



**การออกแบบวงจรดิจิทัลฟิลเตอร์แบงก์โดยใช้ทรัพยากรร่วม  
แบบพิจารณาภาพรวมบน FPGA  
A Digital Filter Bank Design using Global Resource Sharing  
Techniques on an FPGA**

**เกรียงไกร ทศนวิภาส  
Kriangkrai Tassanavipas**

**วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา  
วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า  
มหาวิทยาลัยสงขลานครินทร์**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of  
Master of Engineering in Electrical Engineering  
Prince of Songkla University**

**2552**

**ลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์**

ชื่อวิทยานิพนธ์      การออกแบบวงจรดิจิทัลเฟลเตอร์เบงก์โดยใช้ทรัพยากรร่วมแบบพิจารณา  
ภาพรวมบน FPGA  
ผู้เขียน              นายเกรียงไกร ทิศนวิภาส  
สาขาวิชา            วิศวกรรมไฟฟ้า

---

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

คณะกรรมการสอบ

..... ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.ณัฐฐา จินดาเพ็ชร) (รองศาสตราจารย์ ดร.มนตรี กาญจนะเดชะ)

อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.ณัฐฐา จินดาเพ็ชร)

..... กรรมการ  
(ดร.วรรณรัช สันตือมรทัต) (ดร.วรรณรัช สันตือมรทัต)

..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.พรชัย พฤกษ์ภัทรานนต์) (รองศาสตราจารย์ ดร.วัฒนพงษ์ เกิดทองมี)

บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ อนุมัติให้บัณฑิตวิทยาลัยฉบับนี้เป็น  
ส่วนหนึ่งของการศึกษา ตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....  
(รองศาสตราจารย์ ดร.เกริกชัย ทองหนู)

คณบดีบัณฑิตวิทยาลัย

ชื่อวิทยานิพนธ์	การออกแบบวงจรดิจิทัลฟิลเตอร์แบงก์โดยใช้ทรัพยากรร่วมแบบ พิจารณาภาพรวมบน FPGA
ผู้เขียน	นายเกรียงไกร ทศนวิภาส
สาขาวิชา	วิศวกรรมไฟฟ้า
ปีการศึกษา	2552

### บทคัดย่อ

งานวิจัยนี้นำเสนอระเบียบวิธีการออกแบบวงจรดิจิทัลฟิลเตอร์ขนาด 8 ช่วง  
ความถี่ โดยใช้กระบวนการฟิลเตอร์แบงก์เป็นกระบวนการในการประมวลผลสัญญาณเสียง  
เนื่องจากเมื่อเปรียบเทียบกระบวนการต่างๆในการสร้างเครื่องช่วยฟังขนาด 8 ช่วงความถี่  
กระบวนการดิจิทัลฟิลเตอร์แบงก์ใช้ตัวคูณเพียง 27 ตัวซึ่งนับว่าใช้ตัวคูณน้อยมากเมื่อเทียบกับ  
กระบวนการ FFT (Fast Fourier Transform) ที่ใช้ตัวคูณถึง 557 ตัว การออกแบบวงจรประกอบด้วย  
2 ส่วนคือ ส่วนวงจรข้อมูล (Data-path circuit) สำหรับประมวลผลข้อมูล และ ส่วนวงจรควบคุม  
(Control circuit) สำหรับการควบคุมจังหวะการทำงาน ในส่วนวงจรข้อมูลมีการลดขนาดวงจรให้  
เล็กลงด้วยการใช้เทคนิคการใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม ซึ่งสามารถช่วยลดขนาด  
ของวงจรได้มากกว่าการใช้ทรัพยากรร่วมกันแบบลำดับชั้นมีการวิเคราะห์ถึงความซับซ้อนที่เพิ่มขึ้น  
ของวงจรที่มีผลมาจากการใช้ทรัพยากรร่วมกัน เพื่อให้ใช้มัลติเพล็กซ์เซอร์ (Multiplexer) น้อยที่สุด  
และความซับซ้อนของวงจรมีน้อยที่สุดด้วยผลที่ตามมาคือมีการเปลี่ยนระดับของสัญญาณ (Switching  
activity) น้อยจึงเป็นผลทำให้วงจรสูญเสียพลังงานน้อย นอกจากนี้วงจรประมวลผลนี้เป็นแบบเลข  
ทศนิยม 26 บิต ซึ่งลดจาก 32 บิต ตามมาตรฐาน IEEE 754 ทำให้วงจรเล็กลงเนื่องจากใช้ตัวคูณ  
MUL 18x18 ในวงจรคูณเลขทศนิยม 1 วงจรลดลงจาก 4 ตัวเหลือเพียง 1 ตัว ในส่วนของวงจร  
ควบคุมมีการใช้ FSM (Finite State Machine) แบบ Moore Machine ในการควบคุม จากการพัฒนา  
วงจรนี้บนชิพ Xilinx Spartan3f4000fg900-4 ผลที่ได้จากการสังเคราะห์วงจรในการออกแบบใช้  
พื้นที่ 7396 LUTs 13863 FFs ซึ่งมีขนาดลดลง 44 % เมื่อเทียบกับการใช้ทรัพยากรร่วมกันแบบ  
ลำดับชั้นที่มีขนาด 32 บิตและความเร็วในการทำงาน 2.76 ไมโครวินาทีและมีการใช้พลังงานลดลง  
46% เมื่อเทียบกับเทคนิคการใช้ทรัพยากรร่วมแบบลำดับชั้น

<b>Thesis Title</b>	A Digital Filter Bank Design using Global Resource Sharing Technique on an FPGA
<b>Author</b>	Mr.Kriangkrai Tassanavipas
<b>Major Program</b>	Electrical Engineering
<b>Academic Year</b>	2009

### **ABSTRACT**

This research presents a design of an 8-band digital filter. The digital filter bank was selected as the method in estimating a voice signals because it used less coefficients when compared to another methods for implementing an 8-band digital hearing aid. The filter bank used only 27 multipliers whereas the FFT (Fast Fourier Transform) method uses 557 multipliers. The circuit was divided into two parts: a data-path part for data processing, and a control part for controlling data-path operations. The data-path circuit part was designed by using a global resource sharing technique to reduce the area of circuit. The increasing complexity of circuit resulted from global resource sharing was analyzed for using minimal multiplexers and interconnection. As a result, the smaller switching activity was obtained to reduce power consumption. Furthermore the circuit was 26-bit floating-point reduced from the 32-bit format of IEEE 754 standard. The circuit was smaller because the MUL 18x18 for one floating-point multiplier was decreased from four to one. The control part was implemented by FSM (Finite State Machine) in the Moore Machine model. From the design on Xilinx Spartan 3f4000fg900-4, the synthesis result used 7396 LUTs and 13863 FFs. The area had decreased 44 % when compared with the 32-bit implementation based on the hierarchical resource sharing technique, the latency was 2.76  $\mu$  s and the power consumption was 46 % reduced

## กิตติกรรมประกาศ

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.ณัฐภา จินดาเพ็ชร์ ประธานกรรมการที่  
ปรึกษางานวิจัย ที่ได้เสียสละเวลาในการให้คำปรึกษา แนวคิดในการทำวิจัย รวมถึงการช่วยเหลือ  
แก้ไขปัญหาที่เกี่ยวกับการวิจัย ตลอดจนตรวจสอบและแก้ไขวิทยานิพนธ์ให้ดำเนินไปอย่างลุล่วง  
สมบูรณ์

ขอขอบพระคุณ ดร.วรรณรัช สันตือมรทัต ที่ได้กรุณาให้คำปรึกษา คำแนะนำ  
และให้ความช่วยเหลือในงานวิจัย ตลอดจนช่วยตรวจทานแก้ไขวิทยานิพนธ์ให้ดำเนิน ไปด้วยดี

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร.พรชัย พุกภัยภัทรานนต์ ที่ได้กรุณาให้  
คำปรึกษา คำแนะนำ และให้ความช่วยเหลือในงานวิจัย ตลอดจนช่วยตรวจทานแก้ไขวิทยานิพนธ์  
ให้ดำเนินไปด้วยดี

ขอขอบพระคุณ องศาสตราจารย์ ดร.มนตรี กาญจนะเดชะ ที่ได้กรุณาเสียสละ  
เวลาเป็นประธานกรรมการสอบวิทยานิพนธ์และตรวจทานแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์

ขอขอบพระคุณ รองศาสตราจารย์ ดร.วัฒนพงศ์ เกิดทองมี ที่กรุณาเสียสละเวลา  
เป็นกรรมการสอบวิทยานิพนธ์ อีกทั้งตรวจทานและแก้ไขวิทยานิพนธ์ให้มีความสมบูรณ์ยิ่งขึ้น

ขอขอบพระคุณ บัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์ วิทยาเขตหาดใหญ่  
ที่ให้การสนับสนุนทุนในการทำวิจัยและให้ความช่วยเหลือด้านการประสานงานต่างๆ

ขอขอบพระคุณ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ที่กรุณาให้  
ทุนผู้ช่วยวิจัยแก่ข้าพเจ้า

ขอขอบพระคุณ คณาจารย์ บุคลากร และนักศึกษาปริญญาโทภาควิชา  
วิศวกรรมไฟฟ้าทุกคนที่ได้ให้คำปรึกษา และกำลังใจในการทำงานเป็นอย่างดีเสมอมา

และสุดท้าย ข้าพเจ้าน้อมรำลึกถึงพระคุณของ บิดามารดา และครอบครัว ที่  
ส่งเสริมและสนับสนุนข้าพเจ้าในทุกๆเรื่องตลอดมาจนสำเร็จการศึกษา

เกรียงไกร ทัศนวิภาส

## สารบัญ

	หน้า
สารบัญ	(6)
รายการตาราง	(10)
รายการภาพประกอบ	(11)
บทที่	
1. บทนำ	1
1.1 ความสำคัญและที่มาของหัวข้อวิจัย	1
1.2 การทบทวนเอกสารที่เกี่ยวข้อง	2
1.3 วัตถุประสงค์	5
1.4 ขอบเขตของการวิจัย	5
1.5 ระเบียบวิธีวิจัย	5
1.6 ประโยชน์ที่คาดว่าจะได้รับ	6
2. ทฤษฎีและหลักการเบื้องต้น	7
2.1 เครื่องช่วยฟัง	7
2.1.1 เครื่องช่วยฟังชนิดคล้อง	7
2.1.2 เครื่องช่วยฟังชนิด Behind the ear hearing aid (BTE)	8
2.1.3 เครื่องช่วยฟังชนิด Eyeglass the ear hearing aid	8
2.1.4 เครื่องช่วยฟังชนิด All-in the ear hearing aid (ITE)	9
2.2 ลักษณะทางวงจรไฟฟ้าของเครื่องช่วยฟัง (Electro acoustic)	9
2.2.1 Gain	9
2.2.2 Saturated sound pressure level (SSPL)	9
2.2.3 Frequency Response	10
2.3 ส่วนประกอบเครื่องช่วยฟัง	11
2.31 ไมโครโฟน (Microphone)	11
2.32 ตัวขยายเสียง (Amplifier)	11
	(6)

## สารบัญ (ต่อ)

	หน้า
2.3.3 ภาครับสัญญาณ (Receiver)	11
2.3.4 แบตเตอรี่	12
2.3.5 Telecoil	12
2.3.6 Tone control	12
2.3.7 Ear mould (EM) หรือแบบหุพลาสติค	12
2.4. วงจรดิจิทัลฟิลเตอร์แบงก์ (Digital Filter Bank Circuit)	13
2.5 การเลือกใช้ค่าถ่วงน้ำหนัก (Gain) ในวงจรเครื่องช่วยฟัง	17
2.6 สถาปัตยกรรมของวงจร	19
2.6.1 ส่วนวงจรข้อมูล (Datapath) สำหรับประมวลผลข้อมูล	19
2.6.2 ส่วนวงจรควบคุม (Controller) สำหรับควบคุม	20
2.6.2.1 Mealy FSM	20
2.6.2.2 Moore FSM	21
2.7 เทคนิคการใช้ทรัพยากรร่วม (Resource sharing)	22
2.8 ข้อควรคำนึงในการใช้เทคนิคการใช้ทรัพยากรร่วม	26
2.8.1 วิเคราะห์ปัญหา Combination feedback loop ของการใช้ทรัพยากรร่วม	26
แบบต่างๆ	
2.8.1.1 การวิเคราะห์วงจรโดยใช้ทรัพยากรร่วมวิธีการที่ 1	27
2.8.1.2 การวิเคราะห์วงจรโดยใช้ทรัพยากรร่วมวิธีการที่ 2	27
2.8.1.3 การวิเคราะห์วงจรโดยใช้ทรัพยากรร่วมวิธีการที่ 3	28
2.9 ปัจจัยที่เกี่ยวข้องกับการบริโภคพลังงานและเทคนิคการลดค่าพลังงาน	29
2.10 เทคโนโลยี Field Programmable Gate Array (FPGA)	30
3 การออกแบบวงจร	32
3.1 เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมบนวงจรฟิลเตอร์แบงก์	32
3.1.1 การใช้ทรัพยากรร่วมแบบลำดับชั้น (Hierarchical Resource-Sharing)	33
3.1.1.1 การใช้ทรัพยากรร่วมระหว่างบล็อก (Inter-Block Sharing)	33
3.1.1.2 การใช้ทรัพยากรร่วมภายในบล็อก (Intra-Block Sharing)	35

