



## รายงานการวิจัยฉบับสมบูรณ์

การออกแบบวงจรรวมถอดรหัสสัญญาณเสียงแบบบิทเรทต่ำมาก

สำหรับโครงข่ายอินเทอร์เน็ตของสรรพสิ่ง

**Very Low Bitrate Voice Decoder IC Design  
for Internet of Things Applications**

ดร.สุเมธ วิชาญทัชชัย

ได้รับทุนสนับสนุนงานวิจัยจากเงินกองทุนวิจัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปีงบประมาณ 2561

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## รายงานการวิจัยฉบับสมบูรณ์

การออกแบบวงจรรวมถอดรหัสสัญญาณเสียงแบบบิทเรทต่ำมาก

สำหรับโครงข่ายอินเทอร์เน็ตของสรรพสิ่ง

**Very Low Bitrate Voice Decoder IC Design  
for Internet of Things Applications**

ดร.สุเมธ วิชาญทัณ

ได้รับทุนสนับสนุนงานวิจัยจากเงินกองทุนวิจัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปีงบประมาณ 2561

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงการ (ภาษาไทย) การออกแบบวงจรรวมถอดรหัสสัญญาณเสียงแบบบีทเรทต่ำมากสำหรับโครงข่าย  
อินเทอร์เน็ตของสรรพสิ่ง

แหล่งเงิน เงินกองทุนวิจัยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประจำปีงบประมาณ 2561 จำนวนเงินที่ได้รับการสนับสนุน 100,000 บาท

ระยะเวลาทำการวิจัย 1 ปี 6 เดือน ตั้งแต่ 15 กรกฎาคม 2561 ถึง 14 มกราคม 2563

หัวหน้าโครงการ ดร.สุเมฆ วิศยทักษิณ หน่วยงานต้นสังกัด ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

### บทคัดย่อ

ปัจจุบันอินเทอร์เน็ตแห่งสรรพสิ่ง (IoT) ได้เข้ามามีบทบาทสำคัญในการเชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์  
ทั้งหลายให้สามารถสื่อสารกันได้ สิ่งสำคัญคืออุปกรณ์เหล่านั้นต้องมีราคาไม่แพง แต่ในขณะเดียวกันต้องมี  
ประสิทธิภาพสูงและใช้พลังงานต่ำ โดยส่วนใหญ่ข้อมูลที่ส่งผ่านโครงข่ายเหล่านี้มันเป็นสัญญาณง่าย ๆ ที่วัด  
จากค่าเซ็นเซอร์แบบต่าง ๆ ดังนั้นในการส่งสัญญาณที่ซับซ้อนเช่นสัญญาณเสียงจะติดข้อจำกัดของโครงข่าย  
แบบนี้เนื่องจากต้องการหน่วยประมวลผลที่มีความซับซ้อนและใช้พลังงานสูง งานวิจัยนี้มุ่งเน้นการออกแบบ  
วงจรรวมเฉพาะสำหรับการถอดรหัสสัญญาณเสียงแบบบีทเรทต่ำมาก Codec2 เพื่อใช้ลดภาระของหน่วย  
ประมวลผลกลาง โดยใช้เทคนิคของการประมวลผลแบบขนานและ ไปป์ไลน์ อีกทั้งสามารถลดความซับซ้อน  
ของการถอดรหัสจากการคำนวณแบบจุดลอยตัวมาเป็นแบบจุดตรึง อีกทั้งยังสามารถคำนวณฟังก์ชันทาง  
คณิตศาสตร์ที่จำเป็นเช่นการแปลงฟูเรียร์แบบเร็ว, ฟังก์ชันตรีโกณมิติ, เอ็กโปเนนเชียลและลอการิทึมได้อย่าง  
รวดเร็ว จากผลการทดลองพบว่าสามารถถอดรหัสสัญญาณเสียงขนาด 1200 บิตต่อวินาทีด้วยสัญญาณนาฬิกา  
เพียง 4 MHz หรือน้อยกว่าไมโครโปรเซสเซอร์ Cortex-M4 ถึง 20 เท่า และใช้พลังงาน 0.093 มิลลิวัตต์จาก  
การทดลองผ่านวงจรใน FPGA Xilinx Artix-7 หรือน้อยกว่าไมโครโปรเซสเซอร์ประมาณ 89 เท่า

คำสำคัญ วงจรรวมดิจิทัล, วงจรถอดรหัสสัญญาณเสียง, อินเทอร์เน็ตแห่งสรรพสิ่ง

**Research Title: Very Low Bitrate Voice Decoder IC Design for Internet of Things Applications**

**Researcher: Dr.Sumek Wisayataksin**

**Faculty: Engineering Department: Electronics Engineering**

## ABSTRACT

Nowadays, the internet of thing (IoT) plays an important role by connecting a lot of electronic devices together into a big communication network. The important factors for those devices are the high performance, low power consumption, but it must be low cost. Usually, the transmitted data in IoT network is a simple signal from a sensor. There is a limitation for sophisticated signals such as human voice to be processed in this kind of network, since the complicated signal processing are required which leads very high power consumption. In this research, the high performance integrated circuit for decoding very low bitrate human voice in Codec2 scheme was introduced in order to reduce the load of main CPU. We proposed the techniques of parallel computation in pipeline fashion to reduce the bottleneck of data processing. Moreover, the floating point arithmetic was optimized to fixed point computation, which can reduce the complexity of ALU and circuit size a lot. The fast fourier transform, trigonometry, exponential and logarithm function can be computed efficiently by the proposed architecture. The experimental results revealed that 1200 bps voice signal can be decoded by using only 4 MHz clock signal or 20 times lower than a conventional Cortex-M4 CPU. In addition, the circuit implementation on FPGA Xilinx Artix-7 XC7A100T showed that it consumed 0.093 milliwatt or less than the microprocessor around 89 times.

**Keywords : Digital Integrated Circuits, Voice Decoder Circuit, Internet of Things**

## กิตติกรรมประกาศ

ขอขอบคุณภาควิชาวิศวกรรมอิเล็กทรอนิกส์ รศ.ดร.สุรพันธุ์ เอื้อไพบูลย์สำหรับความช่วยเหลือในฐานะหัวหน้าภาควิชาและนักวิจัยที่เลี้ยง ขอขอบคุณ Dr.David Rown ผู้คิดค้นการเข้ารหัสสัญญาณเสียงแบบ Codec2 สำหรับข้อเสนอในการประมวลผลสัญญาณเสียงคน ขอขอบคุณ ดร.ตรีสุคนธ์ ตรีบุพชาติสกุล สำหรับความช่วยเหลือด้านเอกสารวิจัยและอื่น ๆ การวิจัยครั้งนี้ได้รับทุนสนับสนุนการวิจัยจากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง จากแหล่งทุนเงินกองทุนวิจัยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ประจำปีงบประมาณ พ.ศ. 2561 เลขที่ทุน KREF186103

ผู้วิจัย ดร.สุเมฆ วิศยทักษิณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของการวิจัย.....	1
1.3 ขอบเขตของการวิจัย.....	2
1.4 วิธีดำเนินการวิจัย.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง.....	3
2.1 แนวคิด ทฤษฎีที่เกี่ยวข้อง.....	3
2.2 งานวิจัยที่เกี่ยวข้อง.....	19
บทที่ 3 วิธีดำเนินการวิจัย.....	21
3.1 วิธีดำเนินการวิจัย.....	21
3.2 แผนการดำเนินงานวิจัย.....	21
3.3 อุปกรณ์ที่ใช้ในการวิจัย.....	22
3.4 กระบวนการออกแบบสถาปัตยกรรมและวงจรถอดรหัสสัญญาณเสียง Codec2.....	23
บทที่ 4 ผลการวิจัย.....	39
4.1 ผลการทดสอบการทำงานของวงจรแปลงคู่ของเส้นสเปกตรัม.....	39
4.2 ผลการทดสอบการทำงานของวงจรแปลงฟูเรียร์และการแปลงฟูเรียร์ผกผัน.....	39
4.3 ผลการทดสอบการทำงานของวงจรสร้างแอมพลิจูดฮาร์โมนิก.....	41
4.4 ผลการทดสอบการทำงานของวงจรสังเคราะห์เฟสของสัญญาณ.....	41
4.5 ผลการทดสอบการทำงานของวงจรกรองสัญญาณ.....	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.6 ผลการทดสอบการทำงานของการบวกแบบซ้อนทับ.....	43
4.7 ผลการทดสอบการทำงานของวงจรสื่อสารอนุกรมแบบอะซิงโครนัส.....	43
4.8 ผลการทดสอบการทำงานของวงจรสื่อสารอนุกรมแบบซิงโครนัส.....	44
4.9 ผลการเปรียบเทียบประสิทธิภาพของการถอดรหัส Codec2.....	44
4.10 ผลการสังเคราะห์วงจรและกำลังงานของวงจร.....	45
4.11 ผลการสร้างวงจรต้นแบบบนบอร์ด FPGA.....	46
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	48
5.1 สรุปผลการวิจัย.....	48
5.2 ข้อเสนอแนะ.....	48
บทที่ 6 สรุปผลผลิตงานวิจัย.....	49
6.1 รายชื่อและรายละเอียดผลผลิตงานวิจัยที่ผลิตได้.....	49
เอกสารอ้างอิง.....	50
ภาคผนวก.....	52
ภาคผนวก ก เอกสารการนำเสนอผลงาน.....	52
ภาคผนวก ข สรุปค่าใช้จ่ายการดำเนินโครงการวิจัย.....	59
ประวัตินักวิจัย.....	62

## สารบัญตาราง

ตารางที่	หน้า
2.1 เปรียบเทียบบิตเรทที่ต้องการของการบีบอัดสัญญาณเสียงแบบต่าง ๆ.....	5
2.2 ตารางเปรียบเทียบคุณสมบัติที่สำคัญของ LoRA และ Zigbee.....	15
3.1 แผนงานวิจัย.....	21
4.1 แสดงจำนวนสัญญาณนาฬิกาที่ใช้ในการถอดรหัสสัญญาณเสียงแบบ Codec2.....	45
4.2 ผลการสังเคราะห์วงจรด้วยไลบรารี GF 0.13um.....	45
4.3 แสดงกำลังงานที่ใช้ในการถอดรหัสสัญญาณเสียงแบบ Codec2.....	46
4.4 เพอร์เซ็นต์การใช้งานฮาร์ดแวร์ของ FPGA ต้นแบบและความเร็วสัญญาณนาฬิกา.....	47



## สารบัญภาพ

ภาพที่	หน้า
2.1 สัญญาณเสียงของมนุษย์ที่ถูกวิเคราะห์ด้วยสเปกตรัม	4
2.2 การสังเคราะห์สัญญาณเสียงด้วยการใช้วงจรกรองสัญญาณ	6
2.3 แผนภาพแสดงการถอดรหัสสัญญาณเสียงด้วยวิธี LPC	7
2.4 แผนภาพกระบวนการถอดรหัสสัญญาณเสียงด้วย Codec2	8
2.5 แผนผังการแปลงฟูรีเยร์แบบเร็ว (FFT) ของสัญญาณจำนวน 16 จุด	9
2.6 การรวมกำลังงานของสัญญาณ LPC ให้เท่ากับความถี่ของสัญญาณ Pitch	10
2.7 โปรแกรมสร้างเฟสของสัญญาณกระตุ้นแบบสุ่มสำหรับเสียง Unvoiced	11
2.8 แผนผังการแปลงฟูรีเยร์ผกผันแบบเร็ว (IFFT) ของสัญญาณจำนวน 16 จุด	12
2.9 สัญญาณเอาท์พุท 512 จุดจาก ifft จะถูกแบ่งออกเป็นสองช่วงหัวและท้าย	12
2.10 วินโดว์สามเหลี่ยมสำหรับกรองสัญญาณก่อนการบวกแบบซ้อนทับ	13
2.11 รูปแบบโครงสร้าง IoT ของบริษัท IBM	14
2.12 แสดงบิทเรทที่ส่งได้เปรียบเทียบกับระยะทาง	15
2.13 โมดูล LoRA กับบอร์ดชิพยูนิตต่าง ๆ และตัวโมดูลเดี่ยว	16
2.14 บล็อกไดอะแกรมแสดงการสื่อสารแบบ LoRA ซึ่งสามารถต่อตรงหรือผ่านเรทเตอร์ก็ได้	16
2.15 กระบวนการออกแบบวงจรรวมคิจิตอล	17
2.16 ตัวอย่างซอฟต์แวร์ที่ใช้ในการสอบทานวงจร สังเคราะห์วงจร และสร้าง Prototype บน FPGA	18
3.1 วงจรแปลงคู่ของเส้นสเปกตรัมไปเป็นสัมประสิทธิ์พายกรณ์เชิงเส้น	24
3.2 การวิเคราะห์เวลาของข้อมูลในการแปลง LSP ไป LPC	25
3.3 การคำนวณแบบ Butterfly สำหรับการแปลงฟูรีเยร์แบบเร็ว (FFT)	25
3.4 ลำดับการคำนวณ Butterfly หนึ่งครั้งด้วยการใช้วงจรบวกและวงจรถูคูณเพียงหนึ่งชุด	26
3.5 ขั้นตอนการทำ Butterfly ด้วยการรวมการอ่านและเขียนหน่วยความจำ	26
3.6 วงจรสลับข้อมูลเพื่ออ่านค่าในหน่วยความจำสำหรับการคำนวณ FFT	27
3.7 ค่า Twiddle Factor สำหรับการคำนวณการแปลงฟูรีเยร์แบบเร็ว	28
3.8 การประมาณค่าแบบเชิงเส้นที่ใช้ในการคำนวณค่าตรีโกณมิติ	29
3.9 การคำนวณหาพลังงานของสัญญาณจากผลการแปลงฟูรีเยร์แบบเร็ว	29

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.10 อัลกอริทึมการคำนวณการหารแบบ Goldschmidt.....	31
3.11 อัลกอริทึมการคำนวณฟังก์ชันเอ็กโปเนนเชียลแบบ Goldschmidt.....	32
3.12 อัลกอริทึมการคำนวณฟังก์ชันลอการิทึมแบบ Goldschmidt.....	33
3.13 โปรแกรมคำนวณสัญญาณสุ่มต้นแบบจากภาษาซี.....	34
3.14 ขั้นตอนการคำนวณสัญญาณสุ่มในรูปที่ 29 ใช้เวลาเพียง 2 สัญญาณนาฬิกา.....	34
3.15 การคูณสัญญาณเชิงซ้อนสำหรับสัญญาณกระตุ้นและสัญญาณอินพุท.....	34
3.16 ขั้นตอนการกรองสัญญาณด้วยวินโดว์แบบสามเหลี่ยมและบวกซ้อนทับสัญญาณ.....	35
3.17 โมดูล LoRa SX1278 ที่ใช้ในการทดลอง.....	35
3.18 โปรโตคอล SPI ที่ใช้ในการติดต่อโมดูล LoRa SX1278.....	36
3.19 การเชื่อมต่อกับคอมพิวเตอร์โดยการใช้โปรโตคอล UART.....	37
3.20 สถาปัตยกรรมโดยรวมของวงจรถอดรหัสสัญญาณเสียงแบบ Codec2.....	37
4.1 ผลการทดสอบการทำงานวงจรแปลงคู่ของเส้นสเปกตรัม.....	39
4.2 ผลการทดสอบการทำงานวงจรแปลงฟูเรียร์แบบเร็ว.....	40
4.3 ผลการทดสอบการทำงานวงจรแปลงฟูเรียร์แบบเร็ว.....	40
4.4 ผลการทดสอบการทำงานของวงจรสร้างแอมพลิจูดฮาร์โมนิก.....	41
4.5 ผลการทดสอบการทำงานของวงจรสังเคราะห์เฟสของสัญญาณ.....	42
4.6 ผลการทดสอบการทำงานของวงจรกรองสัญญาณ.....	42
4.7 ผลการทดสอบการทำงานของการบวกแบบซ้อนทับ.....	43
4.8 ผลการทดสอบการทำงานของวงจรสื่อสารอนุกรมแบบอะซิงโครนัส.....	44
4.9 ผลการทดสอบการทำงานของวงจรสื่อสารอนุกรมแบบซิงโครนัส.....	44
4.10 ผลการสร้างวงจรถอดรหัสเสียง Codec2 ต้นแบบด้วยบอร์ด FPGA.....	46

# บทที่ 1

## บทนำ

### 1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอินเทอร์เน็ตแห่งสรรพสิ่งได้เข้ามามีบทบาทสำคัญในการเชื่อมต่ออุปกรณ์อิเล็กทรอนิกส์ขนาดเล็กทั้งหลายให้สามารถสื่อสารกันได้ผ่านโครงข่ายอินเทอร์เน็ต ปัญหาที่สำคัญคืออุปกรณ์เหล่านั้นจำเป็นต้องมีค่าถูก, มีประสิทธิภาพสูง, แต่ในขณะเดียวกันต้องกินพลังงานต่ำเนื่องจากใช้พลังงานจากแบตเตอรี่ ในปัจจุบันข้อมูลที่ส่งผ่านโครงข่ายเหล่านี้ ถูกจำกัดเป็นค่าสัญญาณที่ไม่มีความซับซ้อนและข้อมูลน้อยมากเช่นค่าที่วัดจากเซ็นเซอร์แบบต่าง ๆ ข้อมูลที่ต้องการแบนวิธสูง ๆ เช่นข้อมูลภาพและเสียงติดข้อจำกัดจากโครงข่ายแบบนี้อยู่สองประการหลักคือ

1. โครงข่ายถูกออกแบบมาให้ส่งสัญญาณที่บิตเรทต่ำมาก ๆ เช่น 10 kbps เพื่อประหยัดพลังงาน
2. การประมวลผลที่ซับซ้อนเช่นการเข้ารหัสหรือถอดรหัสสื่อผสม (Multimedia) จำเป็นต้องใช้หน่วยประมวลผลที่ราคาแพงและกินพลังงานมาก

อย่างไรก็ดี หากสามารถออกแบบวงจรรวมแบบเฉพาะ (Application Specific Integrated Circuits) ที่สามารถประมวลผลที่ซับซ้อนได้แต่ใช้พลังงานต่ำ อีกทั้งเป็นวงจรรวมที่ทำการถอดรหัสสัญญาณที่บิตเรทต่ำมาก ๆ ได้ จะทำให้ข้อจำกัดของการสื่อสารผ่านโครงข่ายอินเทอร์เน็ตแห่งสรรพสิ่งลดลง เป็นประโยชน์ในวงกว้างและสามารถเพิ่มประสิทธิภาพของการสื่อสารในอนาคตได้ อีกทั้งยังเป็นองค์ความรู้ใหม่ของประเทศในการออกแบบวงจรรวมเพื่อการสื่อสารสามารถต่อยอดออกไปเป็นสินค้าเชิงพาณิชย์ในภาคอุตสาหกรรม ลดการพึ่งพาเทคโนโลยีการออกแบบและสร้างวงจรรวมจากต่างประเทศได้

### 1.2 วัตถุประสงค์ของการวิจัย

1. ออกแบบวงจรรวมแบบเฉพาะ (Application Specific Integrated Circuits) สำหรับถอดรหัสสัญญาณเสียงแบบบิตเรทต่ำมาก แก้ปัญหาที่ต้องใช้หน่วยประมวลผลที่ซับซ้อนและในพลังงานสูง เช่น CPU
2. สร้างระบบต้นแบบการสื่อสารด้วยเสียงผ่านโครงข่ายอินเทอร์เน็ตแห่งสรรพสิ่งที่มีขนาดเล็กประหยัดพลังงานด้วย FPGA
3. เปรียบเทียบประสิทธิภาพที่ได้วงจรรวมที่ออกแบบเทียบกับการใช้ CPU ในการประมวลผล
4. เพิ่มพูนองค์ความรู้ทางด้านการออกแบบวงจรรวมดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ขอบเขตของการวิจัย

1. ทำการศึกษาการเข้ารหัสเสียงแบบ Codec2 คิดค้นสถาปัตยกรรมทางวงจรที่สามารถหลีกเลี่ยงการประมวลผลแบบจุดลอยตัว ออกแบบหน่วยประมวลผลให้สามารถถอดรหัสสัญญาณเสียงด้วยการคำนวณทางคณิตศาสตร์แบบจุดตรึงเท่านั้นเพื่อประหยัดพลังงาน
2. ออกแบบวงจรรวมด้วยภาษา Verilog ทำการทวนสอบวงจร (Logic Verification) เพื่อตรวจสอบความถูกต้องของวงจร
3. สังเคราะห์วงจร (Logic Synthesis) ทำการวัดขนาดของวงจรผ่านการทำวงจรบน FPGA สร้างบอร์ดเพื่อติดต่อกับ โมดูลดรับส่งแบบ LoRa และ วัดกำลังงานที่ใช้เปรียบเทียบกับการประมวลผลด้วย CPU
4. สรุปผลการทดลอง

### 1.4 วิธีดำเนินการวิจัย

1. ศึกษาวิเคราะห์และออกแบบสถาปัตยกรรมเพื่อการถอดรหัสสัญญาณเสียงแบบ Codec2
2. ออกแบบวงจรด้วยภาษา Verilog ทำการทวนสอบวงจร (Verification)
3. จัดทำ Prototype บน FPGA บอร์ดเพื่อการทดสอบจริงกับ Lora Module
4. สังเคราะห์วงจร ประเมินขนาดและการใช้
5. วัดกำลังงานที่ใช้และความเร็วของสัญญาณนาฬิกาเปรียบเทียบกับการใช้ CPU แบบปกติ
6. สรุปผลการทดลอง ปัญหาที่พบ วิธีการแก้ไข เขียนสรุปรายงาน
7. นำเสนอผลงานวิชาการผ่านการประชุมวิชาการ

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

ประโยชน์ที่ได้รับคือ โครงสร้างสถาปัตยกรรมของวงจรรวมสำหรับการถอดรหัสสัญญาณเสียงแบบบิทเรตต่ำมากที่มีประสิทธิภาพสูง สามารถประยุกต์ใช้กับระบบที่มีข้อจำกัดทางด้านพลังงานเช่นระบบอินเทอร์เน็ตของสรรพสิ่ง (IoT) ที่ใช้พลังงานจากแบตเตอรี่เป็นส่วนใหญ่ อีกทั้งยังสามารถต่อยอดออกไปเป็นวงจรรวมสำหรับประมวลสัญญาณดิจิทัลที่หลากหลายในอนาคต ซึ่งสามารถลดภาระของหน่วยประมวลผลกลางได้อย่างมีประสิทธิภาพ

## บทที่ 2

### แนวคิด ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

#### 2.1 แนวคิด ทฤษฎีที่เกี่ยวข้อง

กระบวนการเข้ารหัสสัญญาณเสียงมนุษย์เป็นการประยุกต์การบีบอัดข้อมูลของสัญญาณเสียงดิจิทัลที่เป็นข้อมูลคำพูดของมนุษย์ การเข้ารหัสสัญญาณเสียงนั้นจะใช้คุณสมบัติพิเศษของเสียงมนุษย์เช่นเสียงที่มีการสั่น (Voiced Sound) และเสียงที่ไม่มีการสั่น Unvoiced Sound) ในการบีบอัดที่แตกต่างกัน โดยทั่วไปแล้วจะสามารถแปลงให้อยู่ในรูปของสัญญาณดิจิทัลที่มีขนาดเล็กได้มาก

โดยปกติสัญญาณเสียงของมนุษย์จะมีความถี่อยู่ในช่วง 400 เฮิรตซ์ ถึง 3500 เฮิรตซ์ ซึ่งมีช่วงที่ต่ำกว่าสัญญาณเสียงอื่น ๆ เช่นเสียงจากเครื่องดนตรี ที่มีความถี่ช่วง 20 เฮิรตซ์ ถึง 20000 เฮิรตซ์ ดังนั้นสัญญาณเสียงจำเป็นจะต้องมีกระบวนการเข้ารหัสที่พิเศษกว่าสัญญาณเสียงปกติ เพื่อให้สามารถบีบอัดข้อมูลได้มากกว่า โดยทั่วไปสามารถแบ่งการเข้ารหัสของเสียงออกได้เป็นสองกลุ่มหลักคือ

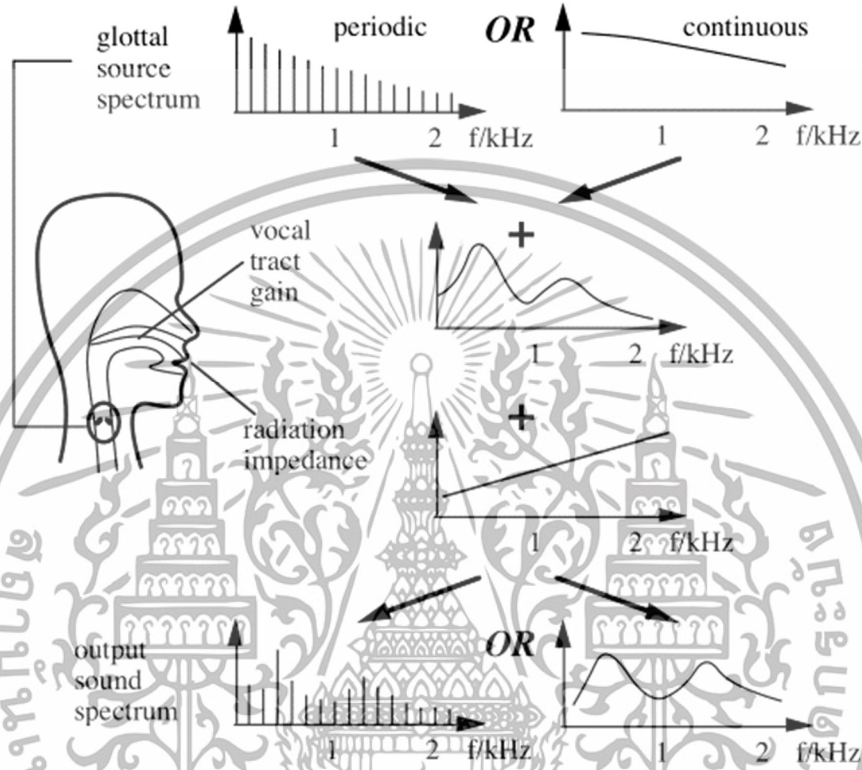
1. การเข้ารหัสรูปสัญญาณ (Waveform coders) สามารถแบ่งออกเป็น 2 แบบย่อยคือ
  - ในรูปแบบของแกนเวลา เช่น PCM หรือ ADPCM
  - ในรูปแบบของแกนความถี่ เช่น ATRAC
2. การเข้ารหัสเสียงโดยเฉพาะ (Vocoder) สามารถแบ่งออกเป็น 2 แบบย่อยคือ
  - การเข้ารหัสแบบพยากรณ์เชิงเส้น (Linear Predictive Coding)
  - การเข้ารหัสด้วยความถี่สั้นพ้อง (Formant Coding)

การเข้ารหัสแบบพยากรณ์เชิงเส้นนั้นเป็นที่นิยมมากในปัจจุบัน เนื่องจากสามารถจำลองเสียงพูดมนุษย์ให้อยู่ในรูปของแบบจำลองที่สามารถสังเคราะห์เสียงกลับมาได้อย่างมีประสิทธิภาพสูงแต่ใช้ข้อมูลน้อย

##### 2.1.1 การเข้ารหัสแบบพยากรณ์เชิงเส้น (Linear Predictive Coding)

การเข้ารหัสแบบพยากรณ์เชิงเส้น เริ่มต้นจากการประมาณเสียงของมนุษย์ด้วยการสร้างเสียงของออด (Buzzer) ที่ปลายของท่อ ซึ่งเรียกเสียงนั้นว่าเสียงที่มีการสั่น (Voiced Sound) สลับไปกับการเพิ่มเสียงของลมจากการทดลองพบว่าแบบจำลองนี้มีลักษณะที่คล้ายกับเสียงของมนุษย์

