียงกริ่งเตือนภัยให้ไม่จำเป็นต้องมีความดังที่ระดับเดียวกันตลอดเวลา แต่ควรมีความดังมากกว่าเสียงของสภาวะแวดล้อมในขณะนั้นเพียงแคให้ผูปฏิบัติงานได้ยินก็เพียงพอ เพื่อลดมลพิษทางเสียงที่มีผลต่อสภาพร่างกายและจิตใจของผูปฏิบัติงานให้เกิดขึ้นน้อยที่สุดและให้เป็นไปตามมาตรฐานของกรมควบคุมมลพิษ (กรมควบคุมมลพิษ, 2536)

หลักการทำงานของระบบอิเล็กทรอนิกส์เตือนภัยแบบดัดแปลงเสียง คือ เมื่อมีการกดปุ่มสัญญาณเตือนภัย หรือเซนเซอร์ชนิดต่างๆ ตรวจจับสิ่งผิดปกติที่เกิดขึ้น (Warning input) ระบบเตือนภัยแบบดัดแปลงเสียงจะเริ่มทำงานโดยสัญญาณเสียงจากสิ่งแวดล้อม (Background Noise) จะ

ผ่านมาจากไมโครโฟนอินพุต และถูกแปลงสัญญาณจากสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC) เพื่อให้ FPGA (Field-Programmable Gate Array) ประมวลผลแล้วสร้างสัญญาณดิจิทัลโดยใช้เทคนิค PWM (Pulse Width Modulation) ปรับความกว้างของสัญญาณพัลส์ (Michael Barr, 2001) และส่งสัญญาณไปยังส่วนเอาต์พุตเพื่อควบคุมการเปิดและปิดของวงจรขับเคลื่อนทำให้ระบบเตือนภัยแบบตัดแปลงเสียงสามารถปรับระดับความดังของกริ่งเตือนภัยให้ดังกว่าเสียงของสถานะแวดล้อมในขณะนั้นประมาณ 5 dB - 10 dB เพียงแค่ให้ผู้ปฏิบัติงานได้ยินก็เพียงพอ เพื่อลดมลพิษทางเสียงที่อาจเกิดจากตัวกริ่งเตือนภัยเอง โดยงานวิจัยชิ้นนี้แบ่งส่วนประกอบหลักของระบบเป็น 3 ส่วนคือ ส่วนอินพุต ส่วนประมวลผล และส่วนเอาต์พุต แสดงดังภาพที่ 2



ภาพที่ 2 ส่วนประกอบระบบอิเล็กทรอนิกส์เตือนภัย

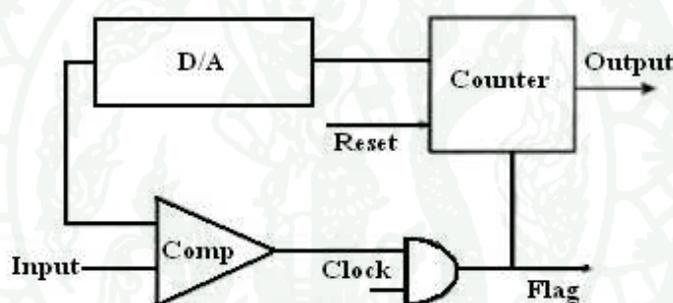
จากภาพที่ 2 แสดงส่วนประกอบหลักของการประยุกต์ใช้ระบบอิเล็กทรอนิกส์มาแก้ปัญหามลพิษทางเสียง โดยมีส่วนประกอบหลัก 3 ส่วนดังนี้

### 3.1 ทฤษฎีการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล

สัญญาณทางไฟฟ้าที่ใช้ในอุปกรณ์อิเล็กทรอนิกส์มี 2 ชนิด คือ สัญญาณอนาล็อก และสัญญาณดิจิทัล สัญญาณอนาล็อกจะใช้ในอุปกรณ์ทั่วไป และใช้ในการควบคุมแบบเก่า ในปัจจุบันมีไมโครโปรเซสเซอร์ และไมโครคอนโทรลเลอร์เข้ามาช่วยในการควบคุมอุปกรณ์ต่างๆ มากมาย ซึ่งทำให้การควบคุมนั้นทำได้ง่าย และรวดเร็วยิ่งขึ้นแต่ในการควบคุมนั้นจำเป็นต้องใช้สัญญาณดิจิทัลในการติดต่อกับไมโครโปรเซสเซอร์ หรือไมโครคอนโทรลเลอร์ ซึ่งอินพุตทางสิ่งแวดล้อมต่างๆยังเป็นสัญญาณอนาล็อกอยู่ ดังนั้นจึงจำเป็นต้องมีการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล แล้วจึงนำสัญญาณดิจิทัลนั้นเข้ามาสู่ไมโครโปรเซสเซอร์

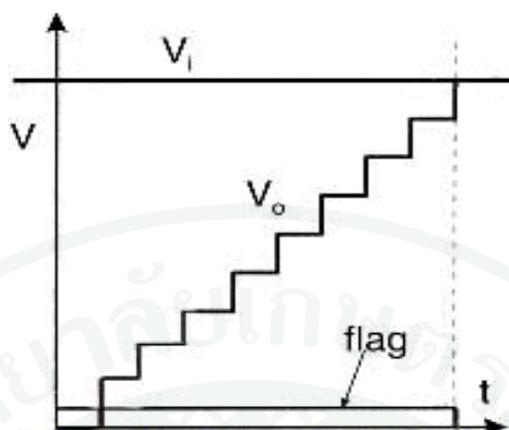
หรือไมโครคอนโทรลเลอร์เพื่อใช้ควบคุมระบบต่อไป (ปนัดดา, 2541) แม้ว่าสัญญาณอนาล็อกนั้นมีความแน่นอน และแม่นยำสูง แต่สัญญาณอนาล็อกนั้นก็ควบคุมได้ยากเนื่องจากในสภาพแวดล้อม มีสัญญาณรบกวนอยู่มาก และการที่จะทำให้การควบคุมแบบอนาล็อกมีความสามารถในการควบคุมเท่ากับการควบคุมแบบดิจิทัลนั้นทำได้ยาก เนื่องจากวงจรควบคุมแบบอนาล็อกจะต้องมีความซับซ้อนสูงอย่างไรก็ตาม สัญญาณดิจิทัลก็ไม่สามารถทดแทนความละเอียดของสัญญาณอนาล็อกได้อย่างสมบูรณ์ แต่ทำให้การควบคุมนั้นทำให้ง่าย และสะดวกยิ่งขึ้น

Counting Converter เป็นวิธีที่ง่ายที่สุดของการแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล โดยการนับค่าเพิ่มขึ้นเรื่อยๆ แล้วนำผลที่ได้จากการนับไปเปรียบเทียบกับค่าที่ต้องการที่ตั้งไว้ ลักษณะการทำงานเป็นดังภาพที่ 3



ภาพที่ 3 วงจร Counting Converter

วงจร Counter เป็นอุปกรณ์นับค่าที่เพิ่มขึ้นทีละหนึ่ง แล้วส่งค่าที่ได้ให้ D/A มีขา Reset รับสัญญาณ Reset เมื่อต้องการให้เริ่มนับใหม่ เมื่อ D/A รับค่าที่นับเพิ่มขึ้นทีละหนึ่งจากตัวนับ ก็แปลงค่าให้เป็นสัญญาณอนาล็อกที่มีค่าความต่างศักย์ค่าๆหนึ่ง แล้วส่งต่อเข้าไปที่อุปกรณ์ตัวเปรียบเทียบ(Comparator) เพื่อทำการเปรียบเทียบค่าความต่างศักย์ของอินพุต และค่าจากที่ตัวนับ ถ้าหากทั้งสองสัญญาณมีค่าเท่ากันจะส่งค่าความต่างศักย์ 0 โวลต์ออกมา (ลอจิก 0) ถ้าไม่เท่ากันก็จะส่งความต่างศักย์ที่ไม่ใช่ 0 โวลต์ออกมา (ลอจิก 1) ซึ่งค่าความต่างศักย์ที่ออกมาจะนำมาเข้าลอจิกเกต ชนิดแอนด์เกตเพื่อทำการ “และ” กับสัญญาณนาฬิกาซึ่งจะได้ค่าลอจิกออกมา ถ้าผลลัพธ์ออกมาเป็นสัญญาณนาฬิกาแสดงว่ายังไม่ได้ผลลัพธ์เท่าที่ต้องการสัญญาณนาฬิกาจะไปทำให้ตัวนับนับเพิ่มขึ้นต่อไป และเมื่อได้ค่าผลลัพธ์ดิจิทัลที่ต้องการแล้ว ค่าที่ได้จากตัวเปรียบเทียบจะให้ค่าความต่างศักย์เป็น 0 (ลอจิก 0) ซึ่งเมื่อนำมาเข้าลอจิกเกต "และ" กับสัญญาณนาฬิกาแล้ว ก็จะทำให้ลอจิก 0 ซึ่งทำให้ตัวนับไม่นับเพิ่มอีก ก็จะได้ค่าดิจิทัลจากตัวนับที่ต้องการแสดงดังภาพที่ 4



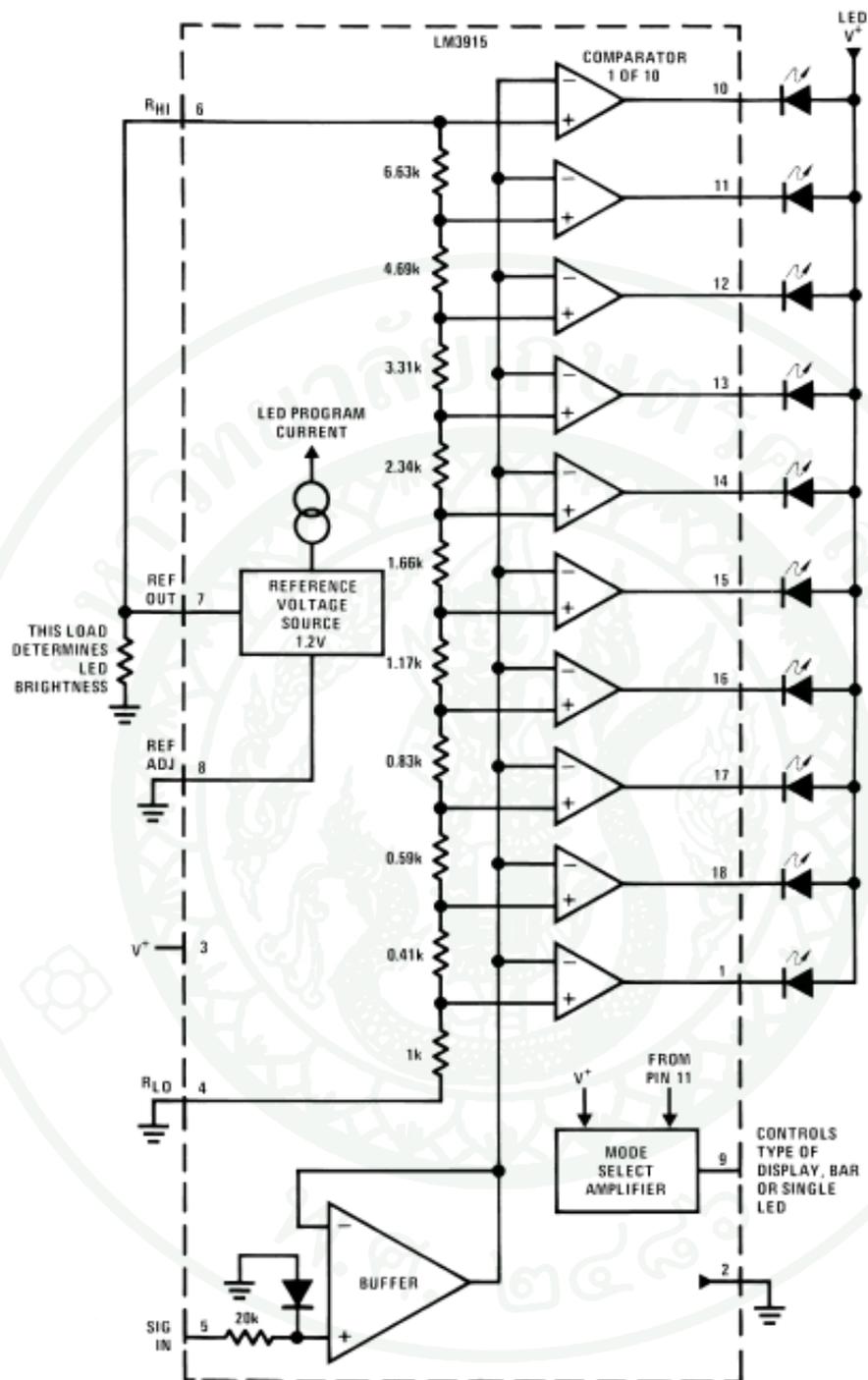
ภาพที่ 4 กราฟเอาต์พุตของวงจร Counter เทียบกับแกนเวลา

ที่มา: ปณัฒนา แสงเทียนชัย การแปลงสัญญาณอนาล็อก-ดิจิตอล 2541

วงจรที่ใช้ในการแปลงสัญญาณอนาล็อกเป็นดิจิตอล โดยทั่วไปแล้ววงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลมีใช้งานอยู่ประมาณ 7 ชนิด (มูลนิธิวิวัฒน์, 2552) คือ

- 1) Parallel Comparator, Simultaneous, หรือ Flash A/D converter
- 2) Single – Ramp หรือ Single – Slope A/D converter
- 3) Dual – Slope A/D converter
- 4) Charge balance A/D converter
- 5) A/D converters using Counters and D/A converters
- 6) Tracking A/D converters
- 7) Successive – Approximation A/D converters

โดยส่วนของอินพุตในงานวิจัยจะทำการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล โดยเลือกใช้ IC (Integrated Circuit) เบอร์ LM3915 ซึ่งเป็นวงจร Parallel Comparator ในตัวเองโครงสร้างภายในของ IC เบอร์ LM3915 แสดงดังรูปภาพที่ 5 เพื่อทำการแปลงเสียงจากสิ่งแวดล้อมโดยแปลงสัญญาณเสียงที่ได้รับจากไมโครโฟนซึ่งเป็นสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลซึ่งเอาต์พุตของ IC LM3915 จาก Datasheet นั้นมีความสัมพันธ์ของแรงดัน (V) กับค่าความดัง (dB) ดังตารางที่ 4



ภาพที่ 5 โครงสร้างภายในของ IC LM3915 ซึ่งแสดงให้เห็นว่ามีวงจร Parallel Comparator ในการแปลงสัญญาณอนาล็อกเป็นดิจิทัลประกอบอยู่ด้วย

ที่มา: National Semiconductor (2001)

ตารางที่ 4 แสดงความสัมพันธ์ระหว่าง แรงดันไฟฟ้าเอาต์พุต (V) ของ IC LM3915 กับค่าความดังของเสียง (dB)

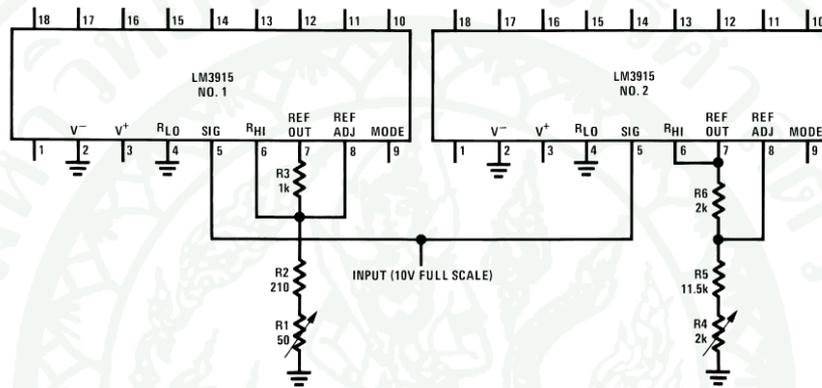
OUTPUT	dB	V <sub>OUT,MIN</sub> (V)	V <sub>OUT,Typ</sub> (V)	V <sub>OUT,MAX</sub> (V)
1	-27	0.422	0.447	0.531
2	-24	0.596	0.631	0.75
3	-21	0.841	0.891	1.059
4	-18	1.189	1.259	1.413
5	-15	1.679	1.778	1.995
6	-12	2.372	2.512	2.819
7	-9	3.35	3.548	3.825
8	-6	4.732	5.021	5.309
9	-3	6.683	7.079	7.498
10	0	9.985	10	10.015

ที่มา : National Semiconductor (2001)

โดยค่าสัมประสิทธิ์ในตารางที่ 4 มีค่าความสัมพันธ์ต่างๆดังนี้

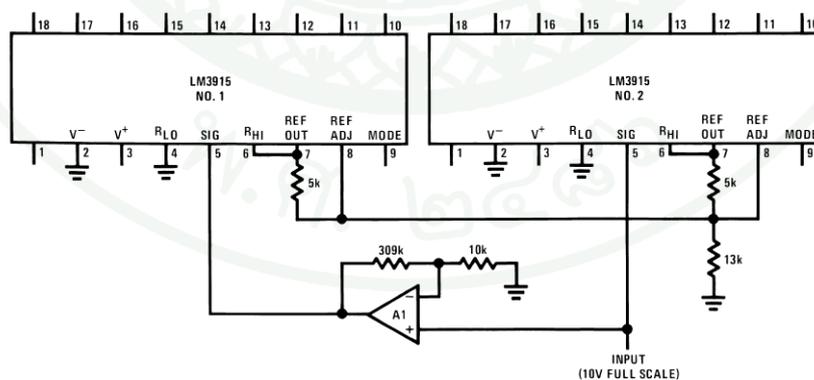
- 1) OUTPUT คือ ช่องสัญญาณเอาต์พุตของ IC
- 2) dB คือ ระดับความดังของเสียง
- 3) V<sub>OUT,MIN</sub> คือ ค่าความต่างศักย์ที่มีค่าต่ำที่สุดของช่องสัญญาณนั้น
- 4) V<sub>OUT,Typ</sub> คือ ค่าความต่างศักย์ที่ IC ทำงานได้ดีที่สุดในช่องสัญญาณนั้น
- 5) V<sub>OUT,MAX</sub> คือ ค่าความต่างศักย์ที่มีค่ามากที่สุดที่เป็นไปได้ของช่องสัญญาณนั้น

การใช้งาน IC LM 3915 เพียงตัวเดียวสามารถรับอินพุตได้เพียง 10 ระดับ แตกต่างกันระดับละ 3 dB จึงมีช่วงในการใช้งานได้แค่เพียง 30 dB เช่น รับเสียงที่มีอินพุตตั้งแต่ 0 dB – 30 dB, 30 dB – 60 dB หรือ 90 dB – 120 dB เป็นต้น แต่อินพุตของเสียงจากสิ่งแวดล้อมในโครงการนี้ควรจะรับค่าของเสียงได้ตั้งแต่ช่วง 60 dB – 120 dB เป็นอย่างน้อย จึงต้องมีการประยุกต์ใช้ IC LM3915 มาต่อกันแบบแคสเคด 2 ตัว และต่อกันโดยประยุกต์ใช้ Op-Amp ในการขยายสัญญาณอินพุต โดยวิธีการต่อ IC LM3915 แสดงดังภาพที่ 6 และภาพที่ 7 ตามลำดับ



ภาพที่ 6 การต่อวงจรสำหรับใช้งาน IC LM3915 จำนวน 2 ตัว แบบแคสเคด

ที่มา: National Semiconductor (2001)



ภาพที่ 7 การต่อวงจรสำหรับใช้งาน IC LM3915 จำนวน 2 ตัว โดยประยุกต์ใช้ Op-Amp ในการขยายสัญญาณอินพุต

ที่มา: National Semiconductor (2001)