## สารบัญ (ต่อ)

	หน้า
3.1.2 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม (Global Resource-Sharing)	35
3.2 ปัจจัยในการออกแบบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	36
3.2.1 การพิจารณาค่าสัมประสิทธิ์	36
3.2.2 พิจารณาความเหมือนของโครงสร้างและความซับซ้อนของสมการ	37
3.2.3 การวิเคราะห์ช่วงเวลา (Time Analysis)	38
3.2.4 การพิจารณาสายการเชื่อมต่อภายในวงจร (Interconnect)	41
3.3 การออกแบบวงจรดิจิทัลเฟลเตอร์แบงก์โดยมีการใช้เทคนิคการใช้ทรัพยากรร่วมในรูปแบบต่างๆ	41
3.3.1 การใช้ทรัพยากรร่วมแบบลำดับขั้น	42
3.3.2 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1	44
3.3.3 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2	47
3.3.4 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3	51
3.3.5 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4	52
4. ผลการวิจัย	56
4.1 ผลการออกแบบวงจรฟิลเตอร์แบงก์	56
4.2 ผลการวิจัยของวงจรฟิลเตอร์แบงก์ด้วยโปรแกรม MATLAB	60
4.3 ผลการวิจัยของวงจรฟิลเตอร์แบงก์ด้วยโปรแกรม Xilinx 10.1	68
4.3.1 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบลำดับขั้น	70
4.3.2 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม	72
4.3.3 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม ร่วมกับการลดมิติข้อมูล	75
4.4 ผลการวัดค่าพลังงานวงจรฟิลเตอร์แบงก์ด้วยโปรแกรม XPower	82
4.4.1 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบลำดับขั้น	82
4.4.2 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม	83
4.4.3 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม ร่วมกับการลดมิติข้อมูล	83



## สารบัญ (ต่อ)

	หน้า
5. บทสรุปและข้อเสนอแนะ	85
5.1 บทสรุป	85
5.2 ข้อเสนอแนะ	87
บรรณานุกรม	88
ภาคผนวก	91
ประวัติผู้เขียน	115

## รายการตาราง

ตาราง		หน้า
3-1	การเปรียบเทียบผลการสังเคราะห์วงจรของการออกแบบการใช้ทรัพยากรร่วม	55
4-1	เปรียบเทียบขนาดพื้นที่ของวงจรรวมทั้งพลังงานที่ถูกบริโภคในแต่ละวงจร	76
4-2	เปรียบเทียบผลการสังเคราะห์วงจรในรูปแบบที่ไม่มีการตั้งค่า Optimization Goal และ resource sharing ใหม่ภายในโปรแกรม Xilinx เทียบกับวงจรที่มีการตั้งค่าเริ่มต้นมาตรฐาน	81
4-3	เปรียบเทียบขนาดพื้นที่ของวงจรรวมทั้งพลังงานที่ถูกบริโภคในแต่ละวงจร	84

## รายการภาพประกอบ

ภาพประกอบ	หน้า
2-1 เครื่องช่วยฟังชนิดคล้อง (body aid)	7
2-2 เครื่องช่วยฟังชนิด Behind the ear	8
2-3 เครื่องช่วยฟังชนิด Eyeglass the ear hearing aid	8
2-4 เครื่องช่วยฟังชนิด ITE ด้านซ้าย all-in the ear ด้านขวา in the canal	9
2-5 Frequency response curve ของเครื่องช่วยฟัง	10
2-6 วงจรดิจิทัลฟิลเตอร์แบงก์แบบเอาต์พุต 8 ช่องสัญญาณ	13
2-7 การประมวลผลของฟิลเตอร์แต่ละตัวภายในฟิลเตอร์แบงก์	14
2-8 ความสัมพันธ์ของค่าผลตอบสนองต่ออิมพัลส์ของฟิลเตอร์แต่ละตัว	15
2-9 โครงสร้างของฟิลเตอร์แบงก์และฟิลเตอร์ย่อยภายในฟิลเตอร์แบงก์	15
2-10 (a) ฟิลเตอร์ย่อย $H_{s1}$ (b) ฟิลเตอร์ย่อย $H_{s2}$	16
2-11 การประมวลผลของฟิลเตอร์แต่ละตัวภายในฟิลเตอร์แบงก์	17
2-12 ภาพแสดงการเปรียบเทียบการเลือกค่าถ่วงน้ำหนักในแต่ละช่วงความถี่เสียงโดยในงานวิจัยนี้เราจะเลือกใช้ช่วงความถี่เสียงเท่ากับ 40 dB คือสัญญาณรูปตัว x โดยอิงกระบวนการทางด้านขวาเนื่องจากมีข้อดีกว่ากระบวนการตรง กลางตรงที่ค่าถ่วงน้ำหนักมีค่าไม่แตกต่างกันมากในแต่ละช่วงความถี่ และในแต่ละแถวเป็นการเปรียบเทียบช่วงความถี่ที่มีปัญหาของผู้พิการทางการได้ยินความถี่ที่ต่างกัน	18
2-13 การออกแบบวงจรสำหรับคำนวณนาฬิกา	19
2-14 บล็อกตัวควบคุม Load-enable และตารางแสดงการทำงานของรีจิสเตอร์	19
2-15 ฟลิป-ฟลอป ชนิด Master-slave	20
2-16 แสดงแผนภาพไคอะแกรมแบบ Mealy Machine	21
2-17 บล็อกไคอะแกรมแบบ Mealy Machine ในมุมมองทางฮาร์ดแวร์	21
2-18 แผนภาพไคอะแกรมรูปแบบ Moore Machine	22
2-19 บล็อกไคอะแกรมแบบ Moore Machine ในมุมมองทางฮาร์ดแวร์	22
2-20 เปรียบเทียบวงจรที่มีการใช้ทรัพยากรการร่วมกันและไม่ได้ใช้ร่วมกันรวมถึงการแบ่งช่วงเวลาที่เหมาะสมกับการทำงานโดยที่ (a) วงจรที่ไม่มีการใช้ทรัพยากรร่วมกันและเลือกแบ่งการทำงานออกเป็น 1 ช่วง	23

## รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
2-20 (b) วงจรที่มีการใช้ทรัพยากรร่วมกันและเลือกแบ่งการทำงานออกเป็น 4 ช่วง (c) วงจรที่มีการใช้ทรัพยากรร่วมกันและเลือกแบ่งการทำงานออกเป็น 5 ช่วง	23
2-21 แผนภาพระดับเกิดของวงจรที่มีการใช้ทรัพยากรร่วมและไม่ใช้ทรัพยากรร่วม	24
2-22 ปัญหา Combination feedback loop	26
2-23 (a) โปรแกรมแสดงวงจรที่นำมาวิเคราะห์ (b) DFG แสดงวงจรที่นำมาวิเคราะห์	26
2-24 วงจรที่มีการใช้ทรัพยากรร่วมวิธีการที่ 1	27
2-25 วงจรที่มีการใช้ทรัพยากรร่วมวิธีการที่ 2	28
2-26 วงจรที่มีการใช้ทรัพยากรร่วมวิธีการที่ 3	28
2-27 การแทรกกรีจิสเตอร์คักข้อมูลลงในวงจร	29
2-28 ขั้นตอนการออกแบบวงจรดิจิทัลบน FPGA	30
3-1 ความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของฟิลเตอร์เบงก์	32
3-2 DFGs ของวงจรฟิลเตอร์เบงก์ 8 ช่วงความถี่	33
3-3 DFG แบบลำดับชั้นของวงจรฟิลเตอร์เบงก์ที่มีการใช้ทรัพยากรระหว่างบล็อก	34
3-4 DFG แบบลำดับชั้นของวงจรฟิลเตอร์เบงก์ (a) ก่อนใช้ทรัพยากรร่วมภายในบล็อก (b) หลังใช้ทรัพยากรร่วมภายในบล็อก	35
3-5 ค่าสัมประสิทธิ์ของวงจรฟิลเตอร์เบงก์ทั้ง 8 บล็อก	37
3-6 A: โครงสร้างการใช้ทรัพยากรร่วมแบบซับซ้อน B: โครงสร้างการใช้ทรัพยากรร่วมแบบไม่ซับซ้อน	37
3-7 การเปรียบเทียบวงจรที่มีการใช้เทคนิคทรัพยากรร่วมโดยพิจารณาการแบ่งช่วงเวลา (a) แบบไม่คำนึงการแบ่งช่วงเวลา (b) แบบคำนึงการแบ่งช่วงเวลา	38
3-8 การแบ่งช่วงเวลาที่เหมาะสมในวงจรฟิลเตอร์เบงก์ที่ได้มีการออกแบบ	39
3-9 การใช้ทรัพยากรร่วมกันภายในบล็อกโครงสร้างบล็อก e	40
3-10 การใช้ทรัพยากรร่วมกันระหว่างบล็อกโครงสร้างบล็อก d	40
3-11 การเปรียบเทียบการวัดค่าสายการเชื่อมต่อภายในวงจร โดยที่ (A) การใช้ทรัพยากรร่วมแบบไม่เหมาะสมมีค่าสายการเชื่อมต่อ 35 สายสัญญาณ	41

## รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
3-11 (B) การใช้ทรัพยากรร่วมแบบเหมาะสมมีค่าสายการเชื่อมต่อ 30 สายสัญญาณ	41
3-12 การใช้ทรัพยากรร่วมแบบลำดับชั้นของวงจรดิจิทัลเตอร์แบงก์	42
3-13 แผนภาพกระแสของการใช้ทรัพยากรร่วมระหว่างบล็อก	43
3-14 แผนภาพกระแสของการใช้ทรัพยากรร่วมภายในบล็อก H3	43
3-15 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1 ของวงจรฟิลเตอร์แบงก์	44
3-16 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน A1	45
3-17 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน A1	45
3-18 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน A3	46
3-19 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน B1	46
3-20 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน B2	46
3-21 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน C1	47
3-22 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2 ของวงจรฟิลเตอร์แบงก์	47
3-23 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A1	48
3-24 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A2	49
3-25 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A3	49
3-26 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A4	50
3-27 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 2 ส่วน B1	50
3-28 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 2 ส่วน B2	50
3-29 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3 ในขั้นแรก	51
3-30 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 3 ส่วน B2	52
3-31 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4	53
3-32 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 2 ส่วน C1	53
3-33 การเปรียบเทียบช่วงเวลาที่ใช้ ระหว่างรูปแบบที่ 3 และรูปแบบที่ 4	54
3-34 แผนภาพกระแสเปรียบเทียบการแบ่งช่วงเวลาของบล็อก e ระหว่างรูปแบบที่ 3 และรูปแบบที่ 4	55
4-1 สัญญาณที่ออกจาก $H_1(z)$ ตามเอกสารอ้างอิงกับที่ได้จากการออกแบบ	56

## รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
4-2 ผลลัพธ์สัญญาณภายในของแต่ละบล็อกภายในวงจรฟิลเตอร์เบงก์	57
4-3 สัญญาณเอาต์พุตของฟิลเตอร์เบงก์	58
4-4 สัญญาณจากเอาต์พุตของ H1 คอนโวลูชันกับสัญญาณภายใน H2 และ H3	58
4-5 สัญญาณเอาต์พุตจากการทำการคอนโวลูชัน	59
4-6 กราฟของเอาต์พุตเมื่อรวมกันทั้ง 8 แถบของฟิลเตอร์เบงก์	59
4-7 กราฟสัญญาณของอินพุตที่ใช้ทำการประมวลผลบน MATLAB	60
4-8 กราฟสัญญาณเอาต์พุต $y_1$ และ $y_{1c}$ ที่ได้จากอินพุตผ่านฟิลเตอร์ H1	61
4-9 กราฟสัญญาณ $y_2$ และ $y_{2c}$ ที่ได้จากเอาต์พุต $y_1$ ผ่านฟิลเตอร์ H2	61
4-10 กราฟสัญญาณ $y_3$ และ $y_{3c}$ ที่ได้จากเอาต์พุต $y_{1c}$ ผ่านฟิลเตอร์ H3	61
4-11 กราฟสัญญาณ $y_4$ และ $y_{4c}$ ที่ได้จากเอาต์พุต $y_2$ ผ่านฟิลเตอร์ H4	62
4-12 กราฟสัญญาณ $y_5$ และ $y_{5c}$ ที่ได้จากเอาต์พุต $y_{2c}$ ผ่านฟิลเตอร์ H5	62
4-13 กราฟสัญญาณ $y_6$ และ $y_{6c}$ ที่ได้จากเอาต์พุต $y_3$ ผ่านฟิลเตอร์ H6	62
4-14 กราฟสัญญาณ $y_7$ และ $y_{7c}$ ที่ได้จากเอาต์พุต $y_{3c}$ ผ่านฟิลเตอร์ H7	63
4-15 กราฟสัญญาณ $y_8$ ที่ได้จากเอาต์พุต $y_4$ ผ่านฟิลเตอร์ H8	63
4-16 กราฟสัญญาณของเอาต์พุตที่ได้จากการประมวลผลบน MATLAB	63
4-17 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงแรกโดยที่ A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)	64
4-18 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 2 โดยที่ A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)	64
4-19 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 3 โดยที่ A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)	65
4-20 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 4 โดยที่ A. วงจรดิจิทัลฟิลเตอร์	65

## รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
4-20 B. ต้นแบบเครื่องช่วยฟัง (วงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์)	65
4-21 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 5 โดยที่ A. วงจรมัลติเพล็กซ์ B. ต้นแบบเครื่องช่วยฟัง (วงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์)	66
4-22 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 6 โดยที่ A. วงจรมัลติเพล็กซ์ B. ต้นแบบเครื่องช่วยฟัง (วงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์)	66
4-23 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 7 โดยที่ A. วงจรมัลติเพล็กซ์ B. ต้นแบบเครื่องช่วยฟัง (วงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์)	67
4-24 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 8 โดยที่ A. วงจรมัลติเพล็กซ์ B. ต้นแบบเครื่องช่วยฟัง (วงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์)	67
4-25 กราฟสัญญาณของเอาต์พุตรวมทุกช่วงความถี่โดยที่ A. วงจรมัลติเพล็กซ์ B. ต้นแบบเครื่องช่วยฟัง (วงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์)	68
4-26 ระบบการทดสอบ (Testbench)	69
4-27 RTL Schematic ของวงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	70
4-28 แสดง Device Utilization Summary ของวงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	71
4-29 แสดง Timing Summary ของวงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	71
4-30 แสดงผลการจำลองการทำงานของวงจรมัลติเพล็กซ์ที่มัลติเพล็กซ์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	71

## รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
4-31 กราฟเปรียบเทียบเอาต์พุตที่ได้จากวงจรฟิลเตอร์เบงก์บน M A T L A B กับ เอาต์พุตวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	72
4-32 RTL Schematicของวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	73
4-33 แสดง Device Utilization Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	74
4-34 แสดง Timing Summary ของวงจรวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	74
4-35 แสดงผลการจำลองการทำงานของวงจรวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	74
4-36 กราฟเปรียบเทียบเอาต์พุตที่ได้จาก เอาต์พุตวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น กับ เอาต์พุตวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	75
4-37 รูปแบบของชุดข้อมูล 32 บิต	75
4-38 รูปแบบของชุดข้อมูล 26 บิต	75
4-39 รูปแบบของชุดข้อมูล 16 บิต	76
4-40 ผลการสังเคราะห์วงจรคุณลักษณะทศนิยมของจำนวนชุดข้อมูล 32 บิต	77
4-41 ผลการสังเคราะห์วงจรคุณลักษณะทศนิยมของจำนวนชุดข้อมูล 26 บิต	77
4-42 ผลการสังเคราะห์วงจรคุณลักษณะทศนิยมของจำนวนชุดข้อมูล 16 บิต	77
4-43 แสดง Device Utilization Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต	78
4-44 แสดง Timing Summary ของวงจรวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต	79
4-45 แสดง Device Utilization Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้นที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx	79



## รายการภาพประกอบ (ต่อ)

ภาพประกอบ	หน้า
4-46 แสดง Timing Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้นที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx	79
4-47 แสดง Device Utilization Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx	80
4-48 แสดง Timing Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx	80
4-49 แสดง Device Utilization Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิตที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx	80
4-50 แสดง Timing Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิตที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx	81
4-51 กราฟสัญญาณของเอาต์พุตโดยที่ A. วงจรดิจิตอลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิตอลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)	82
4-52 พลังงานที่ถูกใช้ในวงจรดิจิตอลฟิลเตอร์เบงก์แบบที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น	82
4-53 พลังงานที่ถูกใช้ในวงจรดิจิตอลฟิลเตอร์เบงก์แบบที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	83
4-54 พลังงานที่ถูกใช้ในวงจรดิจิตอลฟิลเตอร์เบงก์แบบที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต	83

# บทที่ 1

## บทนำ

### 1.1 ความสำคัญและที่มาของหัวข้อวิจัย

เนื่องจากในปี ปัจจุบันจะพบว่า คนพิการจะเพิ่มมากขึ้นเรื่อยๆ โดยเฉพาะอย่างยิ่งคนพิการที่สูญเสียการได้ยิน ซึ่งคนพิการที่สูญเสียการได้ยินที่รุนแรงกว่า 1 พฤศจิกายน 2537 ถึง 30 มิถุนายน 2550 พบว่ามีผู้ป่วยทางด้านการได้ยิน 6 ล้านคน ซึ่งในปี 1993, ๒๗% ของประชากรที่สูญเสียการได้ยินจะพบว่า มีราคาแพงมากซึ่งยกตัวอย่างเช่นเครื่องช่วยฟังของซีเมนส์รุ่น Siemens 284 Super high Power BTE ซึ่งมีราคาประมาณ ๘๐๐ ดอลลาร์หรือเท่ากับ 10000 บาท ซึ่งเมื่อมองราคาที่สูงถึง ๙๐๐๐ บาทก็ถือว่าแพงมากสำหรับผู้ที่มีรายได้น้อย ทำให้ต้องเสียเงินค่าเครื่องที่ทั้งหมดประมาณ 932 ล้านบาท ซึ่งนับว่าเป็นเงินจำนวนมากทีเดียว ซึ่งเครื่องช่วยฟังให้เพียงพอกับผู้ป่วยทั้งหมด

เมื่อศึกษาการทำ งานของเครื่องช่วยฟัง จะพบว่า วิศวกรใช้เทคนิคที่ต่างกันหลายแบบ ยกตัวอย่างเช่น การใช้ FFT (Fast Fourier Transform) หรือ DCT (Discrete Cosine Transform) ซึ่งเมื่อพิจารณาแล้ว จะพบว่า ทั้งสองวิธีนี้ทั้งนี้ไม่มีปัญหาที่อัตราความถี่เสียงจำนวนมากส่งผลให้วงจรที่ออกแบบมีขนาดใหญ่ ยกตัวอย่างเช่นวงจรที่เอาต์พุตความถี่เสียงออก เป็น 8 ช่อง จะพบว่า วิธี FFT จะใช้ตัวคูณที่ 44 ซึ่งเมื่อมาเทียบกับการของดิจิทัลคอลเลกเตอร์แบบก็จะพบว่า ในวิธีการนี้ใช้ค่าที่ 27 ตัวเท่านั้น ผู้วิจัยจึงได้เลือกใช้กระบวนการของดิจิทัลคอลเลกเตอร์แบบในการทำ งานวิจัยนี้

โครงการวิจัยนี้เป็นการพัฒนาของดิจิทัลคอลเลกเตอร์แบบที่ใช้ค่าสัมประสิทธิ์นี้ ออกมา วิธี การอื่น ๆ ที่ใช้วิธีนี้ จะใช้ค่าสัมประสิทธิ์ที่อื่นได้ วงจรดิจิทัลคอลเลกเตอร์แบบที่นี้เป็นส่วนประกอบของต้นแบบเครื่องช่วยฟังที่จะนำไปสู่กระบวนการออกแบบ ASIC ซึ่งจะทำให้สามารถที่จะผลิตเครื่องช่วยฟังในประเทศได้ ทำให้ไม่ต้องนำเข้าจากต่างประเทศเพื่อซื้อเครื่องช่วยฟังในต่างประเทศที่มีผลทำให้ราคาถูกลงกว่าการนำเข้า และในงานวิจัยนี้ ผู้วิจัยได้ใช้ดิจิทัลคอลเลกเตอร์แบบที่เป็น

ส่วนประกอบโดยจะทำ การออกแบบบน FPGA เพื่อที่ จะเน้นไปที่ การออกแบบ ASIC เพื่อที่ จะลดขนาดและลดการใช้ พลังงาน

## 1.2 การทบทวนเอกสารที่เกี่ยวข้อง

1.2.1 Thomas Lunner and Johan Hellgren, “A Digital Filterbank Hearing Aid Design, Implementation and Evaluation”, International Conference on Acoustics, May’1996. บทความนี้ กล่าวถึงวงจรและโครงสร้าง ของเครื่องช่วยฟังแบบดิจิทัล จิตเตอร์แบงก์ โดยในบทความนี้ ได้ บอกสมการที่ ใช้ หาค่า  $\mu$  (close to 1) ที่ หมดจากการแบ่ง ช่วงความถี่ ออกเป็น 8 ช่วงความถี่ และได้ มี การศึกษาที่ ครอบคลุมโดยใช้ คณิตศาสตร์จากโปรแกรม MATLAB

1.2.2 EN-Shou Chang, Daniel D. Gajski and Sanjiv Narayan, “An Optimal Clock Period Selection Method Based on Slack Minimization Criteria”, ACM Transactions on Automation of Electronic Systems, Vol. 1 No. 3, July 1996, Pages 352-370. บทความนี้ จะ กล่าวถึงขั้นตอนสำคัญ ในการสังเคราะห์ฮาร์ดแวร์เพื่อ ใช้ ในอุปกรณ์ การประมวลผลของวงจรนี้ นั่นคือ การเลือก คาบสัญญาณนาฬิกา ให้ แก่ วงจร ซึ่ง การตัดสินใจ เพื่อ เลือก คาบ นี้ มี แนวทางอย่างใด อย่างหนึ่ง ระหว่าง วงเลือกคาบสัญญาณนาฬิกา ที่ แน่ นอนหรือเลือกคาบที่ วงเลือก ใช้ จำนวนคาบสัญญาณนาฬิกา ที่ มากขึ้นซึ่ง จะหาเลือกแนวทางที่ มี ไม่กระทบกะสหนดคาบสัญญาณนาฬิกา แล้ว วนนี้จะก่อ ให้ เกิด การเสีย เวลาโดยเปล่า เป็นประโยชน์ กำหนดคาบสัญญาณที่ ใหญ่เกินไป ดังนั้น ในงานวิจัยนี้ จะนำเสนอวิธี ที่ ใช้ การเลือกคาบสัญญาณนาฬิกา จะทำ ให้ สูญเสีย เวลาที่ เหลือ จากการทำ งานให้ น้อยที่ สุดและสามารถ ใช้ ไปใช้ ในการออกแบบแก่วงจรอื่นที่ วนไปซึ่ง จะก่อ ให้ เกิด ประสิทธิภาพที่ ดี ต่อ การออกแบบ

1.2.3 Kwen-Siong Chong, Bah-Hwee Gwee and Joseph S. Chang, “A Low Energy Asynchronous FFT/IFFT Processor for Hearing Aid Application”, IEEE International Symposium on Circuits and Systems, 2005, pp.751-754 บทความนี้ กล่าว ถึงการออกแบบ เครื่องช่วยฟัง โดยใช้ กระบวนการของ FFT (Fast Fourier Transform) ในการออกแบบ นอกจากนี้ ยังใช้ การออกแบบที่ เป็นแบบ Asynchronous ซึ่ง ได้ มี การเข้าถึงถึงภายในและค่า ตัวคูณเมื่อ มีการแบ่ง สัญญาณเหลือ ออกเป็น 8 ช่วง จะมี ค่า ตัวคูณเท่ากับ 544

1.2.4 Sunitha SL, V. Udaya shankara, “DWT-LMS Speech Enhancement Techniques for Performance Enhancement of Digital Hearing Aid”, ICSCI, Jan 2005. บทความนี้ได้กล่าวถึงกระบวนการออกแบบเครื่องช่วยฟังที่มีความถี่ 8 kHz โดยใช้กระบวนการ DWT (Discrete Wavelet Transform) ซึ่งมีการบอกโครงสร้างของสัญญาณและผลที่ได้ เมื่อมีการป้อนสัญญาณอินพุตในช่วงความถี่ -10, -5, 0, 10 dB ตามลำดับในกระบวนการนี้มีการใช้ตัวคูณเท่ากับ 32 ตัว

1.2.5 Seda Ogreneci Memik, Gokhan Memik, Roozbeh Jafari and Eren Kursun, “Global Resource Sharing for synthesis of control Data Flow Graphs on FPGAs”, DAC 2003, June 2-3 2003. บทความนี้ได้อธิบายเกี่ยวกับปัญหาของฮาร์ดแวร์แบบพิกเจอร์รวมคือ 1. ลดตัวเชื่อมระหว่างวงจรถ่ายที่ไม่จำเป็นถึง 2. พื้นที่ที่เพิ่มขึ้น (Area-gain) 3. พิกเจอร์แต่ละส่วนของทรานซิสเตอร์โดยดูที่การคำนวณการไหลของสัญญาณในแต่ละขั้นตอนวิธีวัดในการจับคู่การใช้ทรานซิสเตอร์รวมกัน 4. พยายามรวมส่วนที่เหมือนกันของทรานซิสเตอร์ทำให้มีขนาดเล็กกลงและลดค่าเวลาหน่วงด้วย 5. การรวมกันของเทคนิคต่างๆที่ได้กล่าวถึงแก้ปัญหาถึงแม้ว่าจะมีการนำเทคนิคการใช้ทรานซิสเตอร์รวมมาใช้ แต่ก็ต้องคำนึงถึงความถี่ที่ผลิตขึ้นบนบล็อกพื้นฐานด้วย

1.2.6 A.B. Hamida, M. samet, N. Lakhoua, M. Drira and J. Mouine “Time-Frequency Speech Processing Strategy Based on and Adjustable Algorithms Dedicated to the Cochlear Prostheses”, ICN’99, December 14-16, 1998, pp.245-248. บทความนี้ได้กล่าวถึงผลลัพธ์ของขอบเขตความถี่ในแต่ละช่วงความถี่ของคิ จิ ตอลฟี สเตอริโอแกรมที่ใช้กระบวนการของ FFT (Fast Fourier Transform) ซึ่งผลลัพธ์ที่ได้ในช่วงความถี่ 0-8,000 Hz โดยสามารถแบ่งช่วงความถี่ในแต่ละช่วงได้ ดังนี้ ช่วงที่ 1 [0-1000Hz], ช่วงที่ 2 [1000-2000Hz], ช่วงที่ 3 [2000-3000Hz], ช่วงที่ 4 [3000-4000Hz], ช่วงที่ 5 [4000-5000Hz], ช่วงที่ 6 [5000-6000Hz], ช่วงที่ 7 [6000-7000Hz] และช่วงที่ 8 [7000-8000Hz] ตามลำดับ

