

โครงการออกแบบวงจรรวม Digital

CU Thai Talk

ผู้ทำโครงการ

นาย อาทิตย์	ธรรมตระการ
นาย อธิวุฒิ	เหลืองนทีเทพ
นางสาว นันทธีรา	อนันตรศิริชัย

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อ

วงจรรวม CU Thai Talk ที่ออกแบบไว้ในรายงานฉบับนี้ จะเป็นวงจรมอนิเตอร์เสียง และสามารถเล่นเสียงที่บันทึกไว้ได้ โดยจะมีวิธีการบีบอัดข้อมูลเสียงด้วยวิธีการมอดูเลตผลต่าง แบบปรับขนาดขั้นได้ (Adaptive Delta Modulation) โดยรับสัญญาณเสียงจากไมโครโฟนเป็นสัญญาณ Analog แล้วแปลงเป็นสัญญาณ Digital 8 บิต ด้วยตัวแปลงสัญญาณแอนะล็อกเป็นดิจิทัลเบอร์ 0820 จากนั้นนำไปบีบอัดด้วยวิธี Adaptive Delta Modulation (ADM) ให้เหลือสัญญาณดิจิทัล 1 บิตก่อนเก็บข้อมูลที่ได้ใน Static Ram เบอร์ HM628128B ซึ่งเป็น SRAM ขนาด 128 kBytes x 8 bits และสามารถนำข้อมูลที่เก็บไว้ออกมาฟังได้ โดยนำสัญญาณที่เก็บไว้มาทีละ 1 บิตแล้วมาปรับกับสัญญาณ Reference เป็นการ Demodulate แล้วจะได้สัญญาณเสียง 8 บิตกลับมา จากนั้นก็ส่งสัญญาณดิจิทัลผ่านตัวแปลงสัญญาณดิจิทัลเป็นแอนะล็อก (DAC) พร้อมกับขยายสัญญาณให้มีขนาดใหญ่พอที่จะฟังได้ยินเมื่อออกจากลำโพง โดย DAC จะใช้เป็น R/2R Ladder ต่อกับ Non-Inverting Amplifier แล้วจึงส่งผ่านไปยัง Low pass filter 4 kHz เพื่อให้ได้สัญญาณเสียงเดิมกลับมาก่อนออกไปยังลำโพง

บทนำ

การใช้งานวงจรดิจิทัลในปัจจุบัน เป็นที่แพร่หลายอย่างมาก และสำหรับในโครงการนี้เป็น การรวมเอาอุปกรณ์ดิจิทัล มาสังเคราะห์รวมกันเป็นวงจรบันทึกเสียง ซึ่งต้องใช้อุปกรณ์จำนวนมาก แต่ในโครงการนี้ นำเอาอุปกรณ์ต่างๆ ที่จำเป็นมาสังเคราะห์รวมกันอยู่ในวงจรรวม (Integrated Circuits) ตัวเดียว โดยอาศัย Static RAM เป็น Storage และเพื่อเพิ่ม Storage ให้มีประสิทธิภาพ ในการจัดเก็บข้อมูลได้มากขึ้น จึงได้นำเอาวิธีการบีบอัดข้อมูล ซึ่งย่อข้อมูลได้อย่างต่ำ 8 เท่า เรียก เทคนิคนี้ว่า Adaptive Delta Modulation (ADM) ซึ่งจะได้กล่าวไว้ในบทต่อไป

กิจกรรมการออกแบบวงจรรวมประเภทดิจิทัลนี้ แม้จะยังไม่เป็นที่แพร่หลายในประเทศไทย แต่ก็มีหลายหน่วยงานและหลายองค์กร ที่ได้พยายามผลักดันเรื่องนี้ให้ประเทศไทยมีศักยภาพใน การแข่งขันในตลาดโลก ซึ่งคาดว่าอีกไม่นานประเทศไทยน่าจะเป็น 1 ในแหล่งผลิตวงจรรวม โดย นักออกแบบเป็นคนไทยได้ หากมีการผลักดันอย่างต่อเนื่อง การทำโครงการนี้เป็นการออกแบบวง จรบันทึกเสียง ซึ่งช่วยเพิ่มศักยภาพ และฝึกทักษะของผู้ออกแบบให้มีความรู้ความชำนาญ เตรียม พร้อมให้เป็นแรงงานของประเทศในอุตสาหกรรมออกแบบวงจรรวมที่มีแนวโน้มจะเติบโต เช่น ประเทศไทย ต่อไป

สารบัญเรื่อง

บทที่ 1 เป้าหมาย แผนการดำเนินงาน และขั้นตอนการพัฒนา

- 1.1 วัตถุประสงค์ และเป้าหมาย
- 1.2 แผนการดำเนินงาน

บทที่ 2 ทฤษฎี และหลักการของ CU Thai Talk

- 2.1 พื้นฐานการบันทึก และเล่นเสียง
- 2.2 ทฤษฎี Adaptive Delta Modulation (ADM)

บทที่ 3 เทคนิคที่ใช้พัฒนา

- 3.1 การนำ ADM มาใช้

บทที่ 4 โครงสร้างการออกแบบวงจรรวม CU Thai Talk

- 4.1 ภาพรวมภายในของ CU Thai Talk
- 4.2 รายละเอียดส่วน Record
- 4.3 รายละเอียดส่วน Play
- 4.4 รายละเอียดส่วน Rew
- 4.5 รายละเอียดตัวหารความถี่
- 4.6 รายละเอียดตัวควบคุมการหยุดเล่น หรืออัดเสียง
- 4.7 รายละเอียดของ Counter กำหนดตำแหน่ง RAM
- 4.8 รายละเอียดของวงจรบวก และวงจรลบ

บทที่ 5 การออกแบบ Board ใช้งาน CU Thai Talk

- 5.1 วงจร Non-Inverting Amplifier
- 5.2 วงจร R/2R Ladder
- 5.3 ภาพรวมของ Board ทั้งหมด
- 5.4 ข้อกำหนด และขั้นตอนการทำงานของ Board

บทที่ 6 ขอบเขตและข้อกำหนดการออกแบบ

บทที่ 7 ผลการจำลอง และสังเคราะห์การทำงานภายใน FPGA

7.1 ผลการจำลองการทำงานของกรอกแบบ

7.2 ผลการสังเคราะห์การทำงานของกรอกแบบ

7.3 ผลการทำงานของกรอกแบบ

บทที่ 8 บทส่งท้าย

8.1 ปัญหา และอุปสรรค

8.2 แนวทางการพัฒนา และประยุกต์ร่วมกับงานอื่นๆ

8.3 ข้อเสนอแนะ

เอกสารอ้างอิง

บทที่ 1

เป้าหมาย แผนการดำเนินงาน และขั้นตอนการพัฒนา

1.1 วัตถุประสงค์และเป้าหมาย

1. เพื่อเป็นการตอบสนองแนวคิดที่ต้องการกระตุ้นให้เกิดการสร้างเทคโนโลยีของประเทศไทยขึ้นมาเอง
2. เพื่อนำความรู้ที่ได้จากวิชาที่เรียน ไปปฏิบัติเพื่อให้เกิดการรู้จริง
3. เพื่อให้เกิดทักษะในการออกแบบวงจรรวมทางด้าน Digital รวมถึงเรียนรู้การสังเคราะห์วงจรรวม Digital ด้วยภาษาขั้นสูง เช่น VHDL หรือ Verilog
4. เป็นจุดเริ่มต้นของการพัฒนาการออกแบบวงจรรวมในประเทศไทย

1.2 แผนการดำเนินงาน และขั้นตอนการพัฒนา

การดำเนินงานจะทำตามแผนดังนี้

1. เลือกเทคนิคที่จะใช้ในการบีบอัดข้อมูล
2. จัดหาอุปกรณ์ เช่น Analog-to-Digital Converter, Digital-to-Analog Converter รวมถึง OpAmp และ Low Pass Filter
3. ออกแบบวงจรรวมดิจิทัลที่ทำหน้าที่เก็บบันทึกเสียง
4. ออกแบบวงจรรวมดิจิทัลที่ทำหน้าที่เล่นเสียง
5. ออกแบบวงจรรวมดิจิทัลที่ทำหน้าที่บีบอัดข้อมูล
6. ออกแบบ และประกอบบอร์ด
7. สังเคราะห์วงจรลงใน FPGA
8. ทดสอบการทำงานทั้งหมด

บทที่ 2

ทฤษฎีและหลักการของ CU Thai Talk

2.1 พื้นฐานการบันทึก และเล่นเสียง

การบันทึกเสียงจะใช้ Counter 17 บิตในการกำหนด Address ของ SRAM ขนาด 128 kBytes x 8 bits ที่ทำหน้าที่เป็น Storage และรับสัญญาณเสียง เป็นสัญญาณ Digital จาก Analog-to-Digital Converter (ADC) แล้วส่งออกทางสาย Data ที่ต่อกับ Static RAM (SRAM)

ส่วนการเล่นเสียง จะใช้ Counter 17 บิตในการกำหนด Address ของ SRAM เช่นเดียวกับขณะบันทึกเสียง แต่จะทำการดึงสัญญาณเสียงที่เก็บไว้จาก SRAM แล้วส่งออกไปเข้า Digital-to-Analog Converter (DAC) แล้วจึงส่งออกมาเป็นเสียง