### 3.2 ส่วนประมวลผลทฤษฎีเบื้องต้นเกี่ยวกับ FPGA

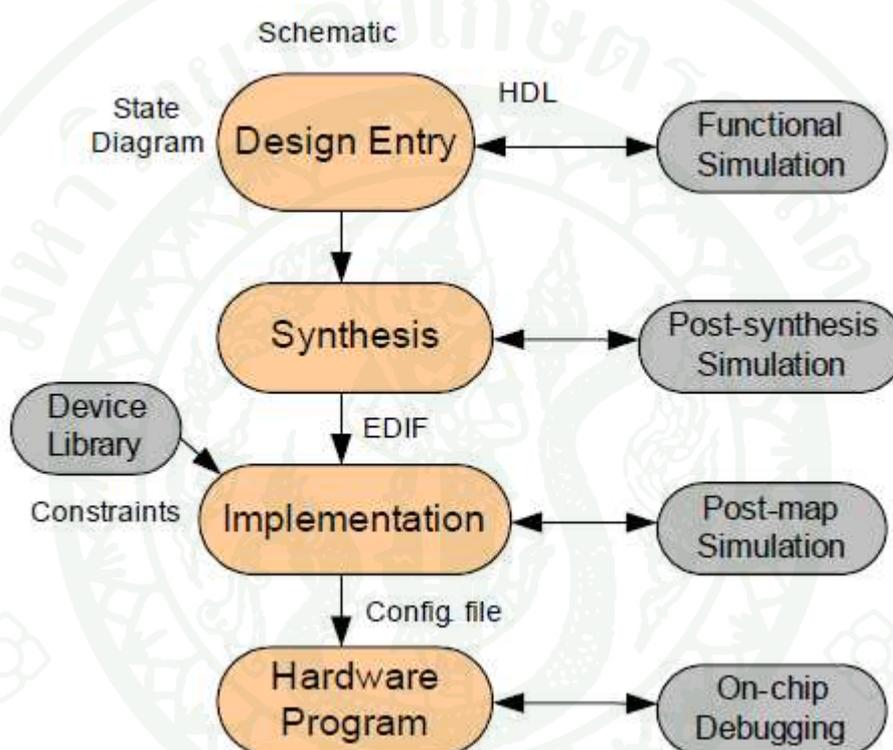
เอฟพีจีเอ (FPGA) ถูกประดิษฐ์ขึ้นตัวแรกในปี ค.ศ. 1985 เพื่อขยายความสามารถให้มากกว่า PLA/PLD (Programmable logic array/Programmable logic device) ในแง่ของจำนวนความจุของเกต และความเร็วสำหรับการรองรับการใช้งานในระบบที่มีความซับซ้อนมากขึ้นเรื่อยๆ ตัว FPGA เองนั้นพูดได้ว่าเป็นการผสมผสานระหว่างฮาร์ดแวร์และซอฟต์แวร์ นั่นคือสามารถนำมาใช้งานเป็นฮาร์ดแวร์ที่สามารถโปรแกรม (Programmable) หรือเปลี่ยนลักษณะของการทำงานได้ (Reconfigurable) เหมือนซอฟต์แวร์ ทำให้มีความยืดหยุ่นในการใช้งานสูง (ธีรยศ, 2548) FPGA จัดเป็น อุปกรณ์สารกึ่งตัวนำชนิดโปรแกรมได้ที่มีโครงข่ายการเชื่อมต่อภายในแบบแมตริกซ์ โครงสร้างภายในของ FPGA นั้นสามารถโปรแกรมให้มีหน้าที่การทำงานเหมือนลอจิกเกตพื้นฐาน เช่น AND, OR, XOR, NOT หรือรวมกันหลายๆ ชนิด (combinational logic) เพื่อให้ทำหน้าที่ที่มีความซับซ้อนเพิ่มขึ้น เช่น decoders หรือฟังก์ชันทางคณิตศาสตร์ ใน FPGA ทั่วไป นอกจากจะประกอบด้วยส่วนของวงจรถลอจิกแบบโปรแกรมได้แล้ว ยังมีบล็อกของหน่วยความจำแสดงดังตารางที่ 5 ซึ่งอาจจะสร้างด้วยฟลิปฟล็อปอย่างง่าย หรือใช้พื้นที่ของสารกึ่งตัวนำสร้างเป็นหน่วยความจำจริงๆ อยู่ภายในก็ได้

ตารางที่ 5 ความสามารถของชิป FPGA

Device	Xilinx Virtex4
จำนวนขา I/O Pin	960
จำนวนลอจิกเซลล์	> 200,000
RAM ภายใน	10Mbits
DSP block (18*18 mult/accu)	192
ตัวจัดการสัญญาณนาฬิกา	20
ความเร็วในการส่งข้อมูลแบบอนุกรม	622Mb/s - 11.1Gb/s

ที่มา: ธีรยศ (2548)

การออกแบบวงจรดิจิทัลด้วย FPGA นั้นเป็นการออกแบบฮาร์ดแวร์ โดยเป็นการทำงานแบบให้ลำดับความสำคัญเท่าๆกันทุกโพรเซส ส่วนการทำงานของไมโครคอนโทรลเลอร์นั้นเป็นการทำงานแบบให้ความสำคัญของโพรเซสก่อนหลัง ซึ่งตรงจุดนี้ทำให้ไมโครคอนโทรลเลอร์มีการประมวลผลการทำงานที่ช้ากว่า FPGA มาก หลักการออกแบบและใช้งาน FPGA โดยทั่วไปมีองค์ประกอบ 4 ส่วนหลัก ๆ ด้วยกันดังภาพที่ 8



ภาพที่ 8 ทฤษฎีขั้นตอนการออกแบบ FPGA มีทั้งหมด 4 ขั้นตอน

ที่มา: ธีรยศ (2548)

จากภาพที่ 8 แสดงให้เห็นว่า Design Entry เป็นขั้นตอนของการรับข้อมูลของการออกแบบเข้าไปในระบบ โดยทั่วไปสามารถทำได้หลายทาง เช่น โดยใช้ Schematic design entry ดึงอุปกรณ์มาจาก Library ของ FPGA หรือโดยใช้ภาษา HDL อย่างเช่น VHDL หรือ Verilog ซึ่งการออกแบบโดยใช้ภาษาชั้นสูงนั้นมีข้อดีก็คือ ไม่ขึ้นกับเทคโนโลยี หรือตัวชิปที่จะถูกโปรแกรมดังนั้นผู้ออกแบบไม่จำเป็นต้องกังวลถึงอุปกรณ์ที่มากับเทคโนโลยีนั้น นอกจากนั้นการอธิบายการทำงานโดยใช้ภาษาชั้นสูง ผู้ออกแบบไม่จำเป็นต้องรู้ถึงลักษณะการเชื่อมต่อของวงจร

ตัวอย่างเช่นถ้าต้องการสร้างตัวบวก (Adder) ที่ทำการบวกข้อมูล a กับ b เข้าด้วยกัน ก็เพียงแค่เพิ่มบรรทัด  $z = a+b$ ; เข้าไปเท่านั้น ส่วนการทดสอบความถูกต้องเป็นในลักษณะการตรวจสอบระดับฟังก์ชันการทำงาน (Functional simulation) โดยใช้ซอฟต์แวร์สำหรับจำลองการทำงาน ซึ่งจะไม่มีการเชื่อมโยงกับดีเลย์ให้เห็น จากนั้นขั้นตอนการ Synthesis ขั้นตอนนี้จะเกี่ยวข้องกับการแปลงแบบที่ได้จาก Schematic หรือ HDL ให้เป็นวงจรลอจิก โดยจะมีขั้นตอนย่อยๆ คือการสังเคราะห์วงจร (Logic synthesis) ซึ่งโดยปกติแล้วจะมีขั้นตอนการ optimization ด้วยเพื่อให้ได้วงจรที่ใช้ทรัพยากรน้อยที่สุด หรือทำงานได้เร็วที่สุด ผลลัพธ์จากขั้นตอนนี้จะเป็น EDIF (Electronic design intermediate form) ซึ่งเป็นไฟล์ที่อธิบายการเชื่อมต่อ (Netlist) และยังไม่ขึ้นกับเทคโนโลยี นั่นหมายความว่า เป็นมาตรฐานที่ตัวเครื่องมือในการออกแบบ (Design tools) ส่วนมากจะสามารถเข้าใจไฟล์ชนิดนี้ได้ การจำลองการทำงานในขั้นตอนนี้จะเป็นลักษณะของ Post-synthesis simulation ซึ่งจะไม่มีการเชื่อมโยงกับดีเลย์ให้เห็นเช่นกัน แต่ผลการจำลองจะได้มาจาก EDIF ไฟล์แล้วจึง Implementation หลังจากได้ Netlist file มาแล้ว จะเป็นการแมป (Map) ให้เข้ากับเทคโนโลยี หรือตัวชิปที่จะใช้งาน (Technology mapping) เช่นโดยส่วนมากแล้ว FPGA จะเป็นเทคโนโลยีแบบ LUT (Look-up table) และหลังจากนั้นก็ทำการวางตำแหน่ง (Placement) ของลอจิกตามทีลอจิกเซลล์ต่างๆ แล้วทำการเชื่อมต่อสายสัญญาณ (Routing) ก่อนที่จะได้ไฟล์สำหรับโปรแกรมลงชิป (Reconfiguration file) การจำลองการทำงานในระดับนี้จะสามารถดูดีเลย์ตามจุดต่างๆ ได้หลังจากมีการสร้างขึ้นจริงๆ ตามเทคโนโลยี หรือตัวชิปที่จะใช้งาน และทำการ Hardware Program การโปรแกรมอุปกรณ์ หรือชิป FPGA นั้นสามารถทำได้เองได้ทันทีรวมทั้งการปรับเปลี่ยนแก้ไขได้ตลอดเวลา โดยกลับไปทำตั้งแต่ต้นก่อนที่จะทำการโปรแกรมทดสอบการทำงานจริงๆ ได้โดยไม่มีค่าใช้จ่ายหรือความเสียหายเกิดขึ้น ซึ่งเหมือนกับไมโครคอนโทรลเลอร์นั่นเอง ดังนั้นชิป FPGA จึงอาจถูกเรียกว่าฮาร์ดแวร์ที่สามารถโปรแกรม(Reprogrammable hardware) ได้

เมื่อพิจารณาประสิทธิภาพทางด้านความเร็วในการประมวลผลต่างๆ FPGA นั้นสูงกว่ามากเมื่อเทียบกับไมโครคอนโทรลเลอร์ที่ทำงานระดับไม่ถึงร้อยเมกะเฮิรตซ์ โดยเปรียบเทียบ FPGA ทำงานที่ความถี่ 300 MHz นั้นเร็วกว่า 1-GHz 32-bit processor (ที่ต้องทำการประมวลผลโปรแกรมทีละบรรทัด) หลายเท่า เนื่องจากการทำงานบน FPGA นั้นเป็นลักษณะของฮาร์ดแวร์ที่สามารถทำได้พร้อม ๆ กันเป็นแบบขนาน ส่วนเรื่องของราคาที่จะสูงไปตามประสิทธิภาพ แต่ถ้าเทียบเวลาในการพัฒนาจากแนวความคิดเป็นผลิตภัณฑ์ (Time-to-market) คิดว่าใกล้เคียงกัน เพราะสามารถออกแบบโดยใช้ภาษาขั้นสูงเหมือนกันทำให้ใช้เวลาค่อนข้างน้อย

ความสามารถทางด้านต่าง ๆ ระหว่างไมโครคอนโทรลเลอร์กับ FPGA นั้นสามารถสรุปได้ในตารางที่ 6 โดยที่มีตัวอุปกรณ์ประเภท ASIC (Application specific IC) มาเปรียบเทียบกับเพิ่มเติม

ตารางที่ 6 แสดงความสามารถ ประสิทธิภาพด้านต่างๆ ระหว่างไมโครคอนโทรลเลอร์ FPGA และ ASIC (Application Specific IC)

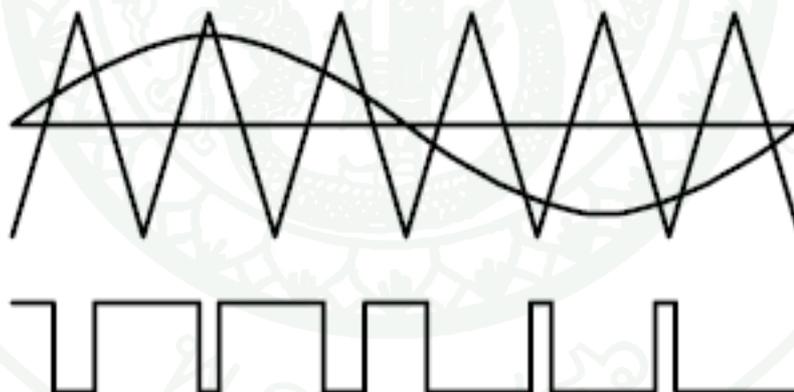
ประสิทธิภาพต่างๆ	Microcontroller	FPGA	ASIC
ความเร็ว	ช้า	เร็ว	เร็วมาก
พลังงาน	น้อย	ปานกลาง	น้อยมาก
ราคา	น้อยมาก	ปานกลาง	มาก
เวลาพัฒนา	น้อย	น้อย	มาก
ความยืดหยุ่น	มาก	ปานกลาง – มาก	น้อย

ที่มา : ธีรยศ (2548)

จากข้อได้เปรียบเสียเปรียบต่างๆ จึงทำให้อุปกรณ์แต่ละชนิดมีความเหมาะสมกับงานประเภทต่างๆ กัน ถ้าหากเป็นงานที่ไม่ต้องการความเร็วมาก และมีงบประมาณที่ค่อนข้างจำกัด เช่น ชุดควบคุมอุปกรณ์ขนาดเล็กก็ควรจะใช้ไมโครคอนโทรลเลอร์ที่มีราคาข้อมเยา แต่ถ้าเป็นงานที่ต้องการความเร็วสูง การส่งผ่านข้อมูลที่เร็ว และใช้การประมวลผลข้อมูลที่มาก เช่น งานทางด้าน DSP (Digital signal processing) งานทางด้านการประมวลผลภาพ (Image processing) การบีบอัดข้อมูล (Datacompression) ก็ควรที่จะเลือกใช้ FPGA ซึ่งปัจจุบันมีการรวมเอาอุปกรณ์ต่างๆ เข้าไว้ในชิป (RISC processor, Memory, DSP block, Multi-gigabit transceiver) ทำให้สามารถออกแบบเป็นลักษณะของระบบใหญ่บนชิปเดี่ยวได้ (System on a programmable chip)

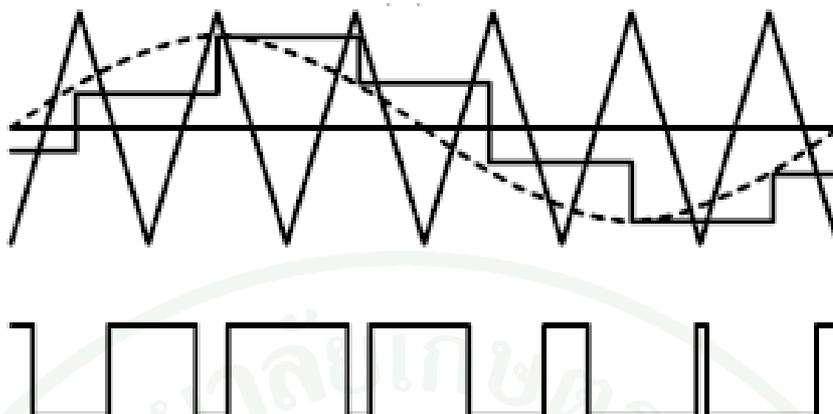
### 3.3 ส่วนเอาต์พุตการประยุกต์ใช้เทคนิค PWM

สัญญาณ PWM ได้มีการนำไปใช้กันอย่างกว้างขวางทั้งทางด้านระบบสื่อสาร โดยเฉพาะอย่างยิ่งการสื่อสารทางแสง เนื่องจาก PWM เป็นกระบวนการมอดูเลตที่สำคัญแบบหนึ่ง (มนตรี, 2539) หรือทางด้านอิเล็กทรอนิกส์กำลัง เพื่อใช้เป็นสัญญาณกระตุ้นการนำกระแสของ อุปกรณ์อิเล็กทรอนิกส์กำลัง หรือในระบบเครื่องมือวัดเพื่อใช้วัดข้อมูลต่างเฟส หรือตัวประกอบกำลัง (Power factor) ในปัจจุบันจึงมีการผลิตวงจรสร้างสัญญาณ PWM ให้อยู่ในรูปแบบวงจรรวม เพื่อความสะดวกในการใช้งาน การกำเนิดสัญญาณ PWM ตามปกติ สัญญาณ PWM สามารถแบ่งได้ เป็น 2 รูปแบบ สัญญาณ PWM แบบแรกคือ สัญญาณ PWM ที่เกิดจากการสุ่มแบบธรรมชาติ (Natural sampling) โดยสัญญาณเรียกว่า NPWM (Natural Pulse Width Modulation) แสดงดังภาพที่ 9 สัญญาณ PWM อีกแบบหนึ่งคือ สัญญาณ PWM ที่เกิดจากการสุ่มแบบยูนิฟอร์ม (Uniform sampling) จะเรียกสัญญาณนี้ว่า UPWM (Uniform Pulse Width Modulation) แสดงดังภาพที่ 10 โครงสร้างของการสร้างสัญญาณ PWM ทั้งสองแบบแสดงดังภาพที่ 11



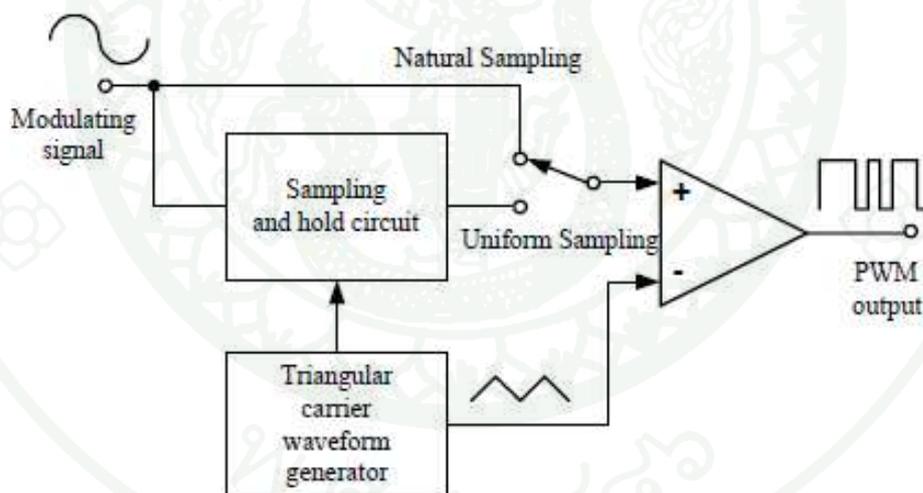
ภาพที่ 9 สัญญาณ PWM แบบ NPWM

ที่มา: มนตรี (2539)



ภาพที่ 10 สัญญาณ PWM แบบ UPWM

ที่มา: มนตรี (2539)



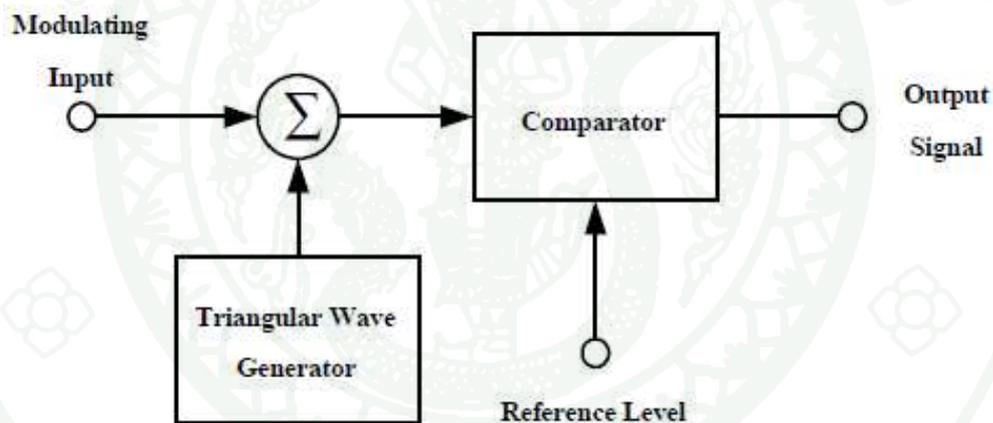
ภาพที่ 11 โครงสร้างของการสร้างสัญญาณ PWM ทั้งสองแบบ

ที่มา: มนตรี (2539)

กรณีที่เป็นการกำเนิดสัญญาณ NPWM สัญญาณข่าวสาร (Modulating signal) จะถูกเปรียบเทียบแรงดันกับสัญญาณสามเหลี่ยมซึ่งเป็นสัญญาณพาห้ เอาท์พุทของวงจรเปรียบเทียบแรงดันก็จะเป็นสัญญาณ NPWM ส่วนในกรณีของสัญญาณ UPWM สัญญาณข่าวสาร