โดยทั่วไปแล้วการเข้ารหัสแบบพยากรณ์เชิงเส้นจะส่งสัญญาณที่เป็นกรอบของสเปกตรัมของเสียง โดยจะส่งเป็นสัมประสิทธิ์ของตัวกรองสัญญาณแบบดิจิทัลเพื่อให้ทนต่อการผิดเพี้ยนจากสัญญาณรบกวนที่เกิดขึ้น



รูปที่ 2.1 : สัญญาณเสียงของมนุษย์ที่ถูกวิเคราะห์ด้วยสเปกตรัม

มาตรฐานของการเข้ารหัสแบบพยากรณ์เชิงเส้นนั้นมีอยู่หลายแบบด้วยกัน เริ่มจากมาตรฐาน LPC-10 ซึ่งออกแบบโดยกระทรวงกลาโหมของสหรัฐอเมริกา มาตรฐานนี้ถือเป็นต้นแบบให้กับอัลกอริทึมต่าง ๆ ในปัจจุบัน ตัวอย่างเช่น Code-excited linear prediction (CELP) หรือ Mixed-excitation linear Prediction (MELP) โดย CELP นั้นได้พัฒนาการบีบอัดสัญญาณด้วยการแบ่งแบบจำลองพยากรณ์เชิงเส้นออกเป็น 2 แบบคือแบบระยะยาวและระยะสั้น อีกทั้งยังสร้าง ตารางรหัสของตัวกระตุ้น (Excitation codebook) ซึ่งสามารถสร้างลำดับของกระตุ้นได้อย่างมีประสิทธิภาพ จึงทำให้มาตรฐาน MPEG-4 เลือกที่จะใช้ CELP เป็นกระบวนการในการบีบอัดสัญญาณเสียง

นอกจาก CELP แล้ว ก็ยังมีผู้พัฒนาวิธีการบีบอัดให้ดียิ่งขึ้นด้วยการใช้การผสมของตัวกระตุ้น เรียกว่า Mixed-excitation linear Prediction (MELP) ซึ่งได้แก้ปัญหของเสียงรบกวนต่าง ๆ จนสามารถเป็นมาตรฐานการบีบอัดสัญญาณเสียงของหน่วยงานทางทหารชั้นนำเช่นองค์การนาโต้

นอกจาก 2 มาตรฐานที่ได้กล่าวมาแล้ว ยังมีอีก 1 วิธีการในการบีบอัดสัญญาณเสียงซึ่งได้พัฒนาโดยวิศวกรชาวออสเตรเลียชื่อ เดวิท โรว์ โดยมุ่งเน้นให้เป็นวิธีการที่ใช้ข้อมูลน้อยที่สุด เรียกว่า Codec2 ซึ่งสามารถบีบอัดสัญญาณเสียงได้ตั้งแต่ 3200 บิตต่อวินาทีไปจนถึง 450 บิตต่อวินาที ซึ่งเป็นวิธีที่มีประสิทธิภาพและเหมาะกับการสื่อสารที่ติดขัดด้วยช่องสัญญาณที่แคบมาก ๆ ได้เป็นอย่างดี

ตาราง 2.1 : เปรียบเทียบบิตเรทที่ต้องการของการบีบอัดสัญญาณเสียงแบบต่าง ๆ

ชนิดของการบีบอัดสัญญาณเสียง	จำนวนบิตเรทที่ต้องการ
Advanced Multiband Excitation (AMBE)	2000 bps
Mixed-excitation linear prediction (MELP)	1200 bps
Codec2	450 bps

ตารางที่ 2.1 แสดงการเปรียบเทียบข้อมูลที่ต้องการใน 1 วินาทีต่อการส่งสัญญาณเสียง จะเห็นได้ว่าวิธีการแบบ Codec2 สามารถบีบอัดสัญญาณได้มากถึง 450 บิตต่อวินาทีเท่านั้น

### 2.1.2 การเข้ารหัสสัญญาณเสียงแบบ Codec2

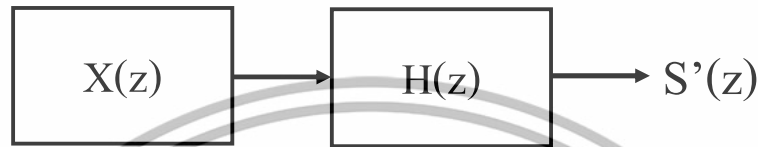
ในหัวข้อนี้จะมาวิเคราะห์การเข้ารหัสสัญญาณเสียงแบบ Codec2 อย่างละเอียด โดย Codec2 นั้นใช้วิธีการจำลองสัญญาณเสียงด้วยการเข้ารหัสเสียงด้วยฮาร์โมนิกของสัญญาณไซน์ (Harmonic sinusoidal speech coding) ซึ่งพัฒนาโดย MIT ในปี 1980 โดยหลักการแล้วเราสามารถจำลองสัญญาณเสียงให้อยู่ในรูปแบบด้วยผลบวกของสัญญาณไซน์ดังสมการที่ 2.1

$$s[n] = \sum_{m=1}^L A[m] \cos(\omega_0 mn + \varphi[m]) \quad (2.1)$$

จากสมการที่ 2.1 จะเห็นว่าสัญญาณเสียงคือผลรวมของสัญญาณไซน์ที่ความถี่เป็นจำนวนเท่าของความถี่มูลฐาน  $\omega_0$  ดังนั้นในแต่ละเฟรมของสัญญาณเราสามารถวิเคราะห์สัญญาณเสียงออกได้ด้วยตัวแปรสามตัวคือ

$$\omega_0, A[m] \text{ และ } \varphi[m] \quad (2.2)$$

โดยที่  $\omega_0$  คือความถี่มูลฐานหรือพีช (pitch) ของเสียง  $A[m]$  คือแอมพลิจูดของความถี่ต่าง ๆ และ  $\varphi[m]$  คือเฟสของสัญญาณ โดยเราจะเลือกค่า  $L$  ให้เท่ากับสัญญาณที่พอดีกับแบนด์วิดท์ที่ 4 กิโลเฮิรซ์ โครงสร้างของแบบจำลองที่ใช้ใน Codec2 นั้นนำมาจากการเข้ารหัสพยากรณ์เชิงเส้น ซึ่งมีรูปแบบดังรูปที่ 2.2



รูปที่ 2.2 : การสังเคราะห์สัญญาณเสียงด้วยการใช้วงจรกรองสัญญาณ

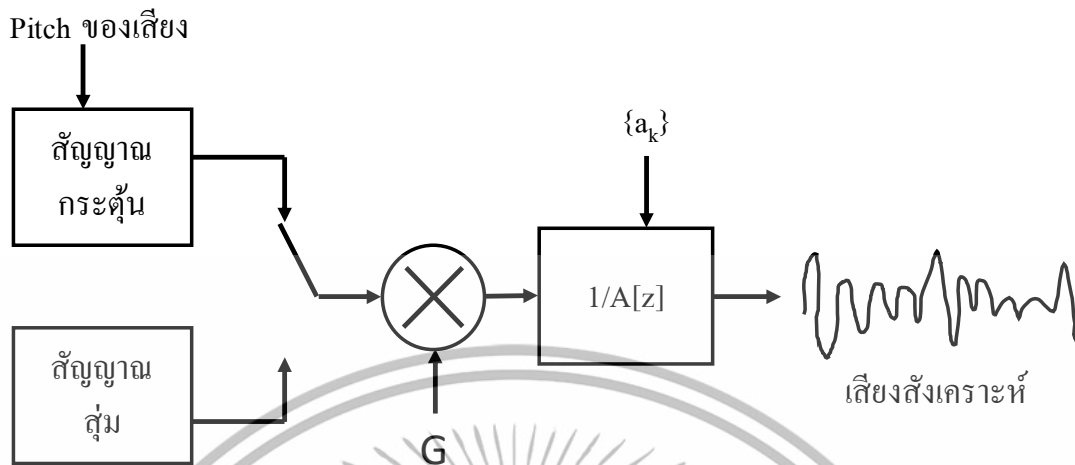
จากรูปที่ 2.2 เราสามารถพิจารณาให้  $H(z)$  เป็นวงจรกรองสัญญาณที่อยู่ในโดเมน  $z$  และ  $X(z)$  คือสัญญาณกระตุ้น (Excitation Source) ที่ป้อนให้กับวงจรกรองสัญญาณ เราจะได้สัญญาณเอาต์พุต  $S'(z)$  เป็นสัญญาณเสียงที่ถูกสังเคราะห์ขึ้น ดังนั้นในการจำลองสัญญาณเสียง เราเพียงแค่เก็บข้อมูลของสัญญาณกระตุ้นและคุณสมบัติของฟังก์ชันถ่ายโอน (Transfer Function) ของ  $H(z)$  ก็จะสามารถสร้างสัญญาณเสียงกลับคืนมาได้ อันนี้คือวิธีการของการบีบอัดสัญญาณเสียงด้วยการเข้ารหัสพยากรณ์เชิงเส้น

จากนิยาม เราจะสามารถหาสมการถ่ายโอนของ  $H(z)$  ได้จากวงจรกรองแบบ All-pole ได้ดังในสมการที่ 2.3

$$H(z) = \frac{G}{1 - \sum_{k=1}^p a_k z^{-k}} = \frac{G}{A(z)} \quad (2.3)$$

โดยที่  $\{a_k\}$  จะมีค่า  $k = 1, 2, \dots, p$  และ  $p$  คือเซตของสัมประสิทธิ์พยากรณ์เชิงเส้น ทั้งนี้คุณภาพของเสียงที่จำลองได้จะขึ้นอยู่กับค่าของ  $p$  โดยหาก  $p$  มีค่ามากจะทำให้สัญญาณเสียงที่ได้มีความคมชัด ซึ่งก็ต้องแลกมาด้วยข้อมูล  $a_k$  ที่มากขึ้น หรือทางตรงข้าม  $p$  ที่มีค่าน้อยจะทำให้สัญญาณเสียงมีความคมชัดต่ำ แต่สามารถประหยัดข้อมูล  $a_k$  ได้

วิธีการในการเข้ารหัสสัญญาณเสียงด้วย Codec2 คือการหาค่าคงที่  $a_k$  ที่เหมาะสมในแต่ละเฟรมของสัญญาณ นำมาประกอบกับความถี่มูลฐานของสัญญาณที่วิเคราะห์ได้และชนิดของเสียงในเฟรมนั้น ๆ ว่าเป็นเสียงแบบสั้น (Voiced Sound) หรือเสียงแบบไม่สั้น (Unvoiced Sound) โดยสามารถจำลองเป็นแผนภาพได้ดังรูปที่ 2.3



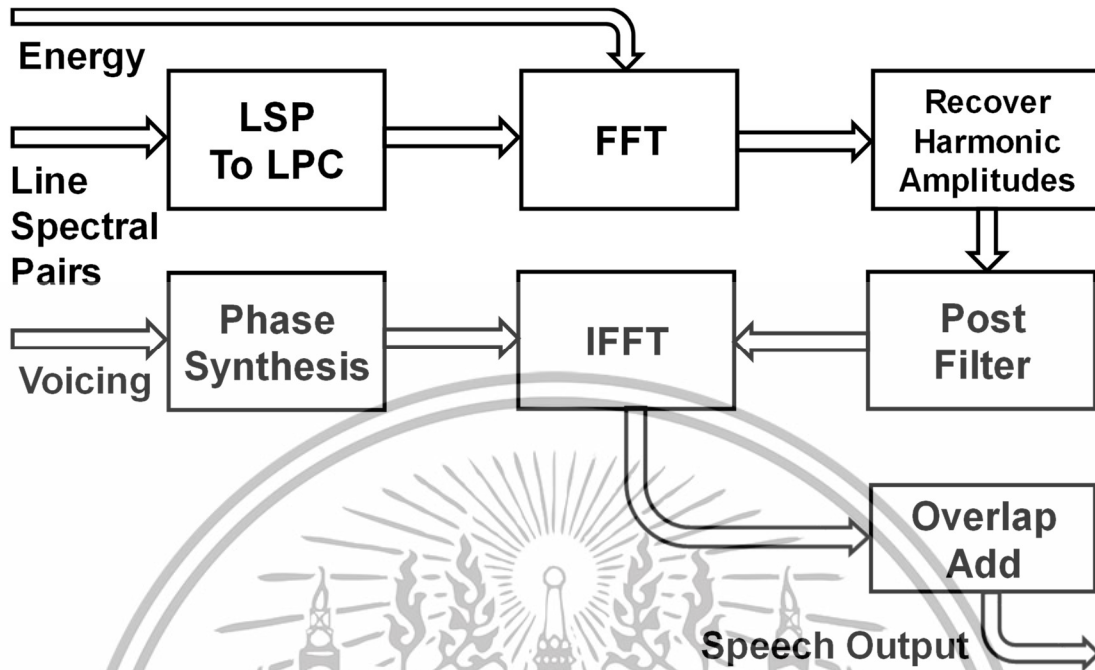
รูปที่ 2.3 : แผนภาพแสดงการถอดรหัสสัญญาณเสียงด้วยวิธี LPC

### 2.1.3 การถอดรหัสสัญญาณเสียงแบบบิตเรทต่ำมากด้วยวิธี Codec2

ในการถอดรหัสสัญญาณเสียงตามรูปที่ 2.3 นั้น พิกซ์ของสัญญาณเสียงจะถูกอ่านจากบิตสตรีมแล้วใช้ในการสร้างสัญญาณกระตุ้นที่เป็นสัญญาณไซน์ ในเวลาเดียวกันสัญญาณสุ่มจะถูกสร้างขึ้น จากนั้นสัญญาณทั้งสองจะถูกเลือกโดยข้อมูลที่ว่าสัญญาณเสียงในเฟรมนั้นเป็นเสียงสั้น (Voiced) หรือเสียงไม่สั้น (Unvoiced) โดยหากเป็นเสียงสั้นจะสัญญาณกระตุ้นค่านบนจะถูกเลือก หากเป็นเสียงไม่สั้นสัญญาณสุ่มจะถูกเลือก จากนั้นจะถูกคูณตัวคูณ  $G$  และเข้าไปทำการคอนโวลูชันในวงจรกรองสัญญาณที่มีค่าสัมประสิทธิ์  $a_k$  ในท้ายที่สุดเราจะได้สัญญาณเสียงที่ถูกสังเคราะห์กลับคืนมา

จากกระบวนการถอดรหัสของวิธีการพยากรณ์แบบเชิงเส้นที่รูปที่ 3 นั้น สามารถสร้างกระบวนการถอดรหัสสัญญาณเสียงด้วยมาตรฐาน Codec2 ได้ดังในรูปที่ 4 โดยสามารถแบ่งกระบวนการถอดรหัสได้เป็น 7 ส่วนดังนี้

1. การแปลงคู่ของเส้นสเปกตรัมไปเป็นสัมประสิทธิ์พยากรณ์เชิงเส้น (LSP to LPC)
2. การแปลงฟูเรียร์แบบเร็ว
3. การสร้างแอมพลิจูดฮาร์โมนิก
4. การสังเคราะห์เฟสของสัญญาณ
5. การแปลงฟูเรียร์กลับแบบเร็ว
6. วงจรกรองสัญญาณ
7. การบวกแบบซ้อนทับของสัญญาณในแต่ละเฟรม



รูปที่ 2.4 : แผนภาพกระบวนการถอดรหัสสัญญาณเสียงด้วย Codec2

คู่ของเส้นสเปกตรัม (Line Spectral Pairs : LSP) นั้นเป็นวิธีการแสดงสัมประสิทธิ์พหุคูณเชิงเส้น (LPC) แบบหนึ่งที่มีประโยชน์มาก เนื่องจากสามารถทนต่อสัญญาณรบกวนที่เกิดขึ้นจากการบิดเบือนได้ดีกว่าการใช้สัมประสิทธิ์พหุคูณเชิงเส้นโดยตรง โดยทั่วไปแล้วจะทำการแบ่ง  $A(z)$  ของวงจรกรองสัญญาณแบบ All-pole ออกเป็นคู่ ๆ ตามสมการที่ 2.4

$$A[z] = 1 - \sum_{k=1}^p a_k z^{-k}$$

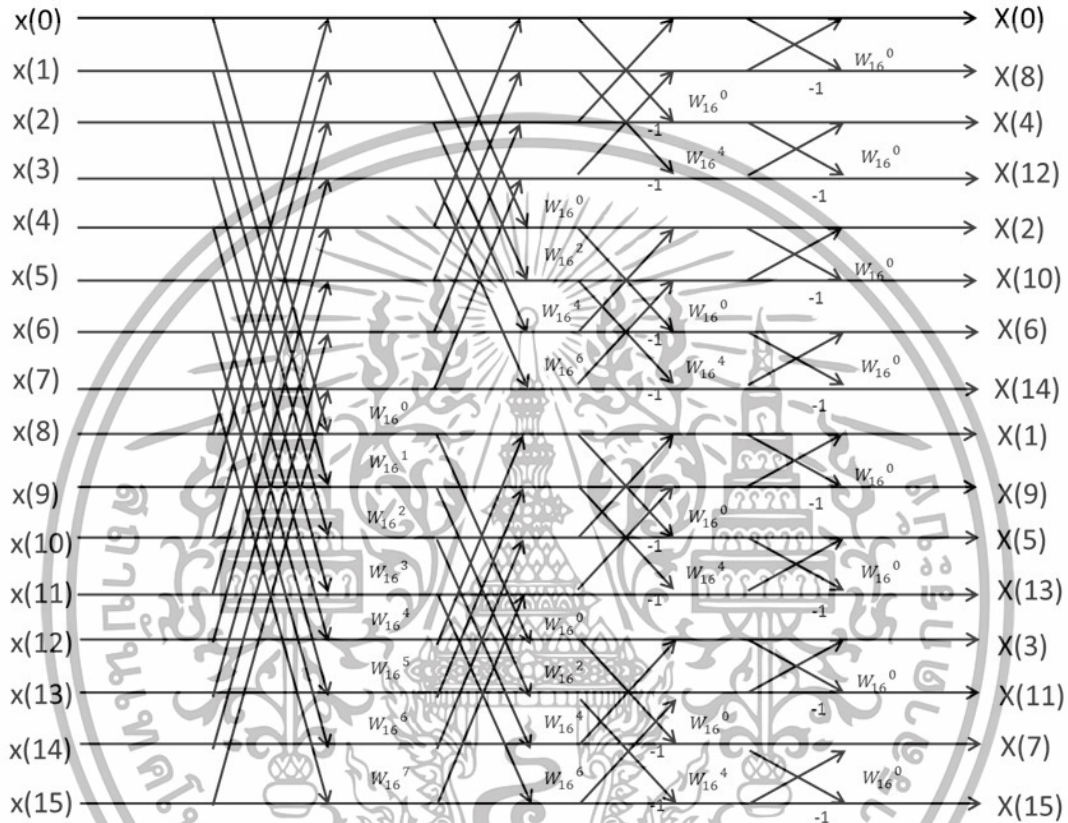
$$P[z] = A[z] + z^{-(p+1)} A[z^{-1}] \quad (2.4)$$

$$Q[z] = A[z] - z^{-(p+1)} A[z^{-1}]$$

โดยที่สามารถแยก  $A[z]$  ออกไปเป็นคู่ ๆ คือ  $P[z]$  และ  $Q[z]$  ก่อนทำการส่งสัญญาณ ดังนั้นในการถอดรหัสของสัญญาณ ขั้นตอนแรกที่ต้องทำคือต้องทำการแปลงค่าคู่ของเส้นสเปกตรัม  $P[z]$  และ  $Q[z]$  กลับมาให้อยู่ในรูปของ  $A[z]$  ก่อนตามสมการที่ 2.5

$$A[z] = 0.5[P(z) + Q(z)] \quad (2.5)$$

จากนั้นสัญญาณ  $A[z]$  ที่ถูกแปลงกลับมาจะต้องถูกแปลงฟูรีเยร์ด้วยขนาด 512 จุด วิธีการที่ใช้คือการแปลงฟูรีเยร์แบบเร็วจำนวน 9 ชั้นตามรูปที่ 2.5

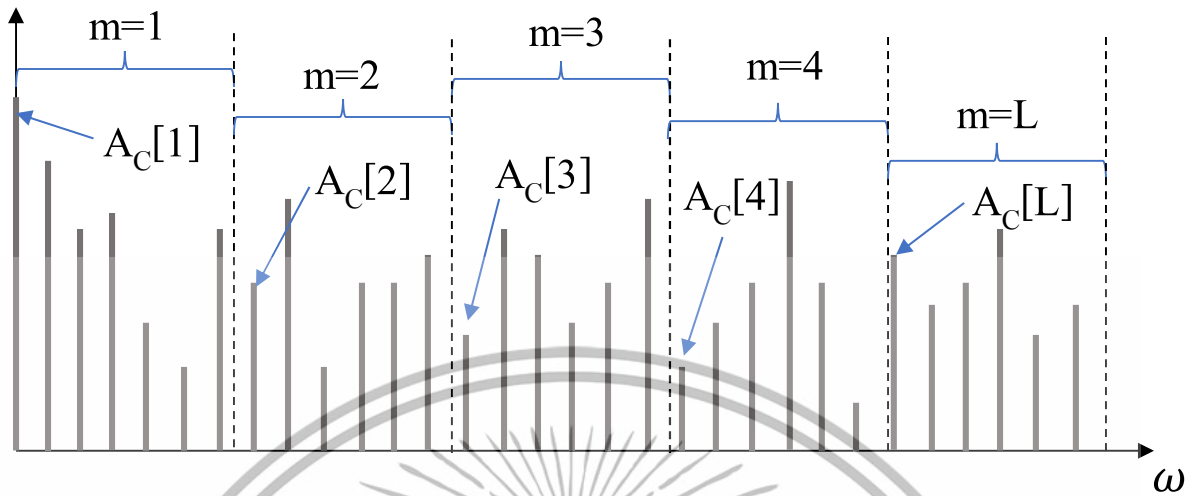


รูปที่ 2.5 : แผนผังการแปลงฟูรีเยร์แบบเร็ว (FFT) ของสัญญาณจำนวน 16 จุด

ขั้นตอนต่อไปเป็นการสร้างแอมพลิจูดฮาร์โมนิกด้วยการหาพลังงานของสัญญาณ LPC ที่ได้จากการแปลงฟูรีเยร์ในขั้นตอนที่แล้ว โดยมีสมการการหาค่าพลังงานตามสมการที่ 2.6

$$P[i] = \frac{1}{X_{Re}[i]^2 + X_{Im}[i]^2} \quad (2.6)$$

จากนั้นจึงทำการรวมกำลังงานของแต่ละความถี่ในสเปกตรัมให้มีฮาร์โมนิกเท่ากับความถี่ของสัญญาณพีช (Pitch) และกำลังงานที่ขอบ  $A_c[m]$  สามารถหาได้ตามรูปที่ 2.6



รูปที่ 2.6 : การรวมกำลังงานของสัมประสิทธิ์ LPC ให้เท่ากับกำลังของสัญญาณ Pitch

สัมประสิทธิ์ของวงจรถอดสัญญาณแบบพหุคูณเชิงเส้นจะมีค่าเท่ากับรากที่สองของกำลังงานที่รวมได้ในรูปที่ 2.6 ตามสมการ

$$A_w[m] = \sqrt{P_w[m]} \quad (2.7)$$

กระบวนการสังเคราะห์สัญญาณกระตุ้น (Excitation) จะเริ่มต้นจากการตรวจสอบว่าสัญญาณเสียงในเฟรมปัจจุบันนั้นเป็นเสียงสั้น (Voiced) หรือไม่สั้น (Unvoiced) หากเป็นเสียงสั้นแล้วจะทำการสร้างสัญญาณไซน์ที่ใช้เป็นตัวกระตุ้นจากสมการ

$$\varphi = \varphi + 80\omega_c \quad (2.8)$$

สัญญาณกระตุ้นในกรณีนี้จะเป็นสัญญาณเชิงซ้อนตามสมการ

$$Ex[m] = e^{j\varphi m} = \cos(\varphi m) + j\sin(\varphi m) \quad (2.9)$$

แต่ในกรณีที่เสียงไม่สั้น (Unvoiced) แล้วเฟสของสัญญาณกระตุ้นจะถูกสุ่มจากวงจรสุ่มตามอัลกอริทึมดังรูปที่ 2.7

```

int codec2_rand(void) {
    unsigned long aa;
    next = next * 1103515245 + 12345;
    aa = (unsigned) (next/65536);
    aa = aa % 32768;
    return ((unsigned) (next/65536) % 32768);
}