1.2.7 วิวัฒน์ บุญสูง, ณัฐรา จินดาเพ็ชร และพรอัษฎภัทรานนท์, “การออกแบบวงจรคิ จิ ตอลฟี สเตอริโอแกรมที่ใช้ทรานซิสเตอร์รวมกันแบบลดอัตราบน FPGAs”, วารสารวิจัย มข. ปีที่ 14 ฉบับที่ 1, มกราคม 2551 หน้า 75-88 งานวิจัยนี้ได้นำเสนอระเบียบวิธี การออกแบบวงจรคิ จิ ตอลฟี สเตอริโอแกรมขนาดใหญ่ สำหรับคิ จิ ตอลฟี สเตอริโอแกรมที่ผลิตขึ้นบน FPGA (Field Programmable Gate Array) ที่มีอยู่อย่างจำกัด ออกแบบได้ มีการแบ่งออกเป็น 2 ส่วนคือ ส่วนวงจรประมวลผลสัญญาณและส่วนควบคุมวงจร



### 1.3 วัตถุประสงค์

เพื่อพัฒนาวงจรดิจิทัลเตอร์เบงก์โดยใช้ เทคโนโลยีทรานซิสเตอร์ รวมแบบ พิจารณาภาพรวมซึ่ง เป็น ส่วนประกอบที่ สำคัญของระบบดิจิทัล จิตตอลสำหรับใช้ ในการ แยกความถี่ ของเสียงเพื่อ ใช้ ในการประยุกต์ใช้ กับผู้ ป่วยที่ได้เป็นปัญหาทางการ

### 1.4 ขอบเขตของการวิจัย

1.4.1 พัฒนางจรดิจิทัลเตอร์เบงก์โดยใช้ เทคโนโลยีทรานซิสเตอร์ รวมแบบ พิจารณาภาพรวมเพื่อ ให้ เข้า ใจถึง ต้นแบบเครื่องช่วยฟัง จิตตอล

1.4.2 ออกแบบวงจรดิจิทัลเตอร์เบงก์ให้ มี ขนาดจริงไปถึงมี ความเร็วและการใช้ พลังงานที่ เหมาะสมเพื่อ ใช้ กับเครื่องช่วยฟัง

1.4.1 สามารถออกแบบวงจรดิจิทัลเตอร์เบงก์โดยใช้ ระเบียบวิธีทรานซิสเตอร์ รวมแบบ พิจารณาภาพรวม (Global resource sharing)

1.4.2 วงจรที่ ออกแบบได้ กับการใช้ ทรานซิสเตอร์ รวมแบบ พิจารณาภาพรวมสามารถ ทำ งานและได้ ผลการทำ งานที่ เหมือนกันแต่ มี ขนาดวงจรเล็กกว่า และของวงจรไม่ ซับซ้อนกว่า การใช้ ทรานซิสเตอร์ รวมแบบลำดับชั้น (Hierarchical resource sharing) ในการออกแบบ

### 1.5 ระเบียบวิธีวิจัย

ในงานวิจัยนี้ เร็ มต้น ได้ มี การศึกษาวงจรดิจิทัลเตอร์เบงก์โดยเทคโนโลยีทรานซิสเตอร์ พบว่า วงจรดิจิทัลเตอร์เบงก์เป็น วงจรขนาดใหญ่ซึ่งได้ดำเนินการนำ เทคโนโลยี การใช้ ทรานซิสเตอร์ รวมมาใช้ โดยในเบื้องต้นนี้ มีการนำ เทคโนโลยี การใช้ ทรานซิสเตอร์ รวมมาใช้ ซึ่ง ผลที่ พบว่า สามารถลดขนาดได้ พอสมควร แต่ ในที่ นี้ ต้น การที่ จะลดขนาดในงานวิจัยนี้ จึงได้ มี การนำ เทคโนโลยี การใช้ ทรานซิสเตอร์ รวมแบบ พิจารณาภาพรวมมาใช้ โดยนี้ สามารถที่จะช่วยลดพื้นที่ ได้ มากกว่า การใช้ ทรานซิสเตอร์ รวมแบบลำดับชั้น เนื่องจากที่ เรจะนำ ส่วน ที่ คล้ายคลึงกัน บางส่วนมาทำ การใช้ ร่วมกันจึงทำ ให้ การใช้ ทรานซิสเตอร์ รวมแบบ พิจารณาภาพรวมที่จะใช้ ทรานซิสเตอร์ รวมได้ มากกว่า นอกจากนั้น การลดบิต ข้อ อนุกรมที่ ได้ ระดับหนึ่ง โดยในที่ นี้ ได้ มี การลดบิต ข้อ อนุกรมจาก 32 บิต ให้ เหลือ 26 บิต ซึ่งให้ ส่วน หนึ่ง ขนาดเล็กลงมากเนื่อง จากตัว คูณ MUL18x18 ลดลงจาก 4 เหลือ 1 ในวงจรคูณ เลขทศนิยมวงจรถัดไป นอกจากนั้น สิ่ง ที่ คาดว่า จะ

ได้รับด้วยเนืองจากวงจรมีขนาดเล็กคือ การบริโภคพลังงานและในแง่ของขนาดได้ออกแบบนั้นได้ มีการแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ ส่วนวงจรมหาศาลที่ประมวลผลข้อมูลและส่วนวงจรมหาคำนวณที่ควบคุมทำหน้าที่ควบคุมจังหวะการทำงานโดยในการออกแบบมีการคำนึงถึงปัจจัยในการออกแบบการใช้ทรัพยากรร่วมเช่น ความซับซ้อน เวลารอคอยของวงจรและมูลเหตุที่ทำให้มีการบริโภคพลังงานสูงขึ้นเป็นต้น โดยผลที่คาดว่าจะได้คือ วงจรที่ออกแบบดี จิตตอลฟิลด์เตอร์แบงก์โดยใช้เทคนิคการใช้ทรัพยากรร่วมและขนาดผลิตภัณฑ์นั้นคาดว่าจะมีขนาดเล็กถึงไม่ถึงกว่า 130% และพลังงานที่ใช้กับวิธีการเดิมก็มีแนวโน้มลดลงโดยการวิเคราะห์พลังงานจะใช้โปรแกรม XPower ซึ่งมีอยู่ในโปรแกรม Xilinx ในการวิเคราะห์พลังงาน

## 1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 ได้เรียนรู้เทคนิคในการออกแบบวงจรที่เหมาะสมที่สุดอันมีต้นทุนน้อยที่เวลาและพลังงาน

1.6.2 ได้ต้นแบบวงจรจิตตอลฟิลด์เตอร์แบงก์ที่เหมาะสมนำไปสู่การวิเคราะห์วงจรช่วยพลังงานต่อและขนาดเล็ก

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 เครื่องช่วยฟัง [สุภาวดี ประคุณหังสิตและสมยศ คุณจักร, 2544]

เครื่องช่วยฟัง (Hearing aid) คือเครื่องขยายเสียงชนิดหนึ่งที่มีขนาดเล็ก ใช้สำหรับขยายเสียงพูด และเสียงต่างๆในสิ่งแวดล้อมให้ดังขึ้น โดยเครื่องช่วยฟังสามารถแบ่งออกได้เป็น 4 ชนิดคือ

##### 2.1.1 เครื่องช่วยฟังชนิดกล่อง (ภาพประกอบ 2-1)

เครื่องช่วยฟังชนิดกล่อง (Body type) มีลักษณะเป็นกล่องสี่เหลี่ยมเล็กๆ ขนาดเท่ากล่องไม้ขีดไฟ ข้างๆกล่องมีตัวหนึบเล็กๆ สำหรับกดตัดปิดกระเปาะเสียง ภายในตัวเครื่องมีไมโครโฟน และ ตัวขยายเสียง (Amplifier) ส่วนภาครับ อยู่ข้างนอก มีลักษณะกลมเท่าเม็ดกระดุม โดยมีเส้นลวดเล็กๆ (cord) เป็นตัวเชื่อม เวลาใช้เครื่องช่วยฟังชนิดนี้ตัวรับต้องติดกับแบบหูพลาสติก (Ear - mould)



ภาพประกอบ 2-1 เครื่องช่วยฟังชนิดกล่อง (body aid)



### 2.1.2 Behind the ear hearing aid (BTE) (ภาพประกอบ 2-2)

เครื่องช่วยฟังชนิดนี้มีขนาดเล็ก ไมโครโฟน ตัวขยายเสียง และภาครับ อยู่ภายใน ตัวเครื่อง ด้านบนมีพลาสติกเล็กๆ โต้คล้ายตะขอ(Hook) สำหรับเกี่ยวไว้กับใบหู รูเปิดของ ไมโครโฟนก็อยู่บริเวณนั้นด้วย การรับเสียงเข้าไปในไมโครโฟนจึงคล้ายคลึงกับธรรมชาติ และมี ขนาดเล็กไม่มีสายรุงรัง



ภาพประกอบ 2-2 เครื่องช่วยฟังชนิด Behind the ear

### 2.1.3 Eyeglass the ear hearing aid (ภาพประกอบ 2-3)

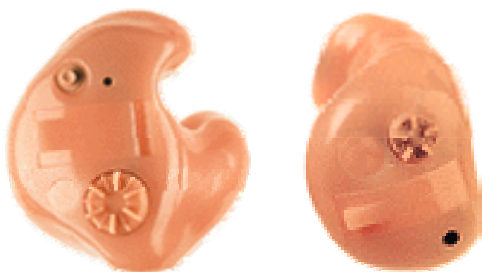
ส่วนประกอบเครื่องช่วยฟังนี้คือ ไมโครโฟน ตัวขยายเสียง และภาครับ อยู่ในก้าน บริเวณขาของแว่นตา มีรูเปิดต่อกับท่อพลาสติกของ Ear mould เพื่อนำเสียงเข้าไปในช่องหู เครื่องช่วยฟังนี้เหมาะสำหรับผู้มีปัญหาทางสายตาด้วย เพราะสามารถใช้ทั้งแว่นตาและเครื่องช่วย ฟังในขณะเดียวกันได้



ภาพประกอบ 2-3 เครื่องช่วยฟังชนิด Eyeglass the ear hearing aid

### 2.1.4 All-in the ear hearing aid (ITE) (ภาพประกอบ 2-4)

ITE เป็นเครื่องฟังขนาดเล็กที่ใส่ไว้ใน Concha ถ้ามีขนาดเล็กใส่ลึกลงไปในช่องหู เรียกว่า In the canal hearing aid (ITC) ทั้ง ITE&ITC นั้น ไมโครโฟน ตัวขยายเสียงและภาครับ บรรจุไว้ใน ear mould ทำให้มีขนาดเล็กใส่ไว้ในช่องหูได้ ลักษณะรับฟังเสียงเหมือนธรรมชาติ เพราะใบหูทำหน้าที่ป้องกันเสียงเข้าไปสู่ไมโครโฟนซึ่งอยู่ในช่องหู ไม่มี head shadow effect เมื่อใช้ ถ้าไม่สังเกตผู้อื่นจะไม่ทราบว่าใช้เครื่องช่วยฟัง เครื่องช่วยฟังชนิดนี้จึงได้ประโยชน์ทางด้านจิตใจและความสวยงาม



ภาพประกอบ 2-4 เครื่องช่วยฟังชนิด ITE ด้านซ้าย all-in the ear ด้านขวา in the canal

## 2.2 ลักษณะทางวงจรไฟฟ้าของเครื่องช่วยฟัง (Electro acoustic)

ลักษณะทางวงจรไฟฟ้า ที่สำคัญประกอบไปด้วย ลักษณะต่างๆดังนี้

### 2.2.1 Gain

Gain หมายถึง กำลังขยายของเครื่องช่วยฟังมีหน่วยเป็นเดซิเบลโดยมีช่วงของการขยายดังนี้ กำลังขยายน้อย (25-40 เดซิเบล) กำลังขยายปานกลาง (40-55 เดซิเบล) และกำลังขยายมาก (60 เดซิเบลขึ้นไป) ซึ่งสามารถเลือกให้เหมาะกับ ผู้สูญเสีย การได้ยินแต่ละคน กำลังขยายของเครื่องช่วยฟังสามารถคำนวณได้จากสมการที่ (2-1)

$$\text{Gain} = \text{Output} - \text{Input} \quad (2-1)$$

### 2.2.2 Saturated sound pressure level (SSPL)

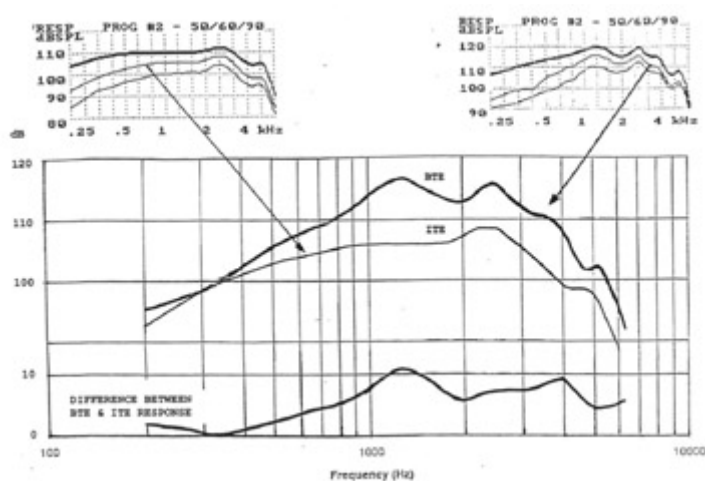
SSPL หรือ MPO หมายถึงกำลังขยายสูงสุดของเครื่องช่วยฟังซึ่งหาได้จากสมการที่ (2-2)

$$\text{Output} = \text{Input} + \text{Gain} \quad (2-2)$$

เครื่องช่วยฟังแต่ละชนิดมี SSPL ต่างๆ กันไป แต่เป็นที่ทราบโดยทั่วไปว่า เสียงที่ดังมากๆ เป็นอันตรายต่อประสาทรับฟังเสียง โดยเฉพาะอย่างยิ่งผู้มีประสาทรับฟังบกพร่อง เมื่อได้รับฟังเสียงดังมากๆ จะทำให้มีอาการปวดหู (residual hearing) ได้ ด้วยเหตุผลนี้ ในการผลิตเครื่องช่วยฟังจึงมีการจำกัดกำลังขยายสูงสุดของเครื่องช่วยฟังไว้เพื่อไม่ให้เป็นอันตรายต่อผู้ใช้ ดังนั้น เครื่องช่วยฟังชนิดนี้ไม่ว่าจะใส่เสียงดังเท่าใดก็ตามเข้าไปในเครื่องช่วยฟัง เสียงที่ออกจากเครื่องช่วยฟังจะถูกจำกัดไม่ให้เสียงออกมาเกินค่า SSPL ของเครื่องช่วยฟังเครื่องนั้นๆ