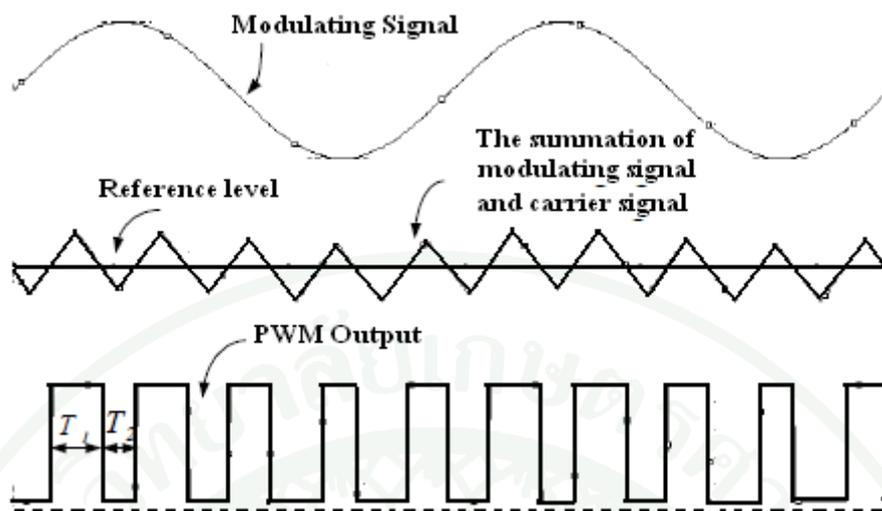
จะผ่านวงจรสุ่มและคงค่าสัญญาณ (Sample and hold) ก่อนเข้าเปรียบเทียบกับสัญญาณสามเหลี่ยม จึงได้สัญญาณ PWM ทั้งสองแบบ สำหรับสัญญาณพาห้สามเหลี่ยมมีอยู่ 2 รูปแบบคือ สัญญาณฟันเลื่อย ซึ่งเรียกว่าสัญญาณ single sided เนื่องจากมีเพียงขอบสัญญาณขอบเดียวที่ถูกมอดูเลต อีกแบบหนึ่งเป็นสัญญาณสามเหลี่ยม ซึ่งจะเรียกว่าสัญญาณ double sided

โครงสร้างของการกำเนิดสัญญาณ PWM ที่ทั่วไปแสดงดังภาพที่ 12 ถ้าหากสัญญาณข่าวสารเป็นสัญญาณรูปคลื่น Sine เหมือนกับสัญญาณด้านบนของภาพที่ 13 ผลรวมทางขนาดของสัญญาณพาห้สามเหลี่ยม และสัญญาณข่าวสารจะได้ตามสัญญาณกลางในภาพที่ 13 ส่วนสัญญาณข้างล่างสุดของภาพที่ 13 เป็นสัญญาณ PWM เอาท์พุทที่ได้จากการเปรียบเทียบแรงดันกับระดับอ้างอิง (Reference level)



ภาพที่ 12 โครงสร้างของการกำเนิดสัญญาณ PWM โดยทั่วไป

ที่มา: มนตรี (2539)



ภาพที่ 13 สัญญาณแต่ละภาคตาม โครงสร้างของการกำเนิดสัญญาณ PWM

ที่มา: มนตรี (2539)

จะเห็นได้อย่างชัดเจนว่าถึงแม้ความถี่พาห้ยังมีค่าไม่มากกว่าความถี่ของสัญญาณข่าวสารมากนัก ความถี่ของสัญญาณ PWM เอ็ดพุตก็จะมีค่าไม่คงที่ดังนั้นในการใช้งานโดยทั่วไป จึงมักกำหนดให้ความถี่พาห้มากกว่าความถี่ข่าวสาร 10 เท่าขึ้นไป

#### ผลงานวิจัยที่เกี่ยวข้อง

เป็นที่ทราบกันดีว่า เมื่อเราอยู่หรือปฏิบัติหน้าที่ในบริเวณที่แวดล้อมด้วยเสียงรบกวน (Noisy environment) เช่น บริเวณโรงอาหาร, สถานบันเทิง, อาคารก่อสร้าง หรือแม้แต่ห้องนักบิน (Cockpit) การสื่อสารด้วยเสียงพูดภายใต้สภาวะดังกล่าว โดยทั่วไปแล้วจะมีความสัมฤทธิ์ผลน้อยกว่าสภาวะปกติกล่าวคือ ผู้พูดอาจจะต้องพูดประโยคเดิมซ้ำๆ หลายๆ ครั้ง หรืออาจต้องเพิ่มความดังในการพูดเพื่อให้ผู้ฟังเข้าใจในขณะที่ผู้ฟังอาจต้องตั้งใจฟังผู้พูดมากกว่าปกติ เมื่ออยู่ในสภาวะการดังกล่าวนานเป็นเวลานาน (จาตุรงค์, 2552) ทั้งผู้พูดและผู้ฟังมักจะเกิดความเหนื่อยล้า ด้วยเหตุผลดังกล่าว ในช่วงหลายทศวรรษที่ผ่านมาได้มีนักวิจัยทำการศึกษาและพัฒนาการเพิ่มสมรรถนะเสียงพูด (Speech enhancement) ให้ยังคงสามารถเข้าใจรับรู้ได้ (High intelligibility and perceptibility) แม้ผู้ฟังอยู่ในสภาวะแวดล้อมด้วยเสียงรบกวน

โดยทั่วไปการเพิ่มสมรรถนะเสียงพูดสามารถแบ่งออกเป็นสองประเภทคือ ประเภทแรกเป็นการเพิ่มสมรรถนะเสียงพูด ด้วยการกำจัดหรือลดผลกระทบจากเสียงรบกวนให้มากที่สุดเท่าที่จะกระทำได้ การเพิ่มสมรรถนะเสียงพูดส่วนใหญ่ในหลายทศวรรษที่ผ่านมาจัดอยู่ในประเภทนี้ เช่น วิธี Signal subspace และวิธี Spectral subtraction แม้นักวิจัยได้ทำการพิสูจน์ว่าเสียงพูดที่ผ่านการเพิ่มสมรรถนะจากวิธีดังกล่าว สามารถเพิ่มประสิทธิภาพการเข้าใจและรับรู้ได้เมื่อเทียบกับสัญญาณเสียงพูดก่อนการเพิ่มสมรรถนะอย่างไรก็ตาม วิธีการเพิ่มสมรรถนะเสียงพูดในประเภทนี้จะตั้งอยู่บนสมมุติฐานของการทราบได้ของคุณสมบัติของเสียงรบกวน เช่น สเปกตรัมของเสียงรบกวน ซึ่งในการใช้งานจริงโดยทั่วไปแล้ว อาจไม่ทราบถึงคุณสมบัติของเสียงรบกวน ส่งผลให้วิธีการเพิ่มสมรรถนะเสียงพูดในประเภทนี้ไม่มีประสิทธิภาพเท่าที่ควร ประเภทสองคือ การเพิ่มสมรรถนะเสียงพูดให้สามารถง่ายต่อการเข้าใจและรับรู้ได้มากขึ้นก่อนที่จะถูกส่งออกไป การเพิ่มสมรรถนะเสียงพูดนี้ เริ่มได้รับความสนใจจากนักวิจัยในช่วงไม่กี่ปีที่ผ่านมาข้อดีของการเพิ่มสมรรถนะของเสียงพูดประเภทนี้คือ ไม่จำเป็นต้องทราบถึงคุณลักษณะหรือคุณสมบัติของเสียงรบกวน Hazen และ Simpson แสดงให้เห็นว่าการเพิ่มขนาดหรือการเน้นสัทลักษณะ (Speech feature) โดยเฉพาะอย่างยิ่งบริเวณพยัญชนะ (Consonant) และบริเวณเปลี่ยนสถานะ (Transition region) ทั้งภายในบริเวณสระ (Within vowel), จากพยัญชนะไปสู่สระ (Consonant to vowel) และจากสระไปสู่พยัญชนะ (Vowel to consonant) มีความสำคัญต่อการเพิ่มประสิทธิภาพการเข้าใจและรับรู้ของเสียงพูด (Speech perception) ในเสียงรบกวน อย่างไรก็ตาม ในงานของ Hazen และ Simpson การได้มาซึ่งสัทลักษณะโดยวิธีนี้ยังต้องใช้การสังเกตด้วยตาและไม่สามารถกระทำได้อย่างอัตโนมัติ

ต่อมา นักวิจัย S.Yoo, C.Tantibundhit และ F.Pernkopf ได้ศึกษาและพัฒนาการแยกสัทลักษณะอย่างอัตโนมัติจากเสียงพูดและเรียกสิ่งที่ได้ว่า องค์ประกอบเสียงพูดชั่วคราว (Transient speech component) อีกทั้งยังแสดงให้เห็นว่าองค์ประกอบดังกล่าวมีความสำคัญอย่างยิ่งต่อการเพิ่มประสิทธิภาพการเข้าใจและรับรู้ได้ของเสียงพูดในเสียงรบกวน เนื่องด้วยการศึกษาและพัฒนาการได้มาซึ่งองค์ประกอบเสียงพูดชั่วคราวเป็นองค์ความรู้ใหม่ (S.Yoo, 2007) ที่นักวิจัยเริ่มให้ความสนใจอย่างจริงจังเมื่อไม่นานมานี้

นักวิจัย Yoo และคณะ ได้พัฒนาวิธีแยกองค์ประกอบเสียงพูดชั่วคราว โดยสัญญาณเสียงพูดดั้งเดิมถูกกรองด้วยตัวกรองผ่านความถี่สูง (High-pass filter) ที่มีความถี่ตัด 700 Hz เพื่อกำจัดฟอร์แมนต์ที่หนึ่ง (First formant) จากนั้นเสียงพูดผ่านตัวกรองความถี่สูง (High-pass filtered speech) ผ่านไปยังตัวกรองผ่านบางความถี่ซึ่งแปรเปลี่ยนตามเวลา (Time-varying bandpass filter)

จำนวนสามตัว เพื่อกำจัดฟอร์แมนต์หลักอีกสามตัว สัญญาณที่ได้จากการผ่านตัวกรองทั้งสาม เรียกว่าองค์ประกอบเสียงพูดกึ่งสถานะคงตัว (Quasi-steady-state component) องค์ประกอบเสียงพูดกึ่งสถานะคงตัวนี้เป็นองค์ประกอบของเสียงพูดที่มีความถี่คงที่หรือมีความถี่เปลี่ยนแปลงช้าๆ ของฟอร์แมนต์สระ (Vowel formant) หรือฮับพยัญชนะ (Consonant hub) จากนั้น องค์ประกอบเสียงพูดกึ่งสถานะคงตัวจะถูกนำไปลบออกจากสัญญาณเสียงพูดผ่านตัวกรองผ่านความถี่สูงผลที่ได้คือ องค์ประกอบเสียงพูดชั่วคราวแสดงดังสมการที่ 1

$$X_{\text{tran}}(t) = X_{\text{hp}}(t) - X_{\text{qss}}(t) \quad (1)$$

ซึ่ง  $X_{\text{tran}}(t)$ ,  $X_{\text{hp}}(t)$  และ  $X_{\text{qss}}(t)$  คือ สัญญาณเสียงพูดชั่วคราว, สัญญาณเสียงพูดผ่านตัวกรองผ่านความถี่สูง และสัญญาณเสียงพูดกึ่งสถานะคงตัว ตามลำดับ อย่างไรก็ตาม เมื่อพิจารณาถึงองค์ประกอบเสียงพูดชั่วคราวจากวิธีนี้ สัญญาณเสียงพูดชั่วคราวยังคงประกอบด้วยความถี่ฟอร์แมนต์คงที่ซึ่งควรปรากฏในในองค์ประกอบเสียงพูดกึ่งสถานะคงตัว

นักวิจัย Tantibundhit และคณะ ได้พัฒนาวิธีการแยกองค์ประกอบเสียงพูดชั่วคราวจากสัญญาณเสียงพูดดั้งเดิมอีกวิธีหนึ่ง โดยการแยกองค์ประกอบของสัญญาณดั้งเดิมออกเป็นสามองค์ประกอบตามขั้นตอนต่อไปนี้ นำการแปลงโคซายน์ดัดแปรแบบไม่ต่อเนื่อง (Modified discrete cosine transform : MDCT) เพื่อแยกข้อมูลของเสียงพูดที่มีความคงที่ (Tantibundhit, 2007) หรือมีการเปลี่ยนแปลงของความถี่อย่างช้าๆ ซึ่งข้อมูลดังกล่าวเรียกว่า องค์ประกอบเสียงพูดโทน (Tonal component) ในขณะที่วิธีการแปลงเวฟเลต (Wavelet transform) เพื่อแยกข้อมูลของเสียงพูดที่มีการเปลี่ยนแปลงอย่างฉับพลัน ซึ่งข้อมูลนี้เรียกว่าองค์ประกอบเสียงพูดชั่วคราว (Transient component) ส่วนที่เหลือจะเป็นองค์ประกอบเสียงพูดตกค้าง (Residual component) ดังแสดงในสมการที่ 2

$$X_{\text{orig}}(t) = X_{\text{tone}}(t) + X_{\text{tran}}(t) + X_{\text{resi}}(t) \quad (2)$$

ซึ่ง  $X_{\text{orig}}(t)$ ,  $X_{\text{tone}}(t)$  และ  $X_{\text{resi}}(t)$  คือสัญญาณเสียงพูดโทน, สัญญาณเสียงพูดชั่วคราว และสัญญาณพูดตกค้างตามลำดับ จากนั้นองค์ประกอบเสียงพูดชั่วคราว ถูกนำมาใช้ในการสร้างเสียงพูดเพิ่มสมรรถนะและทดสอบความสามารถในความเข้าใจและรับรู้ในสภาวะแวดล้อมที่มีเสียงรบกวน

นักวิจัย Tantibundhit, F.Pernkopf และคณะ ได้พัฒนาวิธีการแยกองค์ประกอบชั่วคราวจากการวิเคราะห์ความละเอียดหลากหลาย (Multiresolution analysis) ของสัญญาณไทล์ (Tile) สามารถแบ่งส่วนได้อย่างอัตโนมัติทั้งในเชิงเวลาและความถี่ปรับตามคุณลักษณะของสัญญาณที่กำลังจะทำการวิเคราะห์ โดยสัญญาณเสียงพูดดั้งเดิมจะถูกแยกออกเป็นสององค์ประกอบคือ องค์ประกอบเสียงพูดชั่วคราว (Transient component) และองค์ประกอบไร้เสียงเสียงพูดชั่วคราว (Non-transient component) ดังแสดงในสมการที่ 3

$$X_{orig}(t) = X_{tran}(t) - X_{nont}(t) \quad (3)$$

โดยสัญญาณพูดดั้งเดิมจะถูกแปลงโดยเวฟเลตแพ็คเกจ (Wavelet packet) จากนั้นบล็อกของสัมประสิทธิ์เวฟเลตแพ็คเกจจะถูกแสดงในรูปของไทล์ซึ่งถูกแบ่งส่วนทั้งในเชิงเวลาและความถี่ จากนั้นบล็อกของสัมประสิทธิ์เวฟเลตแพ็คเกจใดมีขนาดความสูงของไทล์มากกว่าหรือเท่ากับความกว้างของไทล์ (Tantibundhit, 2009) บล็อกของสัมประสิทธิ์นั้นจะถูกจัดประเภทเป็นองค์ประกอบเสียงพูดชั่วคราวและจะถูกนำไปใช้สร้างเสียงพูดเพิ่มสมรรถนะเช่นเดียวกัน

จากผลการเปรียบเทียบขององค์ประกอบเสียงพูดชั่วคราวทั้งสามวิธีแสดงดังตารางที่ 7 องค์ประกอบเสียงพูดชั่วคราวที่ได้จากวิธีของ Yoo และคณะ ยังคงมีข้อมูลความถี่สูงที่ฟอร์แมนต์ โดยเฉพาะอย่างยิ่งในฟอร์แมนต์ความถี่สูง องค์ประกอบเสียงพูดชั่วคราวที่ได้จากวิธีของ Tantibundhit และคณะ กำจัดข้อมูลที่มีความถี่สูงที่ได้มีประสิทธิผลกว่าวิธีของ Yoo และคณะ อย่างไรก็ตาม ยังคงเหลือบริเวณความถี่ของสระแทบคงที่ซึ่งไม่ควรมีในองค์ประกอบเสียงพูดชั่วคราว วิธีการของ Tantibundhit, F.Pernkopf และคณะ ที่ถูกพัฒนามาอีกขั้นในองค์ประกอบเสียงพูดชั่วคราวที่มีผลดีที่สุด

นักวิจัย Yoo และคณะ, Tantibundhit และคณะ และ Tantibundhit, F.Pernkopf และคณะ ได้แสดงให้เห็นถึงการเพิ่มสมรรถนะเสียงพูด โดยทดสอบกับภาษาอังกฤษแต่จะไม่มีทดสอบการเพิ่มสมรรถนะขององค์ประกอบเสียงพูดชั่วคราวกับภาษาอื่นๆ ดังนั้นอาจต้องมีการปรับปรุงวิธีการเหล่านี้ให้เหมาะสมกับภาษาโทนมามากยิ่งขึ้น หากต้องการนำไปใช้ในการเพิ่มสมรรถนะเสียงพูดในสภาวะแวดล้อมที่มีเสียงรบกวน

ตารางที่ 7 สรุปข้อดีและข้อด้อยของอัลกอริทึมของนักวิจัยท่านต่างๆ ในการแยกองค์ประกอบเสียงพูดชั่วคราว

นักวิจัย และ คณะ	ข้อดี	ข้อด้อย
Yoo และ คณะ	1. องค์ประกอบเสียงพูดชั่วคราวรวมเอาข้อมูลพยัญชนะซึ่งมีการเปลี่ยนแปลงอย่างฉับพลันคือ ข้อมูลส่วนใหญ่ของ/b/, ข้อมูลส่วนใหญ่ ณ ช่วงสกัดการไหลของอากาศในการออกเสียง /t/ อื่นๆรวมเอาข้อมูลช่วงปลดปล่อยหลังการสกัดกัน, ข้อมูลบริเวณเปลี่ยนสถานะจากพยัญชนะ/b/ ไปสู่สระ /ae/, และข้อมูลบริเวณเปลี่ยนสถานะท้ายสระ /ae/	1. ยังคงมีข้อมูลความถี่ฟอร์แมนต์คงที่เหลืออยู่ โดยเฉพาะอย่างยิ่งในบริเวณความถี่สูง 2. ไม่สามารถรวมเอาเสียงพูดชั่วคราวที่มีความถี่ต่ำกว่า 700 Hz
Tantibundhit และ คณะ	1. ประสิทธิภาพเท่าเทียมวิธีของ Yoo และคณะ 2. รวมเอาเสียงพูดชั่วคราวที่มีความถี่ต่ำกว่า 700 Hz 3. สามารถกำจัดข้อมูลความถี่คงที่ได้โดยมีประสิทธิภาพมากกว่าวิธีของ Yoo โดยสามารถสังเกตได้จากช่องโหว่ (Hole) ในสเปกโตรแกรม	1. ผลกระทบจากความเพี้ยนฟรีเอ็ล็ค โค 2. ยังคงเหลือความถี่แทบคงที่ของสระซึ่งไม่ควรมีข้อมูลของบริเวณดังกล่าวอยู่ในองค์ประกอบของเสียงพูด
Tantibundhit และ F.Pernkopf คณะ	1. มีประสิทธิภาพเทียบเท่าวิธีของ Yoo 2. รวมเอาเสียงพูดชั่วคราวที่มีความถี่ต่ำกว่า 700 Hz 3. สามารถกำจัดข้อมูลความถี่คงที่ได้โดยมีประสิทธิภาพดีกว่าวิธีของ Tantibundhit แบบแรก ไม่มีข้อมูลความถี่คงที่ของสระ /ae/ รวมอยู่ในองค์ประกอบเสียงพูดชั่วคราวตามที่คาดหวังไว้	1. แม้ว่าให้องค์ประกอบเสียงพูดชั่วคราวที่มีผลดีที่สุด แต่อาจจะต้องมีการปรับปรุงอัลกอริทึมหากนำไปเพิ่มสมรรถนะเสียงพูดในสภาวะแวดล้อมที่มีเสียงรบกวนกับภาษาโทน

ที่มา: ไฟฟ้าสาร (2552)