2.2 ทฤษฎี Adaptive Delta Modulation (ADM)

ทฤษฎี Adaptive Delta Modulation สามารถสรุปมาอธิบายเป็นขั้นตอนเพื่อใช้ในการ Implement จริง ได้ดังนี้

1. กำหนดระดับอ้างอิงมา 1 ระดับ และค่าระดับขั้น 1 ค่า
2. ค่ารับสัญญาณข้อมูลจริงๆ เข้ามา
3. นำระดับสัญญาณที่ได้จากขั้นที่ 1. และขั้นที่ 2. มาเปรียบเทียบกัน
 - 3.1 ถ้าระดับอ้างอิงมีระดับน้อยกว่าระดับสัญญาณจริงๆ ให้ส่งค่าบิต 0 แล้วปรับระดับอ้างอิงเพิ่มขึ้น 1 ระดับขั้น
 - 3.2 ถ้าระดับอ้างอิงมีระดับมากกว่าระดับสัญญาณจริงๆ ให้ส่งค่าบิต 1 แล้วปรับระดับอ้างอิงลดลง 1 ระดับขั้น
4. พิจารณาค่าบิตที่ได้จากขั้นที่ 3 เทียบกับค่าบิตที่ได้จากขั้นที่ 3 ในครั้งก่อน
 - 4.1 ถ้าค่าบิตเป็นค่าเดียวกัน แสดงว่าเกิดปัญหา Slope Overload ให้นำค่าระดับขั้นไปคูณด้วย 2 เพื่อแก้ปัญหา Slope Overload ที่เกิดขึ้น
 - 4.2 ถ้าค่าบิตไม่ใช่ค่าเดียวกัน แสดงว่าไม่เกิด Slope Overload ให้กำหนดค่าระดับขั้นเป็นค่าแรกเริ่มเดียวกับในข้อ 1

จะเห็นได้ว่าสัญญาณที่ผ่านวิธีการ ADM จะมีความคลาดเคลื่อนอยู่ระดับหนึ่ง แต่ก็ชดเชยได้ด้วยการปรับความถี่การสุ่มสัญญาณให้สูงขึ้น ก็จะสามารถแก้ปัญหการเกิด Slope Overload ในความถี่สูงๆ ได้

โดยความถี่เวลาที่สามารถเก็บเสียงได้ ขึ้นอยู่กับ RAM ที่ใช้เก็บเสียง ในที่นี้ใช้ RAM 128kBytes x 8 Bits ถ้าใช้กับ Clock ความถี่ 11.7 kHz ก็จะสามารถเก็บเสียงได้นานถึง $(127 \times 1024 \times 8 + 1)/11700 = 88.9$ วินาที

บทที่ 3 เทคนิคที่ใช้พัฒนา

จากบทที่แล้วได้กล่าวถึงวิธีการของ ADM ที่สามารถนำมาใช้งานได้ ในบทนี้ก็จะกล่าวถึงวิธีการใช้ ADM กับวงจรรวมดิจิทัล CU Thai Talk

3.1 การนำ ADM มาใช้

การใช้ ADM ใน CU Thai Talk ก็จะต้องอิงจาก VHDL Code ที่เขียนคือกำหนดระดับอ้างอิงจากข้อมูลไบต์แรกของ SRAM นั่นคือ Address 0000000000000000 นั่นเอง จากนั้นข้อมูลต่อไปจะเริ่มเก็บทีละ 1 บิต โดยการใช้การเปรียบเทียบสัญญาณตามวิธีการ ADM ที่ได้กล่าวไว้ในบทที่แล้ว

สัญญาณที่ถูก Sampling มาครั้งแรกจะถูกเก็บไว้ที่ Address 0000000000000000 ของ SRAM จากนั้น Sample ต่อไปจะถูกนำมาเปรียบเทียบกับสัญญาณอ้างอิงแล้วให้ค่า 0 หรือ 1 ตามวิธี ADM เมื่อได้ค่าบิตที่ต้องการเก็บมาแล้วก็จะใช้ Counter นับบิต 1 ตัวในการนับขึ้น 1 ครั้ง แล้วใช้ Shift Register ในการเลื่อนบิตเพื่อจะต่อเข้ากับขา I/O ของ SRAM และเมื่อ Sampling ได้ครบ 8 ครั้ง Counter นับบิตจะนับจาก 0000 ถึง 0111 แสดงว่าครบ 8 Sample แล้ว ก็จะมีการส่งค่านี้ไปเก็บใน RAM ที่ตำแหน่งนั้น จากนั้นจะนับ Counter ที่ต่ออยู่กับขา Address ของ RAM เพิ่มขึ้น 1 ค่า เป็นการเลื่อน Address ของ RAM แล้ว Reset Counter นับบิตเป็น 0000

ด้วยวิธีการนี้จะบีบอัดข้อมูลได้ถึง 8 เท่าในทุกกรณี และสำหรับการเล่นเสียงก็จะต้องมีการ Reset Counter ด้วยปุ่ม Rew ของบอร์ดก่อน จากนั้นเมื่อเริ่มเล่นจะอ่านข้อมูลจาก SRAM ที่ตำแหน่ง 0 เป็นค่าอ้างอิงแล้วทำการนับ Counter เพื่อเลื่อนไป Address ถัดไปของ SRAM จากนั้นจะอ่านข้อมูลที่ตำแหน่งถัดไปเข้ามา 8 บิต แล้ว Latch เก็บค่าไว้ แล้วจึงดึงค่าจาก MSB เป็นค่าบิตในการถอดรหัสตามวิธี ADM ที่อธิบายไว้ในบทที่ 2 หลังจากถอดรหัสโดยการเพิ่มหรือลดค่าอ้างอิงแล้ว ก็จะเลื่อนบิตไปทางซ้าย 1 ครั้ง พร้อมกับ Counter นับบิตจะนับขึ้น 1 ครั้ง จะทำให้สามารถรู้ได้ว่าตีความครบ 8 บิตหรือยังจาก Counter นับบิต และเมื่อ Counter นับบิต นับถึง 0111 ก็จะเลื่อน Address ของ RAM ไป 1 ค่า แล้วทำการ Reset Counter นับบิตเป็น 0000 เช่นเดียวกับวิธีบันทึกเสียง

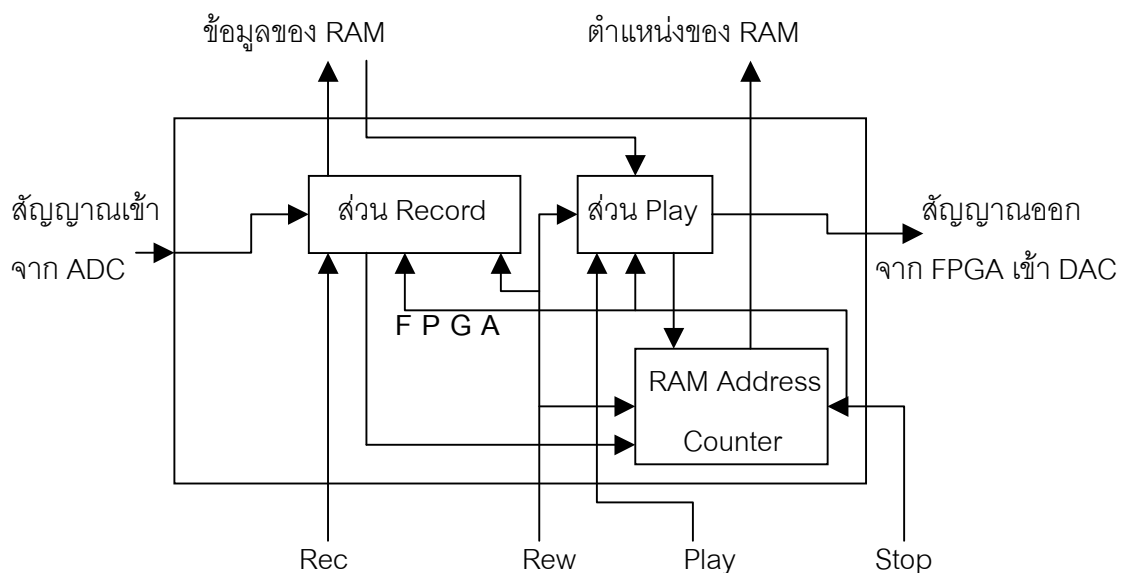
บทที่ 4

โครงสร้างการออกแบบวงจรรวม CU Thai Talk

ในบทที่ผ่านมาได้พูดถึงเพียงหลักการ ทฤษฎีที่ใช้บีบอัด และเทคนิคที่ใช้ในการจัดเก็บข้อมูล แต่บทนี้จะพูดถึงประเด็นหลักของการออกแบบซึ่งจะกล่าวต่อไปตามหัวข้อดังนี้

4.1 ภาพรวมภายในของ CU Thai Talk

ภาพรวมภายในนี้จะอธิบายถึงภาพรวมของวงจรรวมภายใน FPGA โดยจะนำเสนอในรูปแบบของ Block Diagram ดังรูปที่ 4.1 นี้