### 2.2.3 การตอบสนองความถี่ (Frequency Response) (ภาพประกอบ 2-5)

Frequency response คือ กำลังขยายของเครื่องช่วยฟังในความถี่ต่างๆ ซึ่ง Frequency response ของเครื่องช่วยฟังอาจจะขยายในช่วงความถี่กว้างๆ พอๆกันทุกความถี่ (flat frequency response) ขยายเฉพาะช่วงความถี่ต่ำ (low frequency emphasis) หรือขยายเฉพาะช่วงความถี่สูง (high frequency emphasis) ก็ได้แต่โดยทั่วไปเครื่องช่วยฟังมักมี frequency response อยู่ในช่วงความถี่ประมาณ 400-4500 Hz



ภาพประกอบ 2-5 Frequency response curve ของเครื่องช่วยฟัง

แม้ว่าผู้ที่มีความสูญเสียการได้ยินบางส่วนจะได้รับประโยชน์จากการใช้เครื่องช่วยฟังที่มี flat frequency response แต่ส่วนใหญ่แล้วจะได้รับประโยชน์จาก high frequency response มากกว่าทั้งนี้เพราะผู้ที่มีการสูญเสียส่วนใหญ่จะสูญเสียการได้ยินบริเวณความถี่สูง อีกเหตุผลหนึ่งก็คือ เสียงความถี่สูงเป็นเสียงของพยัญชนะซึ่งเป็นเสียงที่สำคัญในการสื่อความหมายของเสียงพูด ส่วนเสียงความถี่ต่ำเป็นเสียงสระ ถ้าขยายให้ดังมากเกินไปจะไปรบกวนความชัดเจนของเสียงความถี่สูง ซึ่งเป็นเสียงของพยัญชนะ และเสียงต่างๆ ที่อยู่รอบตัวในสิ่งแวดล้อมประกอบด้วยเสียง

ความถี่ต่ำเป็นส่วนมาก เมื่อได้รับการขยายให้ดังขึ้นจะไปรบกวนความชัดเจนของเสียงพูดด้วย ดังนั้นโดยทั่วไป frequency response ของเครื่องช่วยฟังจึงมักจะเริ่มมีการขยายเสียงที่มีความถี่ต่ำแล้วค่อยๆ เพิ่มขึ้นเรื่อยๆ และไปขยายเสียงมากบริเวณความถี่สูง

## 2.3 ส่วนประกอบของเครื่องช่วยฟัง (ภาพประกอบ 2-2)

เครื่องช่วยฟังประกอบด้วยส่วนสำคัญต่างๆ ดังต่อไปนี้

### 2.3.1 ไมโครโฟน (Microphone)

ไมโครโฟนมีหน้าที่รับเสียงและเปลี่ยนพลังงานเสียง (Acoustic energy) เป็นพลังงานกล (mechanical energy) โดยการสั่นสะเทือน diaphragm ของไมโครโฟน และเปลี่ยนพลังงานกลให้เป็นพลังงานไฟฟ้า (electrical energy) พลังงานไฟฟ้าที่เกิดขึ้นมีปริมาณน้อยมาก จากนั้นจะส่งต่อไปยังตัวขยายเสียง เพื่อขยายพลังงานไฟฟ้าให้มากขึ้น

ไมโครโฟนจะอยู่ภายในเครื่องช่วยฟังมีรูเปิดที่ด้านนอกของตัวเครื่องเพื่อรับเสียงเข้าไป ซึ่งรูชนิดนี้จะอยู่ด้านหน้า ด้านบน ด้านข้าง หรือด้านหลังก็ได้ แล้วแต่ชนิดของเครื่องช่วยฟังนั้นๆ ปัจจุบันเครื่องช่วยฟังบางชนิดได้ปรับปรุงไมโครโฟนเพื่อรับเฉพาะเสียงที่ต้องการเท่านั้น โดยลดความดังของเสียงรบกวนต่างๆ ในสิ่งแวดล้อม (Ambient noise) ลง เรียกไมโครโฟนชนิดนี้ว่า directional microphone

### 2.3.2 ตัวขยายเสียง (Amplifier)

ตัวขยายเสียง เป็นส่วนที่ขยายสัญญาณหรือพลังงานไฟฟ้าที่ส่งมาจากไมโครโฟนให้ดังขึ้น ซึ่งการขยายจะมีมากน้อยเพียงใดขึ้นอยู่กับ Gain ของเครื่องช่วยฟังแต่ละเครื่องและกำลังขยายสูงสุดก็จะถูกจำกัดโดย SSPL

### 2.3.3 ภาครับสัญญาณ (Receiver)

ภาครับสัญญาณ ทำหน้าที่ตรงกันข้ามกับไมโครโฟน คือทำหน้าที่เปลี่ยนพลังงานไฟฟ้าที่ได้รับการขยายแล้วจาก ตัวขยายสัญญาณ ให้กลับเป็นพลังงานกล และเปลี่ยนพลังงานกลเป็นพลังงานเสียง แต่พลังงานเสียงที่ได้รับการขยายแล้วนั้นคือเสียงดังขึ้น แล้วส่งต่อไปในช่องหูของผู้ใช้เครื่องต่อไป โดยทั่วไปภาครับสัญญาณจะอยู่ในตัวเครื่องช่วยฟังยกเว้น เครื่องช่วยฟังชนิดกล่อง (body type) เท่านั้นที่ภาครับสัญญาณอยู่ภายนอกสามารถมองเห็นได้

### 2.3.4 แบตเตอรี่

เครื่องช่วยฟังจะทำงานได้ต้องอาศัยพลังงานจาก แบตเตอรี่ โดยปกติเครื่องช่วยฟังจะมีอายุการใช้งานประมาณ 200 ชั่วโมง แต่ถ้าใช้กับเครื่องช่วยฟังที่มีกำลังขยายมาก (Powerful hearing aid) อายุการทำงานของแบตเตอรี่จะลดลง โดยทั่วไปแบตเตอรี่เมื่อใช้หมดแล้วจะทิ้งไปเลย แต่มีบางชนิดสามารถนำมาชาร์จใหม่ได้ เนื่องจากแบตเตอรี่มีราคาแพงและต้องใช้เป็นประจำตลอดไปตราบใดที่ยังใช้เครื่องช่วยฟัง

### 2.3.5 Telecoil

ปัจจุบันเครื่องช่วยฟังส่วนใหญ่จะใส่ระบบ Telecoil เข้าไปในเครื่องด้วย ซึ่งระบบนี้จะทำงานเมื่อใช้เครื่องช่วยฟังในสนามแม่เหล็กเท่านั้น เช่น ใน loop system, FM system หรือขณะใช้โทรศัพท์ โดยขณะใช้ให้ปรับปุ่มภายนอกเครื่องช่วยฟังไปที่ Telecoil input คือตัวที่อักษร T ผู้ใช้เครื่องจะได้ยินเฉพาะเสียงที่เกิดขึ้นภายในสนามแม่เหล็ก โดยไม่ได้ยินเสียงรบกวนต่างที่อยู่ภายนอกสนามแม่เหล็กนั้น ทำให้สามารถรับฟังเสียงชัดเจนขึ้น

### 2.3.6 Tone control

ปุ่มปรับ tone control ในเครื่องช่วยฟังจะทำหน้าที่ปรับลักษณะ frequency response ของเครื่องช่วยฟัง เพื่อให้เหมาะกับลักษณะการสูญเสีย การได้ยินของแต่ละคน โดยมีปุ่มปรับเป็นตัวอักษร H (high frequency) L (low frequency) และ N (normal setting) ตัวอย่างเช่นถ้าการสูญเสียการได้ยินเสียงที่ความถี่สูง (high frequency loss) ก็สามารถปรับไปที่ตัวอักษร H หมายถึง ขยายเสียงเฉพาะช่วงความถี่สูงเท่านั้น หรือถ้ากราฟการสูญเสียการได้ยินเป็น flat graph คือมีการสูญเสียการได้ยินพอกๆกัน ตั้งแต่ความถี่ต่ำไปจนถึงความถี่สูงๆ สามารถปรับปุ่ม tone control ไปที่อักษร N เป็นต้น

### 2.3.7 Ear mould (EM) หรือแบบหุพลาสติก (ภาพประกอบ 2-4)

EM เป็นแบบหุที่พิมพ์เฉพาะบุคคล เป็นอุปกรณ์สำคัญชิ้นหนึ่งซึ่งต้องใช้กับเครื่องช่วยฟัง EM มีหลายชนิด ซึ่งสามารถดัดแปลงให้เหมาะสมกับวัตถุประสงค์การใช้แต่ละบุคคล ประโยชน์ของ EM พอสรุปได้ดังต่อไปนี้

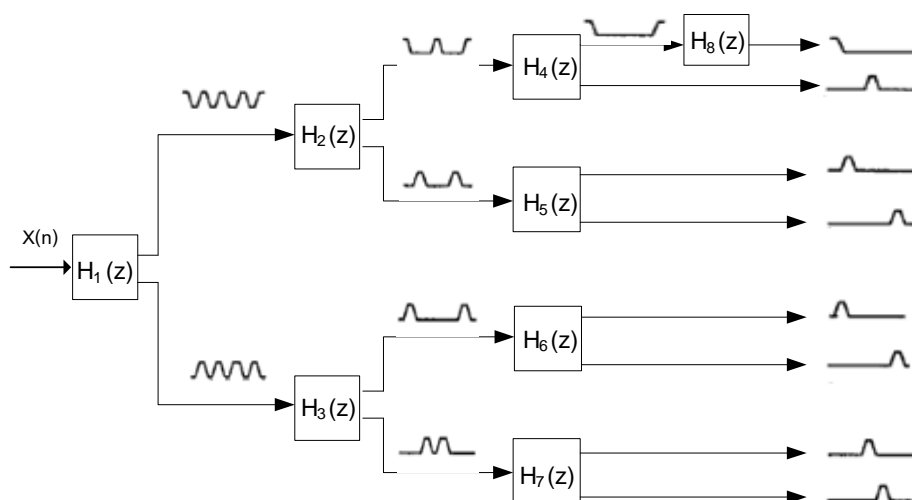
- เป็นอุปกรณ์ซึ่งนำเสียงออกจากเครื่องช่วยฟังส่งต่อเข้าไปในช่องหู
- EM ที่ทำเฉพาะบุคคลจะมีความกระชับพอดีซึ่งช่วยป้องกัน acoustic feedback
- EM ที่ใช้กับเครื่องช่วยฟังชนิด body type จะเป็นตัวเกาะของภากรับ

- สามารถตัดแปลงรูปร่างของขนาดเส้นผ่านศูนย์กลางของ canal EM การเจาะรูเพิ่ม (venting) หรืออื่นๆสามารถเปลี่ยนแปลงลักษณะ electro acoustic ของเครื่องช่วยฟังเพื่อให้เหมาะสมกับลักษณะกราฟการสูญเสียการได้ยินของแต่ละคน

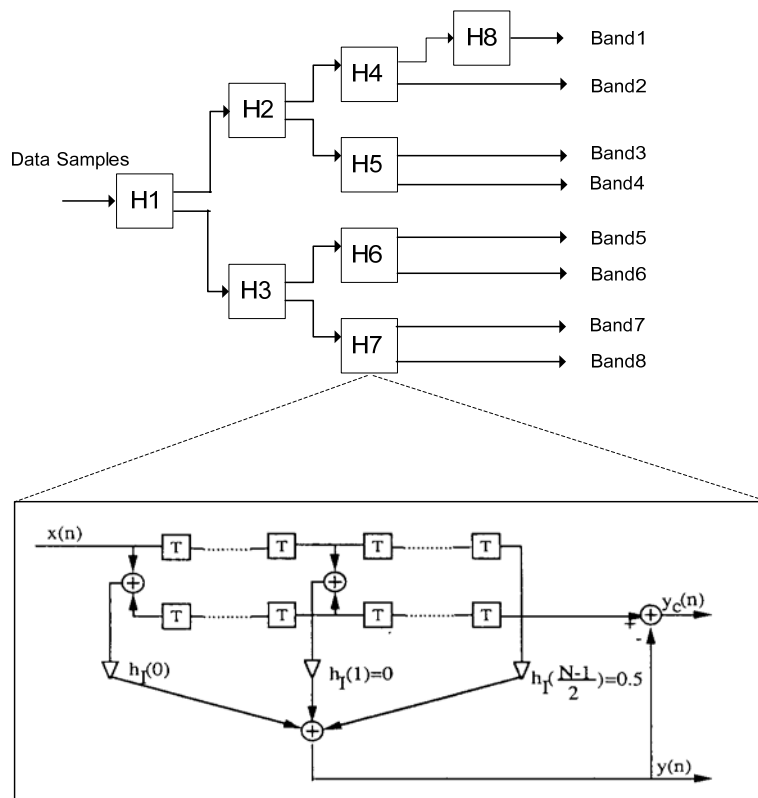
## 2.4 วงจรดิจิทัลฟิลเตอร์แบงก์ (Digital Filter Bank Circuit) [T. Lunner and J. Hellgren, 1991]