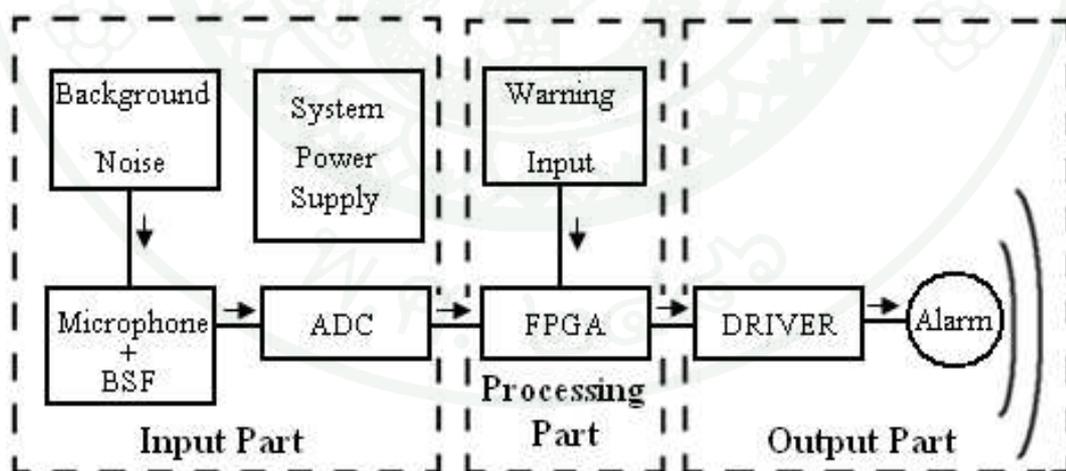
## อุปกรณ์และวิธีการ

### อุปกรณ์

1. อุปกรณ์อิเล็กทรอนิกส์ของระบบอิเล็กทรอนิกส์เตือนภัยแบบคัดแปลงเสียง
  - 1.1 ออปแอมป์ชนิดต่างๆ
  - 1.2 ไอซีสำหรับประมวลผลระบบ FPGA
  - 1.3 ไอซีแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล LM 3915
  - 1.4 กริ่งเตือนภัย
  - 1.5 ไอซีสำหรับขับแรงดันเอาต์พุตชนิดต่างๆ
  - 1.6 คอมพิวเตอร์ และ โปรแกรมออกแบบประมวลผล
  - 1.7 ตัวต้านทาน และ ตัวเก็บประจุ ต่างๆ
  - 1.8 เซนเซอร์ตรวจจับชนิดต่างๆ
  - 1.9 ไอซีสำหรับปรับแรงดันอินพุตชนิดต่างๆ
  - 1.10 สวิตช์ชิงเพาเวอร์ซีพพลาย
  - 1.11 คอนเนกเตอร์ชนิดต่างๆ
  - 1.12 เคสสำหรับใส่ชิ้นงาน
  - 1.13 สายยูเอสบีคาว์โหนดสำหรับใช้โหนดโปรแกรม
  - 1.14 เซเวนเซกเมนต์ และ แอลอีดี สำหรับใช้แสดงสถานะของระบบ
  - 1.15 เครื่องมือหนักในการทำงาน เช่น คีม ไขควง สว่าน อื่นๆ
2. อุปกรณ์เครื่องมือวัดสัญญาณทางไฟฟ้าและเครื่องกำเนิดสัญญาณทางไฟฟ้า
  - 2.1 เครื่องวัดเสียงสามารถติดต่อกับคอมพิวเตอร์ได้
  - 2.2 ดิจิตอลมัลติมิเตอร์
  - 2.3 ดิจิตอลออสซิลโลสโคป
  - 2.4 เครื่องกำเนิดสัญญาณทางไฟฟ้า (Signal Generator)

## วิธีการ

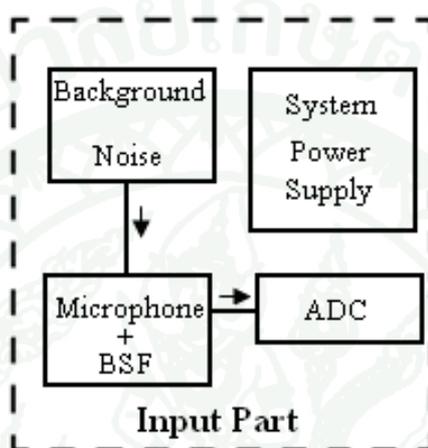
งานวิจัยนี้เป็นการประยุกต์ใช้ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงสำหรับใช้ในโรงงานอุตสาหกรรม เพื่อลดมลพิษทางเสียงที่อาจเกิดจากเสียงของตัวเครื่องเตือนภัยเอง โดยเครื่องเตือนภัยแบบปกติจะถูกตั้งค่าความดังของกริ่งไว้ที่ค่าความดังสูงสุดตลอดเวลา และไม่สามารถปรับระดับความดังของเสียงกริ่งเตือนภัยตามสภาวะแวดล้อมได้ หลักการทำงานของระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงคือ เมื่อมีการกดปุ่มสัญญาณเตือนภัย หรือเซนเซอร์ชนิดต่างๆตรวจจับสิ่งผิดปกติที่เกิดขึ้น (Warning input) ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงจึงเริ่มทำงาน โดยสัญญาณเสียงจากสิ่งแวดล้อม (Background Noise) จะผ่านมาจากไมโครโฟนอินพุต และถูกแปลงสัญญาณจากสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (ADC) เพื่อให้ FPGA ประมวลผลแล้วสร้างสัญญาณดิจิทัลโดยใช้เทคนิค PWM ปรับความกว้างของสัญญาณพัลส์และส่งสัญญาณไปยังส่วนเอาต์พุตเพื่อควบคุมการเปิดและปิดของวงจรขับแรงดันทำให้ระบบเตือนภัยแบบตัดแปลงเสียงสามารถปรับระดับความดังของกริ่งเตือนภัยให้ดังกว่าเสียงของสภาวะแวดล้อมในขณะนั้นประมาณ 10 dB เพียงแค่พอให้ผู้ปฏิบัติงานได้ยินก็เพียงพอ เพื่อลดมลพิษทางเสียงที่อาจเกิดจากตัวเครื่องเตือนภัยเอง โดยงานวิจัยชิ้นนี้แบ่งส่วนประกอบหลักของระบบเป็น 3 ส่วนคือ ส่วนอินพุต ส่วนประมวลผล และส่วนเอาต์พุต ดังภาพที่ 14



ภาพที่ 14 ระบบเตือนภัยแบบตัดแปลงเสียง (Adaptive Auditory Warning System)

## 1. ส่วนอินพุตของระบบเตือนภัยแบบตัดแปลงเสียง (Input Part)

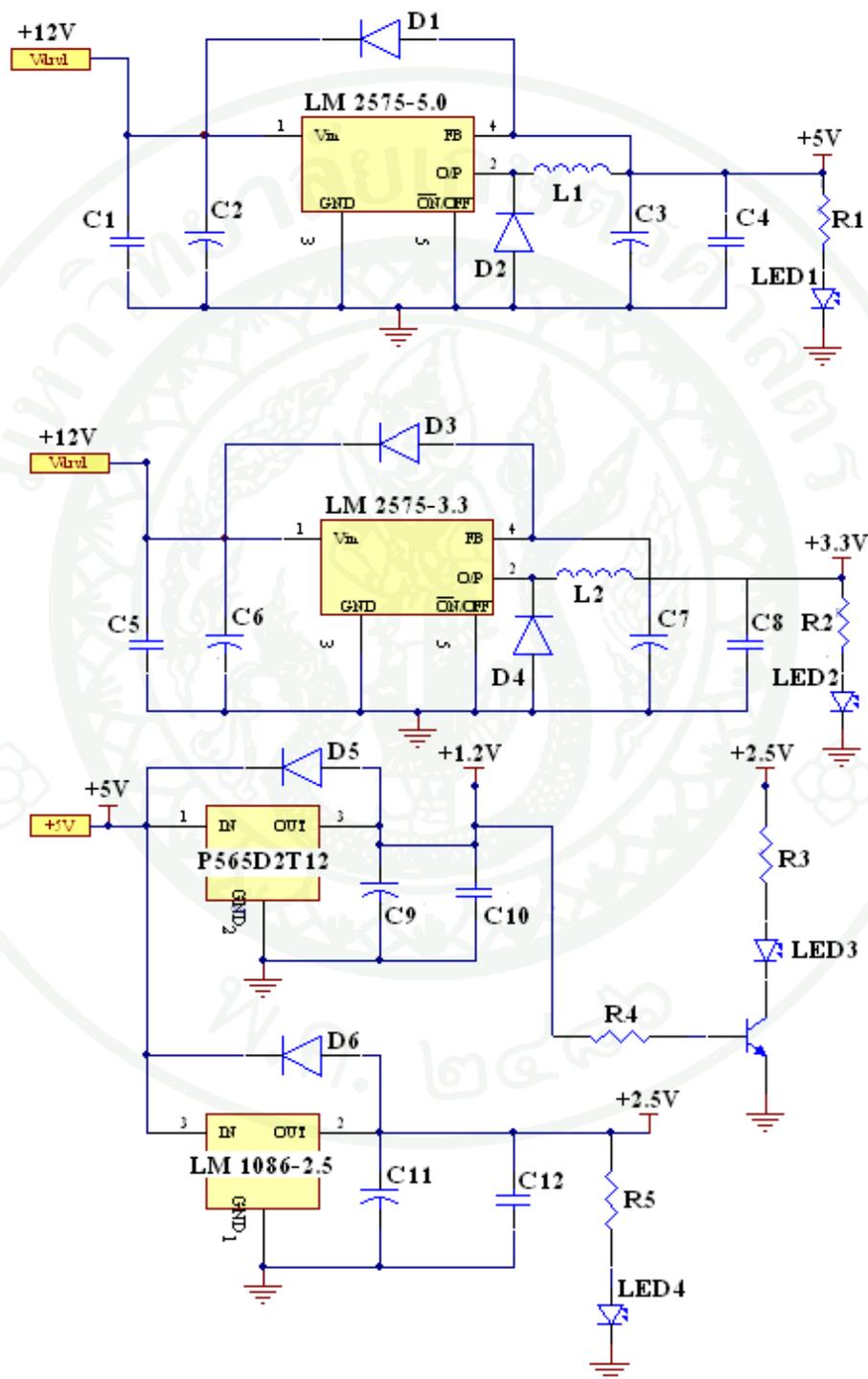
ส่วนอินพุตของระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงแบ่งออกเป็น 2 ส่วนหลักๆ ด้วยกัน คือ ภาคเพาเวอร์ซัพพลายของระบบ และภาคอินพุตในการรับสัญญาณเสียงจากสิ่งแวดล้อม เพื่อแปลงสัญญาณเสียงสิ่งแวดล้อมจากอนาล็อกเป็นดิจิทัลแสดงดังภาพที่ 15



ภาพที่ 15 ส่วนอินพุตของระบบเตือนภัยแบบตัดแปลงเสียง

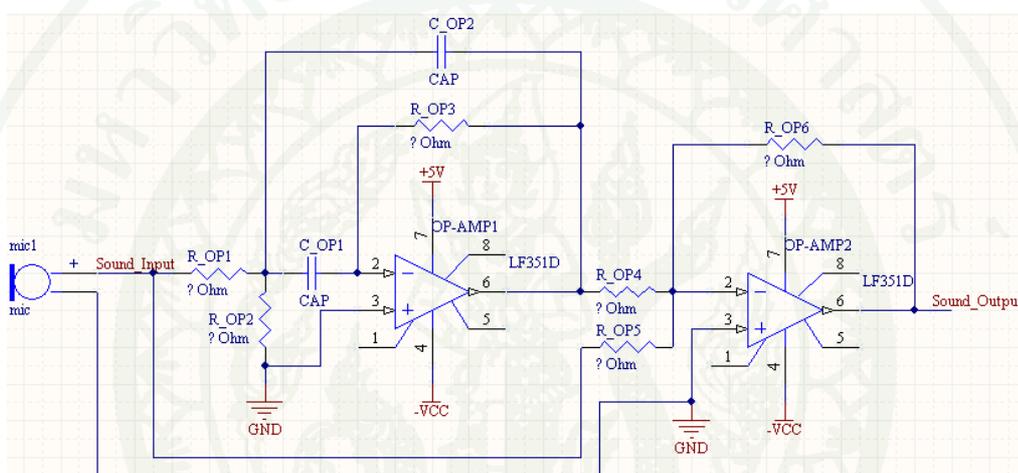
โดยภาคเพาเวอร์ซัพพลายเป็นการแปลงระดับแรงดันไฟฟ้าของระบบจาก 12 V<sub>DC</sub> เป็นแรงดันไฟฟ้าระดับต่างๆ เพื่อใช้เลี้ยงส่วนต่างๆ ของระบบ ระดับแรงดันไฟฟ้าที่ถูกแปลงจาก 12 V<sub>DC</sub> แบ่งออกได้เป็น 4 ระดับ คือ 5 V<sub>DC</sub>, 3.3 V<sub>DC</sub>, 2.5 V<sub>DC</sub> และ 1.2 V<sub>DC</sub> ตามลำดับ แรงดันไฟฟ้า 5 V<sub>DC</sub> เป็นการแปลงแรงดันไฟฟ้าจาก 12 V<sub>DC</sub> ให้เป็นแรงดันไฟฟ้า 5 V<sub>DC</sub> โดยใช้ IC LM 2575-5.0 Switching Regulator สำหรับจ่ายระดับแรงดันไฟฟ้าให้กับ IC ต่างๆ ของระบบที่เป็น IC แบบ TTL (Transistor-Transistor Logic), แรงดันไฟฟ้า 3.3 V<sub>DC</sub> เป็นการแปลงแรงดันไฟฟ้าจาก 12 V<sub>DC</sub> ให้เป็นแรงดันไฟฟ้า 3.3 V<sub>DC</sub> โดยใช้ IC LM 2575-3.3 Switching Regulator สำหรับจ่ายระดับแรงดันไฟฟ้าให้กับ IC ต่างๆ ของระบบที่เป็น IC แบบ CMOS (Complementary Metal-Oxide Semiconductor), แรงดันไฟฟ้า 2.5 V<sub>DC</sub> เป็นการแปลงแรงดันไฟฟ้าจาก 5 V<sub>DC</sub> ให้เป็นแรงดันไฟฟ้า 2.5 V<sub>DC</sub> โดยใช้ IC LM 1086-2.5 Linear Regulator สำหรับโปรแกรม FPGA หรือเป็นแหล่งจ่ายระดับแรงดันไฟฟ้าให้กับ IC Flash Memory, แรงดันไฟฟ้า 1.2 V<sub>DC</sub> เป็นการแปลงแรงดันไฟฟ้าจาก 5 V<sub>DC</sub> ให้เป็น 1.2 V<sub>DC</sub> โดยใช้ IC P565D2T12 Linear Regulator สำหรับจ่ายระดับแรงดันไฟฟ้าไป

ที่ Core ต่างๆ ของ FPGA เพื่อให้ FPGA ทำงานได้อย่างสมบูรณ์ สามารถออกแบบวงจรภาคเพาเวอร์ซัพพลายของระบบอิเล็กทรอนิกส์เดือนกับแบบดัดแปลงเสียงแสดงได้ดังภาพที่ 16



ภาพที่ 16 วงจรภาคเพาเวอร์ซัพพลายของระบบ

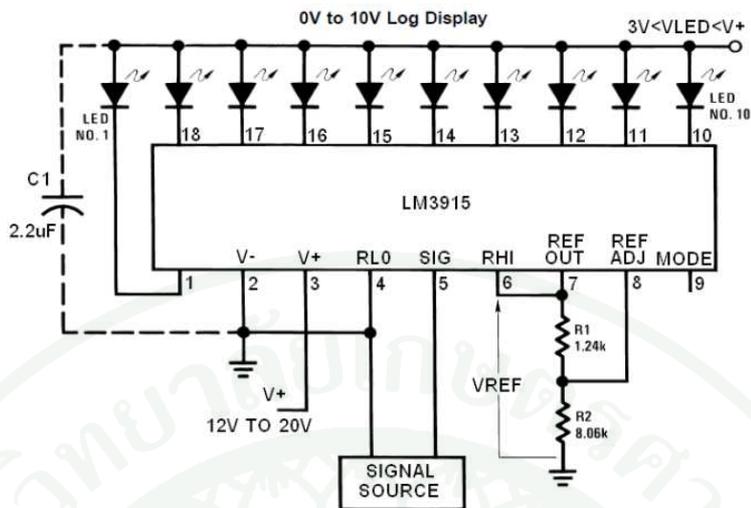
ภาคอินพุตส่วนที่สองเป็นการแปลงสัญญาณเสียงจากสิ่งแวดลอมซึ่งเป็นสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล โดยระบบทำการรับเสียงจากสิ่งแวดลอมด้วยไมโครโฟน และทำการขยายระดับสัญญาณของเสียงจากสิ่งแวดลอมโดยผ่านวงจรขยาย และวงจรกรอง โดยงานวิจัยนี้เลือกใช้วงจรกรองความถี่หยุดผ่าน (Band Stop Filter, BSF) ดังภาพที่ 17 แสดงให้เห็นว่าวงจรที่ทำการออกแบบนั้นใช้โอปแอมป์จำนวน 2 ตัว โดยแบ่งการออกแบบไว้เป็น 2 ส่วน คือ วงจรกรองแถบความถี่ผ่าน (Band Pass Filter, BPF) และวงจรรวมสัญญาณ เพื่อทำการกรองเสียงกริ่งที่มีความถี่ 3.1 kHz ให้ไม่สามารถผ่านได้



ภาพที่ 17 วงจรกรองความถี่หยุดผ่าน (BSF)

วงจรกรองความถี่หยุดผ่านจะอาศัยวงจรกรองแถบความถี่ผ่านมาใช้ในการสร้าง โดยจะนำสัญญาณเอาต์พุตของวงจรกรองความถี่ผ่าน และสัญญาณอินพุตที่ป้อนให้กลับวงจรกรองแถบความถี่ผ่านนำมาต่อเข้าด้วยกันและทำการส่งสัญญาณทั้งสองนี้เข้าไปในวงจรรวมสัญญาณอีกครั้งหนึ่ง ซึ่งสัญญาณที่ออกมาจากวงจรรวมสัญญาณก็จะมีค่าที่ตรงข้ามกับสัญญาณที่ถูกป้อนกลับเข้ามา และวงจรนี้ยังสามารถทำการขยายสัญญาณอินพุตที่เข้ามา (ชนันท์, 2550) ได้อีกด้วย

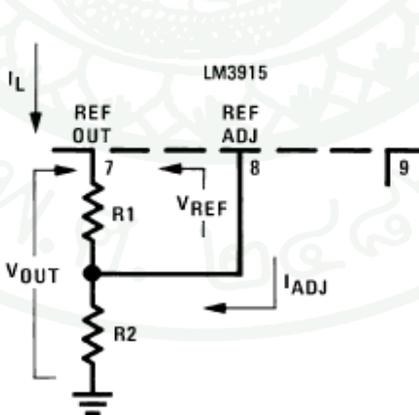
เมื่อได้สัญญาณอินพุตที่ขยายและกรองมาจากสิ่งแวดลอมแล้ว สัญญาณจะถูกส่งต่อมายัง IC LM3915 เพื่อทำการเปลี่ยนสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล การใช้งาน IC LM3915 ทำได้โดยการต่อวงจรดังภาพที่ 18 โดยช่องสัญญาณที่ 1 และ 10-18 แสดงถึงค่าความดังของเสียงอินพุต ในขณะที่นั้นมีระดับความดังเท่าไร ผ่านทาง LED (Light Emitting Diode)



ภาพที่ 18 การต่อวงจรสำหรับการใช้งาน IC LM3915

ที่มา: National Semiconductor (2001)

จากนั้นหาค่า R1, R2 จากสมการปรับค่าเสียงอินพุตให้สัมพันธ์กับแรงดันสมการที่ 4 โดยการต่อวงจร RDIV กับช่องสัญญาณที่ 7, 8 ของ IC LM3915 แสดงได้ดังภาพที่ 19 และใช้ข้อมูลจากตารางที่ 4 ในการกำหนดขอบเขตเอาต์พุต ซึ่งจะได้เอาต์พุต  $V_{OUT}$  เป็นสมการความสัมพันธ์ระหว่าง R1 และ R2 ดัง สมการที่ 4



ภาพที่ 19 วงจร RDIV นำมาต่อกับ IC เบอร์ LM3915

ที่มา: National Semiconductor (2001)