```

รูปที่ 2.7 : โปรแกรมสร้างเฟสของสัญญาณกระตุ้นแบบสุ่มสำหรับเสียง Unvoiced

สัญญาณเอาท์พุทที่ได้จากการสร้างสัญญาณเฟสจะเท่ากับการคูณกันของสังยุคซ้อนของค่า  $A_k[m]$  กับสัญญาณกระตุ้นที่สร้างได้ตามสมการ

$$H[m] = \frac{A_k[m]Ex[m]}{|A_k[m]Ex[m]|} \tag{2.10}$$

วงจรกรองสัญญาณจะทำการตรวจสอบกำลังงานของสัญญาณของสัญญาณ Voiced ว่ามีค่าน้อยกว่าค่าขีดจำกัดหรือไม่ หากมีค่าน้อยกว่าจะเปลี่ยนสัญญาณกระตุ้นให้เป็นค่าเฟสจากสัญญาณสุ่มแทน โดยค่าขีดจำกัด (th) จะถูกปรับเปลี่ยนทุกครั้งที่สัญญาณเป็น Invoiced ตามสมการ

$$e = 10 \log \left( \frac{1}{L} \sum_{m=1}^L (A_w[m])^2 \right)$$

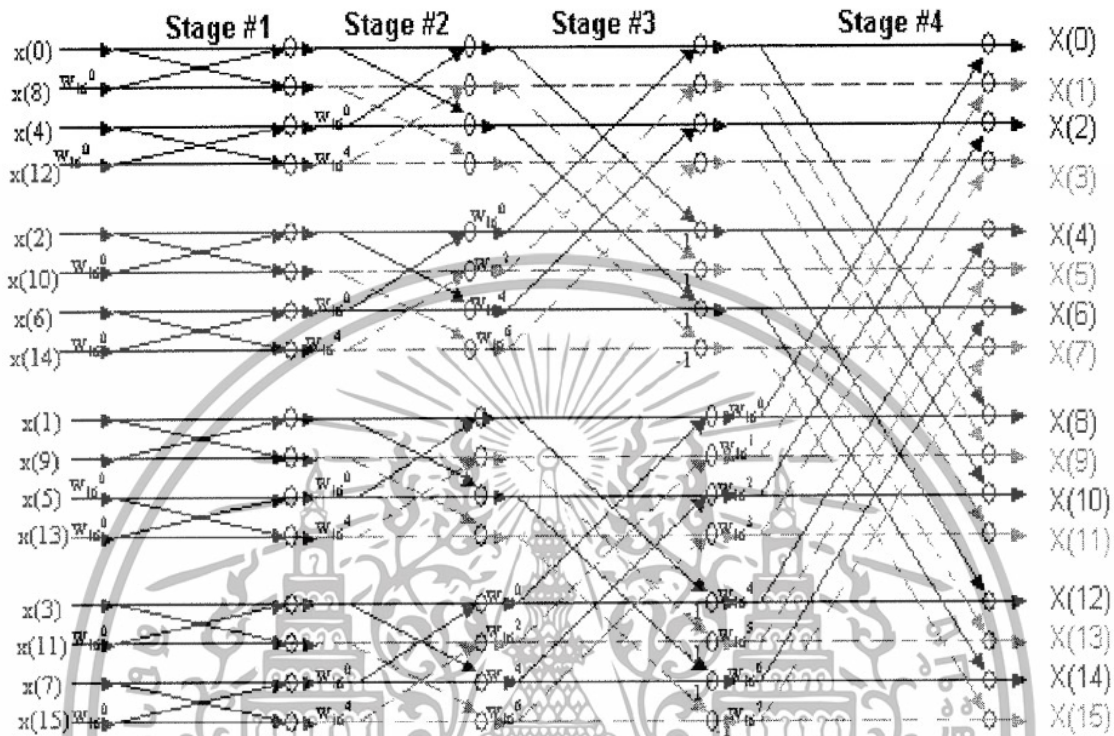
$$bg = bg(1 - \beta) + e\beta \tag{2.11}$$

$$th = 10^{\frac{bg+M}{20}}$$

ในกระบวนการสังเคราะห์สัญญาณเสียงกลับคืนมาในโดเมนเวลานั้นจะเอาสัญญาณสองสัญญาณคือสัญญาณกระตุ้นที่สร้างขึ้นและสัญญาณที่ได้จากการสร้างฮาโมนิกส์มาคูณกัน จากนั้นทำการแปลงฟูเรียร์กลับเพื่อแปลงให้สัญญาณกลับไปสู่โดเมนเวลาตามสมการและรูปที่ 2.8

$$S[m] = A_w[m]Ex[m]$$

$$s[n] = F^{-1}\{S[m]\}$$



รูปที่ 2.8 : แผนผังการแปลงฟูเรียร์ผกผันแบบเร็ว (IFFT) ของสัญญาณจำนวน 16 จุด

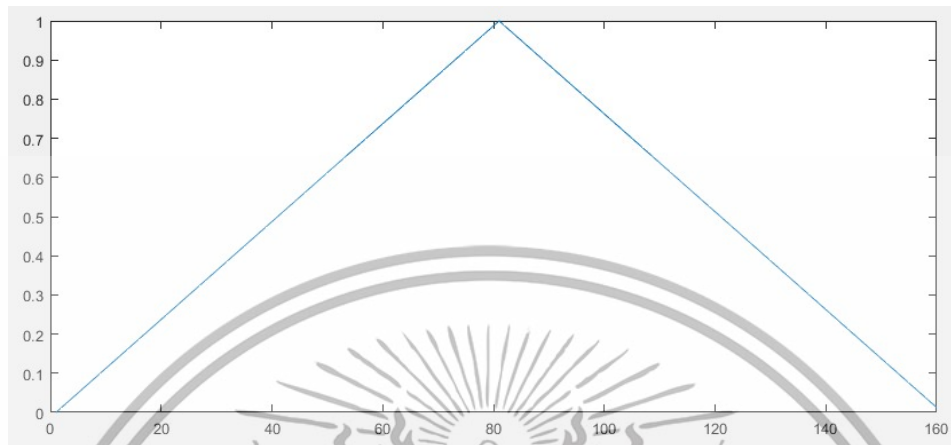
กระบวนการสุดท้ายคือการบวกแบบซ้อนทับของสัญญาณในแต่ละเฟรม โดยสัญญาณ 512 จุดจะถูกแบ่งออกเป็นสองช่วงหัวและท้ายตามรูปที่ 2.9



รูปที่ 2.9 : สัญญาณเอาท์พุท 512 จุดจาก ifft จะถูกแบ่งออกเป็นสองช่วงหัวและท้าย

สัญญาณตำแหน่ง 431-511 จะถูกกรองด้วยวินโดว์แบบสามเหลี่ยมก่อนที่จะถูกบวกเท่ากับสัญญาณต้น 0-79 ของเฟรมที่ผ่านมา ส่วนสัญญาณเฟรมปัจจุบันตำแหน่ง 0-79 จะถูกเก็บไว้รอบวกกับสัญญาณในเฟรม

ถัดไปเพื่อลดความไม่ต่อเนื่องของสัญญาณทั้งสองเฟรม เป็นอันเสร็จสิ้นกระบวนการถอดรหัสสัญญาณเสียง ด้วยวิธี Codec2



รูปที่ 2.10 : วินโดว์สามเหลี่ยมสำหรับกรองสัญญาณก่อนการบวกแบบซ้อนทับ

#### 2.1.4 อินเทอร์เน็ตแห่งสรรพสิ่งและ LoRa (Internet of Things)

Internet of Things หมายถึง การที่อุปกรณ์ต่างๆ สิ่งต่างๆ ได้ถูกเชื่อมโยงทุกสิ่งทุกอย่างสู่โลกอินเทอร์เน็ต ทำให้มนุษย์สามารถสั่งการควบคุมการใช้งานอุปกรณ์ต่างๆ ผ่านทางเครือข่ายอินเทอร์เน็ต เช่น การเปิด-ปิด อุปกรณ์เครื่องใช้ไฟฟ้า (การสั่งการเปิดไฟฟ้าภายในบ้านด้วยการเชื่อมต่ออุปกรณ์ควบคุม เช่น มือถือ ผ่านทางอินเทอร์เน็ต) รถยนต์ โทรศัพท์มือถือ เครื่องมือสื่อสาร เครื่องมือทางการแพทย์ อาคาร บ้านเรือน เครื่องใช้ในชีวิตประจำวันต่างๆ ผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น

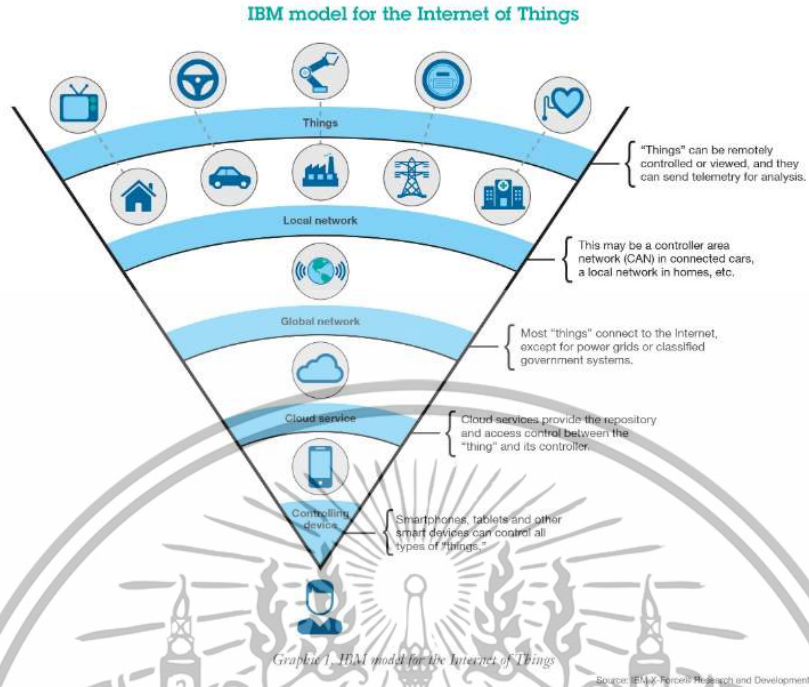
ปัจจุบันมีการแบ่งกลุ่ม Internet of Things ออกตามตลาดการใช้งานเป็น 2 กลุ่มได้แก่

##### 1. Industrial IoT

คือ แบ่งจาก local network ที่มีหลายเทคโนโลยีที่แตกต่างกันในโครงข่าย Sensor nodes โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะเชื่อมต่อแบบ IP network เพื่อเข้าสู่อินเทอร์เน็ต

##### 2. Commercial IoT

คือ แบ่งจาก local communication ที่เป็น Bluetooth หรือ Ethernet (wired or wireless) โดยตัวอุปกรณ์ IoT Device ในกลุ่มนี้จะสื่อสารภายในกลุ่ม Sensor nodes เดียวกันเท่านั้นหรือเป็นแบบ local devices เพียงอย่างเดียวอาจไม่ได้เชื่อมสู่อินเทอร์เน็ต



รูปที่ 2.11 : รูปแบบโครงสร้าง IoT ของบริษัท IBM

### การสื่อสารอินเทอร์เน็ตแห่งสรรพสิ่งด้วย LoRA

Low Power Wide Area Network (LoRA) เป็นเครือข่ายสื่อสารแบบกว้างที่เน้นใช้พลังงานที่ต่ำซึ่งเป็นหัวใจของการสื่อสารด้วยระบบอินเทอร์เน็ตแห่งสรรพสิ่ง จุดเด่น LoRA คือ

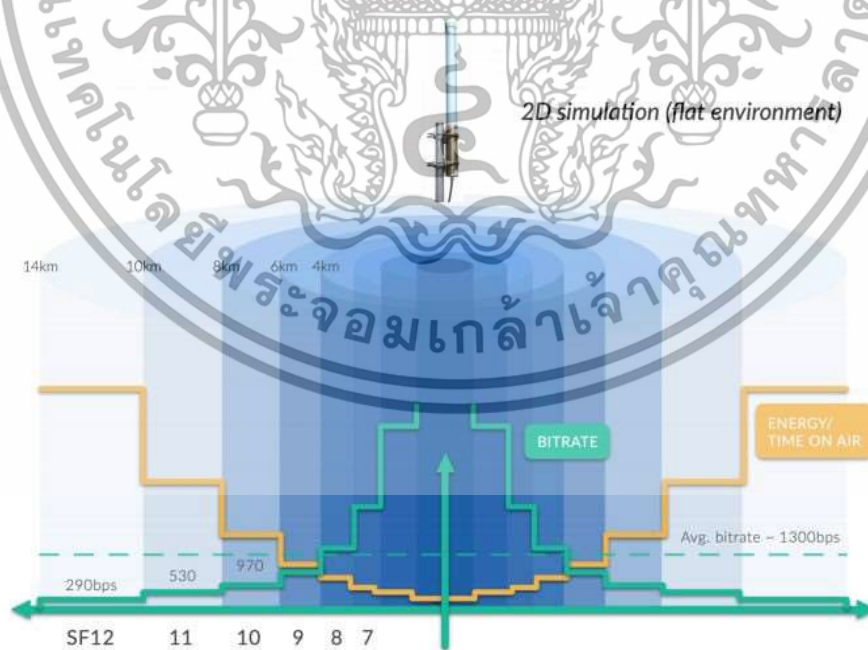
1. เป็นการสื่อสารแบบสองทาง
2. โครข่ายไม่ซับซ้อนไม่จำเป็นต้องมีเร้าเตอร์ หรือจะมีก็ได้
3. บิทเรตต่ำ
4. ประหยัด
5. ใช้พลังงานต่ำมาก

จากตารางที่ 2 จะเห็นว่า LoRA ถูกออกแบบมาให้สามารถส่งสัญญาณได้ไกลและประหยัดพลังงานมากกว่า Zigbee ซึ่งเหมาะสมในการส่งสัญญาณที่บิทเรตต่ำ

ตารางที่ 2.2 : ตารางเปรียบเทียบคุณสมบัติที่สำคัญของ LoRA และ Zigbee

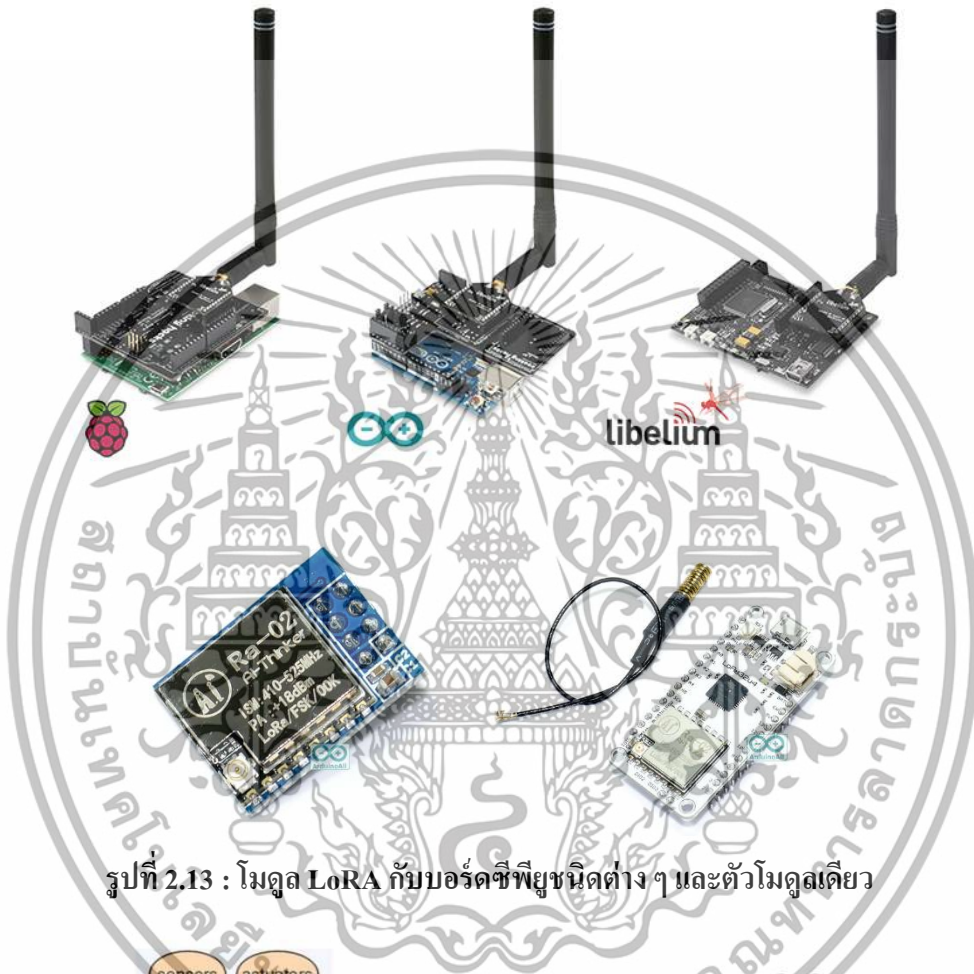
Specifications	LoRa	Zigbee
Frequency Bands	863 to 870 MHz, 902 to 928 MHz, 779 to 787 MHz	868MHz, 915 MHz, 2450 MHz
Coverage distance	2-5 Km (urban areas), 15 Km ( suburban areas )	10 to 100 meters
Power consumption	lower compare to zigbee	low
Modulation technique	LoRa modulation (CSS modulation) , FSK or GFSK	BPSK, OQPSK modulation. Also uses DSSS technique to convert bits to chips.
Data rate	0.3 to 22 Kbps (LoRa modulation) and 100 Kbps (using GFSK)	20 kbps (868 MHz band ) , 40Kbps (915 MHz band ) , 250 kbps (2450 MHz band)

รูปที่ 2.12 แสดง Spreading Factor (SF) กับบิตเรทที่ส่งได้ที่ระยะทางต่าง ๆ จะเห็นได้ว่าหากเราสามารถลดข้อมูลลงได้มากเท่าไรก็ จะสามารถส่งสัญญาณได้ระยะไกลขึ้นในขณะที่ใช้พลังงานได้ต่ำมาก จากข้อมูลของการสื่อสารเสียงด้วย Codec2 จะเห็นได้ว่ามีบิตเรทเท่ากับ 700 บิตต่อวินาทีซึ่งหากส่งด้วยการสื่อสารแบบ LoRA จะสามารถใช้ Spreading Factor (SF) ที่ระดับ 10 โดยสามารถส่งได้ไกล ทำให้เป็นการสื่อสารที่มีประสิทธิภาพสูงมาก

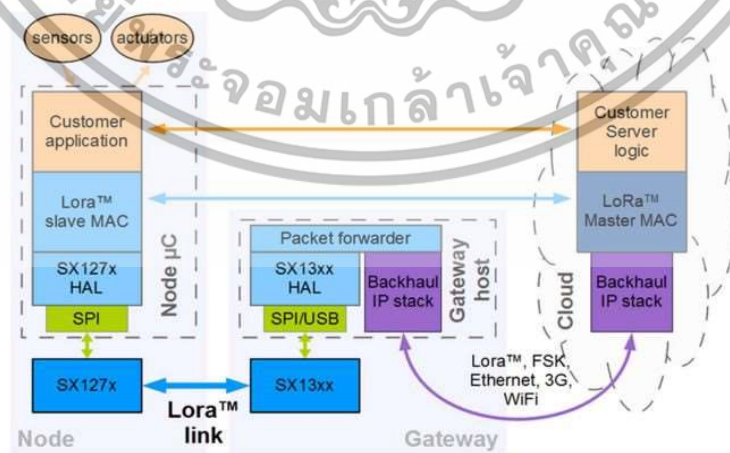


รูปที่ 2.12 : แสดงบิตเรทที่ส่งได้เปรียบเทียบกับระยะทาง

เนื่องจากตัวโมดูลของ LoRA เองมีคุณสมบัติที่กินพลังงานต่ำอยู่แล้ว หากแต่จำเป็นต้องมีไมโครโปรเซสเซอร์มาควบคุม ซึ่งโดยปกติทั่วไปไมโครโปรเซสเซอร์จะใช้พลังงานสูงในการประมวลผลทำให้อายุการใช้งานของแบตเตอรี่สั้น ในปัจจุบันจึงนิยมนำมาใช้ในการส่งสัญญาณจากเซ็นเซอร์ด้วยข้อมูลน้อยๆ เท่านั้น



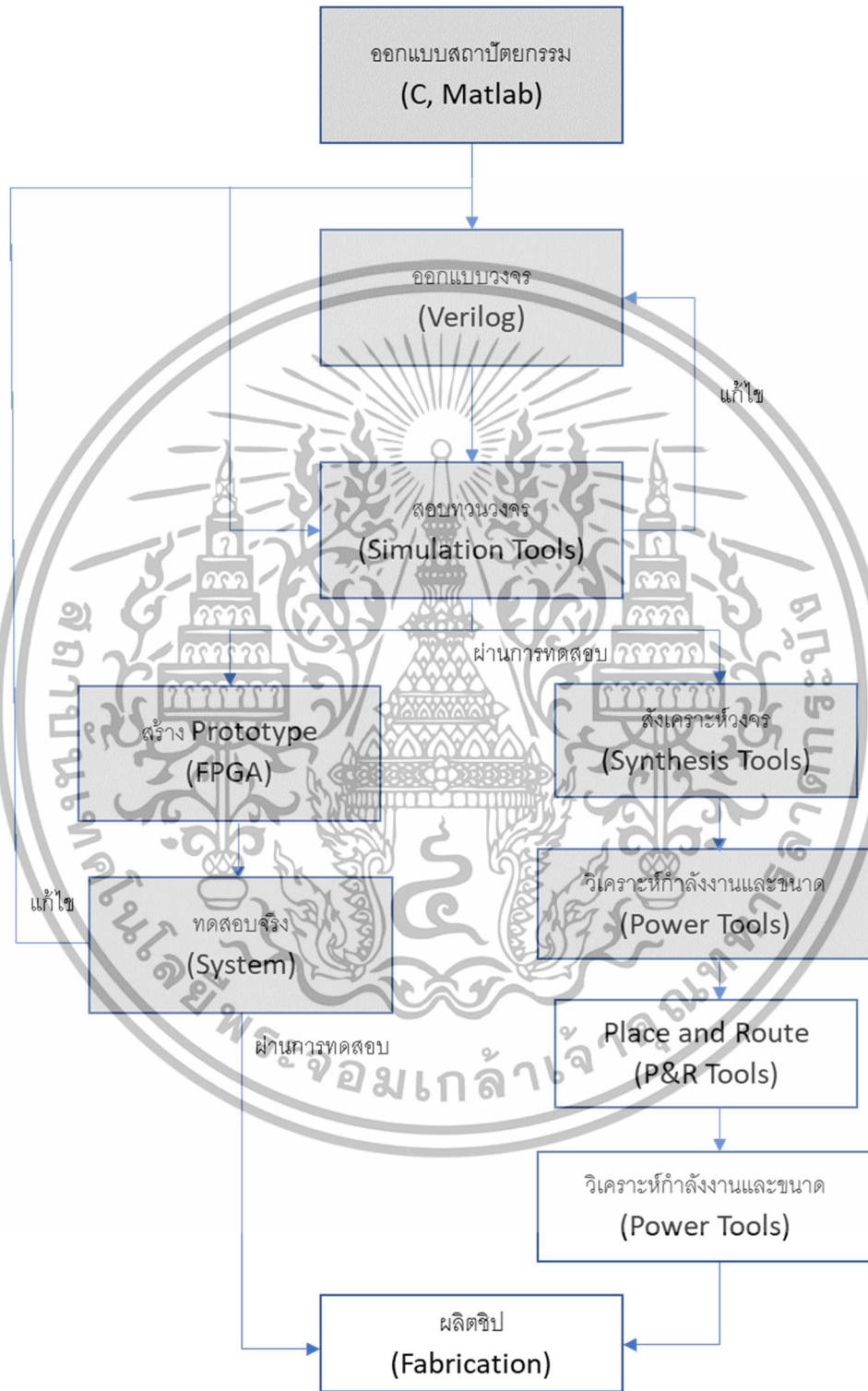
รูปที่ 2.13 : โมดูล LoRA กับบอร์ดชิพยูนิตต่างๆ และตัวโมดูลเดี่ยว



รูปที่ 2.14 : บล็อกไดอะแกรมแสดงการสื่อสารแบบ LoRA ซึ่งสามารถต่อตรงหรือผ่านเราเตอร์ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

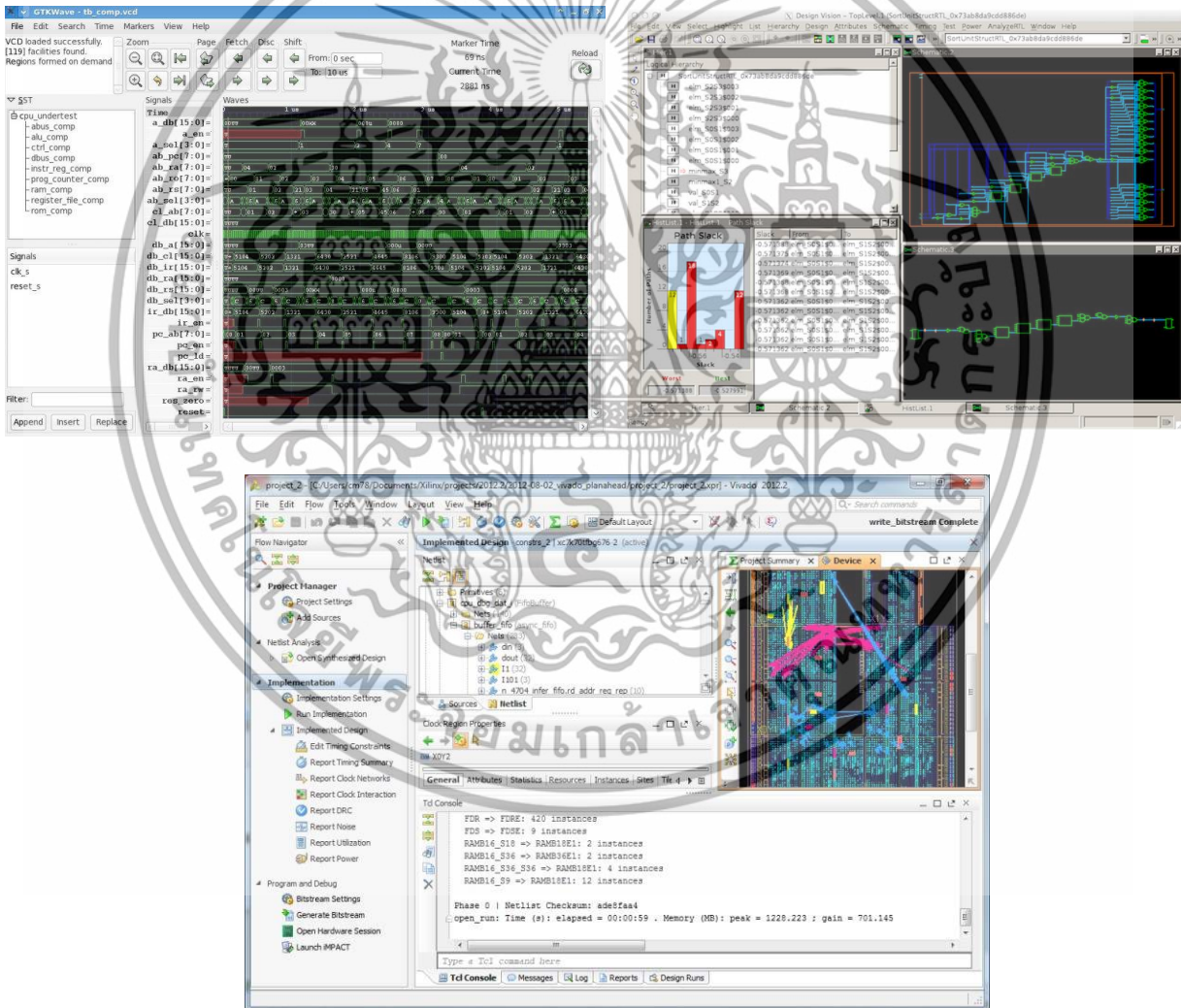
### 2.1.5 กระบวนการออกแบบวงจรดิจิทัล



รูปที่ 2.15 : กระบวนการออกแบบวงจรรวมดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจรรวมนั้นจะต้องเริ่มต้นจากการออกแบบแบบจำลองระบบ โดยทั่วไปมักเป็นภาษาซี หรือ Matlab เพื่อกำหนดโครงสร้างสถาปัตยกรรม ไปป์ไลน์ และวิธีการควบคุมระบบ หลังจากนั้นจะทำการออกแบบวงจรด้วยภาษา Verilog และทำการสอบทวนวงจร (Logic Verification) เปรียบเทียบค่าที่ได้จากวงจร และแบบจำลองที่สร้างไว้ เมื่อวงจรทำงานได้ถูกต้องจะเข้าสู่กระบวนการสังเคราะห์วงจร วิเคราะห์ขนาดและพลังงานที่ใช้ อีกทางหนึ่งจะเริ่มทำการสร้าง Prototype ลงบน FPGA เพื่อทดสอบระบบในสภาพจริง เมื่อผ่านกระบวนการทั้งหมดแล้ว จะเข้าสู่กระบวนการสุดท้ายคือการ Place & Route วงจรก่อนที่จะทำการส่งไปสร้างชิปต่อไป ในงานวิจัยนี้จะครอบคลุมการออกแบบวงจรรวมทั้งหมดยกเว้นขั้นตอนสุดท้ายคือการทำ Place & Route และการสร้างชิปจริงโดยใช้ผลจากการสร้าง Prototype และ Simulation ที่มีความแม่นยำสูงแทน



รูปที่ 2.16 : ตัวอย่างซอฟต์แวร์ที่ใช้ในการสอบทวนวงจร สังเคราะห์วงจร และสร้าง Prototype บน FPGA