รูปที่ 4.1 รูปแสดง Block Diagram ภายใน FPGA ของ CU Thai Talk

จากรูปสามารถแจงออกมาได้ว่าวงจรรวมนี้ประกอบด้วย

1. ส่วน Record
2. ส่วน Play
3. สัญญาณ Rewind ซึ่งจะไป Reset Counter ทุกตัวภายใน FPGA
4. ส่วน Frequency Divider ทำหน้าที่หารความถี่ที่รับมาจาก Clk ภายนอก เพื่อใช้เป็น Clk ภายในให้ส่วนต่างๆ ของ FPGA
5. ส่วนควบคุมการหยุดเล่น หรืออัดเสียง และควบคุมส่วนอื่นๆ
6. Counter กำหนดตำแหน่งของ RAM

ส่วนต่างๆ นี้จะอธิบายถึงรายละเอียดต่อไป

4.2 รายละเอียดส่วน Record

จะแยกออกเป็นรายละเอียดการออกแบบ ข้อกำหนด และขั้นตอนการทำงานของส่วนนี้

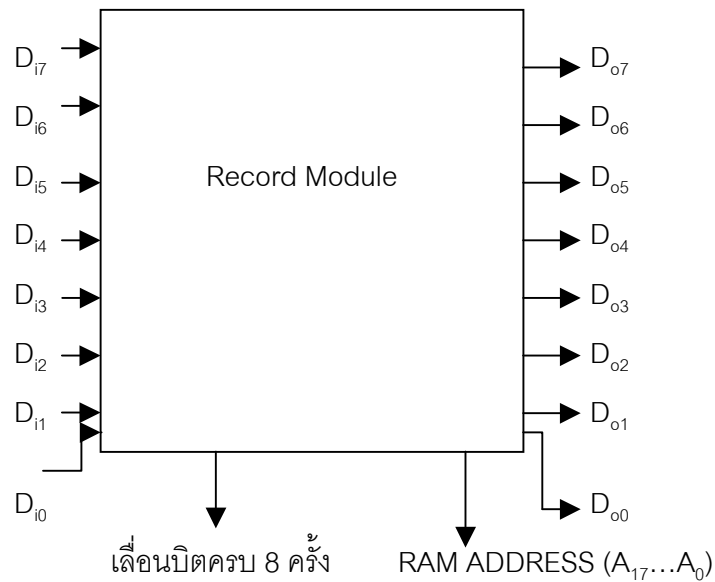
4.2.1 ข้อกำหนดส่วน Record

เริ่มจากการพิจารณาการวิ่งของสัญญาณภายในส่วนนี้ คือสัญญาณเสียง Digital จะวิ่งเข้ามา แล้วมีการพิจารณาว่า Address ของ RAM เป็น 0 หรือไม่ ถ้าเป็น 0 ก็จะนำสัญญาณไปเก็บเลยทั้ง 8 บิต และกำหนดค่าสัญญาณนี้เป็นค่าอ้างอิง แต่ถ้าไม่เป็น 0 จะนำสัญญาณไปเปรียบเทียบกับค่าอ้างอิง แล้วพิจารณาว่ามีค่ามาก หรือน้อยกว่าค่าอ้างอิง ถ้ามากกว่าจะนำค่า 0 ส่งไปเป็น Serial Input ของ Shift Register แต่ถ้าน้อยกว่าค่าอ้างอิงจะนำค่า 1 ส่งไปเป็น Serial Input ของ Shift Register และ Shift Register จะทำการ Shift ไปทางซ้ายจนครบ 8 ครั้ง ก็จะส่งสัญญาณออกไปให้เลื่อนตำแหน่งของหน่วยความจำ

ต่อมาก็พิจารณาว่า RAM และ ADC จำเป็นต้องมี Acknowledge หรือการควบคุมอะไรหรือไม่ ก็พบว่า ADC ไม่ต้องการการควบคุมจาก FPGA แต่ RAM ต้องการการกำหนดให้เป็น Mode เขียนข้อมูล ดังนั้นต้องมีสัญญาณควบคุมคือ Write Enable (WE) และ Output Enable (OE) เพื่อกำหนดโหมดการอ่านและเขียน ซึ่งจาก Specification ของ RAM ต้องส่ง 0 ให้ WE (เพราะขา WE และ OE ของ RAM เป็น Active Low) และส่ง 1 ให้ OE เพื่อกำหนดให้เป็น Write Mode

จากการพิจารณาการวิ่งของสัญญาณภายในจะกำหนดทางเข้าออกของสัญญาณในส่วนนี้ได้ตามรูป 4.2 ซึ่งจะทำให้ได้ข้อกำหนดสำหรับการออกแบบในส่วนนี้ คือ

1. ให้มีการเขียนข้อมูลใน RAM ตลอดเวลา
2. ให้ส่งสัญญาณออกไปเพื่อเลื่อนตำแหน่งของหน่วยความจำ เมื่อรับ Sample ครบ 8 ตัวอย่างแล้ว
3. บิต MSB ของข้อมูลที่ออกไปจะเป็นสัญญาณของ Sample ที่ 1

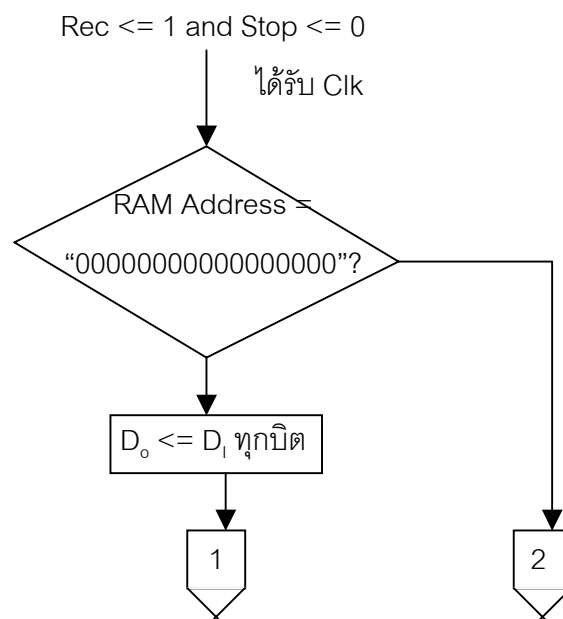


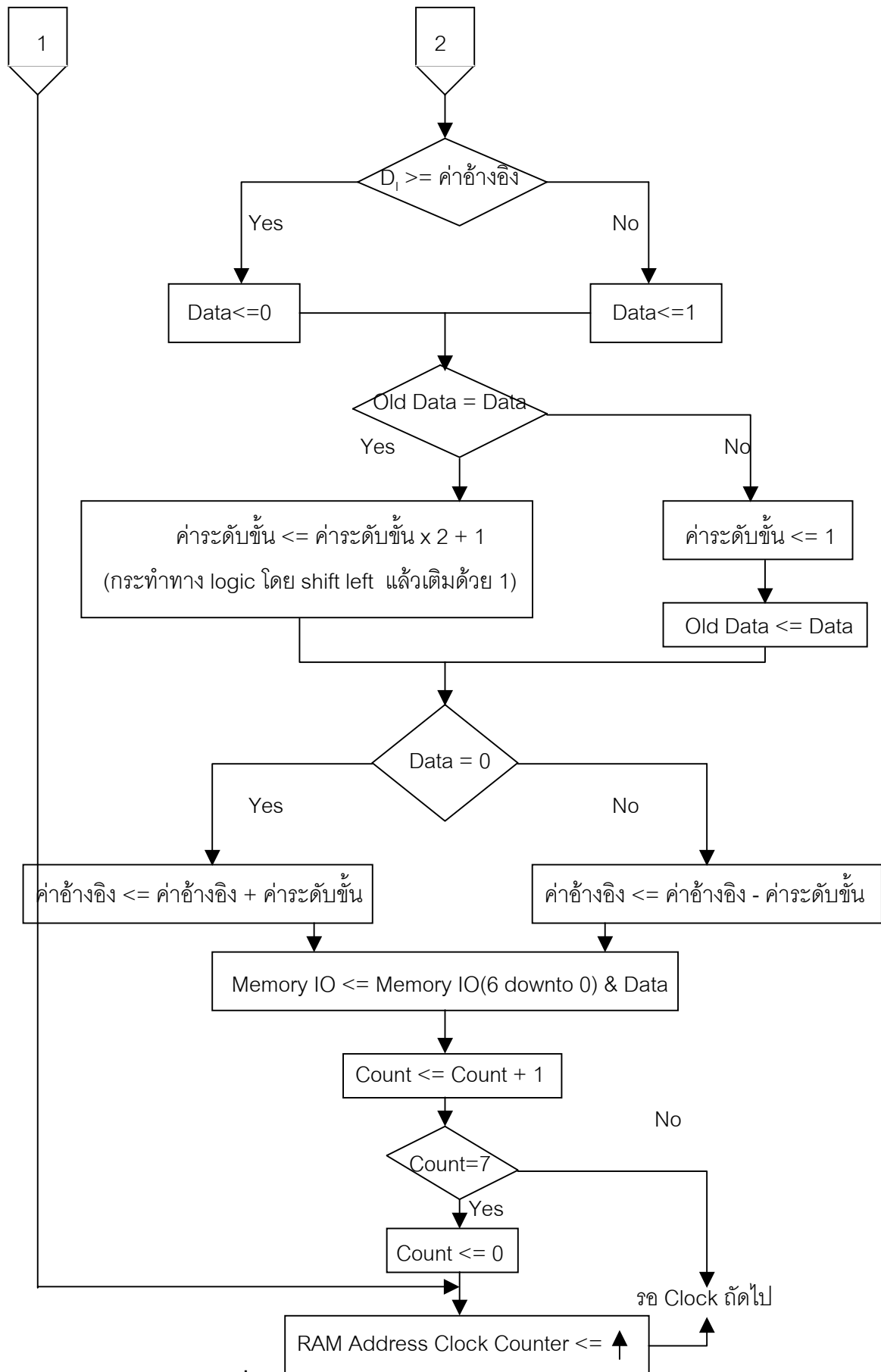
รูปที่ 4.2 ทางเข้าออกของสัญญาณส่วน Record