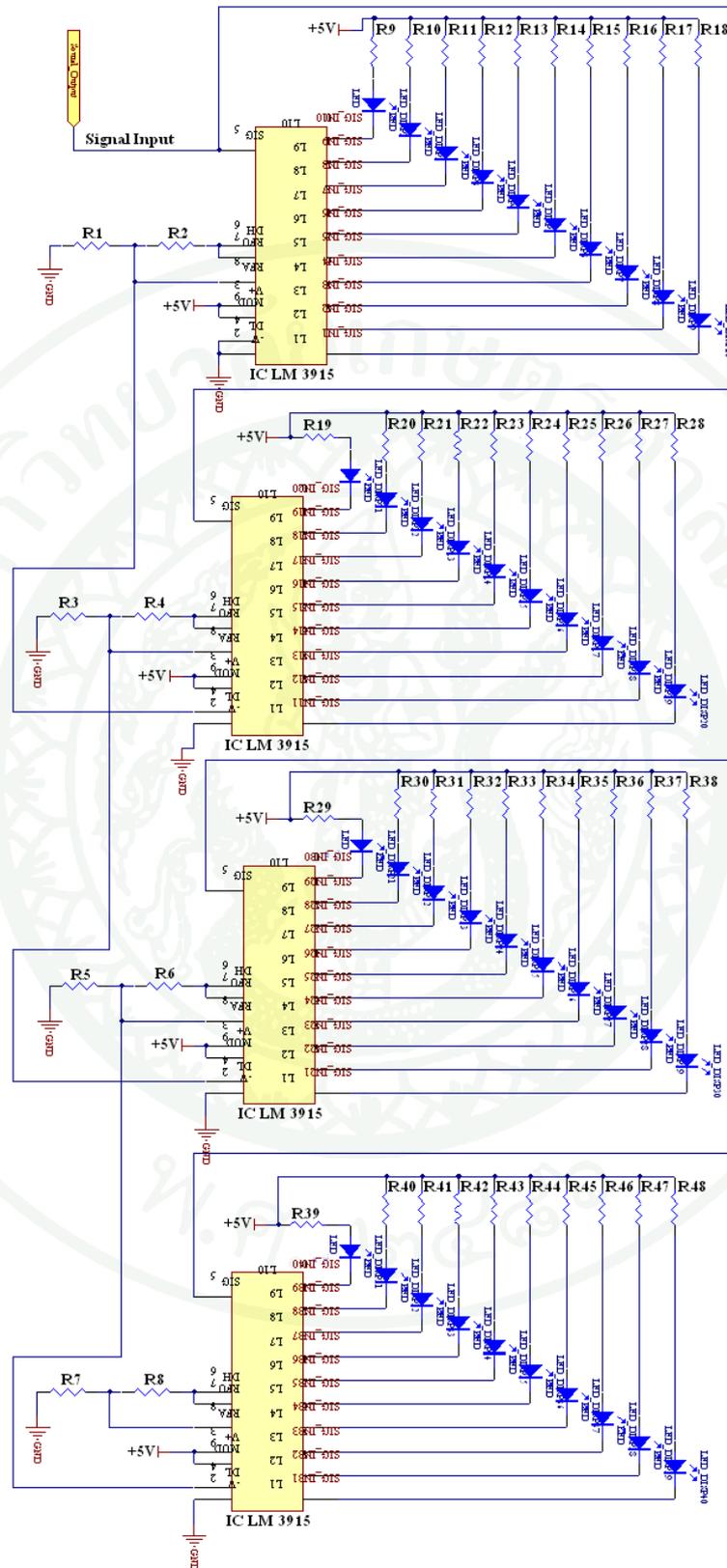
$$V_{OUT} = V_{REF} (1 + (R2/R1)) + (I_{ADJ} * R2) \quad (4)$$

โดยค่าของ

$V_{OUT}$	คือ ค่าแรงดันเอาต์พุตของ IC LM3915 ในหน่วย V
$V_{REF}$	คือ ค่าแรงดันคงที่ภายใน IC LM3915 มีค่าเท่ากับ 1.25 V
$I_{ADJ}$	คือ ค่ากระแสที่มีผลต่อ R1 และ R2 มีค่าตั้งแต่ 120 $\mu$ A – 5 mA
R1 และ R2	คือ ค่าความต้านทานที่ใช้ในการปรับค่า $V_{OUT}$ มีหน่วยเป็น k $\Omega$

เมื่อปรับค่า  $V_{OUT}$  ให้เป็นไปตามความสัมพันธ์ดังสมการที่ 4 แล้ว จากนั้นทำการปรับค่าเสียงที่ดังที่สุดที่สนใจโดยให้  $V_{OUT}$  มีค่าเท่ากับ 10 V และปรับค่าเสียงที่เบาที่สุดที่สนใจโดยให้  $V_{OUT}$  มีค่าเท่ากับ 0.447 V เช่น อินพุตของสัญญาณเสียงจากสิ่งแวดล้อมมีค่าตั้งแต่ 0 dB – 30 dB ในกรณีนี้เราจะทำการปรับ เสียงที่ความดัง 30 dB ให้มีค่า  $V_{OUT}$  เท่ากับ 10 V และเสียงที่ความดัง 0 dB ให้มีค่า  $V_{OUT}$  เท่ากับ 0.447 V ดังนั้น ค่าความดังของเสียงที่ 27 dB และ 24 dB จะมีค่า  $V_{OUT}$  เท่ากับ 7.079 V และ 5.021 V ตามลำดับ ซึ่งเป็นไปตามตารางความสัมพันธ์ระหว่างแรงดันไฟฟ้าเอาต์พุตกับค่าความดังของเสียงจากตารางที่ 4

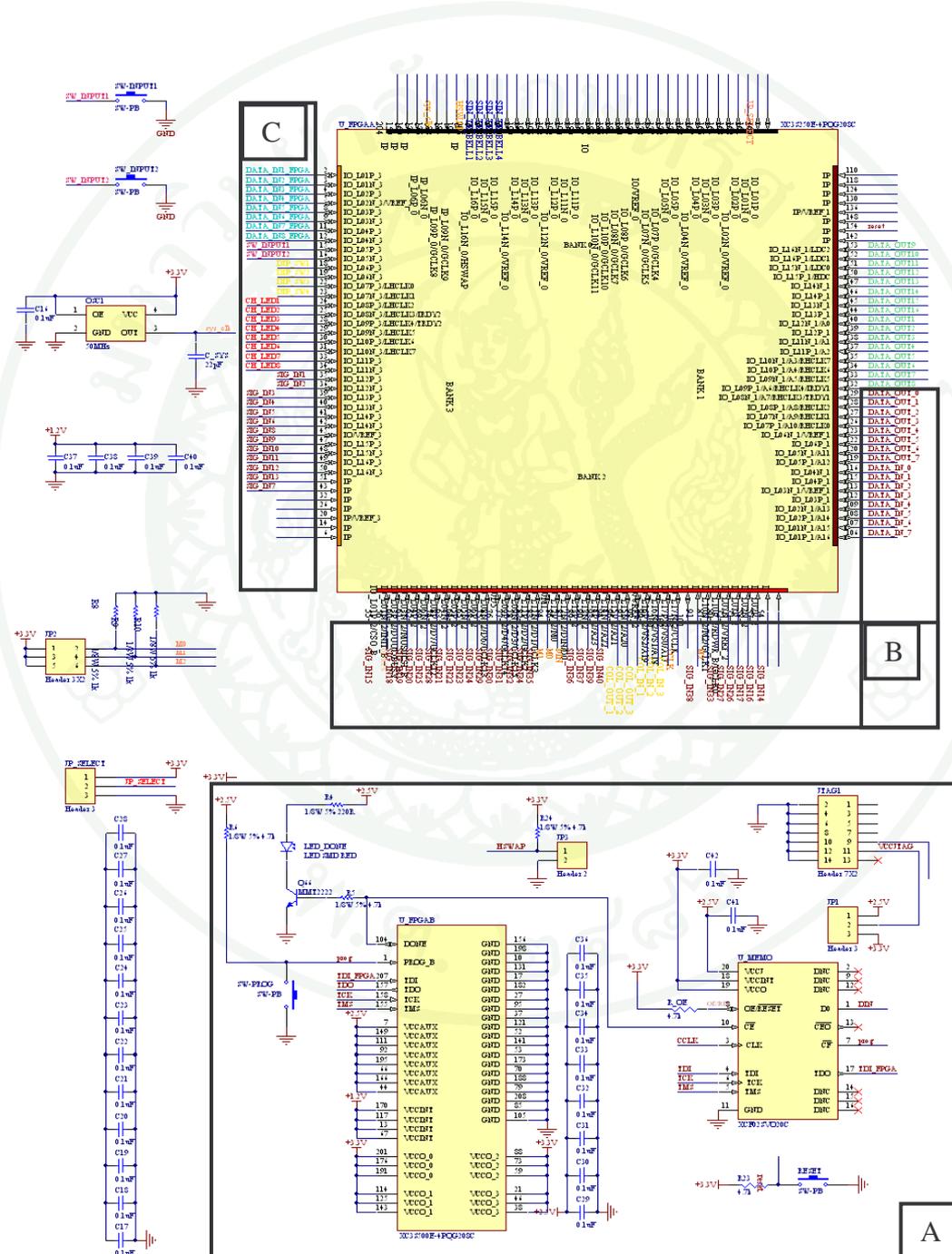
ในงานวิจัยระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงนี้ ได้ออกแบบให้ส่วนอินพุตของระบบสามารถรับอินพุตสัญญาณเสียงจากสิ่งแวดล้อมโดยกำหนดกรอบการรับค่าความดังของเสียงจากสิ่งแวดล้อมไว้ตั้งแต่ 0–120 dB และรับค่าความดังสูงสุดได้ไม่เกิน 120 dB (ตามมาตรฐานของเสียงในโรงงานและเป็นค่าความดังสูงสุดของกริ่งเตือนภัยแบบปกติ) ดังนั้นจึงต้องต่อ IC LM3915 ซึ่งเป็น IC สำหรับแปลงสัญญาณเสียงจากสิ่งแวดล้อมที่เป็นสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลจำนวน 4 ตัว เพื่อให้รับอินพุตของเสียงจากสิ่งแวดล้อมได้ตามกรอบที่กำหนดไว้ โดยอ้างอิงจากทฤษฎี Parallel Comparator และไม่ได้มีการต่อวงจรตามวิธีของ datasheet ของตัว IC LM 3915 เอง เมื่อส่วนอินพุตของระบบรับค่าความดังของเสียงจากสิ่งแวดล้อม ณ วินาทีนั้นๆ แล้วค่าความดังของเสียง ณ วินาทีนั้นๆ จะแสดงผลผ่านทาง LED ทั้ง 40 ดวง ซึ่งค่าที่แสดงผลนี้เป็นค่าดิจิทัลแล้ว สามารถทำการออกแบบชุดวงจรภาคแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลนี้ด้วยโปรแกรม Altium designer แสดงดังภาพที่ 20



ภาพที่ 20 แสดงการต่อ IC LM 3915 โดยทฤษฎี Parallel ออปแอมป์

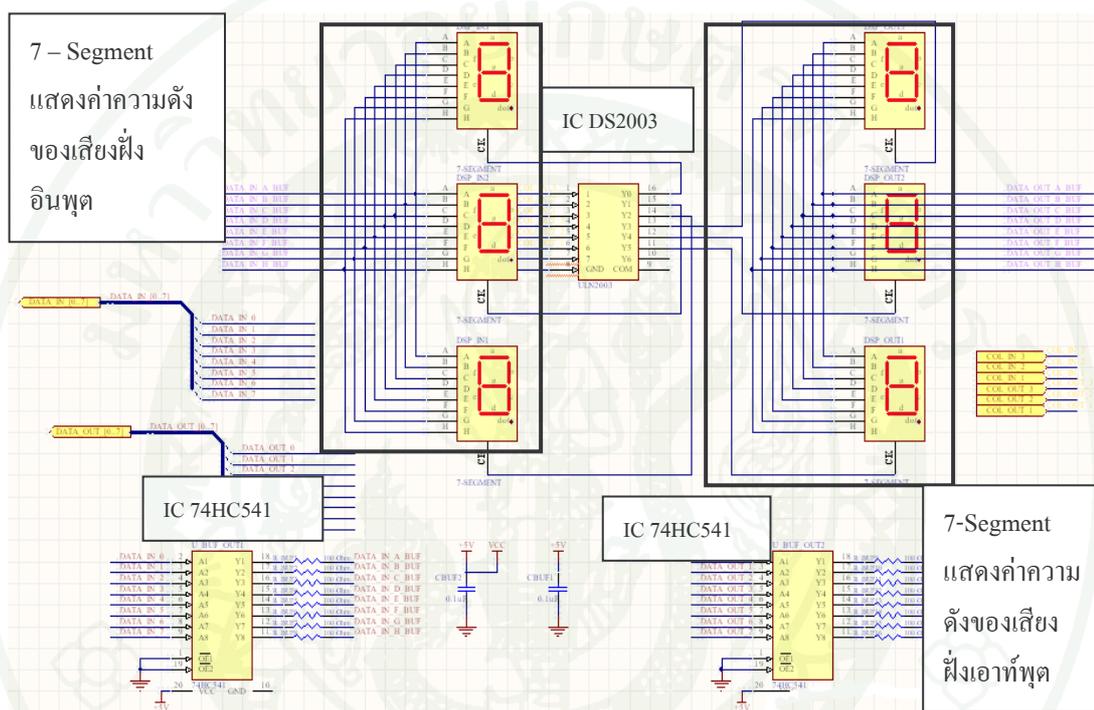
## 2. ส่วนประมวลผลของระบบเตือนภัยแบบตัดแปลงเสียง (Processing Part)

การออกแบบส่วนประมวลผลของระบบ FPGA ได้ถูกนำมาใช้ในการวิเคราะห์ข้อมูลสัญญาณดิจิทัล และควบคุมการทำงานส่วนต่างๆ ของระบบแสดงดังภาพที่ 21



ภาพที่ 21 วงจรส่วนประมวลผล และแฟลชเมมโมรี่

ส่วน A ของวงจรส่วนประมวลผลจากภาพที่ 21 เป็นส่วนที่ใช้สำหรับโปรแกรมลงบน FPGA และทำหน้าที่ติดต่อกับแพลตฟอร์มโมรีภายนอกสำหรับกรณีที่ต้องการ โปรแกรมลงบน FPGA แบบถาวร ส่วน B ของวงจรส่วนประมวลผลจากภาพที่ 21 เป็นส่วนที่ใช้สำหรับติดต่อกับ 7-Segment ฝั่งอินพุต และ 7-Segment ฝั่งเอาต์พุตเพื่อแสดงค่าความดัง ณ ขณะนั้น โดยวงจรแสดงสถานะค่าความดังของเสียงภาคอินพุตเอาต์พุตแสดงดังภาพที่ 22

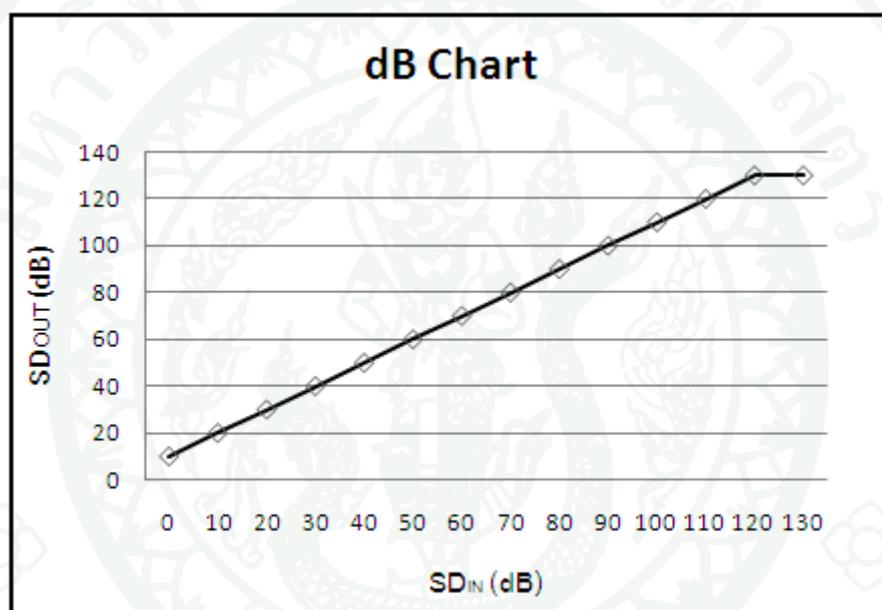


ภาพที่ 22 วงจรแสดงค่าความดังของเสียงภาคอินพุต และภาคเอาต์พุต

ส่วน C ของวงจรส่วนประมวลผลจากภาพที่ 21 เป็นส่วนรับสัญญาณอินพุตจากช่องสัญญาณที่ 1, 10-18 ของ IC LM3915 โดย IC LM3915 ได้ทำการแปลงสัญญาณเสียงให้เป็นสัญญาณทางไฟฟ้าเพื่อให้ FPGA สามารถประมวลผล และควบคุมการทำงานของส่วนต่างๆ ของระบบได้ จากนั้นจะใช้ความสัมพันธ์ระหว่างอินพุต และเอาต์พุตในการคำนวณ ด้วยการอ้างอิงสมการคณิตศาสตร์สมการที่ 5 ในการออกแบบชุดคำสั่งในการทำงาน

$$SD_{OUT} = SD_{IN} + SD_{CONST} ; 0 < SD_{OUT} < 130, 0 < SD_{IN} < 130, 5 < SD_{CONST} < 10 \quad (5)$$

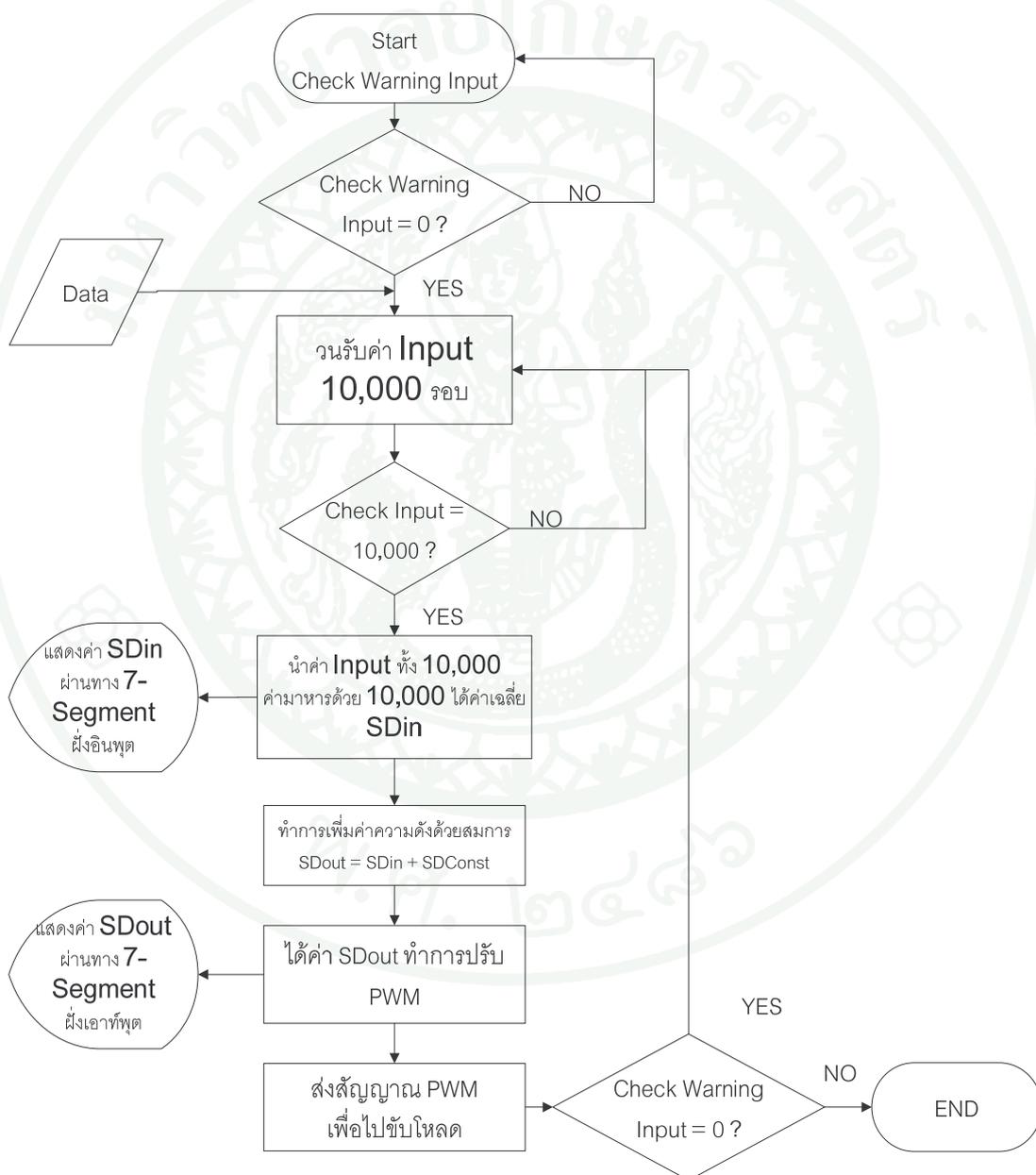
โดย  $SD_{IN}$  คือ สัญญาณดิจิตอลฟังอินพุตมีหน่วยเป็นเดซิเบล (dB),  $SD_{OUT}$  คือ สัญญาณดิจิตอลฟังเอาต์พุตของ FPGA มีหน่วยเป็น dB,  $SD_{CONST}$  คือค่าคงที่สำหรับเพิ่มความดังสัญญาณดิจิตอลฟังอินพุตมีหน่วยเป็น dB เมื่อ FPGA รับค่าความดังของสัญญาณเสียงอินพุต ( $SD_{IN}$ ) ณ.วินาทีนั้น FPGA จะทำการเพิ่มความดังของเสียงเอาต์พุต ( $SD_{OUT}$ ) ด้วยค่าของ ( $SD_{CONST}$ ) โดยค่าตัวแปร  $SD_{OUT}$  ณ.วินาทีนั้น จะถูกนำมาเปรียบเทียบและสร้างสัญญาณพัลส์เพื่อควบคุมความดังของกริ่งเตือนภัย นำกราฟของสมการคณิตศาสตร์สมการที่ 5 มาแสดงค่าความสัมพันธ์ระหว่าง อินพุต และเอาต์พุต โดยไม่สนใจความถี่ได้ผลดังแสดงในภาพที่ 23