## 2.2 งานวิจัยที่เกี่ยวข้อง

กระบวนการบีบอัดสัญญาณเสียงมนุษย์ได้ถูกพัฒนาอย่างยาวนาน โดยเริ่มต้นจากความต้องการทางการทหารของรัฐบาลสหรัฐอเมริกาในสงครามโลกครั้งที่ 2 [1] โดยตัวแบบเริ่มต้นคือมาตรฐาน LPC-10 [2] จากนั้นแนวคิดของมาตรฐาน LPC-10 จึงได้รับการต่อยอดออกมาเป็นหลาย ๆ มาตรฐานในภายหลังเช่น Code-excited Linear Prediction (CELP) [3] ที่มีประสิทธิภาพสูงกว่าด้วยการเพิ่มการพยากรณ์ในระยะยาวและสั้น (Long-term and Short-term Prediction) และการใช้ตารางสร้างสัญญาณกระตุ้น (Excitation Codebook) เป็นครั้งแรก ซึ่งสามารถให้คุณภาพของสัญญาณกระตุ้นได้ดี และทำให้ CELP ถูกเลือกให้เป็นมาตรฐานการบีบอัดสัญญาณเสียงของ MPEG-4 [4] จากนั้น มาตรฐาน Mixed-excitation Linear Prediction (MELP) [5] ก็ได้ถูกพัฒนาต่อยอดโดยสามารถแก้ปัญหานี้สัญญาณรบกวนที่เกิดขึ้นด้วยการใช้สัญญาณกระตุ้นแบบผสมและมีความปลอดภัยต่อการโจรกรรมข้อมูล และทำให้องค์การ Nato และ National Security Agency ของสหรัฐอเมริกายอมรับมาตรฐาน MELP เพื่อใช้ในกิจการทางทหาร

ในปี 2012 ดร. David Rowe วิศวกรชาวออสเตรเลียได้พัฒนากระบวนการบีบอัดสัญญาณเสียงที่สามารถลดขนาดของข้อมูลได้อย่างมาก เรียกว่า Codec2 [6] ข้อดีของมาตรฐานนี้คือไม่มีค่าลิขสิทธิ์ในการใช้งาน เป็นมาตรฐานเปิด และสามารถบีบอัดข้อมูลได้อย่างมีประสิทธิภาพมากกว่ามาตรฐานแบบเดิม ๆ โดยสามารถเลือกระดับการบีบอัดได้ตั้งแต่ 450-3200 บิตต่อวินาที [7] จากความสามารถในการบีบอัดข้อมูลให้มีขนาดเล็กนั้น ทำให้มีความเป็นไปได้ที่จะส่งข้อมูลเสียงผ่านเครือข่ายที่มีแบนวิธต่ำ ๆ เช่น โครงข่าย IoT ตัวอย่างเช่น LoRa เพราะความสามารถของ LoRa เองนั้นสามารถส่งสัญญาณที่บิตเรต 530 บิตต่อวินาทีได้ไกลถึง 10 กิโลเมตรโดยใช้พลังงานที่ต่ำกว่าการส่งข้อมูลด้วยวิธีแบบเดิม ๆ [8]

อย่างไรก็ตามเนื่องจากอัลกอริทึมของการพยากรณ์แบบเชิงเส้นค่อนข้างมีความซับซ้อนมาก เช่นการประมวลผลจำนวนเชิงซ้อนจำนวนมาก ประกอบกับต้องการการคำนวณฟังก์ชันทางคณิตศาสตร์เช่นตรีโกณมิติ เอกโปเนนเชียล ลอการิทึม และวงจรรองสัญญาณดิจิทัลที่มีความซับซ้อนและใช้หน่วยประมวลผลที่มาก ดังนั้นนักวิจัยส่วนใหญ่จึงออกแบบระบบด้วยการใช้ซอฟต์แวร์ในการทำงานบนหน่วยประมวลผลหรือไม่โครโปรเซสเซอร์แบบธรรมดา หรือใช้ Digital Signal Processor [9]

อย่างไรก็ดี มีนักวิจัย P.C.S Abhijna และทีมงานได้สร้างฮาร์ดแวร์โดยเฉพาะเพื่อใช้ในการเข้ารหัสและถอดรหัสสัญญาณเสียงตามมาตรฐาน CELP ด้วยการสังเคราะห์วงจรจากภาษาซีด้วยการใช้การสังเคราะห์วงจรด้วยภาษาขั้นสูง (High Level Synthesis) ในโปรแกรม Vivado ของบริษัท Xilinx [10] นอกจากนี้ S. Shivanagi ก็ได้สร้างวงจรเฉพาะเพื่อทำการเข้ารหัสและถอดรหัสมาตรฐาน MELP เช่นกัน [11] [12]

แต่เนื่องจากวงจรทั้งสองถูกสร้างจากข้อมูลเบื้องต้นซึ่งเป็นภาษาซี ทำให้เป็นการยากที่จะวิเคราะห์ความซับซ้อนและประสิทธิภาพของวงจรที่ถูกสร้างขึ้นด้วยการสังเคราะห์วงจรจากภาษาขั้นสูง อีกทั้งวงจรทั้งสอง

ยังไม่ถูกเปรียบเทียบคุณสมบัติและประสิทธิภาพการทำงานเมื่อเทียบกับการประมวลผลด้วยซอฟต์แวร์แบบ  
ธรรมดา

จากข้อมูลข้างต้น ความท้าทายของการสร้างวงจรประมวลผลสัญญาณเสียงจึงมีหลายส่วนด้วยกัน เช่น  
สามารถออกแบบระบบให้มีประสิทธิภาพสูง ประมวลผลได้รวดเร็ว อีกทั้งมีขนาดเล็กและประหยัดพลังงาน  
ทั้งนี้ทางผู้วิจัยมีความเห็นว่า ระบบมีความจำเป็นจะต้องลดความซับซ้อนของการประมวลผลในหลาย ๆ ส่วน  
เช่นการประมวลผลด้วยจุดลอยตัว (Floating Point) นั้นหากสามารถปรับให้ใช้เป็นการประมวลผลด้วยด้วย  
ตรึงแทน (Fixed Point) ก็จะสามารถลดการคำนวณที่ซับซ้อนลงไปได้ อีกทั้งกระบวนการในการคำนวณ  
ฟังก์ชันทางคณิตศาสตร์ต่างๆ จะต้องมีการออกแบบอย่างรัดกุมเพื่อให้สามารถใช้ทรัพยากรฮาร์ดแวร์ร่วมกัน  
ในการคำนวณแบบต่าง ๆ ได้อย่างมีประสิทธิภาพมากที่สุด ในบทที่ 3 การออกแบบวงจรถอดรหัส  
สัญญาณเสียงแบบ Codec2 จะถูกอธิบายเพื่อให้บรรลุวัตถุประสงค์ข้างต้น



### บทที่ 3

#### วิธีดำเนินงานวิจัยและกระบวนการออกแบบ

##### 3.1 วิธีดำเนินการวิจัย

3.1.1 ออกแบบสถาปัตยกรรมของวงจรถอดรหัสสัญญาณเสียงมาตรฐาน Codec2 โดยการวิเคราะห์จากซอฟต์แวร์ต้นแบบ

3.1.2 ทำการสร้างวงจรด้วยภาษา Verilog และทำการจำลองการทำงาน (Logic Simulation) เปรียบเทียบผลที่ได้จากซอฟต์แวร์ต้นแบบ

3.1.3 ตั้งเคราะห์วงจรในโปรแกรม Vivado เพื่อสร้างวงจรต้นแบบลงใน Xilinx FPGA

3.1.4 สร้างบอร์ดทดลองต้นแบบเพื่อทำการทดลองจริง

3.1.5 เปรียบเทียบประสิทธิภาพและกำลังงานที่ใช้เทียบกับการใช้ไมโครโปรเซสเซอร์แบบปกติ

3.1.6 สรุปผลการทดลองและเขียนรายงานการวิจัยเพื่อนำเสนอในงานประชุมวิชาการนานาชาติและตีพิมพ์ใน Proceeding ในฐานะข้อมูล Scopus

##### 3.2 แผนการดำเนินงานวิจัย

ตารางที่ 3.1 แผนงานวิจัย

กิจกรรม	เดือน															ผู้รับผิดชอบ	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15		
1. เตรียมข้อมูลทำการวิเคราะห์โปรแกรมต้นแบบ	←	→															ดร.สุเมฆ



### 3.1.12 บอร์ด STM32

### 3.1.13 ซอฟต์แวร์สำหรับการออกแบบ จำลองการทำงาน และสังเคราะห์วงจร

## 3.4 กระบวนการออกแบบสถาปัตยกรรมและวงจรถอดรหัสสัญญาณเสียง Codec2

ขั้นตอนการออกแบบวงจรถอดรหัสสัญญาณเสียง Codec2 เริ่มต้นจากการตรวจสอบอัลกอริทึมที่ใช้ด้วยการวิเคราะห์ซอฟต์แวร์ต้นแบบที่เป็นภาษาซี จากการวิเคราะห์พบว่าสามารถแบ่งกระบวนการแปลงสัญญาณได้ออกเป็น 7 ส่วนดังนี้

1. การแปลงคู่ของเส้นสเปกตรัม ไปเป็นสัมประสิทธิ์พยากรณ์เชิงเส้น (LSP to LPC)
2. การแปลงฟูรีเยร์แบบเร็ว
3. การสร้างแอมพลิจูดฮาร์โมนิก
4. การสังเคราะห์เฟสของสัญญาณ
5. การแปลงฟูรีเยร์กลับแบบเร็ว
6. วงจรกรองสัญญาณ
7. การบวกแบบซ้อนทับของสัญญาณในแต่ละเฟรม

โดยในแต่ละขั้นตอนนี้มีการคำนวณที่คล้ายคลึงกัน ซึ่งสามารถแยกออกมาเป็นหน่วยประมวลผลแบบย่อย ๆ ได้ดังนี้

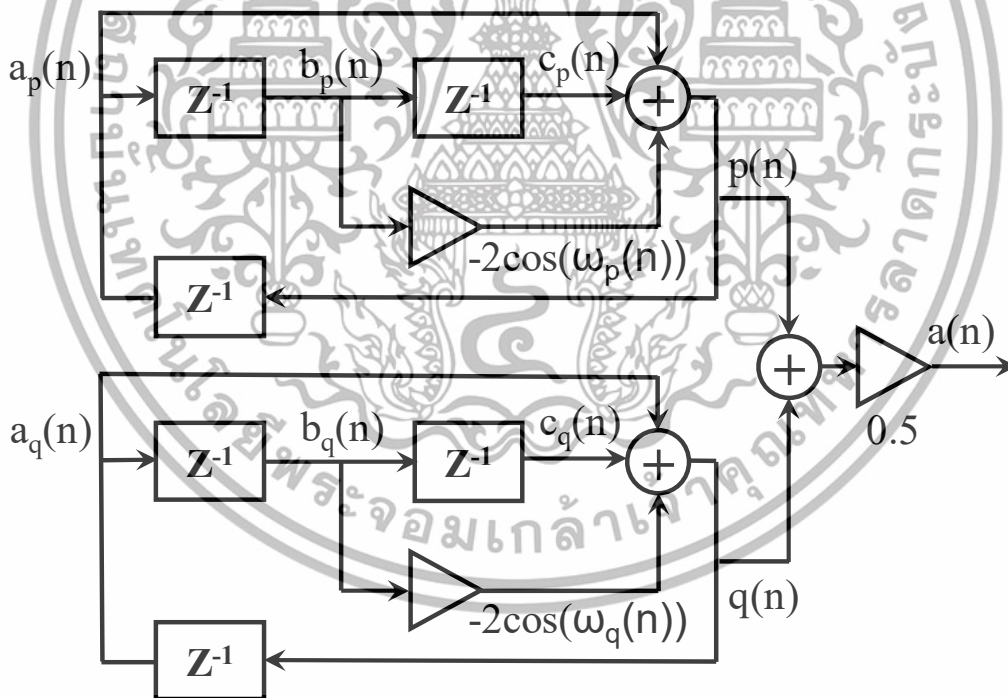
1. วงจรกรองสัญญาณแบบ Finite Impulse Response (FIR)
2. วงจรกรองสัญญาณแบบ Infinite Impulse Response (IIR)
3. วงจรแปลงแปลงฟูรีเยร์และแปลงฟูรีเยร์ผกผัน (FFT)
4. วงจรบวก,ลบ,คูณ,หาร,ยกกำลังและหารากที่สองของสัญญาณ (Arithmetic Unit)
5. วงจรหาพลังงานของสัญญาณ
6. วงจรหาค่าตรีโกณมิติ
7. วงจรหาค่าเอกโปเนนเชียล
8. วงจรหาค่าลอการิทึม

จากการวิเคราะห์หน่วยประมวลผลทั้งหมด ทำให้สามารถสรุปได้ว่าวงจรที่เหมาะสมควรมีลักษณะเป็น วงจรที่คล้ายคลึงกับสถาปัตยกรรมแบบ Very Long Instruction Word (VLIW) เนื่องจาก

1. สามารถปรับเปลี่ยนวงจรให้ทำงานได้หลากหลาย
  2. สามารถทำงานแบบขนานและต่อเป็นไปป์ไลน์ได้โดยที่วงจรยังมีขนาดเล็ก
  3. สามารถออกแบบฟังก์ชันการทำงานของแต่ละส่วนได้โดยการเพิ่ม State Machine ทำให้สามารถ ใช้วงจรร่วมกันได้อย่างดี
  4. ง่ายต่อการทวนสอบวงจรเนื่องจากแต่ละฟังก์ชันแยกกันอย่างชัดเจน
- ในหัวข้อถัดไปจะเป็นการอธิบายการออกแบบแต่ละฟังก์ชันโดยละเอียด

### 3.4.1 วงจรแปลงคู่ของเส้นสเปกตรัมไปเป็นสัมประสิทธิ์พยากรณ์เชิงเส้น (LSP to LPC)

จากบทที่สองในการแปลงคู่ของเส้นสเปกตรัมไปเป็นสัมประสิทธิ์พยากรณ์เชิงเส้นนั้น มีลักษณะคล้ายกับการทำงานของวงจรกรองสัญญาณแบบ Infinite Impulse Response (IIR) โดยมีโครงสร้างดังรูปที่ 3.1



รูปที่ 3.1 : วงจรแปลงคู่ของเส้นสเปกตรัมไปเป็นสัมประสิทธิ์พยากรณ์เชิงเส้น

จากรูปจะเห็นได้ที่เราสามารถเพิ่มความเร็วในการคำนวณได้ด้วยการคำนวณส่วน p และ q ไปพร้อม ๆ กัน อีกทั้งในส่วน p และ q ยังสามารถออกแบบให้เป็นไปป์ไลน์เพื่อเพิ่มความเร็วได้มากขึ้นอีก จากการวิเคราะห์เวลาของข้อมูล (Data Life Time) นั้นสามารถเขียนออกมาอยู่ในรูปที่ 3.2

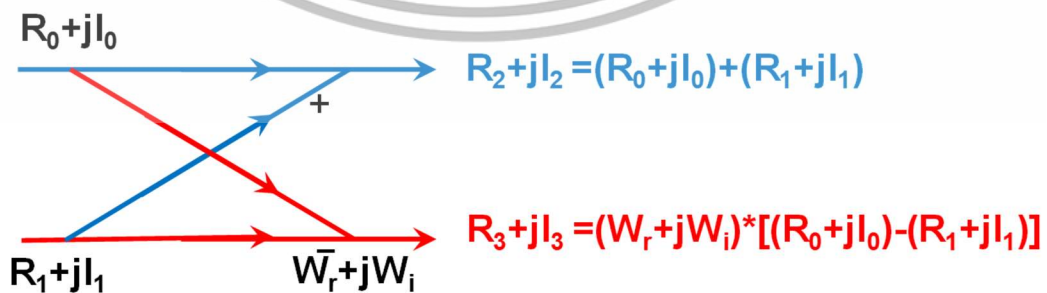
	0	1	2	3	4	5	6	7	8
Mem RD	RD C	RD n1	RD n2	RD C	RD n1	RD n2	RD C	RD n1	RD n2
Reg WR		R2	R0	R1	R2	R0	R1	R2	R0
Multiplier				$R0=2*R0$	$R0=R0*R2$		$R0=2*R0$	$R0=R0*R2$	
Adder					$R3=R3+R1$	$R3=R3-R0$		$R3=R3+R1$	$R3=R3-R0$
Mem WR				$WR n2=R0$	$WR n1=R3$		$WR n2=R0$	$WR n1=R3$	
Reg WR				$R4=0, R3=1$					

รูปที่ 3.2 : การวิเคราะห์เวลาของข้อมูลในการแปลง LSP ไป LPC

จากรูปที่ 3.2 จะเห็นว่าในช่วงเวลาที่ 0 จะทำการอ่านข้อมูลสัญญาณโคไซน์และถูกเก็บไว้ในรีจิสเตอร์ 2 (R2) ในช่วงเวลาที่ 1 จากนั้นจะทำการอ่านค่า n1 และ n2 ไปเก็บใน R0 และ R1 ตามลำดับในช่วงเวลาที่ 2 และ 3 วงจรคูณจะทำการคูณในช่วงเวลาที่ 3 และ 4 จากนั้น วงจรบวกจะทำการบวกในช่วงเวลาที่ 4 และ 5 ในเวลาเดียวกันวงจรจะทำการเขียนค่าสัญญาณที่ประมวลได้ไปยังหน่วยความจำในช่วงเวลาที่ 3 และ 4 จากการคำนวณด้านบนจะเห็นว่าใช้การคำนวณทั้งหมด 6 สัญญาณนาฬิกา โดยสามารถทำเป็นไปป์ไลน์ได้ จึงลดเวลาคำนวณได้เหลือเพียง 3 สัญญาณนาฬิกาด้วยการใช้วงจรบวกและวงจรถคูณอย่างละหนึ่งชุดเท่านั้น ใช้หน่วยความจำแบบ 2 พอร์ตจำนวนหนึ่งชุดและรีจิสเตอร์เพียง 5 ตัวก็สามารถแปลงคู่ของเส้นสเปกตรัมได้อย่างมีประสิทธิภาพ โดยมีอัตราการใช้งานวงจรถคูณและวงจรถบวก 67% การอ่านหน่วยความจำ 100% และการเขียนหน่วยความจำ 67%

### 3.4.2 วงจรแปลงฟูรีเยร์แบบเร็ว (FFT)

ในการแปลงฟูรีเยร์แบบเร็วนี้ สามารถออกแบบวงจรเพิ่มจากวงจรในหัวข้อ 3.4.1 ได้โดยตรง โดยปกติแล้วนั้นการแปลงฟูรีเยร์จะใช้การคำนวณแบบผีเสื้อ (Butterfly) โดยที่มีการคำนวณตามรูปที่ 3.3



รูปที่ 3.3 : การคำนวณแบบ Butterfly สำหรับการแปลงฟูรีเยร์แบบเร็ว (FFT)

จากรูปที่ 3.3 จะเห็นได้ว่ามีค่าคำนวณโดยใช้การคูณทั้งหมด 4 ครั้ง การบวก 6 ครั้ง การอ่านข้อมูลจากหน่วยความจำ 4 ครั้งและเขียนข้อมูลกลับหน่วยความจำ 4 ครั้งเช่นกัน และเนื่องจากมีวงจรบวกเพียงหนึ่งชุด ดังนั้นความเร็วของการคำนวณที่สูงที่สุดคือ 6 สัญญาณนาฬิกา โดยมีอัตราการใช้วงจรบวกเป็น 100% จากการวิเคราะห์ด้านบนทำให้สามารถจัดการการคำนวณ FFT และ iFFT ได้ดังรูปที่ 20

<b>Adder</b>					
$R_a=R_0-R_1$	$R_2=R_0+R_1$	$I_a=I_0-I_1$	$I_2=I_0+I_1$	$R_3=R_0-R_1$	$I_3=I_0+I_1$
<b>Multiplie</b>	$R_b=W_r*R_a$	$I_b=W_i*R_a$	$R_c=W_i*I_a$	$I_c=W_r*I_a$	
$t_0$	$t_1$	$t_2$	$t_3$	$t_4$	$t_5$

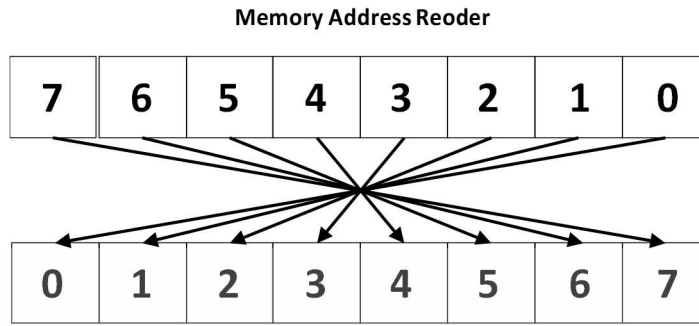
รูปที่ 3.4 : ลำดับการคำนวณ Butterfly หนึ่งครั้งด้วยการใช้วงจรบวกและวงจคูณเพียงหนึ่งชุด

จากรูปที่ 3.4 หากทำการรวมการอ่านและเขียนข้อมูลการหน่วยความจำเข้าไปด้วยจะได้ลำดับการคำนวณดังรูปที่ 21 โดยจะมีการอ่านข้อมูลจำนวน 4 สัญญาณนาฬิกา การบวก 6 สัญญาณนาฬิกาและการคูณจำนวน 4 สัญญาณนาฬิกา การเขียนข้อมูลกลับ 4 สัญญาณนาฬิกา แต่สามารถทำขนานกันและไปป์ไลน์ได้ โดยจะเหลือเวลาการคำนวณเพียง 6 สัญญาณนาฬิกาสำหรับการคำนวณ 1 ครั้งเท่านั้น อีกทั้งยังใช้รีจิสเตอร์เพียง 6 ตัวซึ่งสามารถใช้ร่วมกับวงจร LPC ได้ นับว่าเป็นสถาปัตยกรรมที่มีประสิทธิภาพสูงและมีขนาดเล็ก เนื่องจากมีอัตราการใช้ฮาร์ดแวร์สูงมาก เช่น วงจรบวก 100%

	Butterfly 1					Butterfly 2					
Mem Rd	LD Re0	LD Re1	LD Im0	LD Im1		LD Re0	LD Re1	LD Im0	LD Im1		
Reg Wr	R0=Re0	R1=Re1	R2=Im0	R3=Im1		R0=Re0	R1=Re1	R2=Im0	R3=Im1		
Adder			R4=R0-R1	R0=R0+R1	R0=R2-R3	R2=I0+I1	R3=R1-R3	R4=R5+R2	R4=R0-R1	R0=R0+R1	R0=R2-R3
Multiplier				R1=R4*W <sub>r</sub>	R5=R4*W <sub>i</sub>	R3=R0*W <sub>i</sub>	R2=R0*W <sub>r</sub>			R1=R4*W <sub>r</sub>	R5=R4*W <sub>i</sub>
Mem Wr				ST R0		ST R2	ST R3	ST R4		ST R0	

รูปที่ 3.5 : ขั้นตอนการทำ Butterfly ด้วยการรวมการอ่านและเขียนหน่วยความจำ

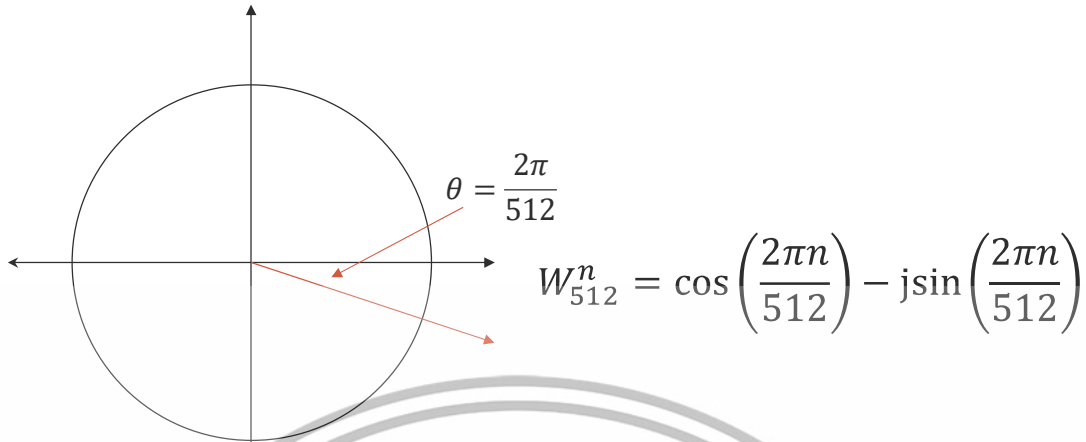
นอกจากนี้ ขั้นตอนการทำ FFT ที่นับว่าใช้เวลามากคือการต้องสลับลำดับของข้อมูลก่อนการคำนวณ Butterfly ซึ่งโดยปกติทั่วไปจะทำการอ่านแล้วสลับไปเขียนที่หน่วยความจำที่ใหม่ ขั้นตอนนี้มีข้อเสียใหญ่สองประการคือ สิ้นเปลืองหน่วยความจำ และเสียเวลาในการอ่านและเขียนข้อมูล ดังนั้นในวงจรนี้จึงทำการออกแบบวงจรสลับการอ่านหน่วยความจำอัตโนมัติที่สามารถควบคุมได้ด้วยการเลือกการใช้งานว่าจะสลับหรือไม่ ทำให้ลดเวลาในการสลับข้อมูลและประหยัดหน่วยความจำไปจำนวนมาก โดยโครงสร้างการอ่านหน่วยความจำเป็นไปดังรูปที่ 3.6 ซึ่งสามารถใช้ได้ทั้งการแปลงฟูเรียร์และการแปลงฟูเรียร์ผกผัน



**รูปที่ 3.6: วงจรสลับข้อมูลเพื่ออ่านค่าในหน่วยความจำสำหรับการคำนวณ FFT โดยไม่ต้องเสียเวลาสลับแบบซอฟต์แวร์**

### 3.4.3 วงจรคำนวณค่าตรีโกณมิติ

ในกระบวนการถอดรหัสสัญญาณเสียงแบบ Codec2 นั้นมีหลายส่วนที่ใช้ฟังก์ชันค่าตรีโกณมิติ ตัวอย่างเช่นการแปลงฟูเรียร์แบบเร็ว และการสังเคราะห์เฟสของสัญญาณเป็นต้น ดังนั้นการออกแบบการคำนวณฟังก์ชันตรีโกณมิติจึงมีความสำคัญอย่างมาก โดยทั่วไปแล้วความละเอียดของฟังก์ชันที่ต้องการจะเป็นตัวกำหนดวิธีการคำนวณที่เหมาะสม ในกรณีนี้มีความจำเป็นจะต้องใช้ทศนิยมเลขฐานสองละเอียดถึง 12 บิตหรือ  $1/1024$  เพื่อให้สัญญาณรบกวนจากการบิดเบือน (Quantization Noise) มีค่าน้อย โดยสามารถหาค่าดังกล่าวได้ 2 แบบหลัก ๆ คือ การใช้ตาราง (Look Up Table) หรือใช้การคำนวณโดยวิธีการประมาณค่าเช่น Cordic หรืออนุกรมเทเลอร์ ทั้งสองวิธีมีข้อดีและเสียต่างกันคือ การใช้ตารางสามารถใช้เวลาคำนวณน้อยที่สุดแต่สิ้นเปลืองเนื้อที่ของหน่วยความจำมากที่สุด ในกรณี Codec 2 นั้นมีความจำเป็นจะต้องใช้หน่วยความจำจำนวนถึง  $6,437 \times 14$  บิตด้วยกัน ส่วนวิธีการประมาณค่าเช่น Cordic จะใช้การคำนวณจนกระทั่งค่าความผิดพลาดมีค่าน้อยที่สุด ซึ่งโดยทั่วไปแล้วใช้เวลาประมาณ 5-10 สัญญาณนาฬิกา ทำให้การคำนวณค่าตรีโกณกลายเป็นคอขวดของวงจร โดยเฉพาะการแปลงฟูเรียร์แบบเร็วที่ใช้การคำนวณตรีโกณที่มาก โดยจากการวิเคราะห์ฟังก์ชันตรีโกณของการแปลงฟูเรียร์นั้น พบว่าค่า Twiddle Factor ของการแปลงแบบ 512 จุดจะใช้ค่า Sine และ Cosine จำนวน 512 จุดด้วยกันดังแสดงในรูปที่ 3.7