4.2.2 รายละเอียดการออกแบบ และขั้นตอนการทำงานส่วน Record

จากข้อกำหนดเมื่อต้องการเขียนข้อมูลลงใน RAM ตลอดเวลาจึงส่ง MEMR เป็น 1 และ MEMW เป็น 0 ตลอดเวลา จากข้อกำหนดต่อไปนี้จะต้องมี Counter 1 ตัว อยู่ภายในส่วนนี้ เพื่อทำการนับเมื่อครบ 8 จึงส่งสัญญาณออกไปให้เลื่อนตำแหน่ง Address ของ Memory

นอกจากนี้สัญญาณที่ออกจากส่วนนี้ MSB จะเป็น Sample แรก และ LSB เป็น Sample สุดท้าย จึงต้องเลื่อนบิตไปทางซ้ายทีละ 1 บิต โดยข้อมูลของแต่ละ Sample ได้มาจากการเปรียบเทียบสัญญาณเข้ากับสัญญาณอ้างอิง จะได้ว่า Signal ภายในส่วนนี้มีสัญญาณอ้างอิงด้วย ดังนั้น ในส่วนนี้จะสามารถออกแบบได้ตาม Flow Chart ดังรูปที่ 4.3 นี้





รูปที่ 4.3 Flow Chart แสดงการออกแบบในส่วน Record

ส่วนขั้นตอนการทำงานก็จะเป็นไปตาม Flow Chart ในรูปที่ 4.2.2 โดยจะเริ่มจากการกดปุ่ม Rec ทำให้ขา Rec ของวงจรรวมที่ออกแบบเป็น 1 จากนั้นเมื่อมี Clock เข้ามาในส่วนนี้ก็จะอ่านค่า 8 บิตจากด้านเข้าแล้วเปรียบเทียบกับค่าอ้างอิง ถ้าค่าของสัญญาณเข้ามากกว่าค่าอ้างอิงก็จะเก็บสัญญาณ 0 เข้าไปใน RAM โดยจะเติมไปที่ LSB พร้อมกับเลื่อนข้อมูลที่จะเก็บเข้า RAM ไปทางซ้าย จากนั้นก็ให้ Counter ตัวหนึ่งนับขึ้นทีละ 1 ทุกครั้งที่มีการเปรียบเทียบสัญญาณ และเมื่อ Counter นับจาก 0 ถึง 7 แสดงว่าเก็บครบ 8 Samples แล้วก็ส่งสัญญาณออกไปบอกให้เลื่อน RAM Address พร้อมกับ Reset Counter นับ 8 ด้วย

และในกรณีที่ RAM Address เป็น 0 ตอนที่สัญญาณเข้ามา ให้ยึดค่านั้นเป็นค่าอ้างอิง แล้วไม่ต้องเปรียบเทียบอะไรอีก ให้ใช้ Byte แรกของ RAM เป็นตัวเก็บค่าอ้างอิง

4.3 รายละเอียดส่วน Play

ส่วน Play นี้จะมีหลักการคล้ายกับส่วน Record แต่จะเป็นส่วนกลับกันกับส่วน Record ดังนี้

4.3.1 ข้อกำหนดส่วน Play

จากหลักการที่กล่าวมาข้างต้นจะได้ข้อกำหนดการออกแบบส่วน Play คือ

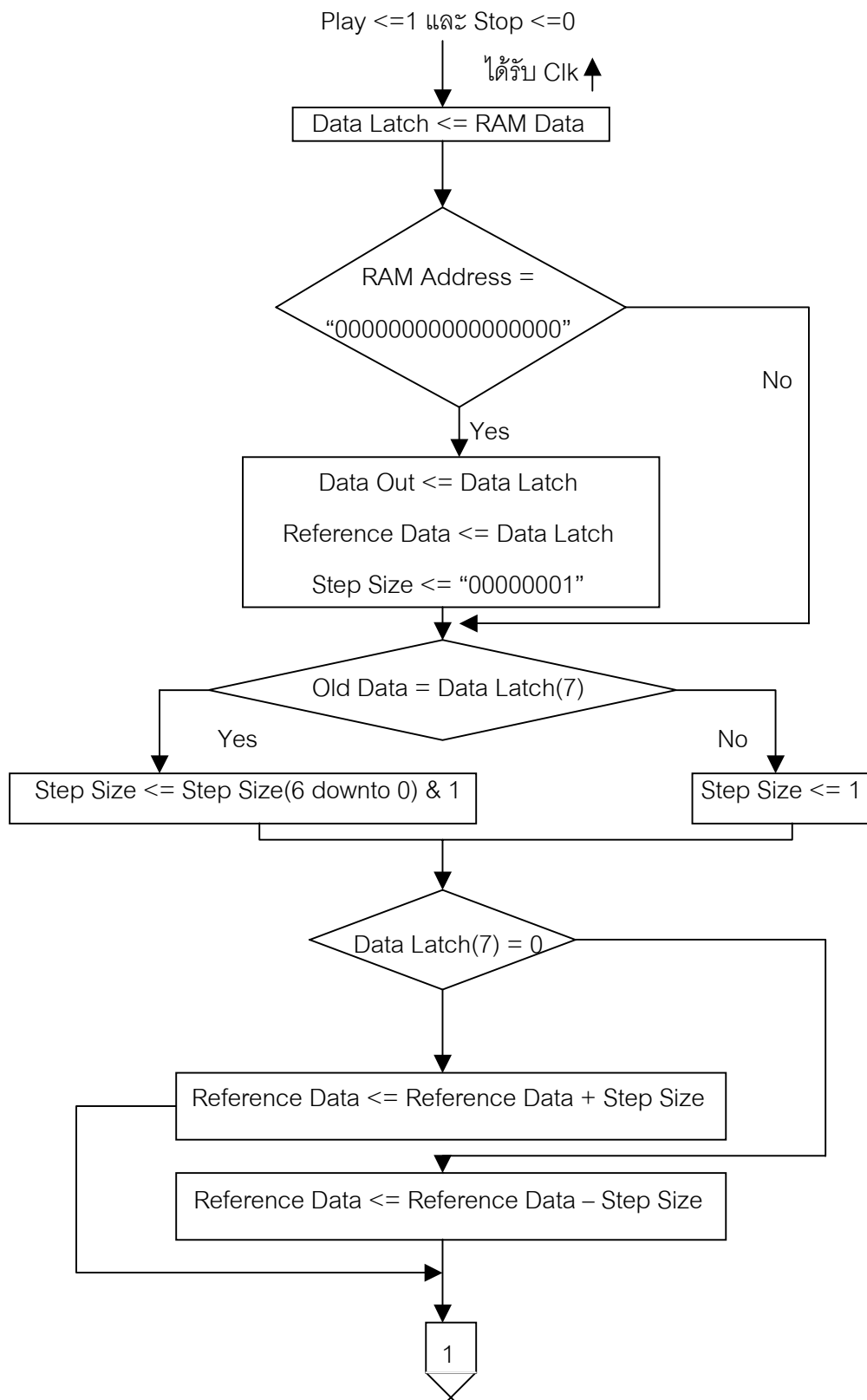
1. อ่านสัญญาณจาก RAM ทีละ 1 บิต โดย MSB เป็น Sample แรกของ Byte นั้น เพื่อให้สามารถ Demodulate สัญญาณจากการ Record ได้อย่างถูกต้อง
2. ให้ Output 8 บิต โดย Demodulate จาก 1 บิต อาศัยการเทียบค่าอ้างอิง
3. ถ้า RAM address เป็น 0 ให้ถือว่าข้อมูลจาก RAM 8 บิตนี้เป็นค่าอ้างอิง