ภาพที่ 23 กราฟความสัมพันธ์ระหว่างความดังอินพุตและความดังเอาต์พุตโดยไม่สนใจความถี่

หลักการทำงานของส่วนประมวลผลของระบบแสดงเป็น Flow Chart ได้ดังภาพที่ 24 โดยระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงจะเริ่มทำงานเมื่อมีการกดปุ่มสัญญาณเตือนภัย หรือเซนเซอร์ชนิดต่างๆตรวจจับสิ่งผิดปกติที่เกิดขึ้นได้แล้ว ส่วนประมวลผลจะเริ่มทำการเก็บค่าสัญญาณเสียงจากสิ่งแวดล้อม (DATA) โดยทำการวนรับข้อมูล และเปรียบเทียบหาค่าความดังที่มากที่สุด 10,000 ค่าความดังใน 0.5 วินาที นำค่าความดังที่ได้ทั้ง 10,000 ค่านั้นมาบวกกันทั้งหมด และนำไปหารด้วย 10,000 จะได้ค่าค่าความดังอินพุตที่แท้จริง ณ. วินาทีนั้น ( $SD_{IN}$ ) และนำค่า  $SD_{IN}$  ที่ได้มานี้แสดงผลผ่านทาง 7-Segment ฟังอินพุต เพื่อแสดงสถานะค่าความดังในขณะนั้นว่ามีค่าความดังเท่าไร จากนั้นทำการเพิ่มความดัง  $SD_{IN}$  ด้วยค่าค่าความดังคงที่ ( $SD_{CONST}$ ) ตามสมการคณิตศาสตร์

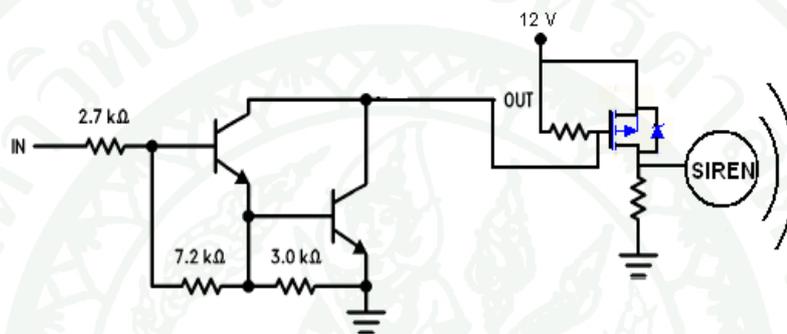
สมการที่ 5 จะได้อ่านค่าความถี่ของเอาต์พุต ณ วินาทีนั้น (SD<sub>OUT</sub>) เมื่อได้อ่านค่า SD<sub>OUT</sub> แล้ว FPGA จะนำค่าความถี่ของเอาต์พุตนี้ไปแสดงสถานะค่าความถี่ผ่านทาง 7-Segment 7-Segment จากนั้นนำค่า SD<sub>OUT</sub> นี้ไปประมวลผลแล้วสร้างสัญญาณดิจิทัลโดยใช้เทคนิค PWM (Pulse Width Modulation) ปรับความกว้างของสัญญาณพัลส์และส่งสัญญาณไปยังส่วนเอาต์พุตเพื่อควบคุมระดับความถี่ของเครื่องเตือนภัยให้ได้อ่านค่าความถี่ตามที่ต้องการ



ภาพที่ 24 Flow Chart หลักการทำงานของส่วนประมวลผลของระบบ

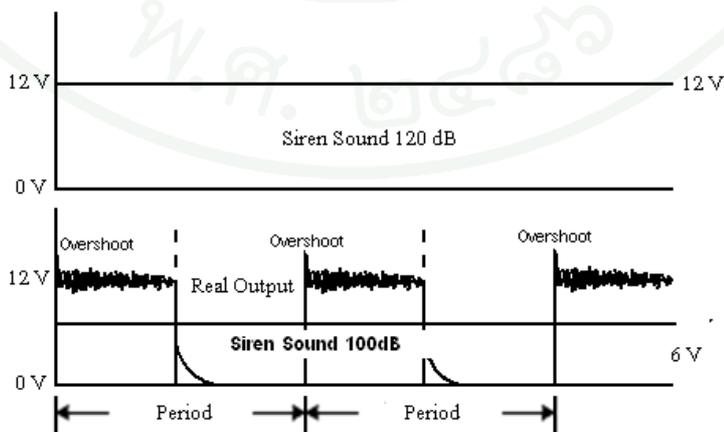
### 3. ส่วนเอาต์พุตของระบบเตือนภัยแบบตัดแปลงเสียง (Output Part)

ส่วนของเอาต์พุตประกอบด้วย IC DS2003 (Voltage Darlington Driver) เป็นวงจรถานซิสเตอร์คาร์ลิงตัน วงจรขับแรงดันมอสเฟต และกริ่งเตือนภัยแสดงดังภาพที่ 25 สามารถควบคุมการเปิดและปิดของมอสเฟตโดยส่งสัญญาณพัลส์ผ่านขา IN ของวงจรขับแรงดันเพื่อควบคุมระดับความดังของกริ่งเตือนภัย



ภาพที่ 25 วงจรถานซิสเตอร์คาร์ลิงตัน วงจรขับแรงดันมอสเฟต และ กริ่งเตือนภัย

เทคนิค PWM ได้ถูกนำมาประยุกต์ใช้ในการปรับระดับความดังของกริ่งสัญญาณเตือนภัยที่ Duty Cycle ต่างๆ เช่นกริ่งเตือนภัยมีระดับความดัง 120 dB ที่ระดับแรงดันไฟฟ้าอินพุต 12 V<sub>DC</sub> ทดลองปรับสัญญาณพัลส์ที่ Duty Cycle 50% ระดับแรงดันไฟฟ้าอินพุต 6 V<sub>DC</sub> ระดับค่าความดังที่ได้ 100 dB แสดงดังภาพที่ 26



ภาพที่ 26 PWM 50 % Duty cycle ค่าความดังที่ได้ 100 dB

จากนั้นนำกริ่งเตือนภัยที่มีค่าความดัง 120 dB ที่ระดับแรงดันไฟฟ้าอินพุต 12 V<sub>DC</sub> นำมาทำการปรับระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยโดยใช้เทคนิค PWM ให้เหลือค่าระดับแรงดันไฟฟ้าอินพุตระดับต่างๆ เช่น 10V<sub>DC</sub>, 9V<sub>DC</sub>, 6V<sub>DC</sub> และ 3V<sub>DC</sub> แล้วใช้ Sound Level Meter ตรวจวัดค่าความดังของกริ่งเตือนภัยเพื่อหาความสัมพันธ์ระหว่าง Duty Cycle ของระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยกับค่าความดังของกริ่งเตือนภัย (dB) แสดงดังตารางที่ 8

ตารางที่ 8 ตารางแสดงความสัมพันธ์ระหว่าง Duty Cycle ของระดับแรงดันไฟฟ้ากับค่าความดังของกริ่งเตือนภัย (dB)

Duty Cycle (%)	ค่าความดัง (dB)
100	120
90	114 – 111
80	107 – 106
70	105 – 104
60	103 – 102
50	101 – 100
40	98 – 96
30	92 – 90
20	87 – 85
10	70 – 68

โดยกริ่งเตือนภัยที่ใช้ทำการหาค่าความสัมพันธ์ระหว่าง Duty Cycle ของระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยกับค่าความดังของกริ่งเตือนภัยนั้น คือกริ่งเตือนภัยตามมาตรฐานชนิดหนึ่งรุ่น C-red1 เป็นกริ่งเตือนภัยแบบอิเล็กทรอนิกส์ที่ใช้กันอย่างแพร่หลายของบริษัท Leoton และเครื่องวัดเสียงที่ใช้ในการวัดค่าความดังของเสียงจากสิ่งแวดล้อมเป็นเครื่องวัดเสียงรุ่น DT-8552 ของบริษัท CEM แสดงดังภาพที่ 27



ภาพที่ 27 กริ่งเตือนภัย และเครื่องวัดเสียง

ส่วน A จากภาพที่ 27 คือกริ่งเตือนภัยแบบอิเล็กทรอนิกส์รุ่น C-red1 และส่วน B จากภาพที่ 27 คือเครื่องวัดเสียงรุ่น DT-8552 จากนั้นเมื่อทราบค่าความสัมพันธ์ระหว่าง Duty Cycle ของระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยกับค่าความดังของกริ่งเตือนภัยจากตารางที่ 8 แล้วสามารถนำค่าที่ได้ไปใช้เขียน โปรแกรมโดยใช้ภาษา VHDL ในการออกแบบชุดคำสั่งเพื่อควบคุมระดับความดังของกริ่งเตือนภัยได้ตามที่ต้องการ

## ผลและวิจารณ์

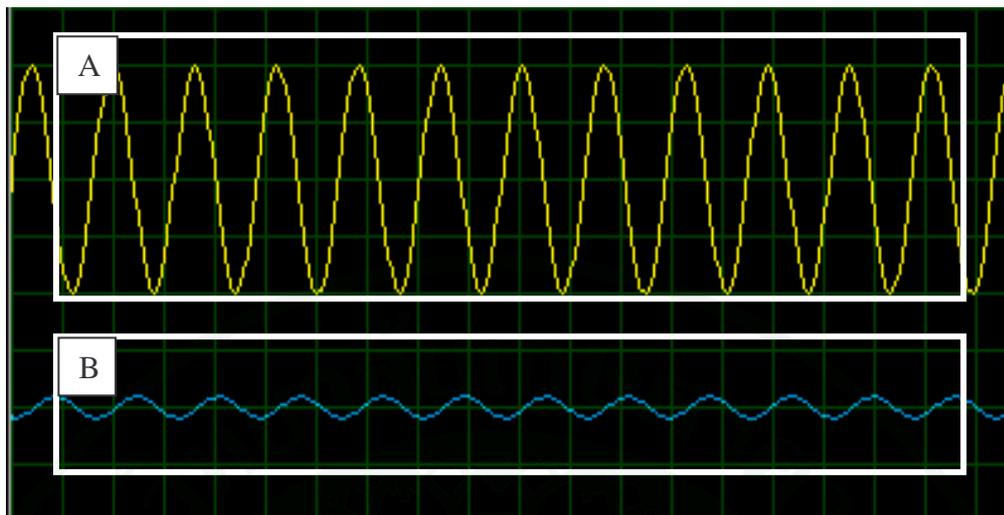
### ผล

ส่วนอินพุตรับสัญญาณเสียงจากสิ่งแวดล้อมโดยใช้ไมโครโฟนความไวสูงรัศมีการรับเสียงประมาณ 2 เมตร และภายในตัวไมโครโฟนมีการขยายสัญญาณเสียงจากสิ่งแวดล้อมมาแล้วระดับหนึ่ง รูปไมโครโฟนแสดงดังภาพที่ 28



ภาพที่ 28 ไมโครโฟนสำหรับใช้รับสัญญาณเสียงจากสิ่งแวดล้อม

จากการทดลองใช้ออสซิลโลสโคปวัดสัญญาณเสียงจากสิ่งแวดล้อมโดยวัดที่เอาต์พุตของไมโครโฟน พบว่าไมโครโฟนสามารถตรวจจับสัญญาณเสียงที่เปลี่ยนแปลงไปจากสิ่งแวดล้อม ณ ขณะนั้นๆ ได้ จากนั้นสัญญาณเสียงจากสิ่งแวดล้อมที่ผ่านไมโครโฟนนี้ถูกส่งผ่านไปยังวงจร BSF หลักการทำงานของวงจร BSF คือความถี่ที่ 3.1 kHz ซึ่งเป็นความถี่ของกริ่งเตือนภัยไม่สามารถผ่านได้ แต่ความถี่ที่ไม่ใช่ 3.1 kHz สามารถผ่านได้เพื่อทำการกรองเฉพาะสัญญาณเสียงของตัวกริ่งเตือนภัยเอง ผลตอบสนองต่อความถี่ของวงจร BSF แสดงดังภาพที่ 29



ภาพที่ 29 ผลตอบสนองต่อความถี่ของวงจร BSF ที่ความถี่ 3.1 kHz

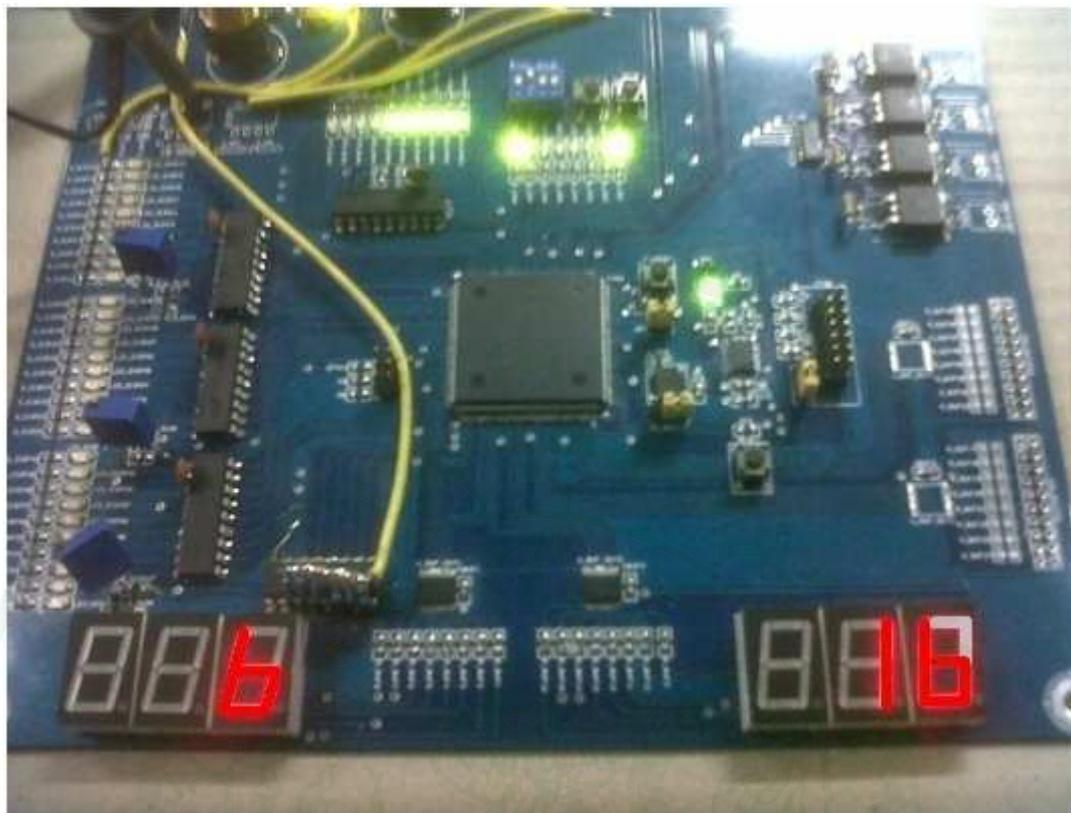
ส่วน A จากภาพที่ 29 คือ สัญญาณความถี่ของกริ่งเตือนภัยซึ่งมีความถี่ 3.1 kHz ส่วน B จากภาพที่ 29 คือ สัญญาณความถี่ของกริ่งเตือนภัยที่ถูกกรองโดยวงจร BSF แล้ว ดังนั้นจากการทดลองพบว่าวงจร BSF ที่ถูกออกแบบสำหรับกรองความถี่ของกริ่งเตือนภัยที่มีความถี่ 3.1 kHz พบว่าวงจร BSF นี้ให้ผลตอบสนองในการกรองความถี่ของกริ่งเตือนภัยเองได้

จากการทดสอบต่อวงจรแปลงสัญญาณเสียงจากสิ่งแวดล้อม ซึ่งเป็นการแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลด้วย IC LM3915 ดังภาพที่ 18 นั้น ทำได้โดยวัดค่าความต่างศักย์ ( $V_{OUTTyp}$  (Multimeter)) จาก IC LM3915 ด้วยดิจิตอลมัลติมิเตอร์เปรียบเทียบกับค่าความต่างศักย์จาก Datasheet ( $V_{OUTTyp}$  (Datasheet)) มีค่าดังตารางที่ 9 ซึ่งจากผลการทดลองพบว่า การแปลงสัญญาณเสียงซึ่งเป็นสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัลโดยใช้ IC LM3915 ได้ผลถูกต้องแม่นยำในระดับหนึ่ง แม้ว่าค่า ( $V_{OUTTyp}$  (Multimeter)) ที่วัดได้จากดิจิตอลมัลติมิเตอร์ไม่ตรงกับค่า ( $V_{OUTTyp}$  (Datasheet)) ของ Datasheet แต่จะพบว่าค่า ( $V_{OUTTyp}$  (Multimeter)) จากดิจิตอลมัลติมิเตอร์ที่คลาดเคลื่อนไปนั้นแกว่งไปมาไม่เกินค่า  $V_{OUTMIN}$  และ  $V_{OUTMAX}$  ของ Datasheet ในระดับความดังนั้นๆ โดยดูข้อมูลเปรียบเทียบได้จากตารางที่ 4

ตารางที่ 9 ตารางเปรียบเทียบค่า  $V_{OUTTyp}$  จาก Datasheet กับค่า  $V_{OUTTyp}$  ที่วัดได้จากดิจิตอลมัลติมิเตอร์

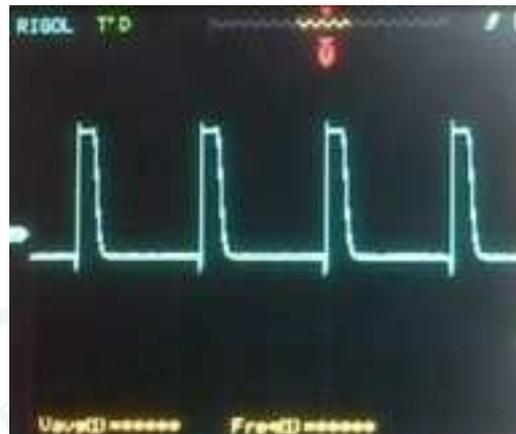
OUTPUT (Channel)	dB	$V_{OUTTyp}$ (V) (Datasheet)	$V_{OUTTyp}$ (V) (Multimeter)
1	-27	0.447	0.43 – 0.46
2	-24	0.631	0.6 – 0.7
3	-21	0.891	0.87 – 0.95
4	-18	1.259	1.2 – 1.3
5	-15	1.778	1.72 – 1.85
6	-12	2.512	2.45 – 2.63
7	-9	3.548	3.45 – 3.65
8	-6	5.021	4.9 – 5.1
9	-3	7.079	6.8 – 7.2
10	0	10	10

จากนั้นทดสอบการทำงานของส่วนประมวลผลของระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียง FPGA ได้ถูกนำมาใช้ทดสอบควบคุมการทำงานส่วนต่างๆ ของระบบ โดยรับค่าความดังของสัญญาณเสียงจากสิ่งแวดล้อมอินพุต จากวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอลรับอินพุตที่แตกต่างกัน 40 ระดับ ระดับความดังตั้งแต่ 0 dB – 120 dB แล้วทำการประมวลผลค่าความดังจากสิ่งแวดล้อมที่รับมา ณ วินาทีนั้นมีค่าความดังเท่าไร จากนั้นแสดงสถานะค่าความดังผ่าน 7-Segment ฟังอินพุต ใช้สมการคณิตศาสตร์จากสมการที่ 5 คำนวณค่าความดังเอาท์พุต และแสดงสถานะค่าความดังเอาท์พุตผ่าน 7-Segment ฟังเอาท์พุตแสดงดังรูปภาพที่ 30 จากการทดสอบพบว่าส่วนประมวลผลของระบบสามารถทำงานได้ถูกต้องตามขอบเขตของงานที่วางไว้