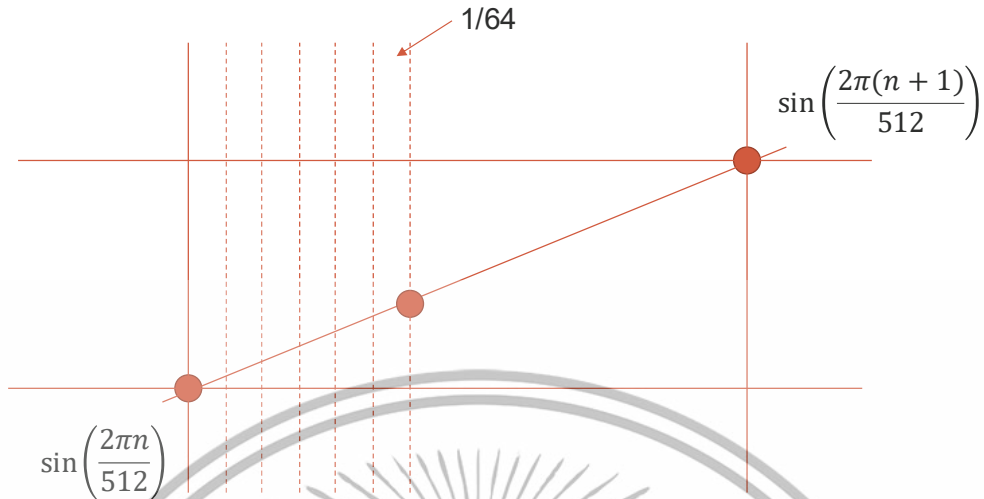


รูปที่ 3.7 : ค่า Twiddle Factor สำหรับจำนวนการแปลงฟูเรียร์แบบเร็ว

เนื่องจากค่าสัญญาณตรีโกณทั้ง 4 จุดภาคมีค่าซ้ำกัน ดังนั้นเราสามารถเก็บค่าเพียง 1 จุดภาคหรือ 128 ค่าแล้วใช้วงจรในการแปลงให้ครบทั้ง 4 จุดภาคแทน อีกทั้งสัญญาณ Sine และ Cosine มีค่าเหมือนกัน เพียงแต่มีค่าเลื่อนเฟสไป ดังนั้นเราสามารถลดขนาดของตารางจาก 6,437 ได้เหลือเพียง 128 ค่าเท่านั้น และสามารถทำให้การแปลงฟูเรียร์สามารถคำนวณ Butterfly โดยการใช้น้ำหนักเพียง 6 สัญญาณนาฬิกา

อย่างไรก็ดี ค่าความละเอียดเพียง 512 ค่าไม่เพียงพอต่อการคำนวณและถอดรหัสในกระบวนการอื่น ๆ ซึ่งต้องการความละเอียดถึง 6437 ค่าดังที่ได้กล่าวมาแล้ว ดังนั้นจึงต้องมีวงจรที่ใช้ในการประมาณค่า (Interpolation) เพื่อเพิ่มความละเอียดของสัญญาณตรีโกณ โดยในกรณีนี้จะใช้การประมาณแบบเชิงเส้น เพราะเป็นวิธีที่ง่ายและซับซ้อนน้อยที่สุด แต่มีข้อเสียคือค่าความผิดพลาดจะมีค่ามากถ้ามีการประมาณที่ละเอียดเกินไป

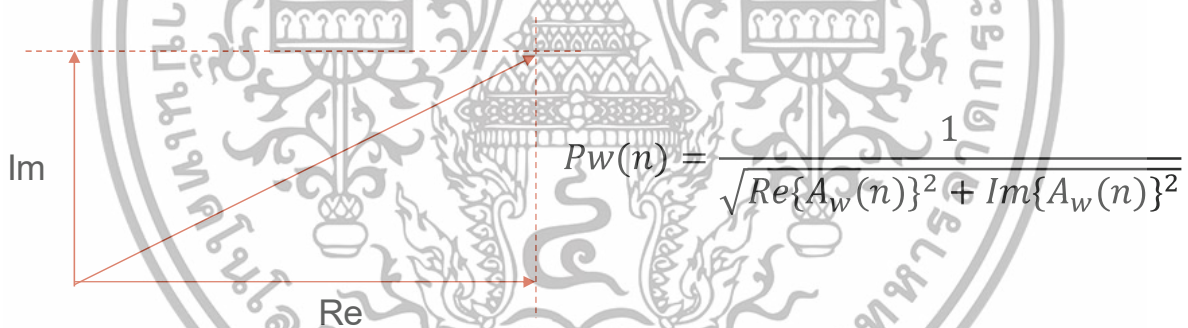
วงจรประมาณค่าแบบเชิงเส้นได้ถูกเพิ่มขึ้นมาสำหรับการคำนวณค่าตรีโกณมิติที่ต้องการความละเอียดมากขึ้น โดยเพิ่มความละเอียดมากขึ้นอีก 64 ส่วน และใช้การประมาณตามรูปที่ 24 แต่เนื่องจากเวลาที่ใช้ในการเปิดตารางจะใช้เวลา 1 สัญญาณนาฬิกา และเวลาที่ใช้การประมาณค่าใช้อีก 1 สัญญาณนาฬิกา ดังนั้นจะมีเวลาในการคำนวณทั้งหมด 2 สัญญาณนาฬิกาด้วยกัน เมื่อการออกแบบทำให้ขั้นตอนการเปิดตารางและการคำนวณสามารถทำเป็นไปป์ไลน์ได้ ดังนั้นในการคำนวณค่าสัญญาณตรีโกณหลาย ๆ ค่าติดกันจะใช้การคำนวณเพียง 1 สัญญาณนาฬิกาเท่านั้น โดยที่ขั้นตอนที่ 1 และ 2 สามารถเหลื่อมกันได้ ในบทที่ 4 จะทำการคำนวณค่าความผิดพลาดที่เกิดขึ้นจากการคำนวณด้วยการประมาณแบบเชิงเส้นว่ามีค่าน้อยหรือมากเท่าใด



รูปที่ 3.8 : การประมาณค่าแบบเชิงเส้นที่ใช้ในการคำนวณค่าตรีโกณมิติ

### 3.4.4 การสร้างแอมพลิจูดฮาร์โมนิก

หลังจากการแปลงสัญญาณด้วยการแปลงฟูรีเยร์แบบเร็วแล้ว ขั้นตอนต่อมาคือการสร้างแอมพลิจูดฮาร์โมนิก โดยเริ่มจากการคำนวณหาพลังงานของสัญญาณดังรูปที่ 3.9



รูปที่ 3.9 : การคำนวณหาพลังงานของสัญญาณจากผลการแปลงฟูรีเยร์แบบเร็ว

จะเห็นว่า การคำนวณประกอบด้วยการหาร ยกกำลัง และการหารากที่สอง ซึ่งการคำนวณพื้นฐานที่จำเป็นจะต้องใช้คือ วงจรหาร วงจรคำนวณค่าเอ็กโพเนนเชียลและลอการิทึม ซึ่งมีความจำเป็นจะต้องหาอัลกอริทึมที่สามารถใช้วงจรร่วมกับขั้นตอนอื่น ๆ ให้ได้มากที่สุดอีกทั้งยังใช้เวลาในการคำนวณน้อย ดังนั้นในหัวข้อถัดไปจะแสดงวิธีการคำนวณค่าดังกล่าว

### 3.4.5 วงจรหารแบบ Goldschmidt

วงจรหารแบบ Goldschmidt เป็นวิธีการคำนวณการหารโดยใช้การคูณและวงจรถ่ายเทียบเท่านั้น โดยมีอัลกอริทึมที่สามารถอธิบายได้ในสมการและรูปที่ 3.10

$$z = \frac{x}{y}$$

$$z = \frac{x \cdot t_1}{y \cdot t_1} = \frac{x \cdot t_1}{1} = x \cdot t_1$$

$$t_0[n] = \frac{2^n}{2^n + 1} \quad t_1[n] = \frac{2^n + 1}{2^n} \quad (3.1)$$

$$t_0[n] = \left\{ \frac{1}{2}, \frac{2}{3}, \frac{4}{5}, \frac{8}{9}, \frac{16}{17}, \frac{32}{33}, \frac{64}{65}, \dots \right\}$$

$$t_1[n] = \left\{ \frac{2}{1}, \frac{3}{2}, \frac{5}{4}, \frac{9}{8}, \frac{17}{16}, \frac{33}{32}, \frac{65}{64}, \dots \right\}$$

ตัวอย่างเช่นการหาค่า  $x/y$  หลักการโดยทั่วไปคือการทำให้ค่า  $y$  มีค่าอยู่ในช่วง 0.5 ถึง 1 ก่อนโดยการใช้การเลื่อนบิตทั้งเศษและส่วน จากนั้นจะใช้  $t_1$  คูณทั้งเศษและส่วนจนกระทั่งส่วนมีค่าเท่ากับ 1 ก็จะได้ค่าการหารคือค่า  $x$  คูณ  $t_1$  นั้นเอง

ในการนี้จำเป็นจะต้องมีค่า  $t_0$  กับ  $t_1$  ในตารางตามสมการข้างบน และอัลกอริทึมการหารทั้งหมดสามารถเขียนอยู่ในรูปโฟลชาร์ทได้ดังรูปที่ 27

### 3.4.6 วงจรคำนวณเอ็ทไปเนนเซียลแบบ Goldschmidt

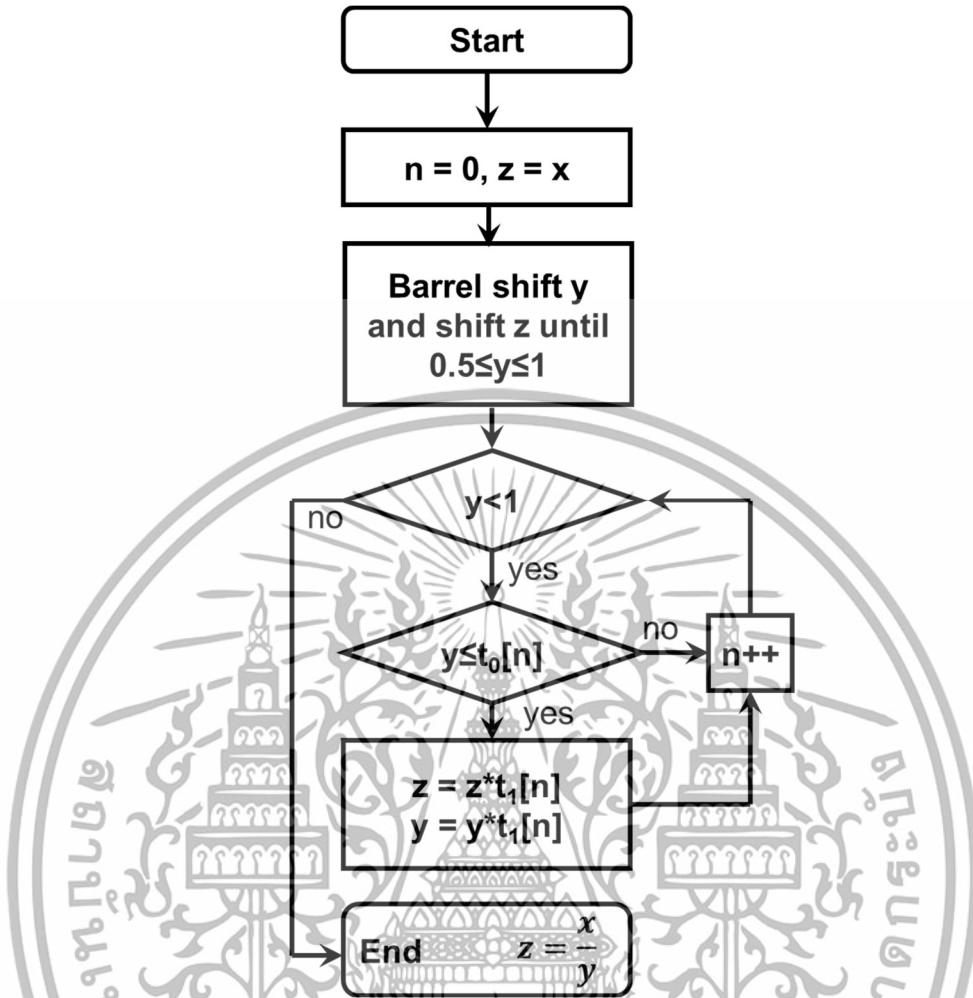
ข้อดีของอัลกอริทึมแบบ Goldschmidt นั้นคือสามารถประยุกต์ใช้กับการคำนวณแบบอื่น ๆ ได้อีก เช่นการคำนวณค่าเอ็ทไปเนนเซียลนั้นสามารถทำได้ดังสมการด้านล่าง

$$z = 2^x; \quad x < 1$$

$$z = 2^{(x-t_2)} \cdot 2^{t_2}$$

$$t_2[n] = \log_2 \left( \frac{2^n + 1}{2^n} \right) \quad (3.2)$$

$$2^{t_2} = \frac{2^n + 1}{2^n} = t_1[n]$$

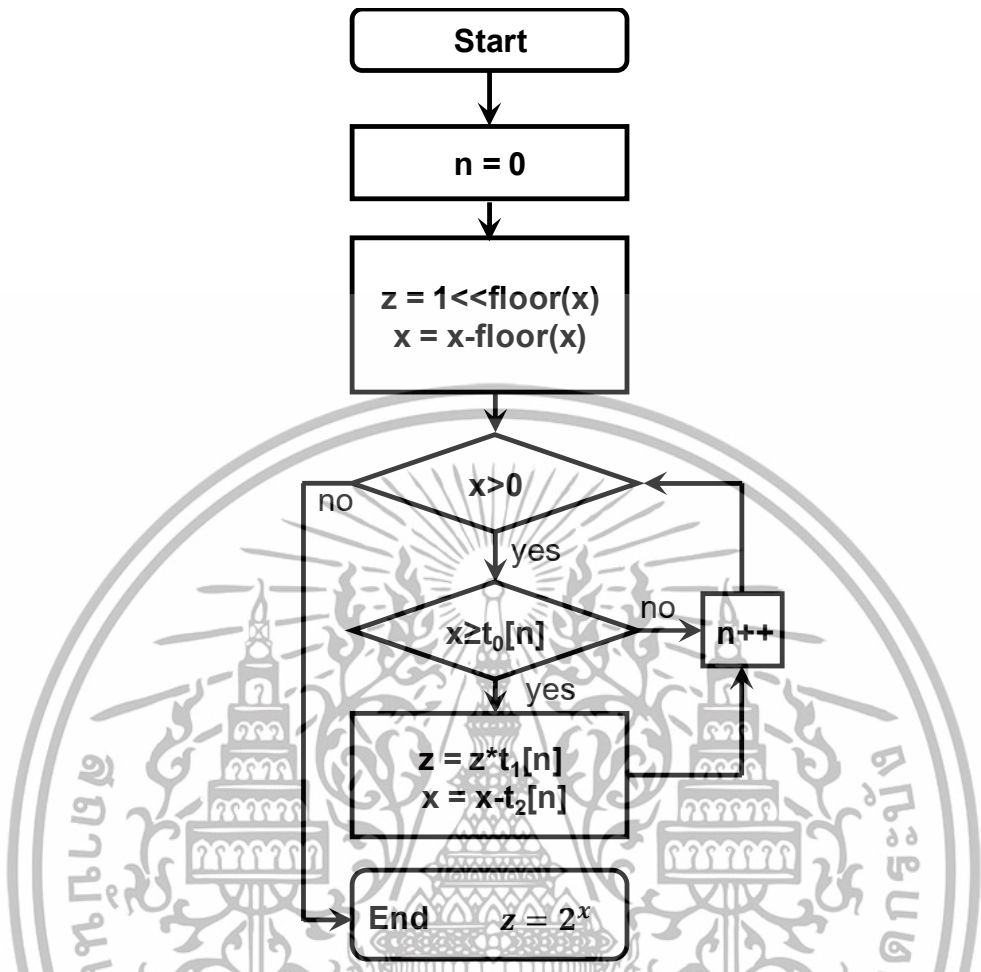


รูปที่ 3.10 : อัลกอริทึมการคำนวณการหารแบบ Goldschmidt

หลักการการคำนวณค่าเอ็กโปเนนเชียลคือการทำให้เลขชี้กำลัง  $x$  มีค่าเป็น 0 โดยการใส่ค่า  $t_2$  ไปลบเพื่อให้ใกล้เคียง 0 มากที่สุด และใช้  $2^{12}$  เข้าไป ดังนั้นวิธีการนี้จะใช้การคูณ, การลบและการเปิดตารางเท่านั้น ทำให้สามารถใช้งานร่วมกับกระบวนการอื่น ๆ ได้มาก เพียงแค่เพิ่ม State Machine ในการคำนวณแบบนี้เท่านั้น

### 3.4.7 วงจรคำนวณลอการิทึม Goldschmidt

การคำนวณแบบลอการิทึมก็ใช้หลักการเดียวกันกับการค่าเอ็กโปเนนเชียลคือพยายามทำให้ค่า  $x$  มีค่าเป็น 1 โดยการคูณ  $t_1$  เข้าไปและลบด้วย  $\log_2(t_1)$  เข้าไป ดังนั้นจึงใช้วงจรคูณ, วงจรลบและการเปิดตารางเท่านั้น มีความใกล้เคียงกับการคำนวณเอ็กโปเนนเชียลและหารมาก



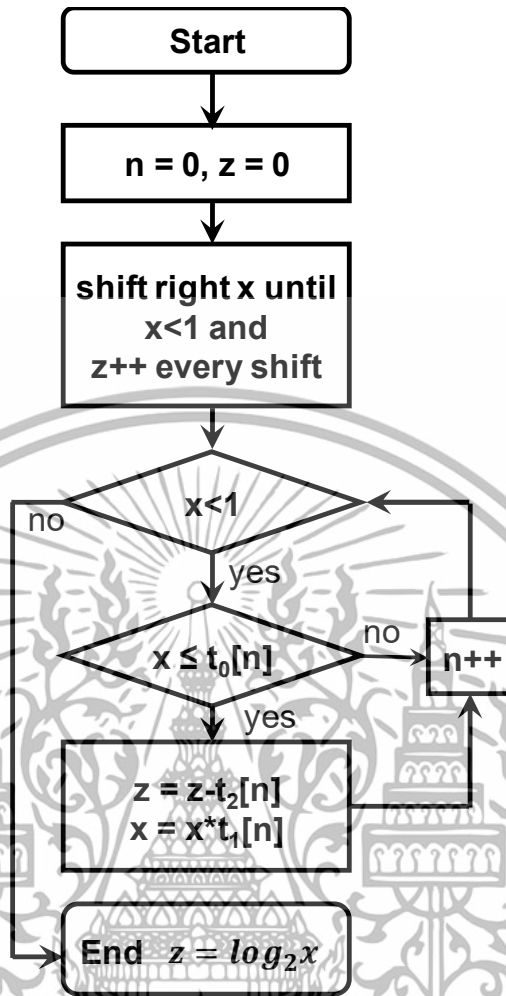
รูปที่ 3.11 : อัลกอริทึมการคำนวณฟังก์ชันเอ็กซ์โปเนนเชียลแบบ Goldschmidt

$$z = \log_2(x); \quad x < 1$$

$$z = \log_2(x \cdot t_1) - \log_2(t_1)$$

$$t_1[n] = \frac{2^n + 1}{2^n} \tag{3.3}$$

$$t_2[n] = \log_2\left(\frac{2^n + 1}{2^n}\right)$$



รูปที่ 3.12 : อัลกอริทึมการคำนวณฟังก์ชันลอการิทึมแบบ Goldschmidt

### 3.4.8 การสังเคราะห์เฟสของสัญญาณ

การสังเคราะห์เฟสของสัญญาณกระตุ้นจะแบ่งออกเป็นสองกรณี หากสัญญาณเฟรมปัจจุบันเป็นสัญญาณ Voiced จะทำการสังเคราะห์สัญญาณไซด์จากเฟสที่ได้คำนวณจากความสัมพันธ์มูลฐาน ซึ่งสามารถใช้วงจรคำนวณสัญญาณไซด์ที่ได้กล่าวในหัวข้อที่ 3.4.3 ได้ แต่หากเป็นสัญญาณ unvoiced แล้วสัญญาณกระตุ้นจะถูกสร้างจากเฟสที่สุ่มขึ้นมาโดยมีวงจรสุ่มตามอัลกอริทึมในรูปที่ 3.13 และมีขั้นตอนการคำนวณตามรูปที่ 3.14 โดยจะเห็นได้ว่าการคำนวณค่าสัญญาณสุ่มจะใช้การคำนวณทั้งหมด 6 สัญญาณนาฬิกาด้วยกันแต่สามารถทำในรูปแบบไปป์ไลน์โดยใช้วงจรคูณ, วงจรเลื่อนบิตและวงจรวกอย่างละ 1 ตัวเท่านั้น ไม่จำเป็นต้องเพิ่มหน่วยประมวลผลใหม่แต่สามารถคำนวณค่าสัญญาณสุ่มได้ทุก 2 สัญญาณนาฬิกาหรือเร็วขึ้นถึง 3 เท่าหากไม่เพิ่มการทำงานแบบไปป์ไลน์และขนานเข้าไป

```

int codec2_rand(void)
{
    unsigned long aa;
    next = next * 1103515245 + 12345;
    aa = (unsigned) (next/65536);
    aa = aa % 32768;
    return ((unsigned) (next/65536) % 32768);
}

```

รูปที่ 3.13 : โปรแกรมคำนวณสัญญาณสุ่มต้นแบบจากภาษาซี

Mem load	RD Rn										
Reg Store		R0 = Rn									
Reg Store	R5 = A0	R5 = A1	R5 = A0	R5 = A1	R5 = A0	R5 = A1	R5 = A0	R5 = A1	R5 = A0	R5 = A1	
Reg Store	R1 = M1	R1 = M0	R1 = M1	R1 = M0	R1 = M1	R1 = M0	R1 = M1	R1 = M0	R1 = M1	R1 = M0	
Mul			R0 = R0 * R1		R0 = R0 * R1	R4 = R1 * R3	R0 = R0 * R1	R4 = R1 * R3			R4 = R1 * R3
Add				R0 = R0 + R5		R0 = R0 + R5	R4 = R4 + R5	R0 = R5 + R5	R4 = R4 + R5		
Shift					R3 = R0 >> 16		R3 = R0 >> 16			R3 = R0 >> 16	
Mem Store								WR R4			WR R4

รูปที่ 3.14 : ขั้นตอนการคำนวณสัญญาณสุ่มในรูปที่ 29 ใช้เวลาเพียง 2 สัญญาณนาฬิกา

ขั้นตอนต่อไปสัญญาณกระตุ้นจะถูกคูณเข้ากับสัญญาณอินพุตจากการแปลงฟูเรียร์ โดยเป็นการคูณแบบจำนวนเชิงซ้อนและมีการหาค่าสัมบูรณ์ของสัญญาณเชิงซ้อนอีกด้วย ดังนั้นจึงต้องใช้การคำนวณการยกกำลัง การหารากที่สอง โดยมีขั้นตอนการคำนวณตามรูปที่ 3.15

Mem Load	RD L	RD Re0	RD Re1	RD Im1	RD Im0	RD Re0	RD Re1	RD Im1	RD Im0				
Reg Store		R2	R0	R1	R2	R3	R0	R1	R2	R3			
Mul					R4 = R0 * R1	R5 = R0 * R2	R1 = R1 * R3	R2 = R2 * R3	R4 = R0 * R1	R5 = R0 * R2	R1 = R1 * R3	R2 = R2 * R3	
Add								R3 = R5 + R1	R5 = R4 - R2			R3 = R5 + R1	R5 = R4 - R2
Mem Store								WR R3	WR R5			WR R3	WR R5

รูปที่ 3.15 : การคูณสัญญาณเชิงซ้อนสำหรับสัญญาณกระตุ้นและสัญญาณอินพุต

### 3.4.9 วงจรกรองสัญญาณ

วงจรกรองสัญญาณจะทำการเทียบพลังงานของสัญญาณกับค่า Threshold โดยมีเงื่อนไขคือจะอัปเดตค่า Threshold เมื่อพลังงานของสัญญาณเสียมมีค่าน้อยกว่าค่า Threshold เริ่มต้นและสัญญาณเป็นสัญญาณ unvoiced โดยจะมีการคำนวณพลังงานให้อยู่ในรูปเดซิเบล ซึ่งต้องใช้การคำนวณลอการิทึมและเอกโปเนนเชียล นอกจากนั้นจะต้องมีการเปรียบเทียบเงื่อนไขว่าจะทำหรือไม่ทำอีกด้วย สุดท้ายหากสัญญาณเป็นสัญญาณเสียมแบบ voiced ที่พลังงานน้อยกว่าค่า Threshold จะทำการเปลี่ยนสัญญาณเป็นสัญญาณไซค์จกเฟสที่ถูกสุ่มขึ้นแทน

### 3.4.10 วงจรแปลงฟูเรียร์กลับแบบเร็ว

จากนั้นสัญญาณที่ถูกกรองแล้วจะถูกแปลงฟูเรียร์กลับแบบเร็ว (Inverse Fast Fourier Transform) โดยสามารถใช้วงจรเดียวกับการแปลงฟูเรียร์แบบเร็วได้

### 3.4.11 การบวกแบบซ้อนทับของสัญญาณในแต่ละเฟรม

ขั้นตอนสุดท้ายคือทำสัญญาณที่ถอดรหัสได้ไปทำการกรองอีกครั้งด้วยวงจรกรองสัญญาณความถี่ต่ำแบบ Finite Impulse Response (FIR) โดยใช้วินโดว์แบบสามเหลี่ยม แล้วจึงทำการบวกซ้อนทับกับสัญญาณเฟรมที่แล้วเพื่อลดรอยต่อระหว่างสัญญาณแต่ละเฟรม

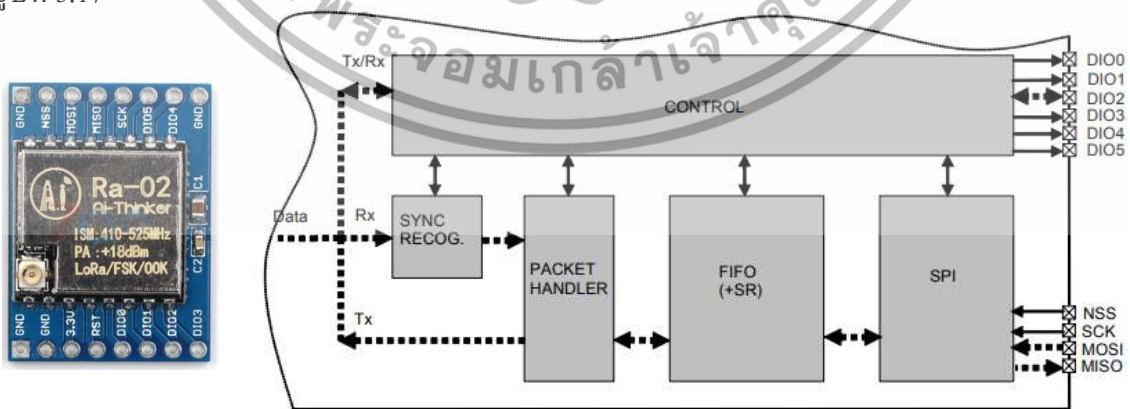
MEM Load	RD sw	RD sn	RD sw	RD sn	RD sw	RD sn	RD sw	RD sn	RD sw
Table Load	RD P0		RD P0		RD P0		RD P0		RD P0
Reg Store		R1=RAM	R3=RAM	R1=RAM	R3=RAM	R1=RAM	R3=RAM	R1=RAM	R3=RAM
Reg Store		R2=TAB		R2=TAB		R2=TAB		R2=TAB	
MUL & ADD			R0=R1*R2	R4=R0+R3	R0=R1*R2	R4=R0+R3	R0=R1*R2	R4=R0+R3	R0=R1*R2
MEM Store					WR R4		WR R4		WR R4