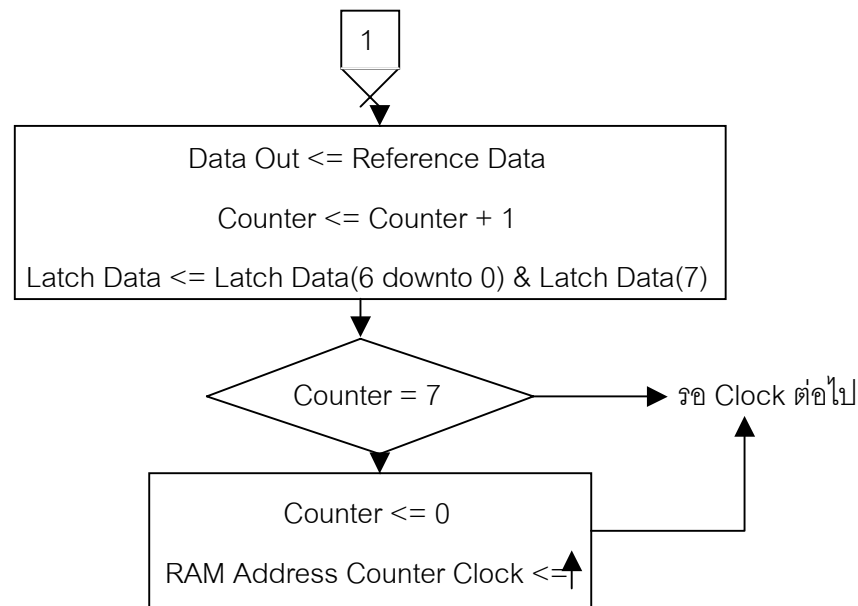
4.3.2 รายละเอียดการออกแบบ และขั้นตอนการทำงานส่วน Play

การอ่านค่าจาก RAM ต้องอ่านเพียงครั้งเดียว แล้ว Latch เก็บไว้ ไม่สามารถทำแบบตอน Record ได้ ทำให้การออกแบบมีข้อแตกต่างกัน โดยจะเริ่มจากอ่านค่าจาก RAM ทันทีที่มี Clock เข้ามาแล้ว Latch ค่านั้นเก็บไว้ จากนั้นอ่านค่าจาก MSB ของสัญญาณที่ได้มาแล้วดูว่าเป็นค่าอะไร

ถ้าเป็นบิต 0 แสดงว่าค่าอ้างอิงมีค่าน้อยเกินไปต้องนำค่าอ้างอิงไปบวกกับสัญญาณระดับขึ้น แต่ถ้าเป็นบิต 1 แสดงว่าค่าอ้างอิงมีค่ามากเกินไป ต้องนำค่าอ้างอิงไปลบกับสัญญาณระดับขึ้น โดยก่อนที่จะทำการบวกหรือลบสัญญาณระดับขึ้นต้องตรวจดูว่าค่าบิตนี้ซ้ำกับบิตก่อนหรือไม่ ถ้าซ้ำก็ให้ปรับค่าสัญญาณระดับขึ้นโดยการ $\times 2 + 1$ ซึ่งอาจใช้เป็นการเลื่อนบิตไปทางซ้ายพร้อมกับใส่ 1 เข้ามาที่ LSB และในขณะเดียวกันจะมีการนับขึ้น 1 ครั้ง เมื่อนับขึ้นครบ 8 ครั้งจะส่งสัญญาณออกไปเพื่อเลื่อนตำแหน่ง RAM ที่จะอ่านข้อมูล แล้วจะทำการ Reset Counter เป็น 0

และในกรณีพิเศษ ที่ RAM Address เป็น 0 ก็ให้ถือว่าค่าที่ได้จาก RAM เป็นค่าอ้างอิงเช่นกันกับวิธี Record จากขั้นตอนการทำงานที่กล่าวมานำมาออกแบบ เป็นลักษณะของ Flow Chart ตามรูปที่ 4.4 เพื่อให้สามารถนำไป Implement ด้วย VHDL ได้สะดวกยิ่งขึ้น





รูปที่ 4.4 Flow Chart แสดงวิธีการทำงาน และการออกแบบส่วน Play

การทำงานเริ่มจากรับข้อมูลจาก RAM Data มา 8 บิต แล้ว Latch เก็บไว้ที่ Data Latch จากนั้นดูว่าเป็น RAM Address เป็น 0 หรือไม่ ถ้าใช่ ก็ให้ Latch Data ส่งออกเป็น Output ทาง Data Out และเก็บไว้เป็นค่าอ้างอิงที่ Reference Data แต่ถ้าไม่ใช่ ก็อ่านค่าจาก MSB ของ Latch Data แล้วดูว่าซ้ำกับข้อมูลก่อนหน้าหรือไม่ ถ้าซ้ำ ให้ shift บิตของ Step Size ไปทางซ้ายและเติม 1 ต่อเข้าไป จะได้ว่า Step Size เพิ่มขึ้น 2 เท่า แต่ถ้าไม่ซ้ำ ให้ Step Size เป็น 1

หลังจากนั้นก็นำเอาค่าอ้างอิง มาบวกหรือ ลบกับ Step Size โดยถ้า Data Latch(7) เป็น 0 ให้บวก และถ้า Data Latch(7) เป็น 1 ให้ลบออก แล้วจึงนำค่าอ้างอิงที่เป็นผลลัพธ์นี้ส่งออกไป DAC ทาง Data Out

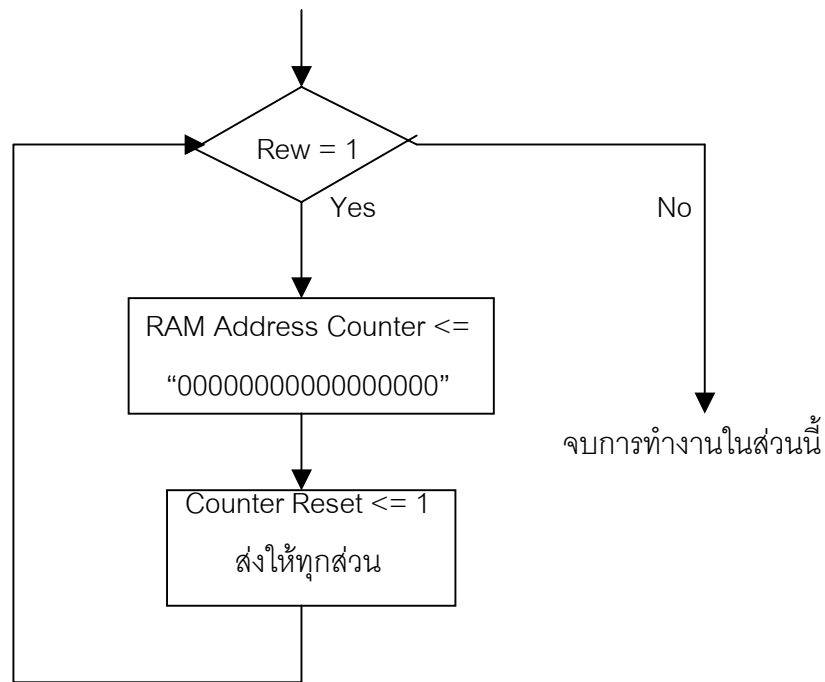
เมื่อ Data Output ถูกส่งออกไปแล้ว Counter ภายในส่วนนี้จะทำการนับค่าขึ้น 1 ครั้ง แล้วเลื่อน Data Latch ไปทางซ้าย 1 บิต เมื่อ Counter นับจาก 0 ถึง 7 ก็จะอ่านวนได้ Data Latch ครบ 8 บิตพอดี จะมีการ Reset Counter และสั่งให้เลื่อน Address RAM เพิ่มอีก 1

4.4 รายละเอียดการออกแบบส่วน Rew

ส่วน Rew นี้จะทำการ Reset1 Counter ทุกตัวใน FPGA ดังนั้นข้อกำหนดของส่วนนี้คือ

1. ยุติการบันทึก หรือเล่น แล้ว Reset RAM Address Counter เป็น 0
2. ส่งสัญญาณให้ Counter ภายในของส่วนอื่นๆ ถูก Reset ด้วย

จึงการออกแบบในส่วนนี้ ง่ายและสั้น สามารถทำตาม Flow Chart ในรูปที่ 4.5



รูปที่ 4.5 Flow Chart แสดงวิธีการทำงาน และการออกแบบส่วน Rew

สำหรับการ Set Counter ให้มีค่าเป็น 0 ทำได้ง่ายเพราะใช้ Model Counter เป็นฟังก์ชัน จึงมีการสามารถกำหนดค่าเป็น 0 ได้ทันที และรวมถึง Counter ภายในส่วนอื่นๆ ก็ใช้ฟังก์ชันเดียวกันเป็นต้นแบบจึงสามารถ Reset Counter ภายในเป็น 0 ได้ทันทีเช่นกัน

4.5 รายละเอียดการออกแบบตัวหารความถี่

ถ้าต้องการหารความถี่ด้วยค่า n วงจรรวมจะรับ Clock หลักจาก Oscillator เข้ามา แล้วส่งให้ Counter นับขึ้นไปเรื่อยๆ จนกระทั่งถึงค่า $n/2$ ก็จะส่งให้ Output มีค่า Invert กับค่าก่อนหน้า ก็จะได้วงจรหารความถี่