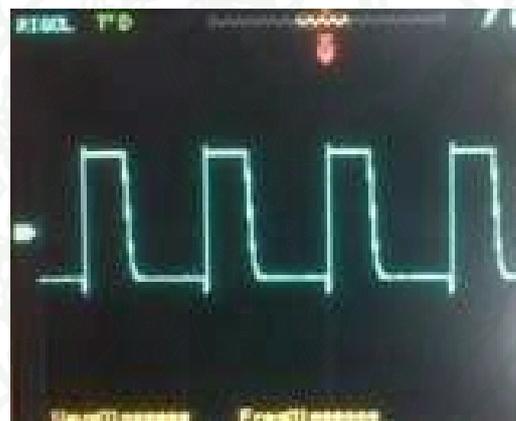


ภาพที่ 30 บอร์ดคอนโทรลของระบบเตือนภัยแบบตัดแปลงเสียง

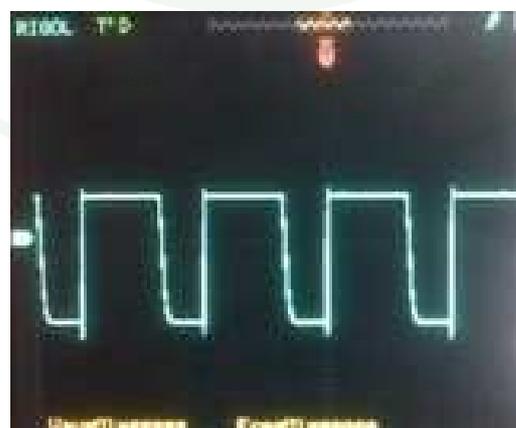
จากการทดสอบการทำงานส่วนเอาต์พุตของระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียง PWM เทคนิคถูกนำมาใช้ในการสร้างสัญญาณพัลส์เพื่อควบคุมระดับความดังของกริ่งเตือนภัย โดยจะทำการทดสอบปรับค่า PWM ที่ Duty Cycle 10 %, Duty Cycle 30 % และ Duty Cycle 70 % จากนั้นใช้ออสซิลโลสโคปทำการวัดระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยได้ผลแสดงดังรูปภาพที่ 31, รูปภาพที่ 32 และรูปภาพที่ 33 ตามลำดับ ซึ่งผลการทดสอบที่ได้พบว่า การปรับค่า PWM ของระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยที่ Duty Cycle 10 %, Duty Cycle 30 % และ Duty Cycle 70 % เทียบกับค่าความดังที่วัดได้จากเครื่องวัดเสียงแล้วนั้น พบว่าค่าความสัมพันธ์ระหว่าง Duty Cycle ของระดับแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยกับค่าความดังของกริ่งเตือนภัยเป็นไปตามตารางแสดงค่าความสัมพันธ์ตารางที่ 8



ภาพที่ 31 สัญญาณ PWM ที่ Duty Cycle 10% เทียบได้ค่าความดัง 70 dB



ภาพที่ 32 สัญญาณ PWM ที่ Duty Cycle 30% เทียบได้ค่าความดัง 90 dB



ภาพที่ 33 สัญญาณ PWM ที่ Duty Cycle 70% เทียบได้ค่าความดัง 105 dB

## วิจารณ์

จากผลการทดลองทั้ง 3 ส่วนของระบบเตือนภัยแบบตัดแปลงเสียงพบว่า ส่วนอินพุตค่า  $V_{OUTTyp}$  ที่วัดจากคิิจิตอลมัลติมิเตอร์มีความคลาดเคลื่อนเกิดขึ้นประมาณ 5 – 10 เปอร์เซ็นต์ เมื่อเทียบกับค่า  $V_{OUTTyp}$  จาก datasheet ซึ่งค่าความคลาดเคลื่อนที่เกิดขึ้นนั้นมีค่าไม่เกินค่าแรงดันมากที่สุด และค่าแรงดันต่ำสุดของแต่ละช่องสัญญาณนั้นรับได้ ส่วนประมวลผลทำงานได้ถูกต้องตามทฤษฎี และส่วนเอาต์พุตจากผลการทดลองโดยทำการปรับระดับความกว้างของสัญญาณพัลส์ ที่ Duty Cycle หลายระดับเพื่อวัดค่าความดังของกริ่งเตือนภัยเทียบกับค่าแรงดันไฟฟ้าอินพุตของกริ่งเตือนภัยที่ Duty cycle ต่างๆ ตามตารางที่ 8 พบว่ากริ่งเตือนภัยแบบอิเล็กทรอนิกส์เป็นโหนดไม่เชิงเส้น เมื่อนำทั้งสามส่วนมาต่อกันเป็นระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงสำหรับใช้ในโรงงานอุตสาหกรรมเพื่อลดมลพิษทางเสียงที่เกิดจากตัวกริ่งเตือนภัยแบบเดิมที่ไม่สามารถปรับระดับความดังของเสียงตามสภาวะแวดล้อมได้ ด้วยแนวคิดที่ว่าเสียงสัญญาณเตือนภัยไม่จำเป็นต้องดังที่ระดับเดียวกันตลอดเวลาแต่ควรดังมากกว่าเสียงของสภาวะแวดล้อมในขณะนั้นเพียงระดับหนึ่งประมาณ 10 dB (เพียงพอให้ผู้ปฏิบัติงานได้ยินก็เพียงพอ) พบว่าระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียง สามารถลดมลพิษทางเสียงที่เกิดจากตัวกริ่งเตือนภัยเองได้

## สรุปและข้อเสนอแนะ

### สรุป

ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงสำหรับใช้ในโรงงานอุตสาหกรรมได้ถูกนำมาใช้เพื่อแก้ปัญหามลพิษทางเสียงที่เกิดจากระบบเตือนภัยแบบเดิมซึ่งไม่สามารถปรับระดับความดังของกริ่งเตือนภัยตามสภาวะแวดล้อมได้ โดยส่วนประกอบหลักของระบบอิเล็กทรอนิกส์เตือนภัยแบ่งออกเป็น 3 ส่วนคือ ส่วนอินพุตเป็นการแปลงเสียงจากสิ่งแวดล้อมโดยเป็นการแปลงสัญญาณอนาล็อกที่ได้รับจากไมโครโฟนให้เป็นสัญญาณดิจิทัล ส่วนประมวลผลกระบวนการทำงานของระบบเริ่มจากเมื่อมีการกดปุ่มสัญญาณเตือนภัย หรือเซนเซอร์ชนิดต่างๆ ตรวจจับสิ่งผิดปกติที่เกิดขึ้นเช่น ความร้อนสูง หรือ ปริมาณควันที่มากเกินไป ส่วนประมวลผลจะเริ่มทำการวิเคราะห์ข้อมูลสัญญาณดิจิทัลและทำหน้าที่ควบคุมการทำงานทุกส่วนของระบบ และส่วนเอาต์พุตเป็นการสร้างสัญญาณพัลส์โดยใช้เทคนิค PWM เพื่อไปควบคุมระดับความดังของกริ่งเตือนภัย ระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงสามารถรับเสียงจากสิ่งแวดล้อมที่ค่าความดังตั้งแต่ 0–120 เดซิเบล รวมทั้งแสดงค่าความดังของเสียงอินพุตจากสิ่งแวดล้อมในขณะนั้นผ่านทาง 7-Segment ฝั่งอินพุตได้ผลใกล้เคียงกับเครื่องวัดเสียงรุ่น DT-8852 สามารถปรับระดับค่าความดังของกริ่งเตือนภัยให้มากกว่าค่าความดังของเสียงจากสิ่งแวดล้อมในขณะนั้น 10 dB และตรวจจับเสียงจากสิ่งแวดล้อมขณะนั้นได้โดยมีรัศมีการตรวจจับ 2 เมตร คณะผู้วิจัยหวังเป็นอย่างยิ่งว่างานวิจัยระบบอิเล็กทรอนิกส์เตือนภัยแบบตัดแปลงเสียงโครงการนี้สามารถนำไปประยุกต์ใช้กับระบบอิเล็กทรอนิกส์แบบตัดแปลงเสียงชนิดอื่นๆ ได้ เช่น เสียงเตือนคนข้ามถนนสำหรับคนตาบอด เสียงเตือนประตูเปิดปิดอัตโนมัติ หรือโครงการต่างๆ สำหรับผู้พิการทางสายตาที่จำเป็นต้องใช้เสียงในการสื่อสารกับผู้พิการ เพื่อควบคุมมลพิษทางเสียงที่อาจเกิดขึ้นจากการใช้เสียงเตือนหรือสื่อสารในลักษณะต่างๆ ให้เกิดขึ้นน้อยที่สุด

### ข้อเสนอแนะ

จากข้อมูลที่ได้ศึกษาระหว่างการดำเนินโครงการวิจัย และจากผลของการดำเนินโครงการทางคณะผู้วิจัยขอเสนอแนะแนวทางอันคาดว่าจะจะเป็นประโยชน์ในการพัฒนาโครงการวิจัยที่เกี่ยวข้องดังต่อไปนี้

1. ในส่วนของวงจรองความถี่ วงจรที่นำมาใช้เป็นแบบอนาล็อกสำหรับกรองสัญญาณความถี่ของตัวจริงเตือนภัย อาจสามารถประยุกต์ใช้วงจรองความถี่แบบดิจิทัลมาแทนที่วงจรองความถี่แบบอนาล็อก เนื่องจากวงจรองความถี่แบบดิจิทัลนั้นมีความยืดหยุ่น และสามารถเปลี่ยนแปลงค่าความถี่ที่ต้องการกรองทิ้งไปได้ง่ายกว่าวงจรองความถี่แบบอนาล็อก

2. พิจารณาในเรื่องของต้นทุนการผลิต หรือต้นทุนในการวิจัย ผู้วิจัยและพัฒนาอาจจะสามารถเลือกใช้ตัวประมวลผลประเภทอื่นเช่น MCS-51 หรือ PIC ที่มีราคาต่ำกว่า FPGA มาประยุกต์ใช้ในงานแทน แต่ MCU ที่เลือกใช้ควรมี I/O มากเพียงพอสำหรับการใช้งาน

3. จากการศึกษาผลของความคลาดเคลื่อนและปัญหาที่เกิดจากส่วนต่างๆของระบบพบว่า ไมโครโฟนแต่ละชนิดที่เลือกใช้มีลักษณะการให้สัญญาณที่แตกต่างกัน แม้ว่าเสียงที่รับมาจะเป็นเสียงเดียวกัน ซึ่งปัญหาที่พบนี้สามารถแก้ไขได้โดยการเลือกไมโครโฟนอินพุตชนิดคุณภาพดี และไม่ควรเป็นไมโครโฟนที่กรองสัญญาณความถี่บางชนิดทิ้งไป หรืออาจแก้ปัญหาได้ด้วยการเปลี่ยนค่าตัวต้านทานสำหรับปรับค่าความดังของเสียงจากสิ่งแวดลอมใหม่เพียงแต่วิธีนี้จะค่อนข้างยุ่งยากมากขึ้นไป

## เอกสารและสิ่งอ้างอิง

- กรมควบคุมมลพิษ. 2536. รายงานสถานการณ์มลพิษของประเทศไทย. บริษัท อินทริเกรเต็ด โปรโมชัน เทคโนโลยี จำกัด. แหล่งที่มา: [http://www.learners.in.th/file/wiparin\\_taksin/มลภาวะทางเสียง.doc](http://www.learners.in.th/file/wiparin_taksin/มลภาวะทางเสียง.doc), 26 ตุลาคม 2548.
- จตุรงค์ ตันติบัณฑิต. 2552. การเพิ่มสมรรถนะเสียงพูดและการเปรียบเทียบของค้ประกอบเสียงพูดชั่วคราว. ไฟฟ้าสารและคอมพิวเตอร์. แหล่งที่มา: <http://www.eit.or.th>, 6 ธันวาคม 2552.
- ชนันท์ ศรีกุล. 2550. การออกแบบและทดสอบวงจรรองความถี่ BSF. สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ. แหล่งที่มา: <http://www.te.kmutnb.ac.th/~msn/publication.html>, 25 เมษายน 2551.
- ธีรยศ เวียงทอง. 2548. ไมโครคอนโทรลเลอร์กับFPGA. มหาวิทยาลัยเทคโนโลยีมหานคร. แหล่งที่มา: [http://www.ee.mut.ac.th/FPGA/HomeFile/uCon\\_FPGA.pdf](http://www.ee.mut.ac.th/FPGA/HomeFile/uCon_FPGA.pdf), 16 กุมภาพันธ์ 2551.
- ปนัดดา แสงเทียนชัย. 2541. การแปลงสัญญาณอนาล็อก-ดิจิตอล. มหาวิทยาลัยเกษตรศาสตร์. แหล่งที่มา: [http://www.capet1.blogspot.com/2009\\_02\\_01\\_archive.html](http://www.capet1.blogspot.com/2009_02_01_archive.html), 20 กุมภาพันธ์ 2552.
- ประกอบ เพชรรัตน์. 2551. ระบบการเตือนภัยในนิคมมาบตาพุด. บริษัท เอ็นซีพี เซฟตี้ แอนด์ เอ็นไวรอนเมนทอล เซอร์วิส จำกัด. แหล่งที่มา: <http://www.npc-se.co.th>, 12 สิงหาคม 2551.
- มนตรี ศิริปรัชญานันท์. 2539. การกำเนิดสัญญาณผลอนุพันธ์ที่มีความถี่คงที่. มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี. แหล่งที่มา: <http://www.te.kmutnb.ac.th/msn/publication.html>, 11 กุมภาพันธ์ 2552.

มนัส สุวรรณ. 2530. นิเวศวิทยาของมนุษย์. มหาวิทยาลัยเทคโนโลยีสุรนารี. แหล่งที่มา:

<http://clremopac.sut.ac.th/BibDetail.aspx?bibno=40440&keyid=103113&searchresultpage=BibList.aspx&mainresultpage=BasicSearch.aspx&showform=%C1%B9%D1%CA+%CA%D8%C7%C3%C3%B3&recordno=&seqid=7166029&bibcount=4&searchsection=BasicSearch>, 27 มกราคม 2553.

วีระเดช พะเขาศิริพงษ์. 2540. กฎหมายสิ่งแวดล้อมและการรักษาความสะอาด. รวมกฎหมาย

สิ่งแวดล้อมและการรักษาความสะอาด. แหล่งที่มา: <http://www.wachirawit.ac.th/elibrary/CustomViewIlist.asp?start=3121>, 28 พฤษภาคม 2540.

อภิรัตน์ อุปลาระกุล. 2541. การออกแบบระบบสัญญาณเตือนอัคคีภัย. นิวสเด็ป. แหล่งที่มา:

[http://www.ployfah.com/knowledge.php?content\\_id=4](http://www.ployfah.com/knowledge.php?content_id=4), 10 เมษายน 2541.

Michael, B. 2001. **Pulse Width Modulation**. Embedded Systems Programming (2001) 103-104

Tantibundhit, C., J. Boston, C. Li, J. Durrant, S. Shaiman, K. Kovacyk, and A. EI-Jaroudi. 2007.

**New signal decomposition method based speech enhancement**, pp. 2607 – 2628.

Signal Processing, vol. 87, no. 11

Tantibundhit, C., F. Pernkopf, and G. Kubin. 2009. **Speech enhancement based on joint time-**

**frequency segmentation**, pp. 4673 – 4679. In Proc of ICASSP 2009

Yoo, S., J. Boston, A. EI-Jaroudi, C. Li, J. Durrant, K. Kovacyk, and S. Shaiman. 2007. **Speech**

**signal modification to increase intelligibility in noisy environment**, pp. 1138 – 1149.

J. Acoust. Soc. Am, vol. 122, no. 2



ภาคผนวก

## ภาคผนวกโปรแกรม

```
-- Module clk_main
-- Generated by Xilinx Architecture Wizard
-- Written for synthesis tool: XST

library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
library UNISIM;
use UNISIM.Vcomponents.ALL;

entity clk_main is
  port ( CLKIN_IN      : in  std_logic;
        RST_IN        : in  std_logic;
        CLKDV_OUT     : out std_logic;
        CLKIN_IBUFG_OUT : out std_logic;
        CLK0_OUT      : out std_logic;
        LOCKED_OUT    : out std_logic);
end clk_main;

architecture BEHAVIORAL of clk_main is
  signal CLKDV_BUF      : std_logic;
  signal CLKFB_IN      : std_logic;
  signal CLKIN_IBUFG   : std_logic;
  signal CLK0_BUF      : std_logic;
  signal GND_BIT       : std_logic;
begin
  GND_BIT <= '0';
  CLKIN_IBUFG_OUT <= CLKIN_IBUFG;
```

```

CLK0_OUT <= CLKFB_IN;
CLKDV_BUF_INST : BUFG
  port map (I=>CLKDV_BUF,
            O=>CLKDV_OUT);

```

```

CLKIN_IBUF_INST : IBUFG
  port map (I=>CLKIN_IN,
            O=>CLKIN_IBUF);

```

```

CLK0_BUF_INST : BUFG
  port map (I=>CLK0_BUF,
            O=>CLKFB_IN);

```

```

DCM_SP_INST : DCM_SP
  generic map( CLK_FEEDBACK => "1X",
                CLKDV_DIVIDE => 15.0,
                CLKFX_DIVIDE => 1,
                CLKFX_MULTIPLY => 4,
                CLKIN_DIVIDE_BY_2 => FALSE,
                CLKIN_PERIOD => 20.000,
                CLKOUT_PHASE_SHIFT => "NONE",
                DESKEW_ADJUST => "SYSTEM_SYNCHRONOUS",
                DFS_FREQUENCY_MODE => "LOW",
                DLL_FREQUENCY_MODE => "LOW",
                DUTY_CYCLE_CORRECTION => TRUE,
                FACTORY_JF => x"C080",
                PHASE_SHIFT => 0,
                STARTUP_WAIT => FALSE)
  port map (CLKFB=>CLKFB_IN,
            CLKIN=>CLKIN_IBUF,

```

```

DSSSEN=>GND_BIT,
PSCLK=>GND_BIT,
PSEN=>GND_BIT,
PSINCDEC=>GND_BIT,
RST=>RST_IN,
CLKDV=>CLKDV_BUF,
CLKFX=>open,
CLKFX180=>open,
CLK0=>CLK0_BUF,
CLK2X=>open,
CLK2X180=>open,
CLK90=>open,
CLK180=>open,
CLK270=>open,
LOCKED=>LOCKED_OUT,
PSDONE=>open,
STATUS=>open);

```

```
END BEHAVIORAL;
```

---

```
entity input_capture is
```

```

  Port ( clk : in  STD_LOGIC;
        rst : in  STD_LOGIC;
        din : in  STD_LOGIC_VECTOR (39 downto 0);
        dout : out STD_LOGIC_VECTOR (7 downto 0));

```

```
end input_capture;
```

```
architecture Behavioral of input_capture is
```

```

  signal din_reg : std_logic_vector(39 downto 0);

```

```

signal dout_reg : std_logic_vector(7 downto 0);
signal dout_reg_tmp : std_logic_vector(7 downto 0);
signal sum_ct : std_logic_vector(7 downto 0);
signal sum_ct_tmp : std_logic_vector(7 downto 0);
signal loop_ct : std_logic_vector(31 downto 0);
signal clk_ct : integer range 0 to 9 :=0;
signal clk_dv : std_logic;

type ram_type is array (39 downto 0) of std_logic_vector(7 downto 0);
signal tmp_ram: ram_type;

begin

tmp_ram(0) <= X"03";
tmp_ram(1) <= X"06";
tmp_ram(2) <= X"09";
tmp_ram(3) <= X"0c";
tmp_ram(4) <= X"0f";
tmp_ram(5) <= X"12";
tmp_ram(6) <= X"15";
tmp_ram(7) <= X"18";
tmp_ram(8) <= X"1b";
tmp_ram(9) <= X"1e";
tmp_ram(10) <= X"21";
tmp_ram(11) <= X"24";
tmp_ram(12) <= X"27";
tmp_ram(13) <= X"2a";
tmp_ram(14) <= X"2d";