รูปที่ 3.16 : ขั้นตอนการกรองสัญญาณด้วยวินโดว์แบบสามเหลี่ยมและบวกซ้อนทับสัญญาณ

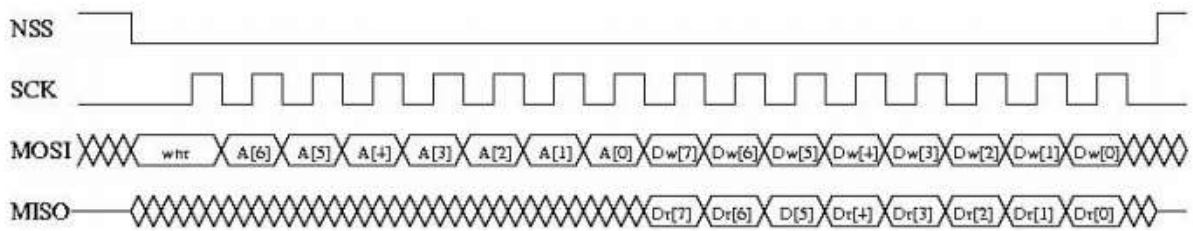
จากรูปที่ 3.16 จะเห็นได้ว่าการกรองสัญญาณและบวกแบบซ้อนทับนั้นใช้เวลาเท่ากับ 5 สัญญาณนาฬิกาสำหรับ 1 ข้อมูล แต่เราสามารถลดเวลาลงด้วยการทำแบบไปป์ไลน์เช่นกัน ทำให้ลดเวลาลงเหลือเพียง 2 สัญญาณนาฬิกาเท่านั้น รวดเร็วกว่าเดิมถึง 2.5 เท่า

### 3.4.12 วงจรเชื่อมต่อกับโมดูลลอรา

โมดูลลอรา sx1278 ใช้การเชื่อมต่อแบบ SPI โดยมีบล็อกโคอะแกรมและโปรโตคอลในการติดต่อ ดังรูปที่ 3.17



รูปที่ 3.17 : โมดูล LoRa SX1278 ที่ใช้ในการทดลอง

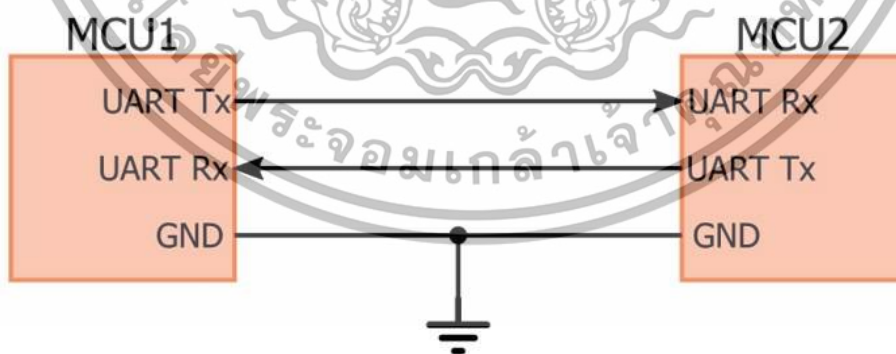


รูปที่ 3.18 : โพรโทคอล SPI ที่ใช้ในการติดต่อโมดูล LoRa SX1278

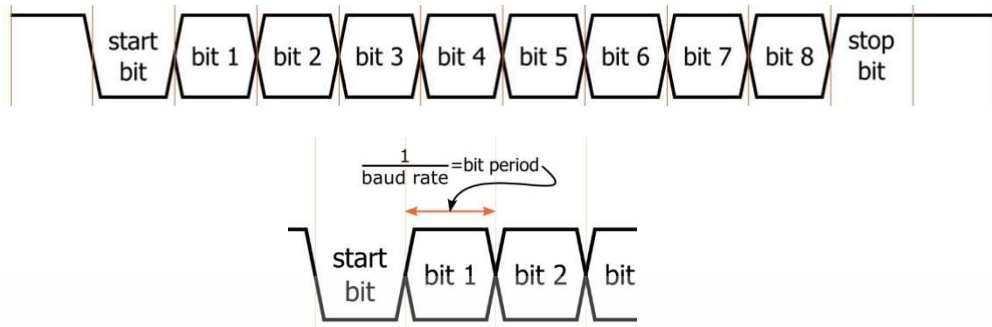
โพรโทคอลที่ใช้ในการติดต่อโมดูล LoRa SX1278 แสดงดังรูปที่ 3.20 โดยเริ่มต้นด้วยให้สัญญาณ NSS เป็น 0 จากนั้นจะเริ่มส่งสัญญาณว่าต้องการเขียนหรืออ่านค่า หากต้องการเขียนบิตแรกบน MOSI จะเป็น 1 แต่หากอ่านจะเป็น 0 จากนั้นจะส่งแอดเดรสขนาด 7 บิตไปที่ละสัญญาณนาฬิกา เริ่มจากบิตที่ 6 ก่อน โดยจะเริ่มส่งเมื่อสัญญาณนาฬิกา SCK เป็นขอบลบ หลังจากนั้นจะเริ่มส่งข้อมูลตั้งแต่บิต 7 ไปจนถึง 0 ในกรณีเขียนค่า หรือกรณีอ่านค่า สัญญาณข้อมูลจะอ่านได้ที่ขา MISO ไปจนกระทั่งครบ 8 บิต และจบด้วยการเปลี่ยนขา NSS เป็น 1 โดยใช้สัญญาณนาฬิกาทั้งหมด 16 รอบสำหรับการอ่านหรือเขียน 1 ไบท์ สัญญาณนาฬิกาความเร็วสูงสุดที่เป็นไปได้คือ 10 MHz

#### 3.4.13 วงจรเชื่อมต่อกับคอมพิวเตอร์

ในการเชื่อมต่อกับคอมพิวเตอร์หรือตัวควบคุมอื่น ใช้การเชื่อมต่ออนุกรมอะซิงโครนัส (UART) เนื่องจากสัญญาณที่สื่อสารไม่ได้ต้องการความเร็วสูง อีกทั้งยังสามารถออกแบบได้ง่าย โดยโปรโตคอลของวงจรเชื่อมต่อแบบ UART มีลักษณะดังรูปที่ 3.19

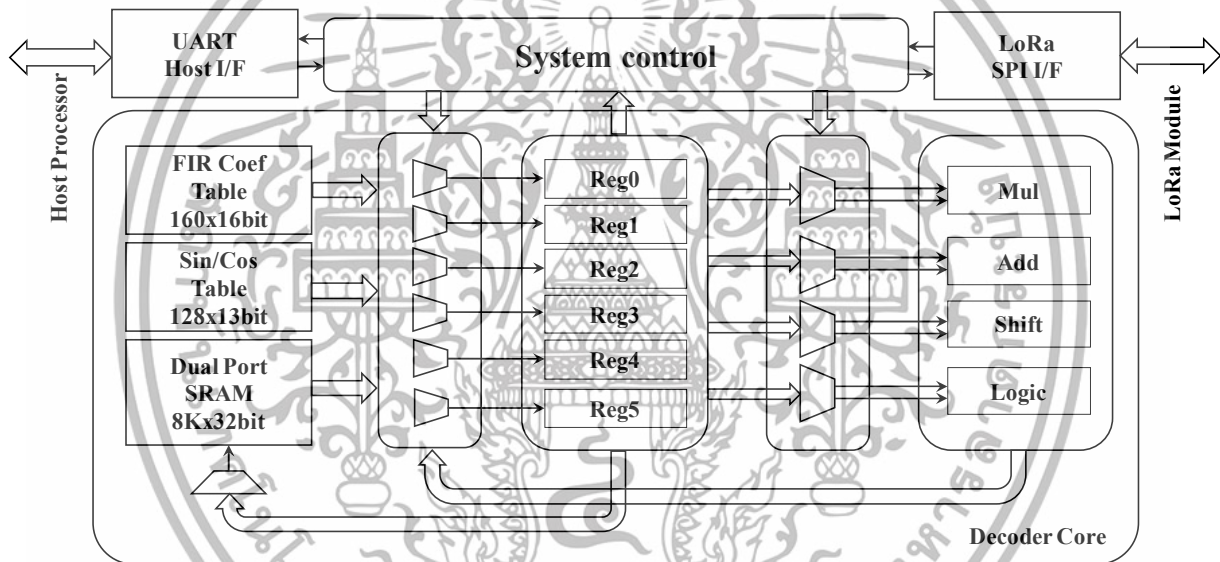


(ก) แสดงการเชื่อมต่อแบบ UART ระหว่างสองอุปกรณ์



รูปที่ 3.19 : การเชื่อมต่อกับคอมพิวเตอร์โดยใช้โปรโตคอล UART

### 3.4.14 สถาปัตยกรรมโดยรวมของวงจรถอดรหัสสัญญาณเสียงแบบ Codec2



รูปที่ 3.20 : สถาปัตยกรรมโดยรวมของวงจรถอดรหัสสัญญาณเสียงแบบ Codec2

โครงสร้างโดยรวมทั้งหมดของวงจรถอดรหัสสัญญาณเสียงแบบ Codec 2 สามารถแสดงได้ดังรูปที่ 3.20 โดยประกอบด้วยส่วนที่เป็นแกนกลางในการถอดรหัสซึ่งประกอบด้วย

1. รีจิสเตอร์ขนาด 32 บิตจำนวน 6 ชุดสำหรับเก็บข้อมูลและพักข้อมูล
2. หน่วยประมวลผลกลางประกอบด้วย วงจรคูณ 32 บิต, วงจรบวก 32 บิต, วงจรเลื่อนบิตและ วงจรคำนวณด้านตรรก จำนวนอย่างละ 1 ชุด
3. หน่วยความจำแบบแรมสองพอร์ตขนาด 8Kx32bit จำนวน 1 ชุด

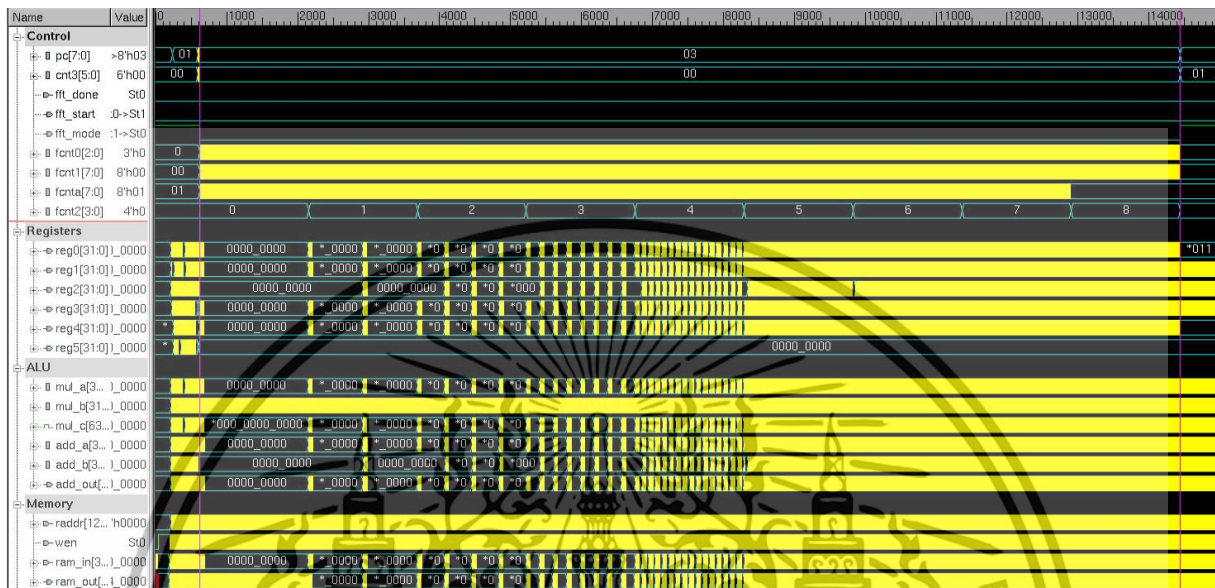
4. หน่วยความจำแบบรอมเพื่อเก็บค่าตารางตรีโกณมิติขนาด 128x13bit จำนวน 1 ชุด
5. หน่วยความจำแบบรอมเพื่อเก็บค่าตารางวงจรรองสัญญาณ FIR ขนาด 160x16bit จำนวน 1 ชุด
6. หน่วยควบคุมกลางเพื่อทำการจัดการตารางการทำงาน (System Control)
7. ตัวสื่อสารกับหน่วยประมวลผลกลาง
8. ตัวสื่อสารกับโมดูลลอรา





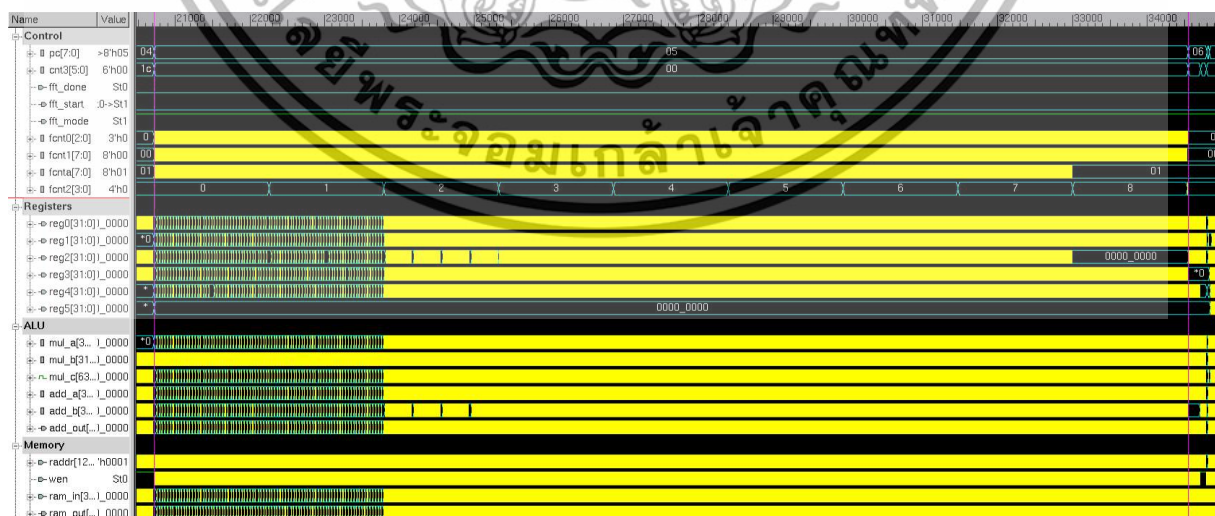
คำนวณด้วยซอฟต์แวร์และฮาร์ดแวร์ที่ออกแบบขึ้นมา  
 จำนวนของวงจรที่ได้ออกแบบ

อีกทั้งยังได้วัดประสิทธิภาพและคุณสมบัติในการ



รูปที่ 4.2 : ผลการทดสอบการทำงานวงจรแปลงฟูเรียร์แบบเร็ว

จากรูปที่ 4.2 และ 4.3 ผลการทดสอบวงจรแปลงฟูเรียร์และการแปลงผกผันแบบเร็ว สามารถคำนวณได้อย่างถูกต้องเมื่อเปรียบเทียบกับผลจากซอฟต์แวร์ต้นแบบ โดยใช้เวลาในการคำนวณทั้งหมด 512 ข้อมูลด้วยการทำคำนวณแบบบิตเตอร์พายจำนวน 9 ชั้น ใช้เวลาทั้งหมด 13831 สัญญาณนาฬิกา

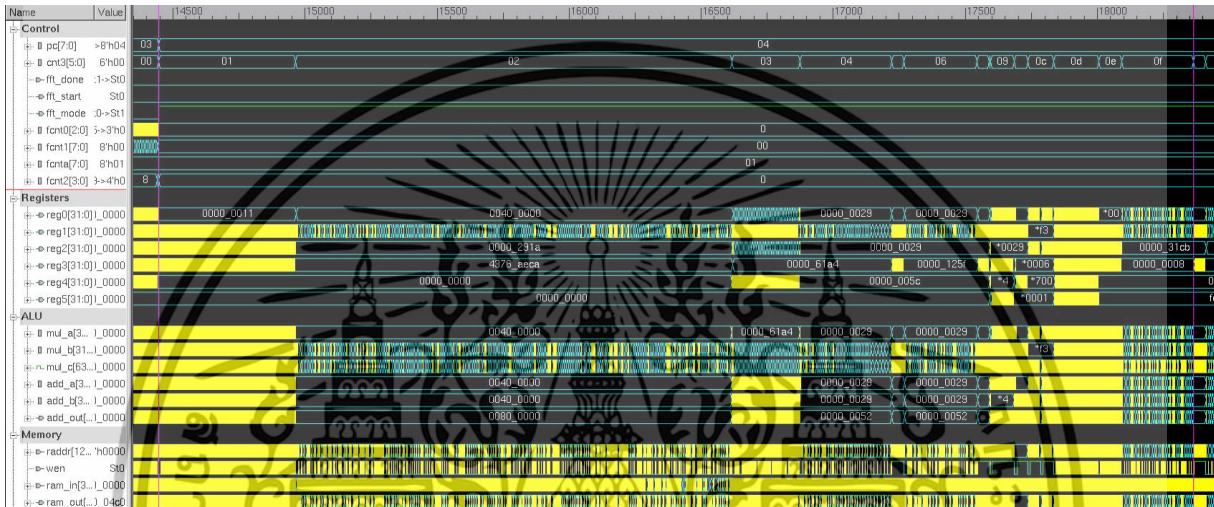


รูปที่ 4.3 : ผลการทดสอบการทำงานวงจรแปลงฟูเรียร์แบบเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 4.3 ผลการทดสอบการทำงานของวงจรสร้างแอมพลิฟายเออร์โมนิค

การทดสอบการทำงานของวงจรแปลงสร้างแอมพลิฟายเออร์โมนิค จะแปลงค่าสัญญาณอินพุตจากซอฟต์แวร์ต้นแบบภาษาซี ทำการทดสอบการทำงานโดยการจำลองการทำงาน และเปรียบเทียบผลที่ได้จากการคำนวณด้วยซอฟต์แวร์และฮาร์ดแวร์ที่ออกแบบขึ้นมา อีกทั้งยังได้วัดประสิทธิภาพและคุณสมบัติในการคำนวณของวงจรที่ได้ออกแบบ



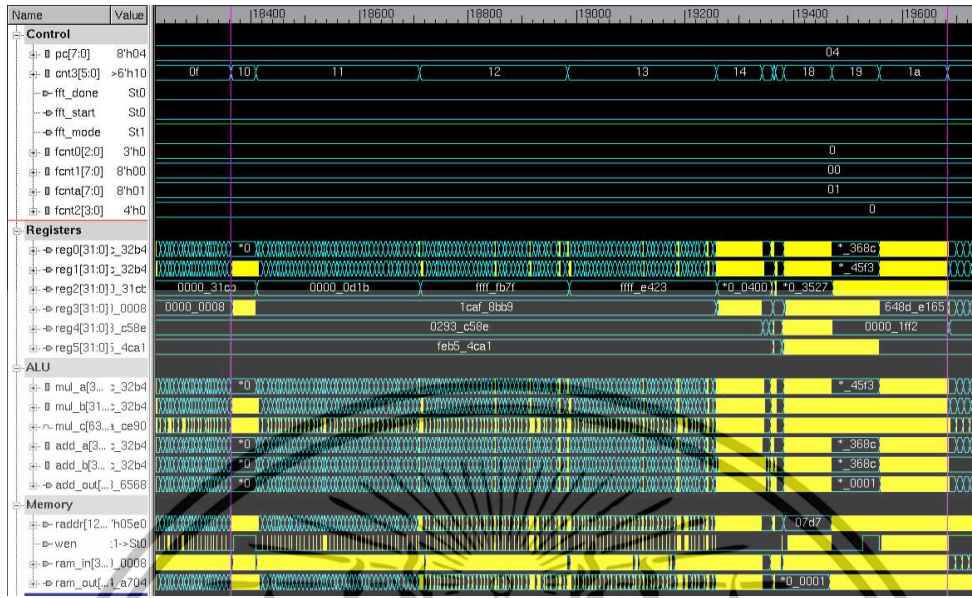
รูปที่ 4.4 : ผลการทดสอบการทำงานของวงจรสร้างแอมพลิฟายเออร์โมนิค

จากรูปที่ 4.4 ผลการทดสอบวงจรสร้างแอมพลิฟายเออร์โมนิค สามารถคำนวณได้อย่างถูกต้องเมื่อเปรียบเทียบกับผลจากซอฟต์แวร์ต้นแบบ โดยใช้การคำนวณมีการคูณ, บวก, ทหาร, เอ็กโปเนนเชียลและลอการิทึมใช้เวลาทั้งหมด 3,766 สัญญาณนาฬิกา

### 4.4 ผลการทดสอบการทำงานของวงจรสังเคราะห์เฟสของสัญญาณ

การทดสอบการทำงานของวงจรสังเคราะห์เฟสของสัญญาณ จะแปลงค่าสัญญาณอินพุตจากซอฟต์แวร์ต้นแบบภาษาซี ทำการทดสอบการทำงานโดยการจำลองการทำงาน และเปรียบเทียบผลที่ได้จากการคำนวณด้วยซอฟต์แวร์และฮาร์ดแวร์ที่ออกแบบขึ้นมา อีกทั้งยังได้วัดประสิทธิภาพและคุณสมบัติในการคำนวณของวงจรที่ได้ออกแบบ

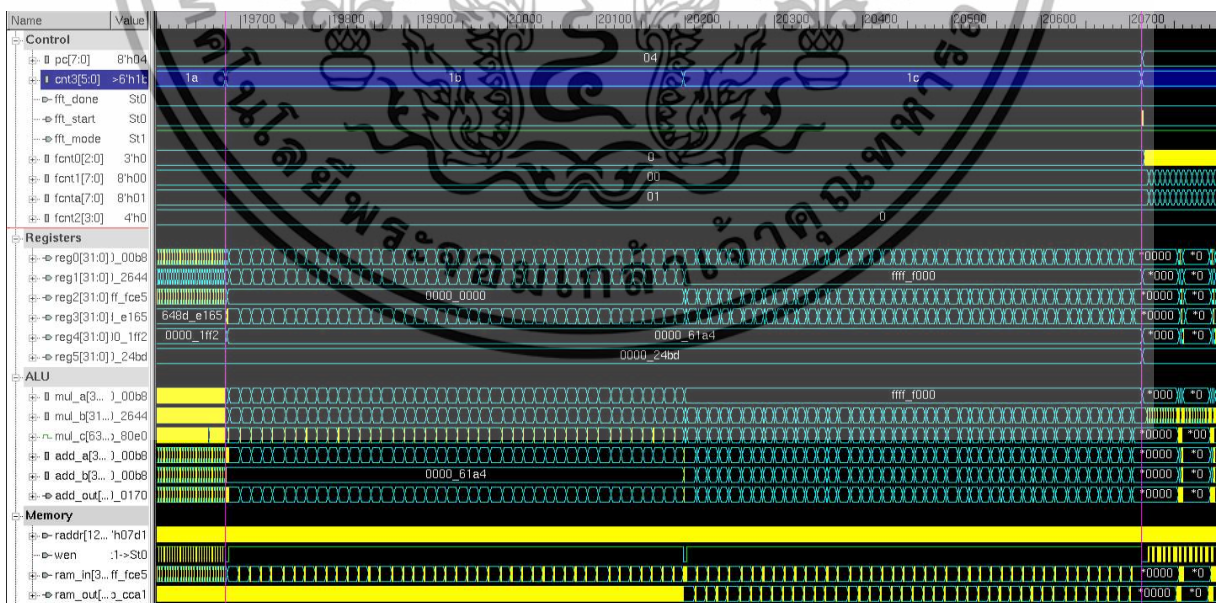
จากรูปที่ 4.5 ผลการทดสอบวงจรสังเคราะห์เฟสของสัญญาณ สามารถคำนวณได้อย่างถูกต้องเมื่อเปรียบเทียบกับผลจากซอฟต์แวร์ต้นแบบ โดยใช้การคำนวณมีการคูณ, บวกและการคำนวณตรีโกณมิติ ใช้เวลาทั้งหมด 3040 สัญญาณนาฬิกา



รูปที่ 4.5 : ผลการทดสอบการทำงานของวงจรสังเคราะห์เฟสของสัญญาณ

#### 4.5 ผลการทดสอบการทำงานของวงจรกรองสัญญาณ

การทดสอบการทำงานของวงจรกรองสัญญาณ จะแปลงค่าสัญญาณอินพุตจากซอฟต์แวร์ต้นแบบภาษาซี ทำการทดสอบการทำงานโดยการจำลองการทำงาน และเปรียบเทียบผลที่ได้จากการคำนวณด้วยซอฟต์แวร์และฮาร์ดแวร์ที่ออกแบบขึ้นมา อีกทั้งยังได้วัดประสิทธิภาพและคุณสมบัติในการคำนวณของวงจรที่ได้ออกแบบ



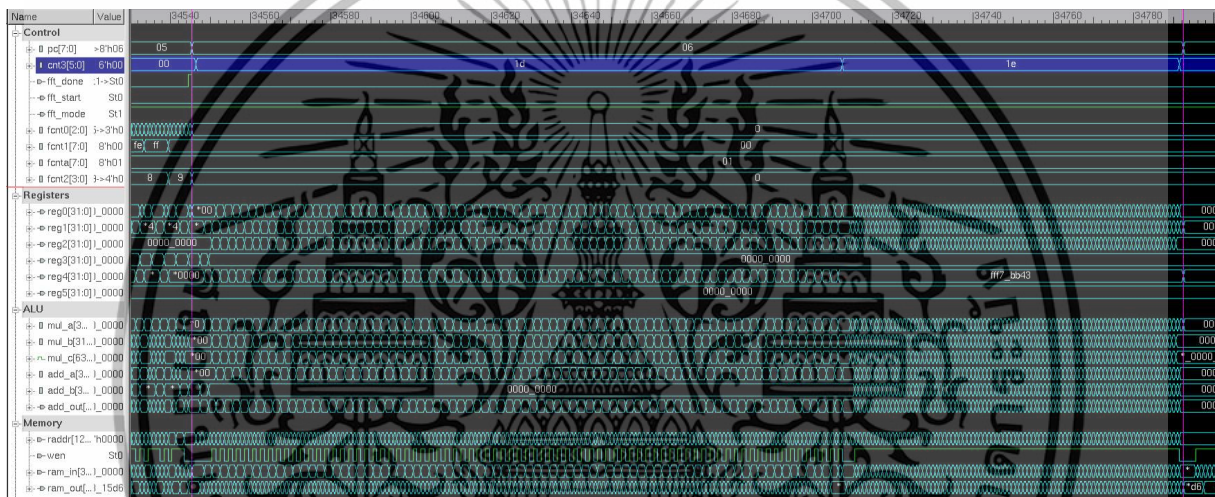
รูปที่ 4.6 : ผลการทดสอบการทำงานของวงจรกรองสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ<sup>42</sup> ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.6 ผลการทดสอบวงจรกรองสัญญาณ สามารถคำนวณได้อย่างถูกต้องเมื่อเปรียบเทียบกับผลจากซอฟต์แวร์ต้นแบบ โดยใช้การคำนวณมีการคูณ,บวก,หาร,เอ็กซ์โปเนนเชียลและลอการิทึม ใช้เวลาทั้งหมด 1797 สัญญาณนาฬิกา