วงจรฟิลเตอร์แบงก์ คือวงจรกรองสัญญาณเชิงเลข (Digital Filter) ประเภทหนึ่งที่ทำหน้าที่แยกสัญญาณด้านเข้าที่ถูกป้อนเข้ามาในวงจรเพื่อปรับปรุงคุณภาพของสัญญาณ โดยสัญญาณที่ถูกป้อนเข้ามาจะถูกแยกหรือกรองเป็นสัญญาณที่มีความถี่ต่างๆกันไปซึ่งเป็นผลมาจากการทำงานของฟิลเตอร์ย่อยที่อยู่ภายในฟิลเตอร์แบงก์และสัญญาณที่ได้จากฟิลเตอร์ย่อยตัวแรกจะถูกป้อนเป็นอินพุตให้แก่ฟิลเตอร์ย่อยอื่นๆที่อยู่ภายในฟิลเตอร์แบงก์ไปตามลำดับจนครบ ซึ่งฟิลเตอร์แต่ละตัวเหล่านั้นก็จะทำหน้าที่แยกหรือกรองสัญญาณที่ได้รับเพื่อปรับปรุงคุณภาพของสัญญาณจากการประมวลผลของฟิลเตอร์แต่ละตัวเช่นกัน สุดท้ายก็จะได้สัญญาณด้านออกของฟิลเตอร์แบงก์

วงจรดิจิทัลฟิลเตอร์แบงก์ที่ใช้ในงานวิจัยนี้เป็นวงจรกรองดิจิทัลที่มีสัญญาณด้านเข้า 1 ช่อง ภายในประกอบไปด้วยบล็อกประมวลผล 8 บล็อก เพื่อทำการแยกสัญญาณที่เข้ามาออกเป็นสัญญาณด้านออก 8 ช่องสัญญาณ โดยแสดงได้ดังภาพประกอบ 2-6 ซึ่งในงานวิจัยนี้เลือกใช้วงจรดิจิทัลฟิลเตอร์แบงก์ที่ซ้อนกันเป็นชั้นๆ เนื่องจากการออกแบบวงจรแบบซ้อนกันเป็นชั้นๆนั้นจะช่วยควบคุมตัวเลขของการคูณได้ดีกว่าการที่ออกแบบวงจรเป็นวงจรแยกออกจากกัน



ภาพประกอบ 2-6 วงจรดิจิทัลฟิลเตอร์แบงก์แบบเอาท์พุท 8 ช่องสัญญาณ [Erkan Onat, 2000]



ภาพประกอบ 2-7 การประมวลผลของฟิลเตอร์แต่ละตัวภายในฟิลเตอร์แบงก์

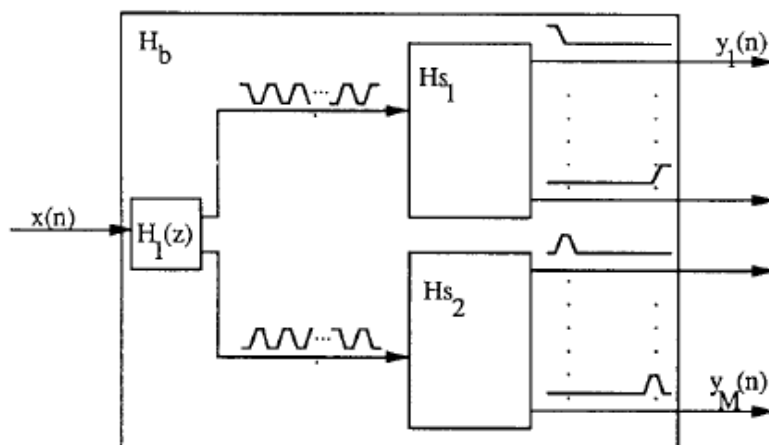
จากภาพประกอบ 2-7 แสดงให้เห็นถึงการทำงานขององค์ประกอบพื้นฐานของ วงจรกรองดิจิทัล โดยหากสังเกตจะพบว่าโครงสร้างภายในของฟิลเตอร์ย่อยนั้นมีโครงสร้างแบบ เดียวกันกับ โครงสร้างของวงจรกรองแบบ FIR

โดยการทำงานของวงจรฟิลเตอร์แบงก์นี้คือ ฟิลเตอร์ย่อยที่อยู่ภายในแต่ละตัวมี การทำงานที่เหมือนกันคือ รับสัญญาณอินพุตที่ด้านเข้าจากนั้นประมวลผลสัญญาณภายใต้สมการ ผลต่างโดยที่ฟิลเตอร์ย่อยแต่ละตัวจะมีสมการเฉพาะ โดยการที่จะสร้างสมการผลต่างของฟิลเตอร์ ได้นั้น จะต้องทราบค่าผลตอบสนองต่ออิมพัลส์ ( $h(n)$ ) ซึ่งภาพประกอบ 2-8 แสดงให้เห็น ความสัมพันธ์ของค่าตอบสนองต่ออิมพัลส์ของฟิลเตอร์แต่ละตัวเพื่อช่วยในการสร้างสมการผลต่าง ของฟิลเตอร์ที่ใช้ประมวลผลสัญญาณภายในวงจรฟิลเตอร์แบงก์นั่นเอง

Filter	H(n)	Decimal Floating point value
<b>H1(z)</b>	h0 : h (0) = h (48)	-0.05062 4178 42547
	h1 : h (16) = h (32)	0.29505933 470299
	h2 : h (24)	0.5
<b>H2(z)</b>	h0 : h (0) = h (24)	-0.05062 4178 42547
	h1 : h (8) = h (16)	0.29505933 470299
	h2 : h (12)	0.5
<b>H3(z)</b>	h0 : h (0) = h (28)	-0.00373 7655 73262
	h1 : h (4) = h (24)	0.0205 6989 4870 10
	h2 : h (8) = h (20)	-0.07232 1904 70689
	h3 : h (12) = h (16)	0.3059 7047 3625 44
	h4 : h (14)	0.5
<b>H4(z)</b>	h0 : h (0) = h (12)	-0.05062 4178 42547
	h1 : h (4) = h (8)	0.29505933 470299
	h2 : h (6)	0.5
<b>H5(z)</b>	h0 : h (0) = h (10)	0.0130 4920 5552 05
	h1 : h (2) = h (8)	-0.06387 1512 1040 5
	h3 : h (4) = h (6)	0.3016 1294 8075 61
	h2 : h (5)	0.5
<b>H6(z)</b>	h0 : h (0) = h (6)	-0.05062 4178 42547
	h1 : h (2) = h (4)	0.29505933 470299
	h2 : h (3)	0.5
<b>H7(z)</b>	h0 : h (0) = h (30)	0.0130 4920 5552 05
	h1 : h (6) = h (24)	-0.06387 1512 1040 5
	h3 : h (12) = h (18)	0.3016 1294 8075 61
	h2 : h (15)	0.5
<b>H8(z)</b>	h0 : h (0) = h (2)	0.29505933 470299
	h1 : h (1)	0.5

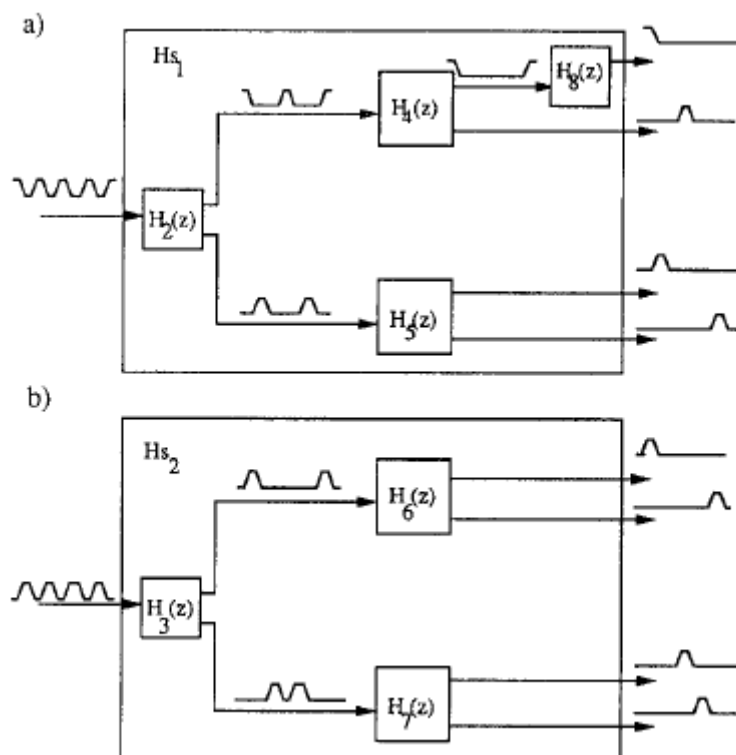
ภาพประกอบ 2-8 ความสัมพันธ์ของค่าผลตอบสนองต่ออิมพัลส์ของฟิลเตอร์แต่ละตัว

การทำงานของฟิลเตอร์เบงก์จะเริ่มจากสัญญาณอินพุต  $x(n)$  ผ่านเข้ามาที่ฟิลเตอร์  $H_1(z)$  จากนั้นสัญญาณจะถูกประมวลผลแล้วส่งค่าเอาต์พุตออกมาเป็นสองค่าคือ  $y_1(n)$  กับ  $y_1c(n)$  โดยที่เอาต์พุตสองค่านี้หากสังเกตจากภาพประกอบ 2-9 จะเห็นว่าเป็นค่าที่ตรงข้ามกันซึ่งกันและกันนั่นเอง จากภาพประกอบเมื่อได้ค่าเอาต์พุตสองค่าดังที่กล่าวมาแล้ว ค่าเอาต์พุตนี้จะถูกส่งไปประมวลผลต่อด้วยฟิลเตอร์ในลำดับถัดไป ( $H_{s1}$  และ  $H_{s2}$ ) จนได้สัญญาณเอาต์พุตของฟิลเตอร์เบงก์ ตั้งแต่  $y_1(n)$  จนถึง  $y_M(n)$



ภาพประกอบ 2-9 โครงสร้างของฟิลเตอร์เบงก์และฟิลเตอร์ย่อยภายในฟิลเตอร์เบงก์





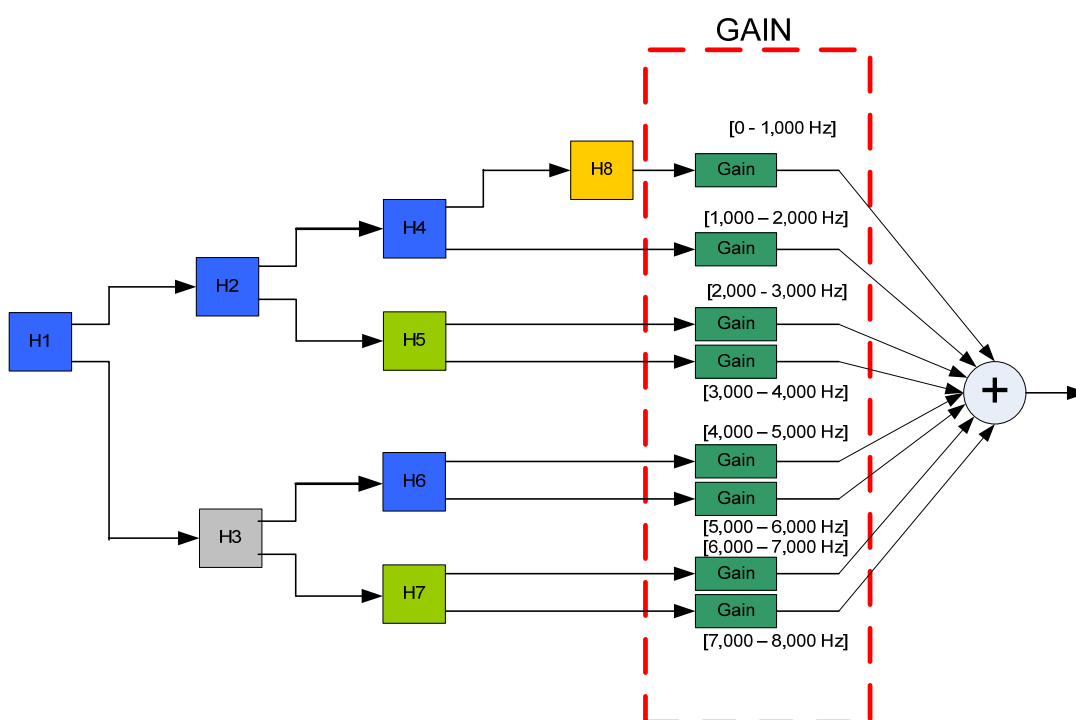
ภาพประกอบ 2-10 (a) ฟิเตอร์ย่อย  $H_{s1}$  (b) ฟิเตอร์ย่อย  $H_{s2}$