```

```
tmp_ram(15) <= X"30";
tmp_ram(16) <= X"33";
tmp_ram(17) <= X"36";
tmp_ram(18) <= X"39";
tmp_ram(19) <= X"3c";
tmp_ram(20) <= X"3f";
tmp_ram(21) <= X"42";
tmp_ram(22) <= X"45";
tmp_ram(23) <= X"48";
tmp_ram(24) <= X"4b";
tmp_ram(25) <= X"4e";
tmp_ram(26) <= X"51";
tmp_ram(27) <= X"54";
tmp_ram(28) <= X"57";
tmp_ram(29) <= X"5a";
tmp_ram(30) <= X"5d";
tmp_ram(31) <= X"60";
tmp_ram(32) <= X"63";
tmp_ram(33) <= X"66";
tmp_ram(34) <= X"69";
tmp_ram(35) <= X"6c";
tmp_ram(36) <= X"6f";
tmp_ram(37) <= X"72";
tmp_ram(38) <= X"75";
tmp_ram(39) <= X"78";
```

```
div_proc : process(clk,rst)
```

```
begin
```

```
    if(rst='1')then
```

```
        clk_ct <= 0;
```

```

        clk_dv <= '1';
    elsif(rising_edge(clk))then
        if(clk_ct=1)then
            clk_ct <= 0;
            clk_dv <= not clk_dv;
        else
            clk_ct <= clk_ct + 1;
        end if;
    end if;
end process;

sum_proc : process(clk_dv,rst)
begin
    if(rst='1')then
        dout <= (others => '0');
        din_reg <= (others => '0');
        dout_reg <= (others => '0');
        dout_reg_tmp <= (others => '0');
        sum_ct <= (others => '0');
        loop_ct <= (others => '0');
    elsif(rising_edge(clk_dv))then
        if(sum_ct=40)then
            sum_ct <= (others => '0');
            din_reg <= din;

            if(loop_ct=10000)then
                loop_ct <= (others => '0');
                dout <= dout_reg_tmp;
                dout_reg_tmp <= (others => '0');
                dout_reg <= (others => '0');
            end if;
        end if;
    end if;
end process;

```

```

else
    if(dout_reg>dout_reg_tmp)then
        dout_reg_tmp <= dout_reg;
    end if;
    loop_ct <= loop_ct + '1';
end if;
else
    if(din_reg(conv_integer(sum_ct))='0')then
        dout_reg <= tmp_ram(conv_integer(sum_ct));
    end if;
    sum_ct <= sum_ct + '1';
end if;
end if;
end process;
END Behavioral;

```

---

```

entity segment_ctrl is

```

```

    Port ( clk : in  STD_LOGIC;
          rst : in  STD_LOGIC;
          din : in  STD_LOGIC_VECTOR (7 downto 0);
          dout : out STD_LOGIC_VECTOR (7 downto 0);
          col : out STD_LOGIC_VECTOR (2 downto 0));

```

```

end segment_ctrl;

```

```

architecture Behavioral of segment_ctrl is

```

```

    signal seg1 : integer range 0 to 12 :=0;

```

```
signal seg2 : integer range 0 to 12 :=0;
```

```
signal seg3 : integer range 0 to 12 :=0;
```

```
type ram_type is array (11 downto 0) of std_logic_vector(7 downto 0);
```

```
signal tmp_ram: ram_type;
```

```
type state_scan_type is (seg1_on,seg2_on,seg3_on,col1_on,col2_on,col3_on);
```

```
signal state_scan : state_scan_type;
```

```
signal ct : integer range 0 to 100 :=0;
```

```
begin
```

```
--tmp_ram(0) <= "00111111";
```

```
--tmp_ram(1) <= "00000110";
```

```
--tmp_ram(2) <= "01011011";
```

```
--tmp_ram(3) <= "01001111";
```

```
--tmp_ram(4) <= "01100110";
```

```
--tmp_ram(5) <= "01101101";
```

```
--tmp_ram(6) <= "01111100";
```

```
--tmp_ram(7) <= "00000111";
```

```
--tmp_ram(8) <= "01111111";
```

```
--tmp_ram(9) <= "01100111";
```

```
--tmp_ram(10) <= "00000000";
```

```
--tmp_ram(11) <= "00000000";
```

```
tmp_ram(0) <= "00111111";
```

```
tmp_ram(1) <= "00000110";
```

```
tmp_ram(2) <= "01011011";
```

```

tmp_ram(3) <= "01001111";
tmp_ram(4) <= "01100110";
tmp_ram(5) <= "01101101";
tmp_ram(6) <= "01111100";
tmp_ram(7) <= "00000111";
tmp_ram(8) <= "01111111";
tmp_ram(9) <= "01100111";
tmp_ram(10) <= "00000000";
tmp_ram(11) <= "00000000";

scan_proc : process(clk,rst)
begin
    if(rst='1')then
        state_scan <= seg1_on;
        ct <= 0;
        col <= "111";
        dout <= (others => '1');
    elsif(rising_Edge(clk))then
        case state_scan is
            when seg1_on =>
                dout <= not tmp_ram(seg1);
                if(ct=20)then
                    ct <= 0;
                    state_scan <= coll_on;
                else
                    ct <= ct + 1;
                end if;
            when coll_on =>
                col <= "110";
        end case;
    end if;
end process;

```

```

if(ct=80)then
    ct <= 0;
    col <= "111";
    dout <= (others => '0');
    state_scan <= seg2_on;
else
    ct <= ct + 1;
end if;
when seg2_on =>
    dout <= not tmp_ram(seg2);
    if(ct=20)then
        ct <= 0;
        --dout <= (others => '0');
        state_scan <= col2_on;
    else
        ct <= ct + 1;
    end if;
when col2_on =>
    col <= "101";
    if(ct=80)then
        ct <= 0;
        col <= "111";
        dout <= (others => '0');
        state_scan <= seg3_on;
    else
        ct <= ct + 1;
    end if;
when seg3_on =>
    dout <= not tmp_ram(seg3);

```

```

        if(ct=20)then
            ct <= 0;
            state_scan <= col3_on;
        else
            ct <= ct + 1;
        end if;
    when col3_on =>
        col <= "011";
        if(ct=80)then
            ct <= 0;
            col <= "111";
            dout <= (others => '0');
            state_scan <= seg1_on;
        else
            ct <= ct + 1;
        end if;
    end case;
end if;

end process;

-- look up
look_proc : process(clk,rst)
begin
    if(rst='1')then
        seg1 <= 11;
        seg2 <= 11;
        seg3 <= 11;
    elsif(rising_edge(clk))then
        case din is

```

```
when X"00" =>
    seg1 <= 11; seg2 <= 11; seg3 <= 0;
```

```
when X"03" =>
    seg1 <= 11; seg2 <= 11; seg3 <= 3;
```

```
when X"06" =>
    seg1 <= 11; seg2 <= 11; seg3 <= 6;
```

```
when X"09" =>
    seg1 <= 11; seg2 <= 11; seg3 <= 9;
```

```
when X"0c" =>
    seg1 <= 11; seg2 <= 1; seg3 <= 2;
```

```
when X"0F" =>
    seg1 <= 11; seg2 <= 1; seg3 <= 5;
```

```
when X"12" =>
    seg1 <= 11; seg2 <= 1; seg3 <= 8;
```

```
when X"15" =>
    seg1 <= 11; seg2 <= 2; seg3 <= 1;
```

```
when X"18" =>
    seg1 <= 11; seg2 <= 2; seg3 <= 4;
```

```
when X"1b" =>
    seg1 <= 11; seg2 <= 2; seg3 <= 7;
```

```
when X"1e" =>
    seg1 <= 11; seg2 <= 3; seg3 <= 0;
```

```
when X"21" =>
    seg1 <= 11; seg2 <= 3; seg3 <= 3;
```

```
when X"24" =>
    seg1 <= 11; seg2 <= 3; seg3 <= 6;
```

```
when X"27" =>
    seg1 <= 11; seg2 <= 3; seg3 <= 9;
```

```

when X"2a" =>
    seg1 <= 11; seg2 <= 4; seg3 <= 2;
when X"2d" =>
    seg1 <= 11; seg2 <= 4; seg3 <= 5;
when X"30" =>
    seg1 <= 11; seg2 <= 4; seg3 <= 8;
when X"33" =>
    seg1 <= 11; seg2 <= 5; seg3 <= 1;
when X"36" =>
    seg1 <= 11; seg2 <= 5; seg3 <= 4;
when X"39" =>
    seg1 <= 11; seg2 <= 5; seg3 <= 7;
when X"3c" =>
    seg1 <= 11; seg2 <= 6; seg3 <= 0;
when X"3f" =>
    seg1 <= 11; seg2 <= 6; seg3 <= 3;
when X"42" =>
    seg1 <= 11; seg2 <= 6; seg3 <= 6;
when X"45" =>
    seg1 <= 11; seg2 <= 6; seg3 <= 9;
when X"48" =>
    seg1 <= 11; seg2 <= 7; seg3 <= 2;
when X"4b" =>
    seg1 <= 11; seg2 <= 7; seg3 <= 5;
when X"4e" =>
    seg1 <= 11; seg2 <= 7; seg3 <= 8;
when X"51" =>
    seg1 <= 11; seg2 <= 8; seg3 <= 1;
when X"54" =>

```

```

        seg1 <= 11; seg2 <= 8; seg3 <= 4;
when X"57" =>
        seg1 <= 11; seg2 <= 8; seg3 <= 7;
when X"5a" =>
        seg1 <= 11; seg2 <= 9; seg3 <= 0;
when X"5d" =>
        seg1 <= 11; seg2 <= 9; seg3 <= 3;
when X"60" =>
        seg1 <= 11; seg2 <= 9; seg3 <= 6;

when X"63" =>
        seg1 <= 11; seg2 <= 9; seg3 <= 9;
when X"66" =>
        seg1 <= 1; seg2 <= 0; seg3 <= 2;
when X"69" =>
        seg1 <= 1; seg2 <= 0; seg3 <= 5;
when X"6c" =>
        seg1 <= 1; seg2 <= 0; seg3 <= 8;
when X"6f" =>
        seg1 <= 1; seg2 <= 1; seg3 <= 1;
when X"72" =>
        seg1 <= 1; seg2 <= 1; seg3 <= 4;
when X"75" =>
        seg1 <= 1; seg2 <= 1; seg3 <= 7;
when X"78" =>
        seg1 <= 1; seg2 <= 2; seg3 <= 0;
when X"7b" =>
        seg1 <= 9; seg2 <= 9; seg3 <= 9;
when others =>
        seg1 <= 9; seg2 <= 9; seg3 <= 9;

```

```

        end case;

    end if;

end process;

END Behavioral;

-----

entity segment_pwm is
    Port ( clk : in  STD_LOGIC;
          rst : in  STD_LOGIC;
          dout : out STD_LOGIC_VECTOR (7 downto 0);
          col : out STD_LOGIC_VECTOR (2 downto 0);
          din : in  STD_LOGIC_VECTOR (3 downto 0));
end segment_pwm;

architecture Behavioral of segment_pwm is

    signal seg1 : integer range 0 to 12 :=0;
    signal seg2 : integer range 0 to 12 :=0;
    signal seg3 : integer range 0 to 12 :=0;

    type ram_type is array (11 downto 0) of std_logic_vector(7 downto 0);
    signal tmp_ram: ram_type;

    type state_scan_type is (seg1_on,seg2_on,seg3_on,col1_on,col2_on,col3_on);
    signal state_scan : state_scan_type;

    signal ct : integer range 0 to 100 :=0;

begin

```

```

tmp_ram(0) <= "00111111";
tmp_ram(1) <= "00000110";
tmp_ram(2) <= "01011011";
tmp_ram(3) <= "01001111";
tmp_ram(4) <= "01100110";
tmp_ram(5) <= "01101101";
tmp_ram(6) <= "01111100";
tmp_ram(7) <= "00000111";
tmp_ram(8) <= "01111111";
tmp_ram(9) <= "01100111";
tmp_ram(10) <= "00000000";
tmp_ram(11) <= "00000000";

scan_proc : process(clk,rst)
begin
  if(rst='1')then
    state_scan <= seg1_on;
    ct <= 0;
    col <= "111";
    dout <= (others => '1');
  elsif(rising_Edge(clk))then
    case state_scan is
      when seg1_on =>
        dout <= not tmp_ram(seg1);
        if(ct=20)then
          ct <= 0;
          state_scan <= coll_on;
        else

```

```

        ct <= ct + 1;
    end if;

    when col1_on =>
        col <= "110";
        if(ct=80)then
            ct <= 0;
            col <= "111";
            dout <= (others => '0');
            state_scan <= seg2_on;
        else
            ct <= ct + 1;
        end if;
    when seg2_on =>
        dout <= not tmp_ram(seg2);
        if(ct=20)then
            ct <= 0;
            state_scan <= col2_on;
        else
            ct <= ct + 1;
        end if;
    when col2_on =>
        col <= "101";
        if(ct=80)then
            ct <= 0;
            col <= "111";
            dout <= (others => '0');
            state_scan <= seg3_on;
        else
            ct <= ct + 1;
        end if;
    end when;
end process;

```

```

        end if;
    when seg3_on =>
        dout <= not tmp_ram(seg3);
        if(ct=20)then
            ct <= 0;
            state_scan <= col3_on;
        else
            ct <= ct + 1;
        end if;
    when col3_on =>
        col <= "011";
        if(ct=80)then
            ct <= 0;
            col <= "111";
            dout <= (others => '0');
            state_scan <= seg1_on;
        else
            ct <= ct + 1;
        end if;
    end case;
end if;

end process;

-- look up
look_proc : process(clk,rst)
begin
    if(rst='1')then
        seg1 <= 11;
        seg2 <= 11;
        seg3 <= 11;

```

```

elsif(rising_edge(clk))then
    case din is
        when X"0" =>
            seg1 <= 11; seg2 <= 11; seg3 <= 0;
        when X"1" =>
            seg1 <= 11; seg2 <= 1; seg3 <= 0;
        when X"2" =>
            seg1 <= 11; seg2 <= 2; seg3 <= 0;
        when X"3" =>
            seg1 <= 11; seg2 <= 3; seg3 <= 0;
        when X"4" =>
            seg1 <= 11; seg2 <= 4; seg3 <= 0;
        when X"5" =>
            seg1 <= 11; seg2 <= 5; seg3 <= 0;
        when X"6" =>
            seg1 <= 11; seg2 <= 6; seg3 <= 0;
        when X"7" =>
            seg1 <= 11; seg2 <= 7; seg3 <= 0;
        when X"8" =>
            seg1 <= 11; seg2 <= 8; seg3 <= 0;
        when X"9" =>
            seg1 <= 11; seg2 <= 9; seg3 <= 0;
        when X"A" =>
            seg1 <= 1; seg2 <= 0; seg3 <= 0;

        when others =>
            seg1 <= 9; seg2 <= 9; seg3 <= 9;
    end case;
end if;

```

```
end process;
END Behavioral;
```

---

```
entity pwm_ctrl is
```

```
  Port ( clk : in  STD_LOGIC;
         rst : in  std_logic;
         din : in  STD_LOGIC_VECTOR (3 downto 0);
         pwm : out STD_LOGIC);
```

```
end pwm_ctrl;
```

```
architecture Behavioral of pwm_ctrl is
```

```
  signal pwm_ct : integer range 0 to 10 := 0;
  signal clk_dv : std_logic;
  signal clk_ct : integer range 0 to 50 :=0;
```

```
begin
```

```
  div_proc : process(clk,rst)
```

```
  begin
```

```
    if(rst='1')then
```

```
      clk_ct <= 0;
```

```
      clk_dv <= '1';
```

```
    elsif(rising_edge(clk))then
```

```
      if(clk_ct=20)then
```

```
        clk_ct <= 0;
```

```
        clk_dv <= not clk_dv;
```

```
      else
```

```
        clk_ct <= clk_ct + 1;
```

```
      end if;
```

```

        end if;
    end process;

    pwm_proc : process(clk_dv,rst)
    begin
        if(rst='1')then
            pwm <= '0';
            pwm_ct <= 0;
        elsif(rising_edge(clk_dv))then
            if(din = "0000")then
                pwm <= '0';
            else
                if(din > pwm_ct)then
                    pwm <= '1';
                else
                    pwm <= '0';
                end if;
            end if;

            if(pwm_ct=9)then
                pwm_ct <= 0;
            else
                pwm_ct <= pwm_ct + 1;
            end if;
        end if;
    end process;

END Behavioral;

```

---

```

entity Top_sCtrol is

```

```

    Port ( clk_in : in STD_LOGIC;

```

```

rst : in STD_LOGIC;

        led_run : out std_logic;
        led_start : out std_logic;

din : in STD_LOGIC_VECTOR (39 downto 0);
pwm : out STD_LOGIC;
start : in STD_LOGIC;
segment_in : out STD_LOGIC_VECTOR (7 downto 0);
col_in : out STD_LOGIC_VECTOR (2 downto 0);
segment_out : out STD_LOGIC_VECTOR (7 downto 0);
col_out : out STD_LOGIC_VECTOR (2 downto 0));
end Top_sCtrl;

architecture Behavioral of Top_sCtrl is

constant db_constant : std_logic_vector(7 downto 0) := "00001001";
signal db_input : std_logic_vector(7 downto 0);
signal din_pwm : std_logic_vector(3 downto 0);
signal sum : std_logic_vector(7 downto 0);
signal clk : std_logic;
--signal rst : std_logic;

COMPONENT input_capture
PORT(
        clk : IN std_logic;
        rst : IN std_logic;
        din : IN std_logic_vector(39 downto 0);
        dout : OUT std_logic_vector(7 downto 0)
);
END COMPONENT;

```

```
COMPONENT pwm_ctrl
```

```
  PORT(
```

```
    clk : IN std_logic;
```

```
    rst : IN std_logic;
```

```
    din : IN std_logic_vector(3 downto 0);
```

```
    pwm : OUT std_logic
```

```
  );
```

```
END COMPONENT;
```

```
COMPONENT segment_ctrl
```

```
  PORT(
```

```
    clk : IN std_logic;
```

```
    rst : IN std_logic;
```

```
    din : IN std_logic_vector(7 downto 0);
```

```
    dout : OUT std_logic_vector(7 downto 0);
```

```
    col : OUT std_logic_vector(2 downto 0)
```

```
  );
```

```
END COMPONENT;
```

```
COMPONENT segment_pwm
```

```
  PORT(
```

```
    clk : IN std_logic;
```

```
    rst : IN std_logic;
```

```
    din : IN std_logic_vector(3 downto 0);
```

```
    dout : OUT std_logic_vector(7 downto 0);
```

```
    col : OUT std_logic_vector(2 downto 0)
```

```
  );
```

```
END COMPONENT;
```

```
COMPONENT clk_main
```

```

PORT(
    CLKIN_IN : IN std_logic;
    RST_IN : IN std_logic;
    CLKDV_OUT : OUT std_logic;
    CLKIN_IBUFG_OUT : OUT std_logic;
    CLK0_OUT : OUT std_logic;
    LOCKED_OUT : OUT std_logic
);
END COMPONENT;

begin

--rst <= not rstn;
led_run <= not rst;
sum <= db_constant + db_input;
led_start <= start;

pwm_decode_proc: process(clk,rst)
begin
    if(rst='1')then
        din_pwm <= (OTHERS => '0');
    elsif(rising_edge(clk))then
        if(start='1')then
            if(sum>114)then
                din_pwm <= X"A";
            elsif(sum>107)then
                din_pwm <= X"9";
            elsif(sum>105)then
                din_pwm <= X"8";
            elsif(sum>103)then

```

```

        din_pwm <= X"7";
    elsif(sum>101)then
        din_pwm <= X"6";
    elsif(sum>98)then
        din_pwm <= X"5";
    elsif(sum>92)then
        din_pwm <= X"4";
    elsif(sum>87)then
        din_pwm <= X"3";
    elsif(sum>70)then
        din_pwm <= X"2";
    else
        din_pwm <= X"1";
    end if;
else
    din_pwm <= (OTHERS => '0');
end if;
end if;
end process;

```

```

Inst_input_capture: input_capture PORT MAP(

```

```

    clk => clk,
    rst => rst,
    din => din,
    dout => db_input

```

```

);

```

```

Inst_pwm_ctrl: pwm_ctrl PORT MAP(

```

```

    clk => clk,
    rst => rst,

```

```

    din => din_pwm,
    pwm => pwm
);

```

```

Inst_segment_ctrl: segment_ctrl PORT MAP(
    clk => clk,
    rst => rst,
    din => db_input,
    dout => segment_in,
    col => col_in
);

```

```

Inst_segment_pwm: segment_pwm PORT MAP(
    clk => clk,
    rst => rst,
    dout => segment_out,
    col => col_out,
    din => din_pwm
);

```

```

Inst_clk_main: clk_main PORT MAP(
    CLKIN_IN => clk_in,
    RST_IN => rst,
    CLKDV_OUT => clk,
    CLKIN_IBUFG_OUT => open,
    CLK0_OUT => open,
    LOCKED_OUT => open
);

```

```

END Behavioral;

```

## ประวัติการศึกษา และการทำงาน

ชื่อ –นามสกุล	อภิสิทธิ์์ สุนทรพะลิน
วัน เดือน ปี ที่เกิด	27 ม.ค 2527
สถานที่เกิด	กรุงเทพมหานคร
ประวัติการศึกษา	วศ.บ. (ไฟฟ้า) มหาวิทยาลัยเกษตรศาสตร์
ตำแหน่งหน้าที่การงานปัจจุบัน	-
สถานที่ทำงานปัจจุบัน	-
ผลงานดีเด่นและรางวัลทางวิชาการ	-
ทุนการศึกษาที่ได้รับ	-