#### 4.6 ผลการทดสอบการทำงานของการทำงานแบบซ้อนทับ

การทดสอบการทำงานของการทำงานแบบซ้อนทับ จะแปลงค่าสัญญาณอินพุตจากซอฟต์แวร์ต้นแบบภาษาซี ทำการทดสอบการทำงานโดยการจำลองการทำงาน และเปรียบเทียบผลที่ได้จากการคำนวณด้วยซอฟต์แวร์และฮาร์ดแวร์ที่ออกแบบขึ้นมา อีกทั้งยังได้วัดประสิทธิภาพและคุณสมบัติในการคำนวณของวงจรที่ได้ออกแบบ



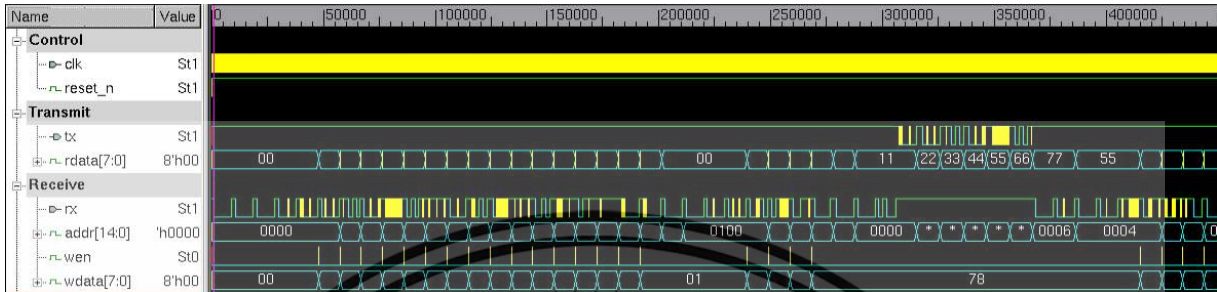
รูปที่ 4.7 : ผลการทดสอบการทำงานของการทำงานแบบซ้อนทับ

จากรูปที่ 4.7 ผลการทดสอบการทำงานแบบซ้อนทับ สามารถคำนวณได้อย่างถูกต้องเมื่อเปรียบเทียบกับผลจากซอฟต์แวร์ต้นแบบ โดยใช้การคำนวณเป็นการคำนวณและวงจรกรองสัญญาณแบบ FIR จำนวน 80 จุดและบวกซ้อนทับกับสัญญาณในเฟรมที่ผ่านมา ใช้เวลาทั้งหมด 248 สัญญาณนาฬิกา

#### 4.7 ผลการทดสอบการทำงานของวงจรสื่อสารอนุกรมแบบอะซิงโครนัส (UART)

การทดสอบการทำงานของวงจรสื่อสารอนุกรมแบบอะซิงโครนัส วงจรสื่อสารได้ถูกออกแบบให้สื่อสารได้ด้วยความเร็วเท่ากับ 230,400 บิตต่อวินาทีเพื่อใช้ในการทดสอบและควบคุมผ่านคอมพิวเตอร์ จากรูปที่ 4.4 ผลการทดสอบสามารถรับส่งสัญญาณได้อย่างถูกต้องและสามารถทำได้พร้อมกัน โดยขั้นตอนในการรับข้อมูล จะทำการรับแอดเดรสของหน่วยความจำที่ต้องการติดต่อ จากนั้นจะระบุจำนวนไบต์ที่ต้องการเขียนและตามด้วยข้อมูลที่ต้องการเขียน ในส่วนการส่งข้อมูล โสสจะต้องทำการกำหนดแอดเดรสของ

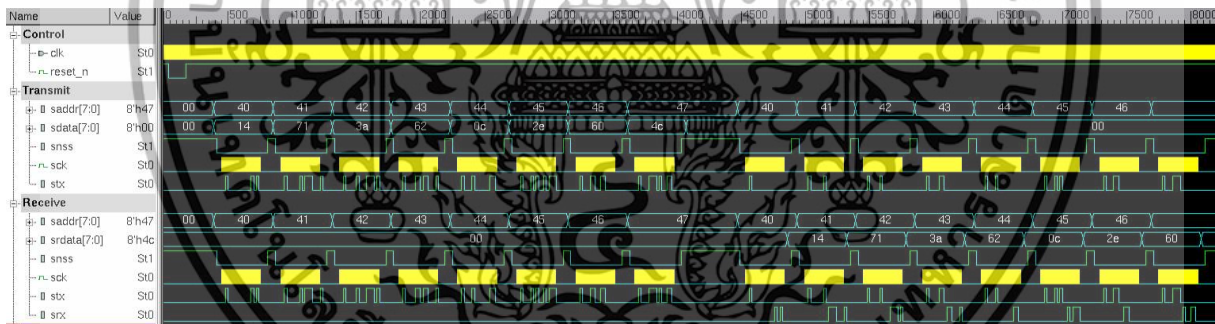
หน่วยความจำที่ต้องการติดต่อ จากนั้นระบุจำนวนไบต์ที่ต้องการเขียน หลังจากนั้นวงจรจะอ่านค่าจากหน่วยความจำและแปลงเป็นสัญญาณอนุกรมส่งออกไปทางพอร์ต TX ตามลำดับ



รูปที่ 4.8 : ผลการทดสอบการทำงานของวงจรถ่ายโอนข้อมูลแบบอะซิงโครนัส (UART)

#### 4.8 ผลการทดสอบการทำงานของวงจรถ่ายโอนข้อมูลแบบซิงโครนัส (SPI)

การทดสอบการทำงานของวงจรถ่ายโอนข้อมูลแบบซิงโครนัส (SPI) เพื่อทำการเชื่อมต่อกับโมดูล LoRa sx1278 ใช้ความถี่สัญญาณนาฬิกาที่ 5 MHz จากผลการทดสอบทั้งด้านรับและส่ง สามารถรับส่งสัญญาณได้ถูกต้องตามโปรโตคอลของ SPI



รูปที่ 4.9 : ผลการทดสอบการทำงานของวงจรถ่ายโอนข้อมูลแบบซิงโครนัส (SPI)

#### 4.9 ผลการเปรียบเทียบประสิทธิภาพของการถอดรหัส Codec2

ตารางที่ 4.1 แสดงผลการเปรียบเทียบความเร็วในการถอดรหัสสัญญาณเสียง Codec2 โดยเปรียบเทียบกับการคำนวณด้วยไมโครโปรเซสเซอร์ Cortex-M4 จะเห็นว่าทุกกระบวนการ วงจรที่ออกแบบสามารถทำงานได้เร็วกว่า โดยเฉพาะอย่างยิ่งการแปลงฟูรีเยร์สามารถคำนวณได้เร็วกว่าถึง 22 เท่า และโดยภาพรวมสามารถถอดรหัสสัญญาณเสียง 1200 bps ขนาด 1 เฟรมได้โดยใช้สัญญาณนาฬิกาเพียง 36,911 ลูกเท่านั้น เร็วว่าไมโครโปรเซสเซอร์ถึง 20 เท่าด้วยกัน ดังนั้นในการประมวลผลแบบเรียลไทม์ซึ่งมีจำนวนเฟรมเท่ากับ 100 เฟรมต่อวินาทีนั้น จะใช้สัญญาณนาฬิกาเพียง 3.7 MHz ก็เพียงพอในขณะที่หากใช้ไมโครโปรเซสเซอร์จะต้อง

ใช้ความเร็วของสัญญาณนาฬิกาถึง 75.2 MHz เลขที่เดียว วงจรที่ออกแบบจึงเหมาะกับการใช้ในระบบฝังตัว เพราะใช้สัญญาณนาฬิกาที่น้อยกว่าและกินพลังงานต่ำกว่าการประมวลผลด้วยไมโครโปรเซสเซอร์โดยทั่วไป

**ตาราง 4.1 : แสดงจำนวนสัญญาณนาฬิกาที่ใช้ในการถอดรหัสสัญญาณเสียงแบบ Codec2**

กระบวนการ	วงจรที่ออกแบบ (cycle)	Cortex-M4 (cycle)
แปลงคู่ของเส้นสเปกตรัม	398	5,328
การแปลงฟูเรียร์แบบเร็ว	13,831	304,622
การสร้างแอมพลิจูดฮาร์โมนิก	3,766	70,739
การสังเคราะห์เฟสของสัญญาณ	3,040	14,452
วงจรกรองสัญญาณ	1,797	14,486
การแปลงฟูเรียร์กลับแบบเร็ว	13,831	337,931
การบวกแบบซ้อนทับ	248	4,650
รวม	36,911	752,208

#### 4.10 ผลการสังเคราะห์วงจรและกำลังงานของวงจร

ในหัวข้อนี้อธิบายถึงผลการสังเคราะห์วงจรและการวัดกำลังของวงจร ซึ่งวงจรได้ถูกเขียนด้วยภาษา Verilog HDL และสังเคราะห์ด้วยไลบรารีของบริษัท Global Foundry (GF) ด้วยเทคโนโลยีทรานซิสเตอร์ขนาด 0.13 ไมโครเมตร ผลการสังเคราะห์วงจรได้ตามตารางที่ 4.2

**ตาราง 3.2 ผลการสังเคราะห์วงจรด้วยไลบรารี GF 0.13um**

ชนิดของเซลล์	ขนาด (um <sup>2</sup> )	จำนวนเกต (Gate)
Combinational Cell	50,010.49	7,365
Buffer/ Inv Cell	1,853.56	273
Sequential Cell	12,543.78	1,847
Total	64,407.83	9,485

จะเห็นได้ว่าวงจรใช้เกตเพียง 9,485 เกตเท่านั้นในการสร้างวงจรเพื่อถอดรหัสสัญญาณเสียงแบบ Codec2 ซึ่งในการวัดกำลังงานที่ใช้จากวงจรที่สร้างขึ้น เปรียบเทียบกับกำลังงานที่ใช้จากการใช้

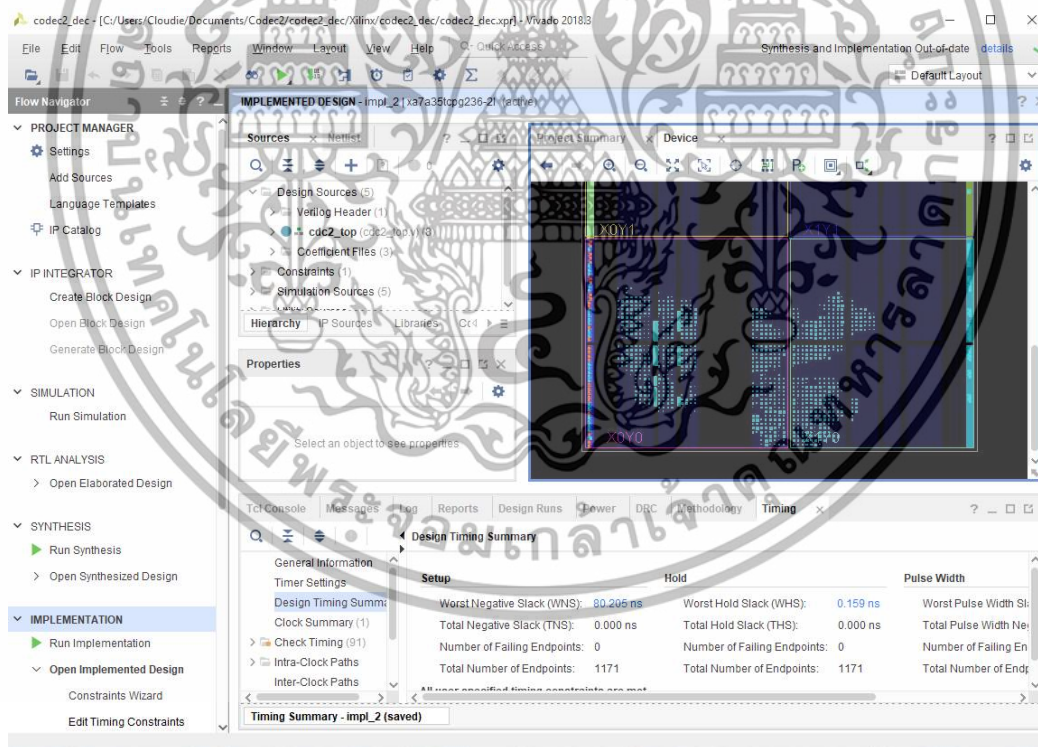
ไมโครโปรเซสเซอร์ในการคำนวณได้ผลการทดลองตามตารางที่ 4.3 ซึ่งสรุปได้ว่าวงจรที่สร้างขึ้นใช้กำลังงานน้อยกว่ามาก จึงมีความเหมาะสมจะใช้งานที่มีพลังงานจำกัดเช่น ใช้พลังงานจากแบตเตอรี่เป็นต้น

ตาราง 4.3 แสดงกำลังงานที่ใช้ในการถอดรหัสสัญญาณเสียงแบบ Codec2

	วงจรที่สร้างขึ้น (GF 0.13um)	Cortex-M4 (STM32F410)
กำลังงาน ไฟฟ้า (uw)	0.093	8.29

#### 4.11 ผลการสร้างวงจรต้นแบบบนบอร์ด FPGA

วงจรที่ได้ออกแบบถูกทดสอบกับบอร์ด FPGA รุ่น Digilent CMOD A7 และได้สังเคราะห์วงจรด้วยโปรแกรม Xilinx Vivado รูปที่ 4.10 แสดงผลการสร้างวงจรบน FPGA และผลการสังเคราะห์วงจร



รูปที่ 4.10 : ผลการสร้างวงจรถอดรหัสเสียง Codec2 ต้นแบบด้วยบอร์ด FPGA

ตาราง 4.4 เปอร์เซ็นต์การใช้งานฮาร์ดแวร์ของ FPGA ต้นแบบและความเร็วสัญญาณนาฬิกา

	จำนวนที่ใช้	จำนวนที่มี	เปอร์เซ็นต์ใช้งาน
LUT	3,312	63,400	5.2%
Flip Flop	712	126,800	0.56%
DSPs	8	240	3.33%
Memory	72,064 bit	4,976,640 bit	1.45%
Max Clock Freq			84.23 MHz



## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

จากผลการออกแบบสถาปัตยกรรมเพื่อถอดรหัสสัญญาณเสียงแบบ Codec2 ด้วยโครงสร้างแบบ Very Long Word Instruction นั้นสามารถคำนวณอัลกอริทึมอันซับซ้อนประกอบด้วยการแปลงฟูเรียร์แบบเร็ว, การแปลงคู่สเปกตรัม, การสังเคราะห์เฟสสัญญาณ, การคำนวณหาพลังงานและกรองสัญญาณ รวมถึงการคำนวณการกรองสัญญาณแบบ Finite Impulse Response ด้วยขนาดวงจรเพียง 9485 เกท ซึ่งเป็นการคำนวณแบบจุดตรึงขนาด 16 บิต ลดความซับซ้อนของซอฟต์แวร์ต้นแบบที่คำนวณด้วยจุดลอยตัวแบบ 32 บิตไปได้อย่างมาก ทำให้การแปลงสัญญาณเสียงขนาด 1200 บิตต่อวินาทีใช้สัญญาณนาฬิกาเพียง 4 เมกะเฮิร์ตเท่านั้น ต่ำกว่าการใช้ไมโครโปรเซสเซอร์ Cortex-M4 ซึ่งต้องใช้สัญญาณนาฬิกาถึง 76 เมกะเฮิร์ต อีกทั้งยังสามารถลดกำลังงานในการถอดรหัสเหลือเพียงประมาณ 0.1 มิลลิวัตต์ เทียบกับไมโครโปรเซสเซอร์ที่ใช้ถึง 8 มิลลิวัตต์

จะเห็นได้ว่างานวิจัยนี้ สามารถการคำนวณอัลกอริทึมที่หลากหลายในการถอดรหัสสัญญาณเสียงได้ด้วยวงจรขนาดเล็กและมีประสิทธิภาพ เป็นโครงสร้างที่เหมาะสมจะนำไปใช้งานกับระบบอินเทอร์เน็ตแห่งสรรพสิ่งซึ่งต้องการการใช้พลังงานที่น้อย มีขนาดเล็กและราคาถูก เพราะสามารถประหยัดพลังงานไปได้อย่างดี

#### 5.2 ข้อเสนอแนะ

อย่างไรก็ตาม โครงสร้างแบบนี้ยังสามารถพัฒนาต่อไปให้มีความยืดหยุ่นมากยิ่งขึ้น โดยทำเป็นโครงสร้างของ Instruction Set Architecture ที่สามารถประยุกต์ใช้กับมาตรฐานการถอดรหัสเสียงอื่น ๆ เช่น CELP เพื่อเพิ่มความสามารถให้กับระบบ ตัวอย่างเช่นนำไปใช้เป็นตัวช่วยในระบบที่มีไมโครคอนโทรลเลอร์ขนาดเล็กเช่น Arduino ให้สามารถประมวลผลสัญญาณเสียงที่ซับซ้อนได้ด้วยการใช้พลังงานไม่มาก ซึ่งจะเป็งานวิจัยที่จะพัฒนาต่อไปในอนาคต

## บทที่ 6 สรุปผลผลิตงานวิจัย

### 6.1 รายชื่อและรายละเอียดผลผลิตงานวิจัยที่ผลิตได้

the 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST 2019), Luang Prabang, Lao PDR, 2<sup>nd</sup> -5<sup>th</sup> July 2019 สำหรับหลักฐานการเข้าร่วมนำเสนอผลงาน ประกอบด้วย Program schedule และ Paper แสดงอยู่ในภาคผนวก ก



## เอกสารอ้างอิง

- [1] J. Srinonchat, "New Technique to Reduce Bit Rate of LPC-10 Speech Coder," IEEE Region 10 Conference, Hong Kong, November 2006, pp.1-4
- [2] W.C. Chu, "Speech Coding Algorithms Foundation and Evolution of Standardized Coders," Wiley-Interscience, 2003
- [3] M. Sadek, S.E. Ramly, A. Tawfik, "A new CELP speech coder for wireless communication," 14th International Conference on Digital Signal Processing Proceedings, Greece, Vol.2, pp.997-1000
- [4] C. Wu, H. Jiang, B. Li, "An Improved MELP Speech Coder," International Conference on Information Technology and Computer Science, Ukraine, August 2009, Vol.2, pp.130-133
- [5] D. Rowetel, "Techniques for Harmonic Sinusoidal Coding," Ph.D. Thesis, 1997
- [6] D. Rowetel, "Codec2", [http://www.rowetel.com/?page\\_id=452](http://www.rowetel.com/?page_id=452)
- [7] J.E. Shuda, A.J. Rix, M.J. Booyesen, "Module-Level Monitoring Of Solar PV Plants Using Wireless Sensor Networks," 26th Southern African Universities Power and Engineering Conference, South Africa, January 2018, pp.1-7
- [8] T.T. Teo, E.C. Tan, "Implementation of 2400 bps MELP vocoder on TMS320C44," Fourth International Conference on Signal Processing, China, October 1998, pp.576-579
- [9] P.C.S Abhijna, N.R Sangeetha, J.R Sagar, R. Rahul, G. Gupta, "Implementation of CELP encoder using Vivado HLS," International Conference on Recent Trends in Electronics, Information & Communication Technology, India, May 2017, pp.1443-1447
- [10] S. Shivanagi, M. Koushik, J. Kumar, G. Gupta, D. Saravanan, "MELP Decoder Implementation on Zynq Kit by High Level Synthesis," International Conference on Current Trends in Computer, Electrical Electronics and Communication, India, September 2017, pp.78-82
- [11] Koushik M, S. Shivanagi, J. Kumar, J. Yadav, D. Saravanan, "Implementation of MELP Encoder on Zynq FPGA using HLS," International Conference on Current Trends in Computer, Electrical, Electronics and Communication, India, September 2017, pp.87-91

- [12] R.E. Goldschmidt, “**Applications of Division by Convergence,**” MSc dissertation, Massachusetts Inst. of Technology, June 1964
- [13] D. Rowetel, “**Codec2 Speech Coding Reference C Source Code,**” <https://svn.code.sf.net/p/freetel/code/codec2/branches>
- [14] LoRa Alliance, “**A Technical Overview of LoRa and LoRaWAN**”, LoRa Alliance, 2015
- [15] Martin Bor; Utz Roedig, “**LoRa Transmission Parameter Selection**”, 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), 2017
- [16] Umber Noreen; Ahcène Bounceur; Laurent Clavier, “**A study of LoRa low power and wide area network technology**”, 2017 IEEE International Workshop on Measurement and Networking (M&N), 2017



## ภาคผนวก

### ภาคผนวก ก

เอกสารการนำเสนอผลงาน

- Program schedule ของการนำเสนอผลงานวิจัยใน International Conference ICEAST ครั้งที่ 5
- เอกสารที่ได้รับการตีพิมพ์เผยแพร่ (Paper) ในงาน International Conference ICEAST ครั้งที่ 5





**5<sup>th</sup>**

**iceast**

**2019**

**Luang Prabang, Lao PDR**

2-5 July 2019  
[www.iceast2019.org](http://www.iceast2019.org)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปตีพิมพ์หรือเผยแพร่โดยไม่ได้รับอนุญาต



- 11:25-11:45 Paper ID 129: An Ultra-violet sterilization robot for disinfection  
*Pacharawan Chanprakon, Tapparat Sae-oung, Treesukon Treebupachatsakul, Pimkhuan Hannanta-anan, Thailand Wibool Piyawattanametha, Thailand, USA*
- 11:45-12:05 Paper ID 13: E-Framework for Tourism Product Scoring  
*Punyavee Visadsoontornsakul, Thailand*

**Session 18: Telecommunications, Computer Science, Instrumentation and Control, Electrical And Electronic Engineering 5**

**Session chair: Khamphong Kongsomboun**

**Room: C**

- 10:25-10:45 Paper ID 61: Phuket Mangrove Gastronomy Information System Development  
*Urairat Makchan and Putsadee Pornphol, Thailand*
- 10:45-11:05 Paper ID 109: Reducing Energy Consumption by using Greedy Three-hop Information over Ad-hoc Network  
*Phonepadith Phoummayong and Somephone Kanthavong, Lao PDR, Hiroshi Ishii, Japan*
- 11:05-11:25 Paper ID 10: Home Appliances-Controlled Platform with HomeKit Application  
*Charoen Vongchumyen, Pakorn Watanachaturaporn, Sarayuth Torthitithum, Jirachaya Khamsopa, Thailand*
- 11:25-11:45 Paper ID 67: An Efficient Hardware Architecture of Codec2 Low Bit-rate Speech Decoder  
*Sumek Wisayataksin, Thailand*
- 11:45-12:05 Paper ID 66: Measuring Icon Recognition Mapping with Automated Decision Making System  
*Pikulkaew Tangtisanon and Yanaphat Khongtrakon, Thailand*

**Session 19: Telecommunications, Computer Science, Instrumentation and Control, Electrical And Electronic Engineering 6**

**Session chair: Theerayod Wiangtong**

**Room: A**

- 13:00-13:20 Paper ID 42: The Development of  $K^N$  Voltage Gain Switched-Capacitor DC-DC Converter Based on Parallel Fibonacci-Type Converter  
*Ratanaubol Rubpongse, Wanglok Do, Kei Eguchi, Japan*
- 13:20-13:40 Paper ID 80: Efficiency Improvement of a Dual-Active-Bridge Isolated Bidirectional DC-DC Converter  
*Nisakorn Pruengprach and Warachart Suwan-ngam, Thailand*
- 13:40-14:00 Paper ID 77: Transient Respond Comparison Between Modified Droop Control and Virtual Synchronous Generator in Standalone Microgrid  
*Darith Leng and Sompob Polmai, Thailand*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา XXXVI อย่างเป็นทางการหรือแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# An Efficient Hardware Architecture of Codec2 Low Bit-rate Speech Decoder

Sumek Wisayataksin

Department of Electronics Engineering, Faculty of Engineering,  
King Mongkut's Institute of Technology Ladkrabang,  
Bangkok, Thailand  
sumek.wi@kmitl.ac.th

**Abstract**— Speech coding algorithms have been developed for years to digitalize human voice to a few binary bits as possible while maintaining reasonable quality. Codec2 vocoder algorithm is one of an efficient sinusoidal coding with very high compression rate down to 450 bit/s. In this paper, an efficient hardware architecture of Codec2 decoder is proposed to increase the performance of voice decoding process and reduce comprehensive tasks from a host processor. Although the sinusoidal decoding algorithm is complicated with many arithmetic operations such as the arithmetic of complex numbers, FFT, FIR filter, division, trigonometry, exponential and logarithm functions, several techniques were explored to optimize and parallelize a datapath of the proposed hardware. The implementation on Xilinx Artix-7 FPGA revealed that the proposed architecture could reduce the processing time up to 20 times, compared to the conventional Cortex-M4 CPU running with the original software.

**Keywords**— Speech Coding, Sinusoidal Coding, Linear Prediction Coding, Codec2, Application Specific Integrated Circuits, FPGA, Digital System Design.

## I. INTRODUCTION

Nowadays, audio coding especially speech vocoder plays an important role in various applications from amateur audio, mobile telephony to voice over IP. Due to the increasing demand for voice communication, speech coding algorithms have been explored to reduce the size of packets, but it still retains good perceptual quality. The linear prediction coding vocoder such LPC-10 created by United States Department of Defense is one of the early standard [1], which has the most influential ideas in speech coding to many algorithms later, for the examples, Code-excited linear prediction (CELP), Mixed-excitation linear prediction (MELP) and Codec2 [2].

Code-excited linear prediction (CELP) improved the encoding performance by the utilization of long-term and short-term linear prediction model and introduced an excitation codebook, which gives the best excitation sequence. The significantly better quality than others in the past made MPEG-4 chose CELP as a compression scheme for its Audio speech coding [3]. After that, Mixed-excitation linear prediction (MELP) was developed to solve the problem of annoying artifacts such as buzzes, thumps and tonal noises. The generation of a mixed excitation signal is the key to this algorithm [4]. The MELP becomes a standard for military applications and secure voice of NSA and NATO.

Furthermore, D. Rowe developed a low-bitrate speech codec called Codec2 as a patent-free, open source standard [5]. The contribution of Codec2 is very low-bitrate scheme

from 3200 to only 450 bit/s. The experiment claimed that the quality of codec2 voice is better than one from the MELP algorithm [6]. According to the tiny size of voice packets from its characteristic, it is a challenge for hardware designers to develop a Codec2 vocoder system to operate in such a low bandwidth and low power environments. For example, to transmit voice data in a rural area with LoRa communication protocol can be possible, since LoRa can broadcast data at bit rate 530 bit/s to 10 km distance by its specification [7].

Due to the complexity of linear prediction algorithms such as complex number operations, sinusoidal, logarithm, exponential functions, digital filter and a lot of branch conditions, voice encoder and decoder systems have been widely implemented in software and run on off-the-shelf processors or in digital signal processors (DSP) [8]. However, some researchers have created the system using hardware to accelerate the coding operations. P.C.S Abhijna and his teams developed the CELP encoder using Vivado High Language Synthesis [9]. S. Shivanagi also implemented the MELP vocoder using the same idea [10][11]. However, it is difficult to analyze the redundancy of generated hardware by using high-level synthesis such as C language. Moreover, the performances of automatic generated RTL and the architecture were not compared to a system running on microprocessors with the original code.

In this research, the algorithm of Codec2 voice decoder was analyzed, the efficient hardware architecture was created by using Verilog HDL and synthesized to the FPGA. In the next section, the Codec2 decoding scheme will be considered in the hardware designer point of view to optimize all operations into simple ASIC structures. Then, the architecture of our proposed decoder will be demonstrated in Section 3. After that, in Section 4, the performance, hardware utilization, and synthesis timing results will be described. Finally, the conclusion will be addressed in Section 5.