จากภาพประกอบที่ 2-10 แสดงลำดับในการประมวลผลของฟิเตอร์ย่อย  $H_{s1}$  และ  $H_{s2}$  เพื่อให้ได้ค่าเอาต์พุตทั้งหมดครบ โดยที่ ฟิเตอร์ย่อย  $H_{s1}$  จะรับค่าอินพุตซึ่งคือค่าเอาต์พุต  $y_1(n)$  ของฟิเตอร์  $H_1(z)$  มาป้อนให้ฟิเตอร์  $H_2(z)$  ซึ่งทำการแยกสัญญาณออกมาได้ค่าเอาต์พุตเป็น  $y_2(n)$  และ  $y_{2c}(n)$  จากนั้นค่าสัญญาณ  $y_2(n)$  จะถูกป้อนเป็นอินพุตให้กับฟิเตอร์  $H_4(z)$  โดยจากฟิเตอร์  $H_4(z)$  นั้นสัญญาณอินพุตจะถูกประมวลผลได้ค่าสัญญาณเอาต์พุตเป็น  $y_4(n)$  กับ  $y_{4c}(n)$  ค่าเอาต์พุต  $y_4(n)$  จะเป็นค่าอินพุตให้แก่  $H_8(z)$  จนสุดท้ายจะได้ค่า  $y_8(n)$  ส่วนค่า  $y_{8c}(n)$  จะไม่นำมาใช้ และสำหรับสัญญาณ  $y_{2c}(n)$  จะป้อนไปยังฟิเตอร์  $H_5(z)$  ได้ค่าเอาต์พุตเป็น  $y_5(n)$  กับ  $y_{5c}(n)$  ดังนั้นสำหรับฟิเตอร์ย่อย  $H_{s1}$  เมื่อทำการประมวลผลเสร็จแล้วนั้นจะได้ค่าสัญญาณทั้งหมด 4 ค่า

ฟิเตอร์ย่อย  $H_{s2}$  มีการทำงานที่เป็นไปในลักษณะเดียวกันกับ ฟิเตอร์ย่อย  $H_{s1}$  โดยที่ค่าอินพุตเริ่มต้นคือค่า  $y_{1c}(n)$  และเมื่อฟิเตอร์ภายในฟิเตอร์ย่อย  $H_{s2}$  ทำงานเสร็จแล้วนั้นจะได้ค่าเอาต์พุตอีก 4 ค่า ซึ่งเมื่อไปรวมกับเอาต์พุตจากฟิเตอร์ย่อย  $H_{s1}$  ก็จะได้ค่าสัญญาณเอาต์พุตทั้งหมด 8 ค่านั่นเอง

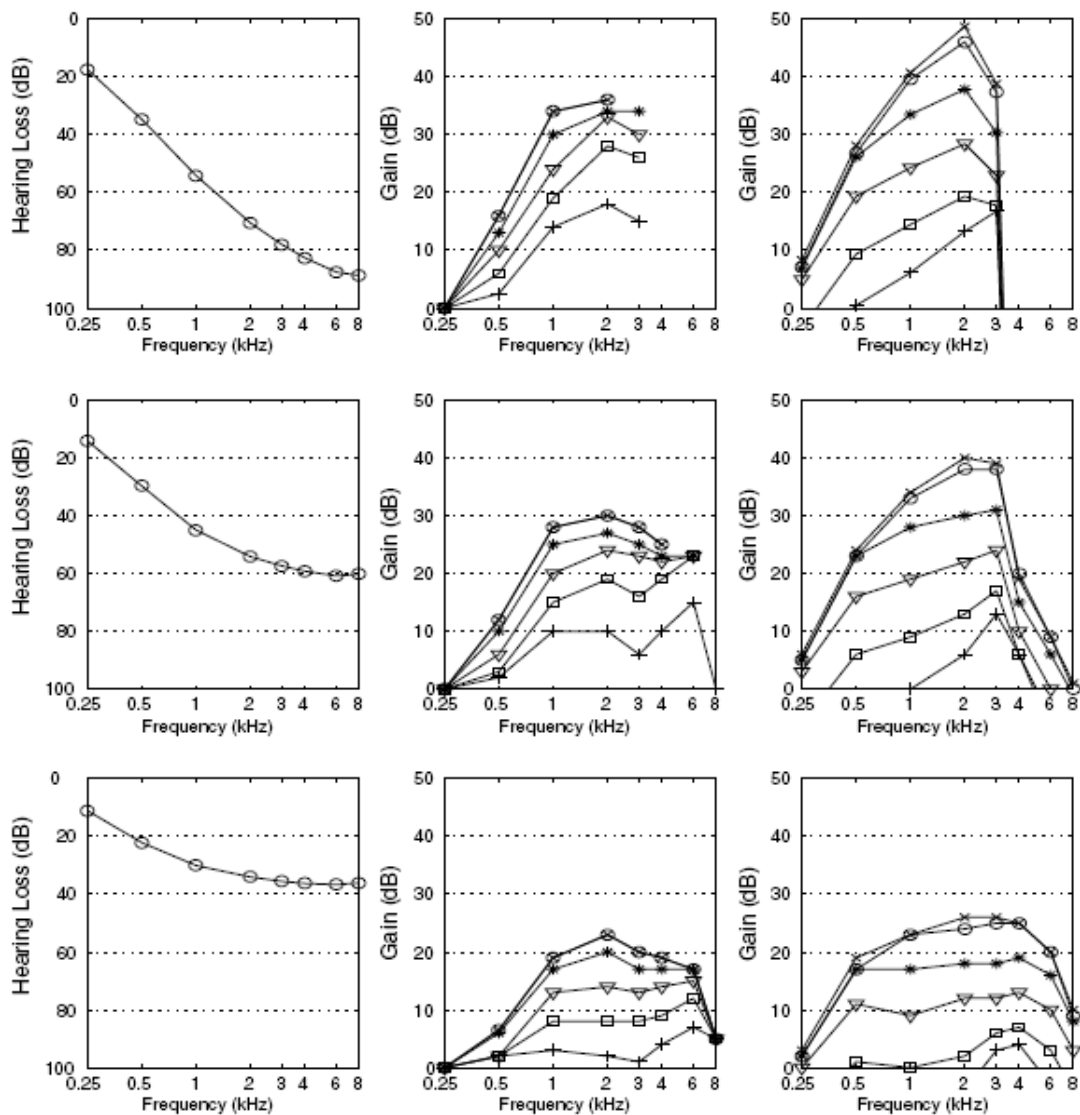
## 2.5 การเลือกใช้กำลังขยาย (Gain) ในวงจรเครื่องช่วยฟัง

เมื่อพิจารณาวงจรเครื่องช่วยฟังภายในส่วนของวงจรข้อมูลจะพบว่าภายในวงจรเมื่อผ่านกระบวนการของวงจรดิจิทัลฟิลเตอร์เบงก์นั้นเสียงจะถูกแยกออกเป็น 8 ช่วงความถี่ โดยในช่วงแรกจะมีความถี่อยู่ระหว่าง 0-1000 Hz ในช่วงที่ 2 มีความถี่อยู่ระหว่าง 1000-2000 Hz ช่วงที่ 3 มีความถี่อยู่ระหว่าง 2000-3000 Hz ช่วงที่ 4 มีความถี่อยู่ระหว่าง 3000-4000 Hz ช่วงที่ 5 มีความถี่อยู่ระหว่าง 4000-5000 Hz ช่วงที่ 6 มีความถี่อยู่ระหว่าง 6000-7000 Hz และช่วงที่ 8 มีความถี่อยู่ระหว่าง 7000 – 8000 Hz [A.B. Hamida, 1998] โดยสามารถแสดงได้ดังภาพประกอบที่ 2-11



ภาพประกอบ 2-11 ตำแหน่งของค่าถ่วงน้ำหนักภายในส่วนวงจรข้อมูลและช่วงความถี่ของเอาต์พุต

เมื่อศึกษาค่ากำลังขยายจะพบว่า การเลือกใช้ค่ากำลังขยายนั้นเราสามารถเลือกใช้ได้ในหลายความเข้มเสียงตั้งแต่ช่วงความเข้มเสียง 40-90 dB ซึ่งสามารถแสดงค่ากำลังขยายในแต่ละความเข้มเสียงและช่วงความถี่ ซึ่งจะพบว่าความเข้มเสียงที่คนใช้ในการสนทนาส่วนใหญ่จะอยู่ในช่วงที่ไม่เกิน 60 dB แสดงได้ดังภาพประกอบ 2-12 ดังนั้นในที่นี้จึงได้มีการเลือกการใช้ค่ากำลังขยายในช่วงความเข้มเสียง 40 dB โดยเราจะนำค่ากำลังขยายในแต่ละช่วงความถี่ไปคูณกับความถี่เสียงในช่วงนั้นๆ ตามภาพประกอบที่ 2- 11



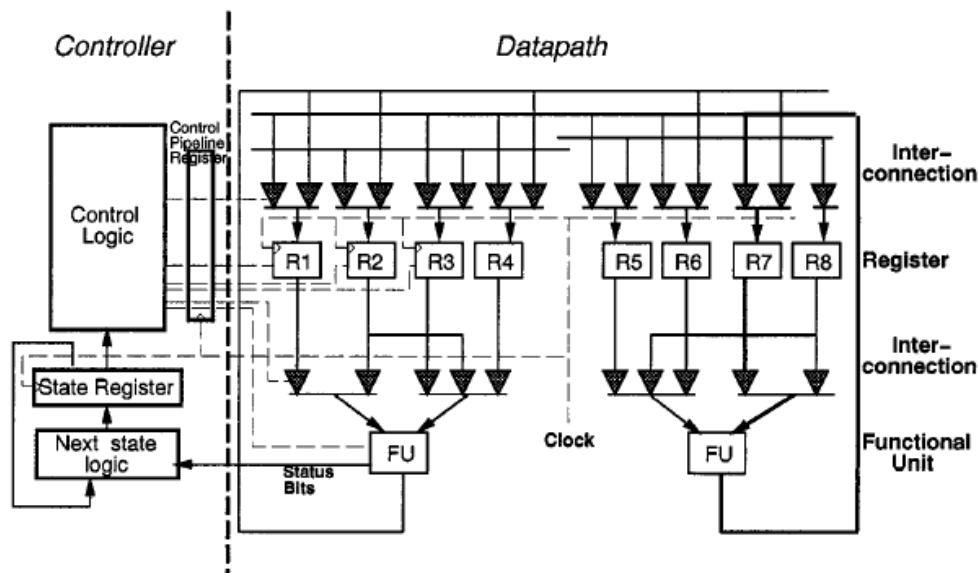
ภาพประกอบ 2-12 ภาพแสดงการเปรียบเทียบการเลือกค่ากำลังขยายในแต่ละช่วงความถี่เสียง โดยในงานวิจัยนี้เราจะเลือกใช้ช่วงความถี่เสียงเท่ากับ 40 dB คือสัญลักษณ์รูปตัว x โดยอิงกระบวนการทางด้านขวาเนื่องจากมีข้อดีกว่ากระบวนการตรงกลางตรงที่ค่ากำลังขยายมีค่าไม่แตกต่างกันมากในแต่ละช่วงความถี่ และในแต่ละแถวเป็นการเปรียบเทียบช่วงความถี่ที่มีปัญหาของผู้พิการทางการได้ยินความถี่ที่ต่าง กัน (Yongsam Kim, 2006)

## 2.6 สถาปัตยกรรมของวงจร

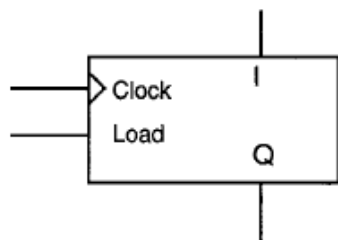
ในการออกแบบวงจรมีการใช้แบบ FSM (Finite State Machine Datapath) ภายในวงจรรวมประกอบด้วย 2 ส่วน ได้แก่

### 2.6.1 ส่วนวงจรข้อมูล (Datapath) สำหรับประมวลผลข้อมูล

วงจรข้อมูลประกอบด้วย รีจิสเตอร์, หน่วยดำเนินการ(Functional units) เช่น วงจรบวก, วงจรลบ, วงจรคูณ และ ตัวเชื่อมต่อข้อมูล (Interconnections) เช่น ตัวเลือกสายสัญญาณ (Selector) หรือ บัส (Buses) โดยรีจิสเตอร์ในวงจรข้อมูลนั้นมี ตัวควบคุม load-enable ดังภาพประกอบที่ 2-14 ในภาพประกอบที่ 2-13 วงจรควบคุม load-enable สำหรับแต่ละรีจิสเตอร์ถูกควบคุมแยกกันโดย Controller ในการออกแบบรูปแบบนี้จะใช้สัญญาณนาฬิกาเพียง 1 เฟสเท่านั้น ในแต่ละรีจิสเตอร์ จะถูกควบคุมด้วย สัญญาณนาฬิกาและสัญญาณ load-enable ควบคู่กันไป