วงจรหารความถี่นี้มีเพื่อใช้ส่ง Clock ที่ความถี่ตามต้องการให้ ADC เพื่อกำหนดความถี่การสุ่มสัญญาณ ซึ่งตามข้อกำหนดทางคณะกรรมการ ให้ต่ำสุดเป็น 11.025 kHz (Telephone Quality) ในที่นี้จะใช้ความถี่การสุ่มสัญญาณ 15.6 kHz แต่รับ Clock จาก Oscillator ความถี่ 32 MHz จึงต้องหารความถี่ด้วยค่า 2048

4.6 รายละเอียดการออกแบบตัวควบคุมการหยุดเล่น หรืออัดเสียง

จะรับสัญญาณจากปุ่ม Stop เข้ามา โดยปกติถ้า Stop เป็น 0 จะให้การทำงานตามปกติ แต่เมื่อ Stop เป็น 1 จะทำให้ Counter ทุกตัวหยุดนับ และไม่มีการทำงานใดๆ เกิดขึ้นอีก ดังนั้นถึงแม้จะมีการ Convert สัญญาณจาก ADC หรือ DAC ก็จะไม่มีการควบคุมจาก FPGA ทำให้ทุก

อย่างหยุดนิ่ง แต่ Address ต่างๆ จะยังคงค่าเดิม โดยแสดงการออกแบบจาก If then statement ดังนี้

If Stop = 0 then

ทำงานตามที่ออกแบบไว้

End If;

ซึ่งจะเห็นว่าถ้า Stop=1 ก็จะไม่ทำงานตามที่ออกแบบไว้จนกว่า Stop = 0 เป็นส่วน Control Path ที่ใช้ควบคุม Data Path

4.7 รายละเอียดการออกแบบของ Counter กำหนดตำแหน่ง RAM

ตัว Counter กำหนดตำแหน่ง RAM เป็น Counter 17 บิตใช้กับ RAM ขนาด 128 kBytes การสร้าง Counter ทำโดยการนำค่า RAM Address ปัจจุบัน มาบวกกับ 1 ทุกครั้งที่ได้รับ Clock ก็จะได้เป็น Counter การบวกของ Counter คือนำ LSB มา xor กับ 1 แล้วนำ LSB กับ 1 มา And กันจะได้ Carry สำหรับบิตถัดไป ทำเช่นนี้ไปเรื่อยๆ จนกว่าบิต Carry จะเป็น 0 ก็จะได้ค่า Address ใหม่

4.8 รายละเอียดการออกแบบวงจรบวก และวงจรลบ

วงจรบวก การทำงานบวกจะใช้วิธีในสมการนี้

Input : A และ B

Output : C

$$C_i = A_i \text{ xor } B_i \text{ xor } \text{Carry}_i$$

$$\text{Carry}_i = (A_i \text{ and } B_i) \text{ or } (A_i \text{ and } \text{Carry}_{i-1}) \text{ or } (B_i \text{ and } \text{Carry}_{i-1})$$

โดย $i = 0, 1, 2, 3, \dots, n-1$ สำหรับข้อมูล n บิต

วงจรลบ การทำงานลบจะใช้วงจรบวกช่วยด้วยโดยทำดังนี้

ถ้าต้องการ Output : C ที่เกิดจาก Input เป็น A-B ให้นำ B ทุกบิตไปผ่าน Inverter จะได้ B เป็น 1's complement แล้วนำ B ไปบวกเพิ่ม 1 ในที่นี้อาจใช้วงจรนับค่า B ขึ้น 1 จะได้ B ใหม่เป็น 2's complement แล้วจึงนำ A กับ B ใหม่ ไปเข้าเป็น Input ของวงจรบวก ผลลัพธ์จากวงจรบวกที่ได้จะเป็นค่า A-B

บทที่ 5

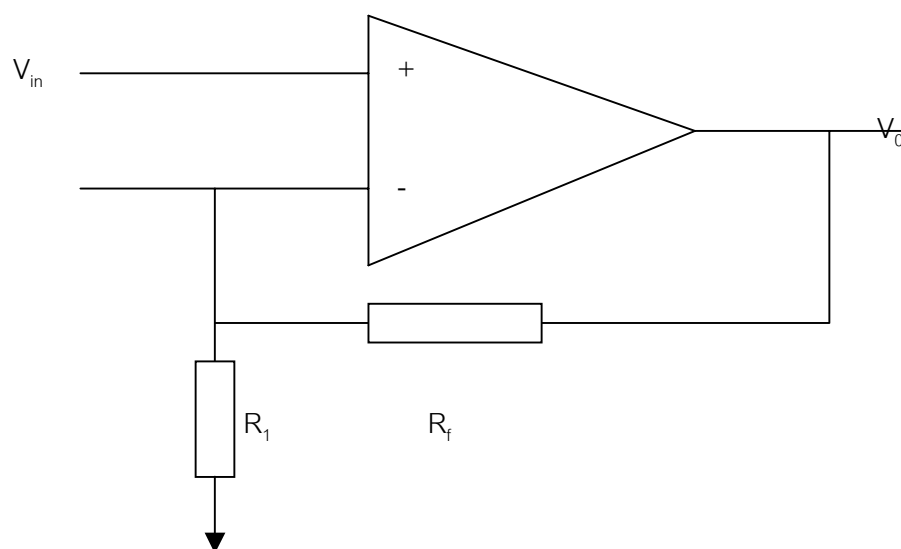
การออกแบบ Board ใช้งาน CU Thai Talk

บอร์ดที่ใช้งาน CU Thai Talk นี้ประกอบไปด้วยส่วนสำคัญๆ คือ

1. ภาครับ และขยายสัญญาณเข้า เป็นสัญญาณ Analog
2. Analog-to-Digital Converter
3. CU Thai Talk (FPGA จาก Xilinx เบอร์ 4010E)
4. Digital-to-Analog Converter และ Amplifier ขยายสัญญาณ Analog

4.1 Non-Inverting Amplifier

จะใช้ OpAmp เบอร์ ICL7611 ของบริษัท MAXIM ในการต่อ Non-Inverting Amplifier ดังรูปที่ 5.1



รูปที่ 5.1 Non-Inverting Amplifier

วิเคราะห์ gain ของวงจรขยาย เนื่องจาก OpAmp มีกระแสไหลเข้าเป็น 0 และแรงดันขั้วบวก และขั้วลบมีค่าประมาณเท่ากัน ดังนั้นได้สมการ Node คือ

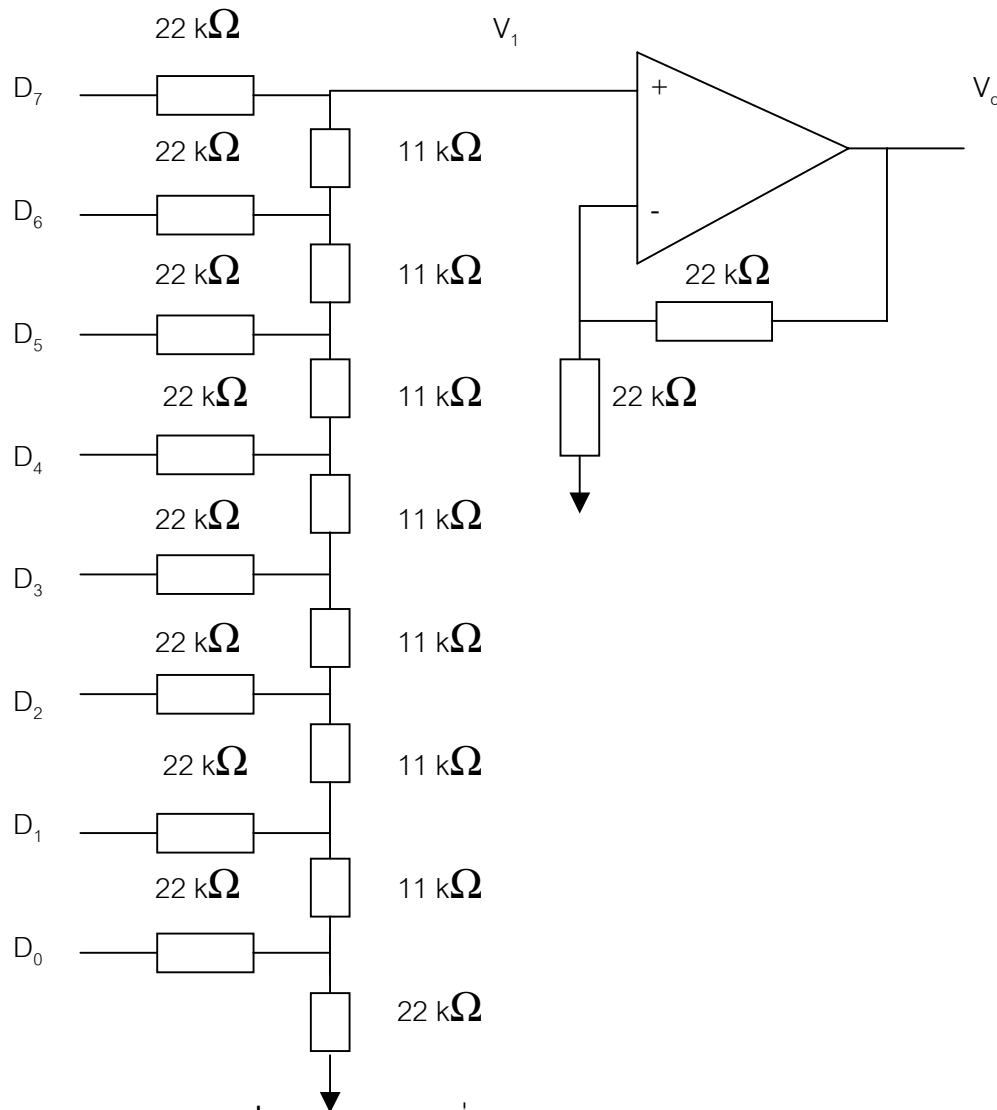
$$(V_o - V_i)/R_f = V_i/R_1$$

ได้

$$A_v = V_o/V_i = (1 + R_f/R_1)$$

5.2 R/2R Ladder

การใช้ วงจร R/2R จะทำหน้าที่เป็น Digital-to-Analog Converter โดยรับสัญญาณแรงดันเป็น Digital และจะได้สัญญาณ Analog เป็นกระแส ต่อ OpAmp แปลงเป็นสัญญาณแรงดันพร้อมกับขยายสัญญาณดังรูปที่ 5.2