## II. CODEC2 DECODER SCHEME

Codec2 applies the harmonic sinusoidal speech coding for efficient voice compression. The speech synthesis process will reconstruct the human voice by generating the sinusoid wave from a fundamental frequency ( $\omega_0$ ) and its harmonics with corresponding amplitudes and phases. Fig.1 illustrates a block diagram of the targeted decoder. The Line spectral pairs (LSP) are parsed from given bitstreams by host CPU and converted to linear prediction coefficients (LPC) by the LSP to LPC block. After that, the Fast Fourier Transform (FFT) is performed to retrieve spectral amplitude samples. Then, the

power spectrum of each band is determined, and post filtering is executed respectively. On the other side, the voiced information is used to create excitation samples in the phase synthesis block. When two operations are completed, the excitations and power spectrum samples are multiplied together, and inversed FFT is determined again. Finally, the output voice is filtered by the FIR filter before overlapped adding to the previous output frame.

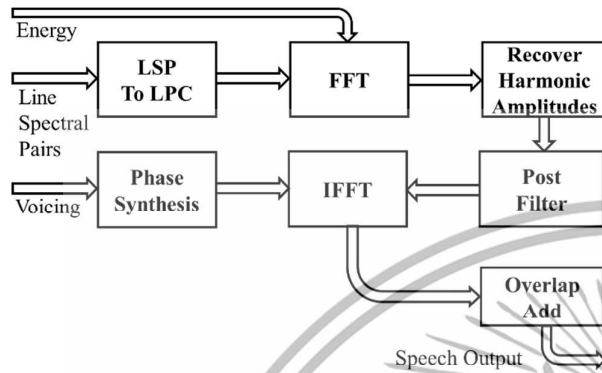


Fig.1 Block diagram of Codec2 decoder

Lines spectrum pairs (LSP) are the way to represent the Linear prediction coefficients (LPC) since they give greater interpolation properties and robustness to quantization noise. Two IIR filters can be deployed to retrieve the LPC from LSP samples as depicted in Fig.2. Pairs of LSP input samples are  $\omega_p(n)$  and  $\omega_q(n)$ . The initial  $a_p(0)$  and  $a_q(0)$  are 1,  $b_p(0)$ ,  $b_q(0)$ ,  $c_p(0)$  and  $c_q(0)$  are 0. This process can be simply executed by using delay registers, an adder, and a multiplier. However, to compute the cosine function will be discussed in detail.

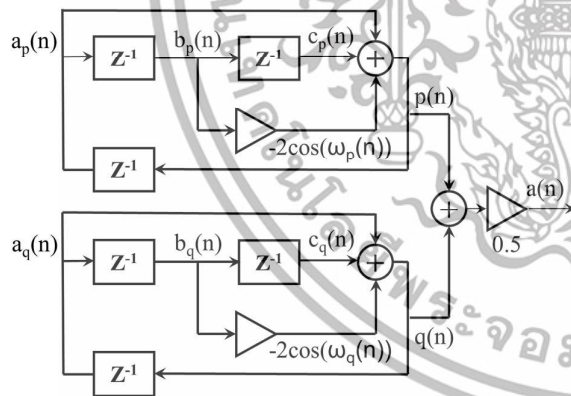


Fig.2 LSP to LPC conversion

Fast Fourier Transform (FFT) and its inverse operation (IFFT) are very computationally intensive algorithms, which are suitable for ASIC structure. In Codec2, 512-point FFT and IFFT are performed every frame. In order to reduce the computation time, but retain a reasonable circuit size, the pipeline scheme was implemented as shown in Fig.3. Since one butterfly operation requires six additions and four multiplications, the fastest scheme with one adder and one multiplier is six clocks not including memory load and store time. However, a twiddle factor ( $W_r+jW_i$ ) was also considered whether we should implement it withCORDIC algorithm or simply make a look-up table. After the

experiment, the second choice was decided because it could reduce a lot of computation time with only 31-bit of 128 words of ROM. This computation flow gives the maximum hardware utilization of one adder and one multiplier.

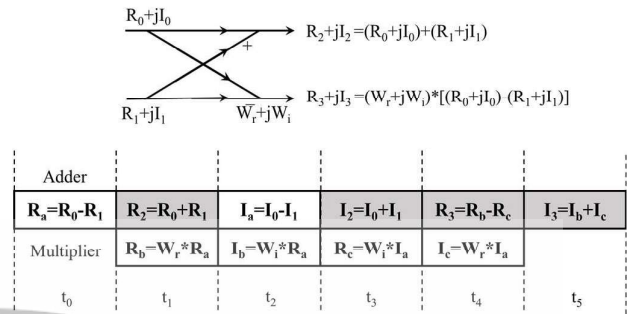


Fig.3 FFT butterfly and its pipeline scheme.

The harmonic amplitude recovery process is mainly to find the power spectrum of each band. In this stage, the mathematic operations are mostly to compute the complex absolute value, number summation, division, and square root. The post-filter process determines the average energy across the spectrum in a decibel unit and decides whether the samples should be filtered or not. Phase analysis generates excitation samples by using trigonometry function. It is obvious that there are many complicated mathematic operations such as number division, exponential, logarithm and trigonometry function required for the rest of the work.

$$t_0[n] = \frac{2^n}{2^n + 1}, \quad t_1[n] = \frac{2^n + 1}{2^n}, \quad t_2[n] = \log_2 \left( \frac{2^n + 1}{2^n} \right)$$

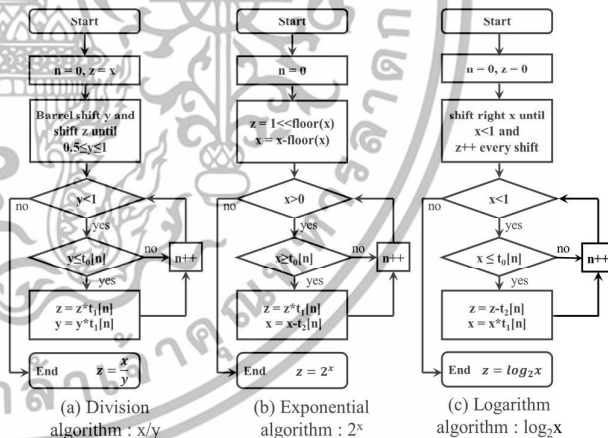


Fig.4 Modified Goldschmidt algorithm for number division, exponential and logarithm calculation

The trigonometry functions including sine and cosine can be simply shared with a look-up table using in the FFT process. However, the depth of sine/cosine tables is only 128 addresses, but other processes need more precise to 6,437 addresses. In order to compromise between ROM size and computation time, the linear interpolation was deployed to increase the resolution without expanding ROM size. The performance of interpolation will be later discussed in Section 4.

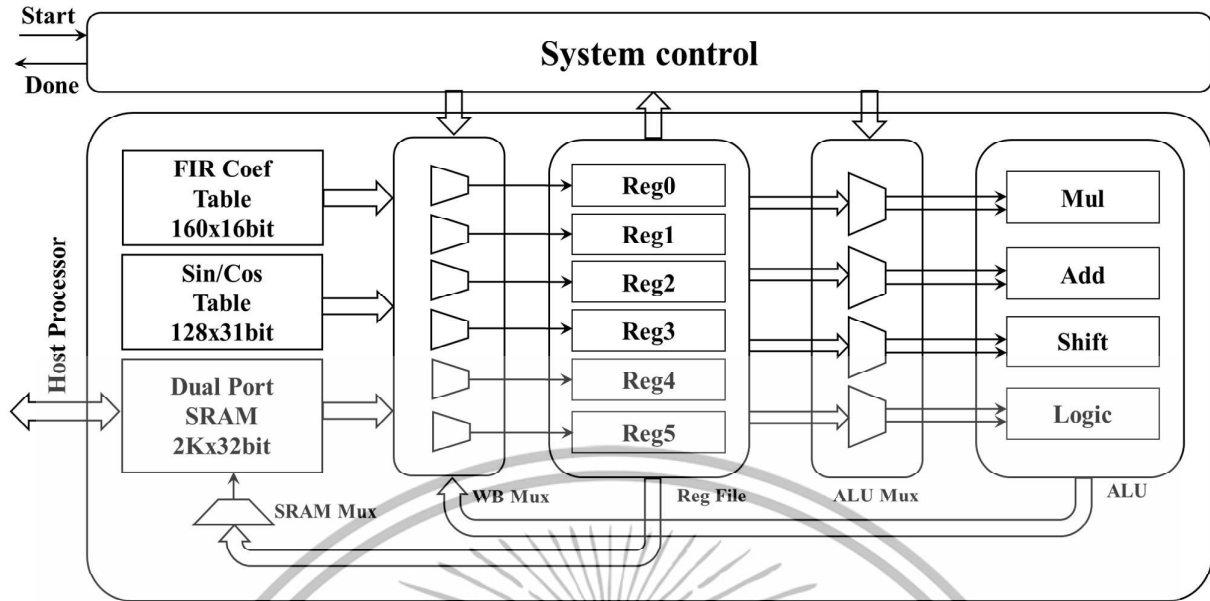


Fig.5 Architecture of proposed Codec2 speech decoder.

For the number division ( $x/y$ ), the Goldschmidt division algorithm is deployed [12]. With some modifications, this algorithm can be applied to the exponential ( $2^x$ ) and logarithm ( $\log_2 x$ ) as well. Fig.4 demonstrates the flowcharts of three similar functions. Three Tables,  $t_0[n]$ ,  $t_1[n]$  and  $t_2[n]$ , are prepared where  $n$  is the number of fraction resolution, for example,  $n=12$  for a 12-bit fraction in fixed-point representation. This iteration method requires only three small tables, the multiplier, and the adder, so the overhead hardware is very small with a few state machines and counters added. After whole functions were analyzed, in the next step, the proposed ASIC architecture will be stated.

### III. HARDWARE ARCHITECTURE

To enhance the parallelism and pipelining of speech signal processing, the very long instruction word (VLIW) architecture was considered. The main concept is to allow multiple data processing and memory load/store at the same time. The sophisticated data flows were determined carefully for each operation in a system control block including FFT, LSP to LPC conversion, complex number functions and so on. The proposed system structure is demonstrated in Fig.5.

Our hardware is a 32-bit fixed-point ASIC with 20-bit integer and 12-bit fraction point. The register file has six of 32-bit general purposed registers employed to store intermediate data during signal processing. The ALU consists of a 32x32 bit multiplier, a 32-bit adder, a barrel shifter, and bitwise logic operations. Moreover, 2 Kword of 32-bit dual-port SRAM is attached to memorized input LSP coefficients, data variables, and output speech samples. The reason for having the dual-port SRAM is to enhance the capability of data load and store at the same time which reduce the issue of memory access bottleneck. In addition, 128x31bit Sine and Cosine table are provided for 512-point FFT and other trigonometry functions. Moreover, 160x16bit FIR coefficient table is also placed for sample overlap addition process at the last stage.

parallel independently and connecting all dataflow in the pipeline seamlessly as much as possible. For example, during input data from SRAM are loaded to specific registers for the next cycle operation, data in other registers can be multiplied and added in parallel, while the output from the previous cycle is being stored to SRAM. This mechanism gives very efficient hardware utilization, whereas the whole structure is still simple. For instance, 512-point FFT requires only six cycles for one butterfly operation including data load and store as shown in Fig.3. The total FFT process consumes  $6 \times 256 \times 9 = 13,824$  clocks plus 7 clocks pipeline overhead. This is the fastest time for single adder and multiplier can archive. A branch operation was also implemented for an if-else decision as same as in a microprocessor.

For the host interface, the CPU can treat this block as a memory slave. The control signals are as same as general memory interfaces with address, data in, write enable and data out signals. Only one additional signal is necessary for starting operation and CPU will get the interrupt signal when one frame decoder is finished.

### IV. PERFORMANCE AND IMPLEMENTATION RESULTS

TABLE 1 Number of clock cycles for one frame decode at 1200 bps

Processing	Proposed ASIC (cycles)	Cortex-M4 [13] (cycles)
LSP to LPC	398	5,328
FFT	13,831	304,622
Harmonic recover	3,766	70,739
Phase synthesis	3,040	14,452
Post filter	1,797	14,486
IFFT	13,831	337,931
Overlap add / FIR	248	4,650
Total	36,911	752,208

The decoding performance was measured and compared to the same processing running on ARM Cortex-M4 processor. Test sequences at 1,200 bps were executed, and the results showed that proposed ASIC could archive around 20 times faster than software running on the CPU. The processing speed of FFT/IFFT, frame overlap addition, and FIR operations resolved a lot of memory load and store redundancy. The experiment revealed that the maximum error from Trigonometry functions calculated by the interpolation method was only  $2^{-12}$  for 12-bit fraction point. Table 1 shows the number of clock cycles for one frame decoder. Since the voice packet consists of 100 frames of samples in a second, the real-time decoder can be archived by only 4 MHz clock speed. It is noticeable that our dedicated hardware can reduce a lot of workload from the host CPU.

The proposed architecture was implemented on Xilinx Artix-7 XC7A100T. Table 2 illustrates the size of circuits with around 5% of look-up table cells and only 627 flip-flops. In addition, 5 DSP slices and 72,064 bits or 9,008 bytes of block RAM are employed. The logic synthesis to place and route results showed that the maximum speed of clock frequency can be up to around 84 MHz. It is much more than 4 MHz requirement for real-time processing. Fig.6 depicts a floorplan of the targeted FPGA.

TABLE 2 FPGA hardware utilization and speed performance.

Hardware Utilization			
	Used	Available	Util %
LUT	3,158	63,400	4.98%
Flip Flop	627	126,800	0.49%
DSPs	5	240	2.08%
Memory	72,064 bit	4,976,640 bit	1.45%
Speed Performance			
Max clk freq	84.23 MHz		

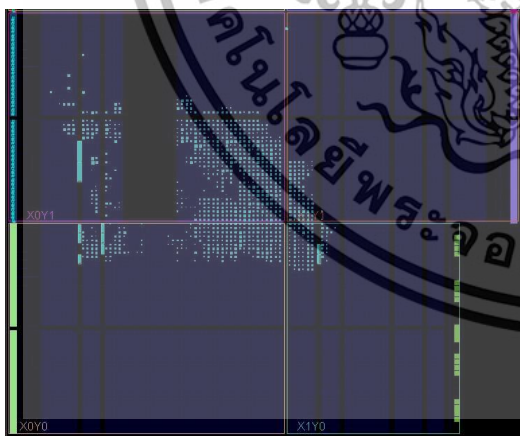


Fig.6 Artix-7 XC7A100T floorplan for proposed Codec2 Decoder

## V. CONCLUSION

The efficient hardware architecture of Codec2 speech decoder was developed by optimizing various dataflow of each processing step. VLIW structure was selected to enhance the signal processing performance and decrease the bottleneck of memory access from conventional CPUs. Many hardware parallelisms and pipelining schedule were carefully organized to handle whole hardware resources with high utilization as much as possible. The results showed that the real-time decoding processing of 1,200 bps speech data can be executed with 4 MHz clock speed or 20 times faster than the Cortex-M4 CPU. In the next step, it is challenged to enhance the capability of this architecture to an application specific microprocessor for general speech vocoder, which gives the high performance, small footprint, yet higher flexibility for other standards such as CELP or MELP as well.

## ACKNOWLEDGMENT

This work was supported by the King Mongkut's Institute of Technology Ladkrabang (KMUTL), Thailand under Grant KREF186103.

## REFERENCES

- [1] J. Srinonchat, "New Technique to Reduce Bit Rate of LPC-10 Speech Coder," IEEE Region 10 Conference, Hong Kong, November 2006, pp.1-4
- [2] W.C. Chu, "Speech Coding Algorithms Foundation and Evolution of Standardized Coders," Wiley-Interscience, 2003
- [3] M. Sadek, S.E. Ramly, A. Tawfik, "A new CELP speech coder for wireless communication," 14th International Conference on Digital Signal Processing Proceedings, Greece, Vol.2, pp.997-1000
- [4] C. Wu, H. Jiang, B. Li, "An Improved MELP Speech Coder," International Conference on Information Technology and Computer Science, Ukraine, August 2009, Vol.2, pp.130-133
- [5] D. Rowetel, "Techniques for Harmonic Sinusoidal Coding," Ph.D. Thesis, 1997
- [6] D. Rowetel, "Codec2", [http://www.rowetel.com/?page\\_id=452](http://www.rowetel.com/?page_id=452)
- [7] J.E. Shuda, A.J. Rix, M.J. Booyesen, "Module-Level Monitoring Of Solar PV Plants Using Wireless Sensor Networks," 26th Southern African Universities Power and Engineering Conference, South Africa, January 2018, pp.1-7
- [8] T.T. Teo, E.C. Tan, "Implementation of 2400 bps MELP vocoder on TMS320C44," Fourth International Conference on Signal Processing, China, October 1998, pp.576-579
- [9] P.C.S Abhijna, N.R. Sangeetha, J.R. Sagar, R. Rahul, G. Gupta, "Implementation of CELP encoder using Vivado HLS," International Conference on Recent Trends in Electronics, Information & Communication Technology, India, May 2017, pp.1443-1447
- [10] S. Shivanagi, M. Koushik, J. Qumar, G. Gupta, D. Saravanan, "MELP Decoder Implementation on Zynq Kit by High Level Synthesis," International Conference on Current Trends in Computer, Electrical Electronics and Communication, India, September 2017, pp.78-82
- [11] Koushik M, S. Shivanagi, J. Qumar, J. Yadav, D. Saravanan, "Implementation of MELP Encoder on Zynq FPGA using HLS," International Conference on Current Trends in Computer, Electrical, Electronics and Communication, India, September 2017, pp.87-91
- [12] R.E. Goldschmidt, "Applications of Division by Convergence," MSc dissertation, Massachusetts Inst. of Technology, June 1964
- [13] D. Rowetel, "Codec2 Speech Coding Reference C Source Code," <https://svn.code.sf.net/p/freetel/code/codec2/branches>

## ภาคผนวก

### ภาคผนวก ข

สรุปค่าใช้จ่ายในการดำเนิน โครงการวิจัย

- แบบรายงานการใช้จ่ายเงิน



บันทึกการยกย่องรับ-จ่ายเงิน โครงการวิจัย สัญญาเลขที่ ..KREF186103..... ตั้งแต่วันที่ ..15 กรกฎาคม 2561..... ถึงวันที่ ...14 มกราคม 2563.....

แหล่งทุน: ทุนวิจัยเริ่มต้นสำหรับอาจารย์ใหม่ กองทุนวิจัยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อโครงการ: การออกแบบวงจรรวมเดอเดรทส์สัญญาณเสียงแบบปรับค่าสำหรับโครงข่ายอินเทอร์เน็ตของสรรพสิ่ง

ชื่อหัวหน้าโครงการ นายสุเมธ วิทยัทธชิน

ว/ด/ป	รายการ	เลขที่อ้างอิง	รายการรับ - จ่าย		รายรับ	รายจ่าย	งบดำเนินงาน				งบลงทุน	รวม	
			รับ	จ่าย			ค่าใช้สอย	ค่าวัสดุ	ค่าสาธารณูปโภค	ค่าครุภัณฑ์			
16 ก.ค. 61	งบประมาณที่ได้รับอนุมัติ (ตามแผน)		100,000.00										
	จำนวนเงินที่ได้รับ (งวดที่ 1)		98,750.00										
	จำนวนเงินที่ได้รับ (งวดที่ 2)		1,250.00										
	จำนวนเงินที่ได้รับ (งวดที่ 3)												
	หัก ค่าใช้จ่าย (ครั้งที่ 1)								100,256.64				100,256.64
	ค่าใช้จ่าย (ครั้งที่ 2)												
	ค่าใช้จ่าย (ครั้งที่ 3)												
	ค่าใช้จ่าย (ครั้งที่ 4)												
	งบประมาณคงเหลือ		100,000.00		0.00								
	<b>รายละเอียดค่าใช้จ่าย</b>												
<b>ครั้งที่ 1</b>		(เลขที่ใบเสร็จรับเงิน)											
12 ก.ย. 61	บอร์ด ASUS Thinker	I-1809120005							4,284.50				4,284.50
12 ก.ย. 61	หน่วยความจำคอมพิวเตอร์	JCP6109350							2,700.00				2,700.00
16 ก.ย. 61	ฮาร์ดดิสก์	302003							4,680.00				4,680.00
16 ก.ย. 61	โหนดรับรีโมเตอร์	IV1802852							4,990.00				4,990.00
22 ก.ย. 61	อุปกรณ์สำนักงาน	11183106004064							542.00				542.00
25 ก.ย. 61	บอร์ด TE0725-03-100-2C FPGA	DEB13408402							34,170.24				34,170.24
4 ม.ค. 62	เซ็นเซอร์และบอร์ด ESP-01S	IV6201040117							1,295.00				1,295.00
4 ม.ค. 62	อุปกรณ์ปฏิบัติการอิเล็กทรอนิกส์	12970							2,054.67				2,054.67
12 ม.ค. 62	อุปกรณ์คอมพิวเตอร์	57-F01-44941							1,757.00				1,757.00
31 ม.ค. 62	ไอซี MCP4922	836-0043							967.12				967.12

เอกสารนี้เป็นเอกสารที่สงวนไว้ใช้ภายในของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บันทึกการจ่ายรับ-จ่ายเงิน โครงการวิจัย สัญญาเลขที่ ..KREF186103..... ตั้งแต่วันที่ .....15 กรกฎาคม 2561..... ถึงวันที่ ...14 มกราคม 2563.....

แหล่งทุน: ทุนวิจัยเริ่มต้นสำหรับอาจารย์ใหม่ กองทุนวิจัยสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ชื่อโครงการ : การออกแบบวงจรรวมแอมพลิฟายเออร์ที่สูญเสียแบบปรับที่ตามกำลังสำหรับโครงข่ายอินเทอร์เน็ตของสรรพสิ่ง  
 ชื่อหัวหน้าโครงการ นายสุเมธ วิทยัทธชัย

ว/ศ/ป	รายการ	เลขที่อ้างอิง	รายการรับ - จ่าย		รายรับ ดอกเบี้ยรับ	รายจ่าย งบบุคลากร	รายจ่าย		รวม รายจ่าย
			รับ	จ่าย			งบดำเนินงาน	งบลงทุน	
31 ม.ค. 62	อุปกรณ์ปฏิบัติการอิเล็กทรอนิกส์	15920					1,306.80		1,306.80
24 เม.ย. 62	อุปกรณ์ปฏิบัติการอิเล็กทรอนิกส์	17942					633.00		633.00
27 เม.ย. 62	อุปกรณ์ปฏิบัติการอิเล็กทรอนิกส์	845-0368					263.90		263.90
31 ส.ค. 62	บอร์ด TE0712-02-200-2C FPGA	DE813408402					14,902.41		14,902.41
31 ส.ค. 62	โมดูล LoRa sX1928	IV6208310006					520.00		520.00
18 กย. 62	หน่วยความจำคอมพิวเตอร์	JCP6207836					1,290.00		1,290.00
20 กย. 62	อุปกรณ์คอมพิวเตอร์	319660					11,960.00		11,960.00
20 กย. 62	อุปกรณ์คอมพิวเตอร์	319659					11,940.00		11,940.00
	<b>รวมทั้งสิ้น 1</b>						100,256.64	-	100,256.64

ลงชื่อหัวหน้าโครงการ .....

หมายเหตุ: ให้แนบสำเนาสมุดบัญชีที่รับเข้ามาพร้อมกับรายงานการดำเนินงานด้วยทุกครั้ง

## ข้อมูลประวัติคณะผู้วิจัย

### ประวัติส่วนตัว

ชื่อ-สกุล นายสุเมธ วิชาทักษิณ

ตำแหน่งปัจจุบัน อาจารย์

### ประวัติการศึกษา

ชื่อย่อปริญญา	สาขา	สถาบันที่จบ	ปีที่จบ
Doctor of Philosophy	Communications and Integrated Systems	Tokyo Institute of Technology, Japan	2010
Master of Engineering	Communications and Integrated Systems	Tokyo Institute of Technology, Japan	2005
Bachelor of Engineering (1 <sup>st</sup> Honor)	Electronics Engineering	King Mongkut's Institute of Technology Ladkrabang, Thailand	2002

สาขาวิจัยที่มีความชำนาญพิเศษ (แตกต่างจากวุฒิการศึกษา)  
การออกแบบวงจรสำหรับปัญญาประดิษฐ์ขนาดเล็กบนระบบฝังตัว

รางวัลด้านวิชาการ/ด้านวิจัย/งานสร้างสรรค์ (ด้านศิลปะ หรืออื่นๆ) ที่ได้รับ

ปี พ.ศ.	ชื่อรางวัล	สถาบันที่ให้

ทุนการศึกษาและทุนวิจัยที่เคยได้รับ

ปี พ.ศ.	ทุนการศึกษาและทุนวิจัย	สถาบันที่ให้
2002	ทุนรัฐบาลญี่ปุ่นศึกษาต่อปริญญาโทที่ประเทศญี่ปุ่น	รัฐบาลประเทศญี่ปุ่น
2005	ทุนรัฐบาลญี่ปุ่นศึกษาต่อปริญญาโทที่ประเทศญี่ปุ่น	รัฐบาลประเทศญี่ปุ่น

## ผลงานวิจัย/งานสร้างสรรค์

### ผลงานวิจัย/งานสร้างสรรค์ที่ตีพิมพ์เผยแพร่ (ระดับชาติและนานาชาติ)

- [1] **Sumek Wisayataksin**, Dongju Li, Tsuyoshi Isshiki, Hiroaki Kunieda: “Low Cost SoC Design of H.264/AVC Decoder for Handheld Video Player”. IEICE Transactions 91-A(4): 1197-1205 (2008)
- [2] **Sumek Wisayataksin**, Dongju Li, Tsuyoshi Isshiki, Hiroaki Kunieda : “Entropy Decoding Processor for Modern Multimedia Applications”. IEICE Transactions 92-A(12): 3248-3257 (2009)

### การเสนอผลงานวิชาการ

- [1] Trio Adiono, Dani Fitriyanto, Akhmad Mulyanto, **Sumek Wisayataksin**, Kazumasa Takeichi, Dongju Li, Tati Rajab Mengko, Hiroaki Kunieda: “New Macroblock Engine Architecture for Video Processing”. MVA 2005: 68-71
- [2] **Sumek Wisayataksin**, Dongju Li, Tsuyoshi Isshiki, Hiroaki Kunieda: “H.264/AVC decoder SoC towards the low cost mobile video player”. SoCC 2007: 151-154
- [3] Yossiri Ariyakul, **Sumek Wisayataksin**, “Data Modulation Technique Using Concentration of Odor Molecules”, Proceeding of IEEE Sensors Conference 2018, 28-31 October 2018, New Delhi, India.
- [4] **Sumek Wisayataksin**, “An Efficient Hardware Architecture of Codec2 Low Bitrate Speech Decoder”, The 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST2019), 2-5 July 2019, Luang Prabang, Lao PDR.
- [5] **Sumek Wisayataksin**, Geranun Boonyuu, “A Programmable Artificial Neural Network Coprocessor for Handwritten Digit Recognition”, The 2nd International Conference on Information and Communications Technology (ICOIACT2019), 24-25 July 2019, Yogyakarta, Indonesia
- [6] **Sumek Wisayataksin**, Panupong Angkasuwan and Yossiri Ariyakul, “4-ary Odor-Shift Keying Using Multi-channel Olfactory Display”, The 11<sup>th</sup> International Conference on Information Technology and Electrical Engineering (ICITEE2019), 10-11 October 2019, Pattaya, Thailand