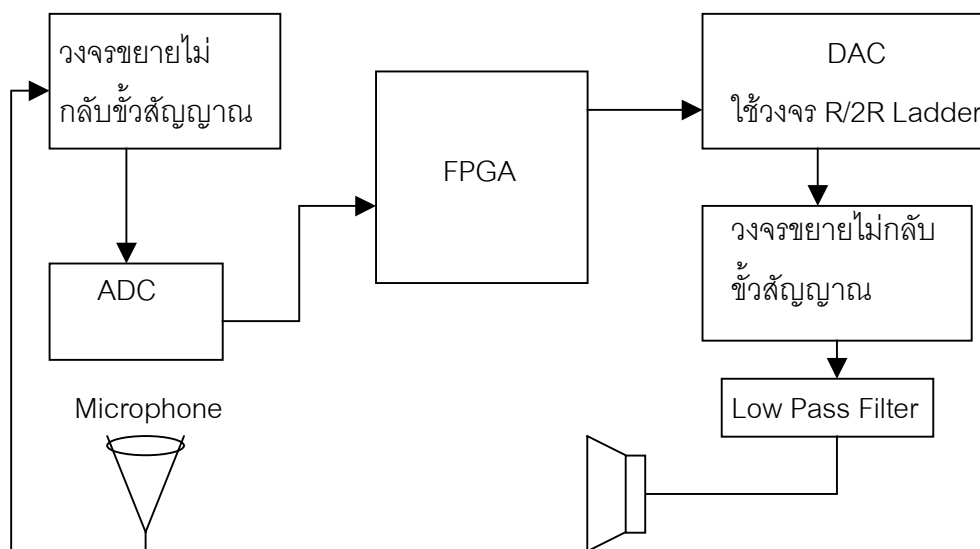
รูปที่ 5.2 วงจร DAC ที่สร้างโดย R/2R Ladder

จากรูปจะได้ V_1 เป็นสัญญาณ Analog ต่อเข้ากับ Non-Inverting Amplifier จะได้ V_0 สำหรับต่อเข้ากับลำโพง

5.3 ภาพรวมของบอร์ดทั้งหมด

จากวงจรพื้นฐานที่ใช้ในบอร์ดตามข้อ 5.1 และ 5.2 รวมถึงใช้อุปกรณ์อื่นๆ เพื่อให้ทำงานได้ตามขั้นตอนการทำงานคือ รับสัญญาณจากไมโครโฟน ส่งเข้า Non-Inverting Amplifier แล้วแปลงเป็นสัญญาณ Analog เป็นสัญญาณ Digital จากนั้นก็ส่งเข้าไปบันทึกเสียงใน FPGA โดย FPGA จะทำการบีบอัดข้อมูลด้วยวิธี ADM แล้วส่งไปยัง SRAM ที่เป็นตัวเก็บ และเมื่อต้องการฟังเสียงที่อัดไว้ก็นำข้อมูลใน SRAM มาคลายการบีบอัดใน FPGA แล้วส่งออกไปยัง DAC ที่เป็นวงจร R/2R Ladder ส่งเข้า Non-Inverting Amplifier แล้วส่งผ่าน Low Pass Filter จะได้สัญญาณ Analog เป็นสัญญาณเสียงสำหรับออกไปทางลำโพงได้

สามารถแสดงในรูป Block Diagram ได้ดังรูปที่ 5.3



รูปที่ 5.3 Block Diagram ของบอร์ด

จากรูปจะกล่าวลงในแต่ละ Block ได้ดังนี้

- วงจรขยายไม่กลับขั้วสัญญาณ ใช้ตามรูปที่ 5.1
- วงจร ADC ใช้ IC เบอร์ ADC0820 ของบริษัท MAXIM
- วงจร DAC ใช้วงจร R/2R Ladder ตามรูปที่ 5.2

5.4 ข้อกำหนด และขั้นตอนการทำงานของบอร์ด

การทำงานของบอร์ดจะขออธิบายในส่วนของ Block Diagram ในรูปที่ 5.3 ลงทีละ Block ดังนี้

วงจรขยายไม่กลับขั้วสัญญาณ

ใช้วงจรตามรูปที่ 5.1 โดยวงจรจะทำการขยายสัญญาณก่อนนำไปเข้าวงจรส่วนอื่น โดยไม่มีการกลับเฟส ซึ่ง OpAmp ที่ใช้จะใช้เบอร์ ICL7611 ของบริษัท MAXIM เมื่อขยายสัญญาณแล้วจะส่งให้วงจรส่วนอื่นต่อไป

ส่วน Analog-to-Digital Converter(ADC)

ใช้ IC เบอร์ ADC0820 ของบริษัท MAXIM ต่อตามรูปใน Schematic ให้เป็นการแปลงต่อเนื่องแบบ Pipeline โดยขา Chip Select ของ ADC จะรับสัญญาณจาก FPGA เพื่อใช้ควบคุมความถี่การสุ่มสัญญาณ ดังนั้นไม่ว่าจะมีสัญญาณอะไรเข้ามาก็ตาม และไม่ว่าจะเข้ามาเมื่อใด ADC ก็จะสามารถสุ่มและแปลงสัญญาณตลอดอย่างต่อเนื่อง แต่สัญญาณ Digital ที่แปลงได้ FPGA จะตรวจจับได้หรือไม่ ขึ้นอยู่กับสวิตช์ควบคุม

ส่วน Digital-to-Analog Converter (DAC)

ใช้วงจร R/2R Ladder ในการแปลงซึ่งเมื่อมี Digital Input เข้ามา จะให้ Analog Output ออกไปที่ไหนที่เพื่อไปเข้าวงจรขยายก่อนส่งออกลำโพง

วงจร Low Pass Filter

ใช้กรองสัญญาณความถี่สูงทิ้งไปเพื่อให้สัญญาณที่ได้มีลักษณะเหมือนสัญญาณเสียงจริงๆ

เมื่อประกอบวงจรทุกส่วนเข้าด้วยกันตาม Block Diagram จะได้วงจรตาม Schematic ในภาคผนวก

บทที่ 6

ขอบเขต และข้อจำกัดการออกแบบ

ขอบเขตและข้อจำกัดการออกแบบสำหรับ CU Thai Talk นี้ได้แก่

1. เมื่อทำการอัดเสียงแล้ว จะเล่นซ้ำกี่ครั้งก็ได้ แต่ไม่สามารถอัดเสียงต่อจากที่อัดไว้ได้ นั่นคือเมื่ออัดเสียงแล้วต้องอัดให้ตลอดจนหมดเท่าที่ต้องการ แต่สามารถหยุดชั่วคราวได้โดยกดปุ่ม Stop และเมื่อปล่อยปุ่ม Stop ก็จะสามารถอัดเสียงต่อได้
2. เมื่อทำการ Play แล้วต้องเล่นให้จบ แต่อาจ Pause ได้โดยการใช้ปุ่ม Stop เช่นกันกับตอนอัดเสียง แต่จะเล่นเสียงแล้วกลับมาอัดเสียงต่อจากตำแหน่งที่เล่นไม่ได้
3. เมื่อทำการ Rewind จะไปเริ่มที่ตำแหน่งแรกของ RAM ดังนั้นจะต้อง Record หรือ Play เท่านั้น
4. เมื่อ Record ต้องเริ่ม Record ที่ตำแหน่งแรกของ RAM คือต้อง Rewind ก่อนจึงจะ Record ได้
5. การ Play ต้องเริ่ม Play จากตำแหน่งแรกของ RAM คือต้อง Rewind ก่อนจึงจะ Play ได้
6. ปุ่ม Stop ใช้ในการหยุดการ Play หรือ Record ชั่วคราวได้ เมื่อปล่อยปุ่ม Stop ก็จะสามารถทำการ Play หรือ Record ต่อไปได้
7. วงจรใช้ Clock ความถี่ 32 MHz และสามารถอัดเสียงด้วย RAM 128 kBytes x 8 bits ได้นานถึง 88.9 วินาที ในทุกกรณี

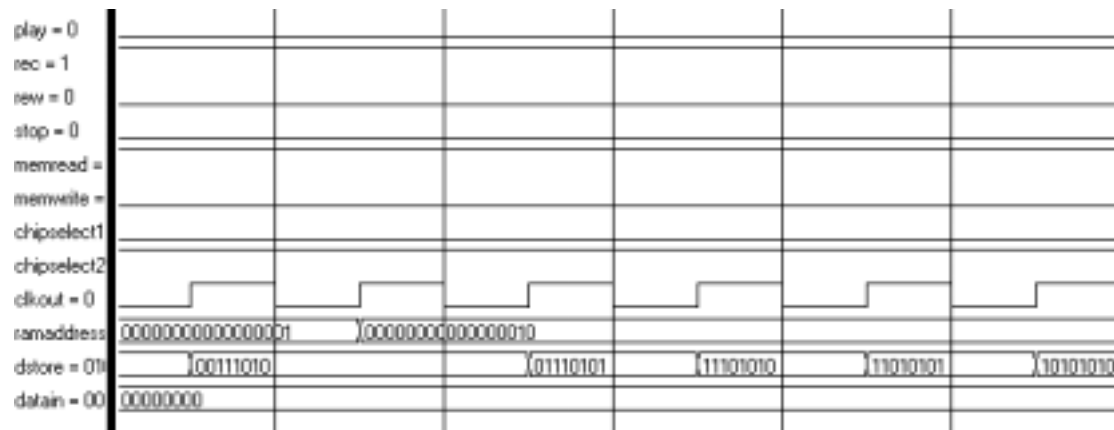
บทที่ 7

ผลการจำลอง และการสังเคราะห์ภายใน FPGA

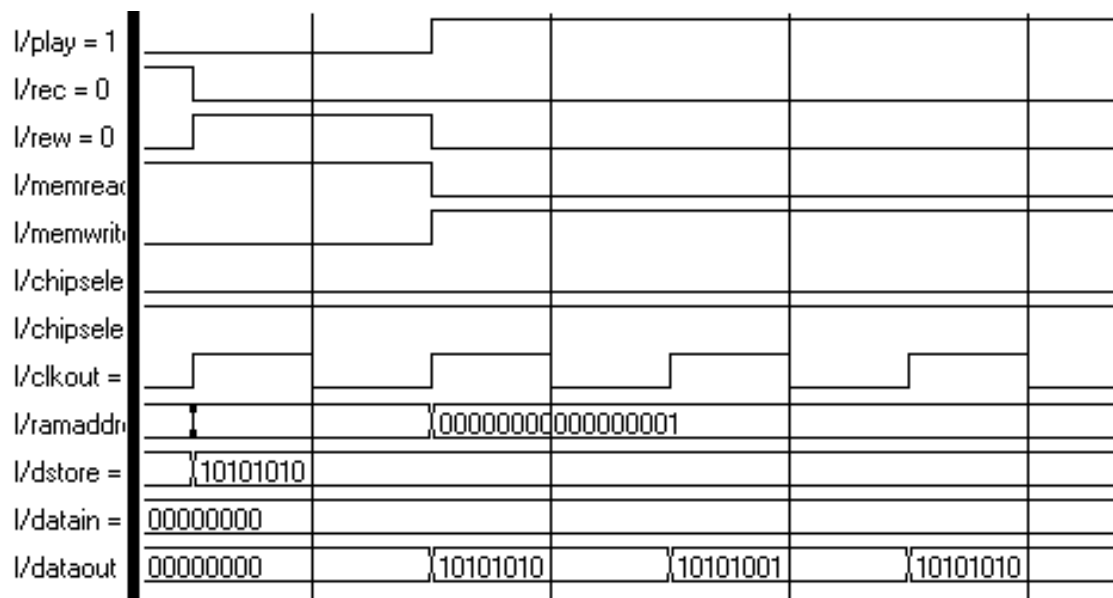
7.1 ผลการจำลองการทำงานของการทำงานการออกแบบ

ทำการจำลองการทำงานการออกแบบด้วย โปรแกรม Modelsim PE ได้รูปคลื่นดังรูปที่

7.1 และ 7.2



รูปที่ 7.1 แสดงผลการจำลองการ Record เสียง



รูปที่ 7.2 แสดงผลการจำลองการ Rewind และ Play เสียง

7.2 ผลการสังเคราะห์การทำงานของการทำงานการออกแบบ

ผลการสังเคราะห์ (Synthesis) ด้วยโปรแกรม Exemplar Leonardo Spectrum Level 3 ได้ File เป็น Format Edif ชื่อว่า Audio9e.Edf ซึ่งอยู่ในแผ่นดิสก์ที่แนบมากับรายงานฉบับนี้ด้วย โดยผลการสังเคราะห์ประกอบด้วย File ต่อไปนี้

- Audio9e.vhd VHDL File
- Audio9e.edf EDIF File
- Audio9e.ncf Report File
- Audio9e.sum Report File

7.3 ผลการทำงานของการทำงาน

ผลการทำงานสามารถทำงานได้ถูกต้องจนกระทั่งสัญญาณออกจาก FPGA ผ่าน DAC แต่เนื่องจากลำโพงมีความต้านทานต่ำ และ output stage ไม่สามารถขับโหลดความต้านทานต่ำมากๆ ได้จึงไม่สามารถให้เสียงออกมาทางลำโพงได้ แต่อาจจับสัญญาณออกได้โดยปลดโหลดแล้วเปรียบเทียบสัญญาณที่ออกจาก DAC กับสัญญาณก่อนเข้า ADC ว่ามีความเพี้ยนต่างกันมากน้อยเพียงใด

และจากเทคนิค ADM ที่กล่าวไว้ในบทแรกๆ จะได้ว่าสามารถบันทึกเสียงได้นานถึง 88.9 วินาที โดยใช้ Static RAM ขนาด 128 kBytes x 8 bits

บทที่ 8

บทส่งท้าย

8.1 ปัญหา และอุปสรรค

การทำโครงการนี้ประสบปัญหาหลายประการ ขอสรุปไว้ดังนี้

1. โปรแกรมที่ทางผู้จัดการแข่งขันจัดส่งช้ามาก ทำให้ไม่สามารถเตรียมงานได้ไม่ดีเท่าที่ควร
2. ศูนย์ติดต่อประสานงานอยู่ไกล และไม่สะดวกในการติดต่อ
3. โปรแกรมที่ทำการ Place & Route มี Bug อยู่ ทำให้ไม่สามารถ Place & Route ได้ ต้องเสียเวลาในการ Debug โปรแกรม จึงสามารถทำงานต่อได้
4. Output Stage ในขณะที่ขับโหลดลำโพง ไม่สามารถขับได้ จึงถูก Pull Down เป็น 0 Volt ตลอดเวลา ทำให้ไม่มีสัญญาณเสียงออกมาทางลำโพง

8.2 แนวทางการพัฒนา และการประยุกต์ร่วมกับงานอื่นๆ

แนวทางในการนำชิ้นงานนี้ไปประยุกต์ร่วมกับงานอื่นสามารถทำได้มากมายหลายอย่าง เพราะงานนี้ทำเป็นงานสำหรับอัดเสียงเฉพาะ สามารถใช้ในสร้างเครื่องอัดเสียงแบบพกพาได้ โดยจะสามารถอัดเสียงได้นานขึ้นกว่าที่เป็นอยู่ ถ้าเพิ่ม Storage ให้มากกว่านี้

8.3 ข้อสรุป และข้อเสนอแนะ

สำหรับโครงการนี้สามารถออกแบบวงจรรวม Digital ที่ทำหน้าที่บันทึกเสียง และเล่นย้อนกลับได้ แต่มีปัญหาเรื่องบอร์ด และ Output Stage ตอนขับโหลดที่เป็นลำโพง จึงทำให้ไม่สามารถเล่นเสียงได้ แต่ผลการทำงานมีความถูกต้องจนถึงสัญญาณที่ออกมาจาก DAC

ข้อเสนอแนะสำหรับโครงการนี้ขอสรุปเป็นข้อๆ ดังนี้

1. ควรจะมีศูนย์ประสานงานอยู่ในตัวเมืองบ้าง เพราะในการติดต่อขอรับอุปกรณ์มีความลำบากมาก เนื่องจากศูนย์ประสานงานอยู่ที่ ลาดกระบัง เพียงแห่งเดียว
2. Software และอุปกรณ์ต่างๆ ควรจัดหาให้ผู้แข่งขันเร็วกว่านี้ เพื่อที่ผู้แข่งขันจะได้ตรวจสอบการทำงาน และแก้ไขได้ตั้งแต่เนิ่นๆ
3. ทางผู้จัดการแข่งขันไม่ระบุ Spec ของอุปกรณ์ ที่ต้องใช้ให้ชัดเจน ซึ่งการหาอุปกรณ์เป็นภาระของ

เอกสารอ้างอิง

ผศ. ดร. วาทีต เบญจพลกุล, “เอกสารประกอบการเรียนวิชา Principle of Communications”, ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ปีพ.ศ. 2542

ผศ. กฤษฎา วิสวธีรานนท์, “เอกสารประกอบการเรียนวิชา Digital Circuit Design”, ภาควิชาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย ปีพ.ศ. 2542