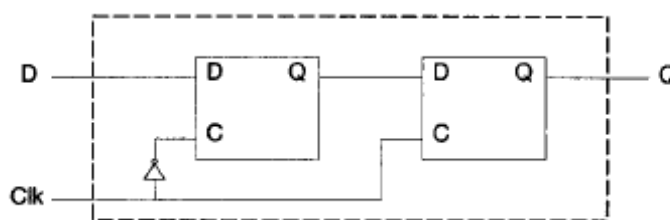
ภาพประกอบ 2-13 การออกแบบวงจรมีจำนวนนาฬิกา



inputs			output
Clock	Load	I	Q
↗	H	L	L
↗	H	H	H
↗	L	X	Q
L	L	X	Q
H	L	X	Q

ภาพประกอบ 2-14 บล็อกตัวควบคุม Load-enable และตารางแสดงการทำงานของรีจิสเตอร์

ในส่วนของตรรกะแสดงสถานะถัดไป จะทำหน้าที่คำนวณ สถานะถัดไปซึ่งถูกเก็บไว้ในรีจิสเตอร์สถานะ โดยที่รีจิสเตอร์สถานะและส่วนควบคุมไปป์ไลน์คือ ฟลิป-ฟลอป ชนิด Master-slave ดังภาพประกอบที่ 2-15 ดังนั้น รีจิสเตอร์สถานะสามารถเก็บสถานะปัจจุบันให้แก่บล็อกตรรกะ แล้ว ตัวควบคุมรีจิสเตอร์ไปป์ไลน์ (Control Pipeline Register) สามารถเก็บสัญญาณควบคุม (Control Signals) ให้แก่ ส่วนประกอบของวงจรข้อมูลเช่นกัน



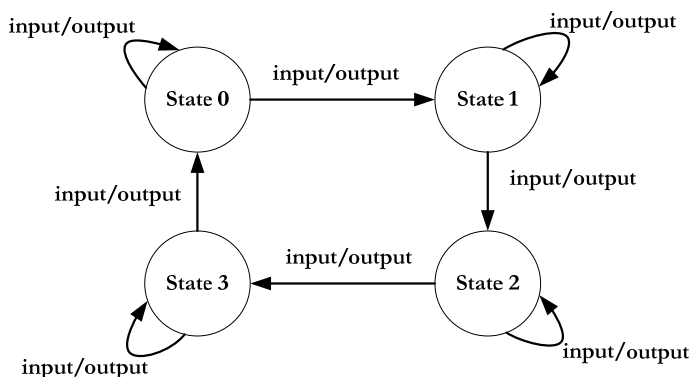
ภาพประกอบ 2-15 ฟลิป-ฟลอป ชนิด Master-slave

## 2.6.2 ส่วนวงจรควบคุม (Controller) สำหรับควบคุม

โดยในส่วนควบคุมจะมีการทำงานแบบ FSM ซึ่งบรรจุ รีจิสเตอร์สถานะ (state register) ส่วนควบคุม (Control logic) และบล็อกตรรกะ (Logic block) แบบต่อเนื่อง 2 ตัว คือ ตรรกะแสดงสถานะถัดไป (Next State Logic) และ ตรรกะเอาต์พุต (Output Logic) โดยตรรกะเอาต์พุตจะทำหน้าที่ขับสายควบคุม (Control line) โดยพื้นฐานจะมีการออกแบบ FSM ได้ 2 รูปแบบคือ

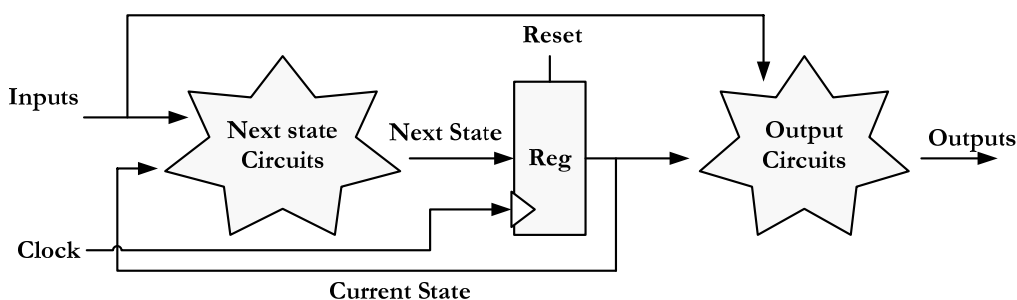
### 2.6.2.1 Mealy FSM

Mealy Machine เป็นรูปแบบการเขียนแผนภาพแสดงสถานะ (State-diagram) แบบที่ค่าของเอาต์พุตในแต่ละสถานะจะขึ้นอยู่กับค่าของสถานะปัจจุบัน (Current state) และค่าของอินพุตในแต่ละสถานะด้วย ซึ่งจะแตกต่างจาก Moore Machine ที่ค่าเอาต์พุตจะเปลี่ยนแปลงขึ้นอยู่กับค่าของสถานะปัจจุบันเท่านั้น เพื่อความเข้าใจให้พิจารณาจากภาพประกอบ 2-16



ภาพประกอบ 2-16 แสดงแผนภาพไคอะแกรมแบบ Mealy Machine [ชำนาญ ปัญญาใส, 2547]

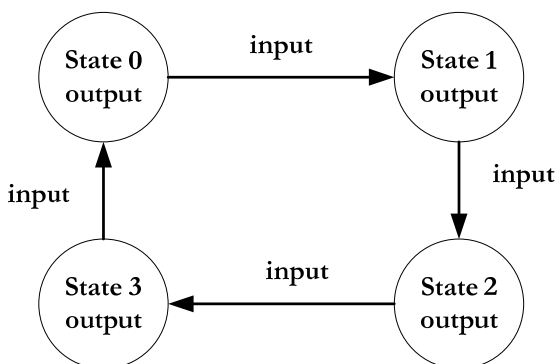
ในแผนภาพไคอะแกรมมีทั้งหมด 4 สถานะ โดยค่าของเอาต์พุตในแต่ละสถานะจะขึ้นอยู่กับค่าของอินพุตและค่าของสถานะ ปัจจุบันนั้นๆ ซึ่งถ้ามองในรูปของฮาร์ดแวร์หรือวงจรสามารถเขียนให้เป็นบล็อกไคอะแกรม ได้ดังภาพประกอบ 2-17 ซึ่งจะประกอบไปด้วยวงจรคอมบิเนชันในส่วนสถานะถัดไป (Next State) และส่วนของรีจิสเตอร์ที่ใช้เก็บค่าสถานะของสถานะ และวงจรคอมบิเนชันในส่วนการสร้างสัญญาณเอาต์พุต ที่นำค่าของอินพุตมาพิจารณาในการเปลี่ยนค่าเอาต์พุตในแต่ละสถานะด้วย



ภาพประกอบ 2-17 แสดงบล็อกไคอะแกรมแบบ Mealy Machine ในมุมมองทางฮาร์ดแวร์ [ชำนาญ ปัญญาใส, 2547]

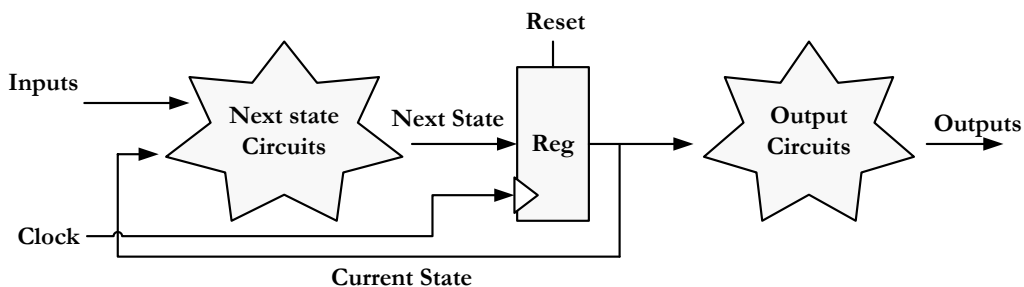
**2.6.2.2 Moore FSM**

Moore Machine เป็นรูปแบบการเขียนแผนภาพแสดงสถานะแบบค่าของเอาต์พุตในแต่ละสถานะจะขึ้นอยู่กับค่าของสถานะปัจจุบัน (Current State) เท่านั้น เพื่อความเข้าใจพิจารณาภาพประกอบ 2-18



ภาพประกอบ 2-18 แสดงแผนภาพไคอะแกรมรูปแบบ Moore Machine [ชำนาญ ปัญญาใส, 2547]

ในรูปไคอะแกรมมีทั้งหมด 4 สถานะ โดยค่าของเอาต์พุตในแต่ละสถานะจะถูกเขียนไว้ในวงกลมของสถานะนั้นๆ และการเปลี่ยนแปลงสถานะจะขึ้นอยู่กับค่าอินพุตที่เข้ามาในแต่ละสถานะซึ่งถ้ามองในรูปของฮาร์ดแวร์หรือวงจรสามารถที่จะเขียนเป็นบล็อกไคอะแกรมได้ดังภาพประกอบ 2-19 ซึ่งจะประกอบไปด้วยวงจรคอมบิเนชันในส่วนของ Next State และส่วนของวงจรรีจิสเตอร์ที่ใช้ในการเก็บค่าสถานะของสถานะและวงจรคอมบิเนชันในส่วนการสร้างสัญญาณเอาต์พุต

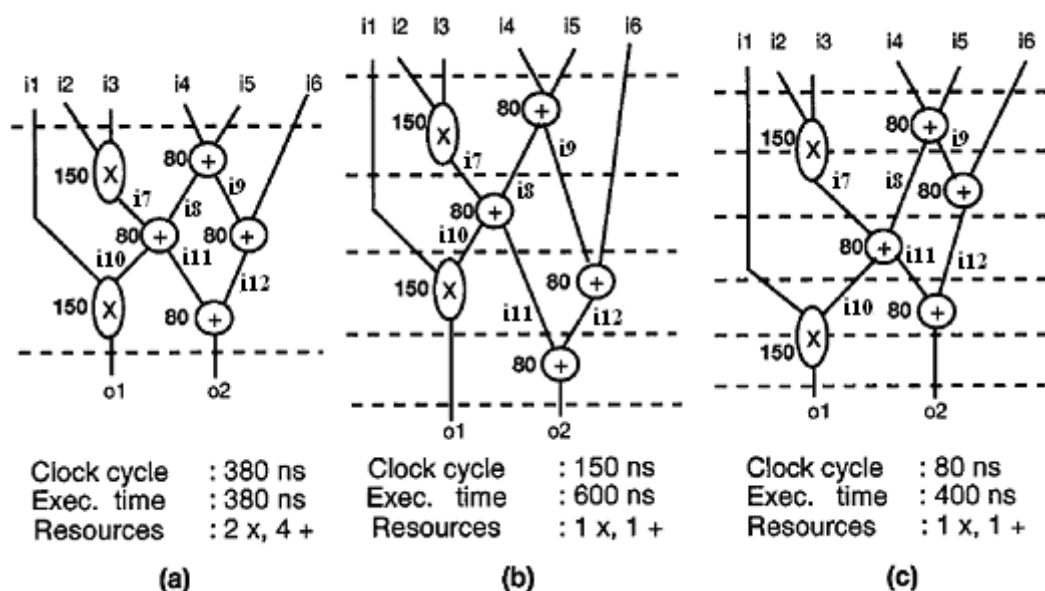


ภาพประกอบ 2-19 แสดงบล็อกไคอะแกรมแบบ Moore Machine ในมุมมองทางฮาร์ดแวร์ [ชำนาญ ปัญญาใส, 2547]

**2.7 เทคนิคการใช้ทรัพยากรร่วม (Resource sharing)**

โดยทั่วไปการออกแบบวงจรมักจะมีปัจจัยในการออกแบบเป็นตัวกำหนดซึ่งได้แก่ขนาด พลังงาน ความซับซ้อนของวงจร และ ประสิทธิภาพเป็นต้น เมื่อก้าวถึงปัจจัยในส่วนของขนาดและพลังงานนั้นก็มีเทคนิคการใช้ทรัพยากรร่วมซึ่งเป็นเทคนิคที่สามารถลดขนาดของวงจรให้มีขนาดเล็กลงได้มากและผลที่ตามมาเมื่อวงจรมีขนาดเล็กลงพลังงานที่ใช้ก็จะลดลงตามไปด้วย โดย

สามารถอธิบายการทำงานได้ดังนี้คือ เทคนิคการใช้ทรัพยากรร่วมกันเป็นขั้นตอนปกติในการสังเคราะห์วงจรโดยจะเกี่ยวข้องกับการลดจำนวนตัวดำเนินการ (Operator) ให้น้อยลงดังนั้นการใช้ทรัพยากรร่วมเพื่อลดขนาดพื้นที่ของวงจรโดยจัดรูปแบบวงจรและทำการใช้ทรัพยากรร่วมในส่วนที่ทำได้ให้มากที่สุดเพื่อให้ได้ประสิทธิภาพที่ดีที่สุด ซึ่งการใช้ทรัพยากรร่วมกันจะเกี่ยวข้องกับจำนวนมัลติเพล็กซ์เซอร์ (Multiplexer) และ การเชื่อมต่อสัญญาณกล่าวคือเมื่อมีการใช้ทรัพยากรร่วมพบว่าต้องมีตัวมัลติเพล็กซ์เซอร์คอยทำหน้าที่สับเปลี่ยนชุดข้อมูลเพื่อใช้ตัวร่วมนั้นโดยจะมีสายสัญญาณเลือก (Select signal) เป็นตัวทำหน้าที่เลือกข้อมูล แต่ละชุดเข้าทำงานในเวลาที่แตกต่างกัน โดยภาพประกอบที่ 2-20 แสดงชุดข้อมูลเปรียบเทียบทรัพยากรที่มีการใช้ร่วมกัน และทรัพยากรที่ไม่ได้มีการใช้ร่วมกันดังภาพประกอบที่ 2-20



ภาพประกอบ 2-20 เปรียบเทียบวงจรที่มีการใช้ทรัพยากรการร่วมกันและไม่ได้ใช้ร่วมกันรวมถึงการแบ่งช่วงเวลาที่เหมาะสมกับการทำงานโดยที่

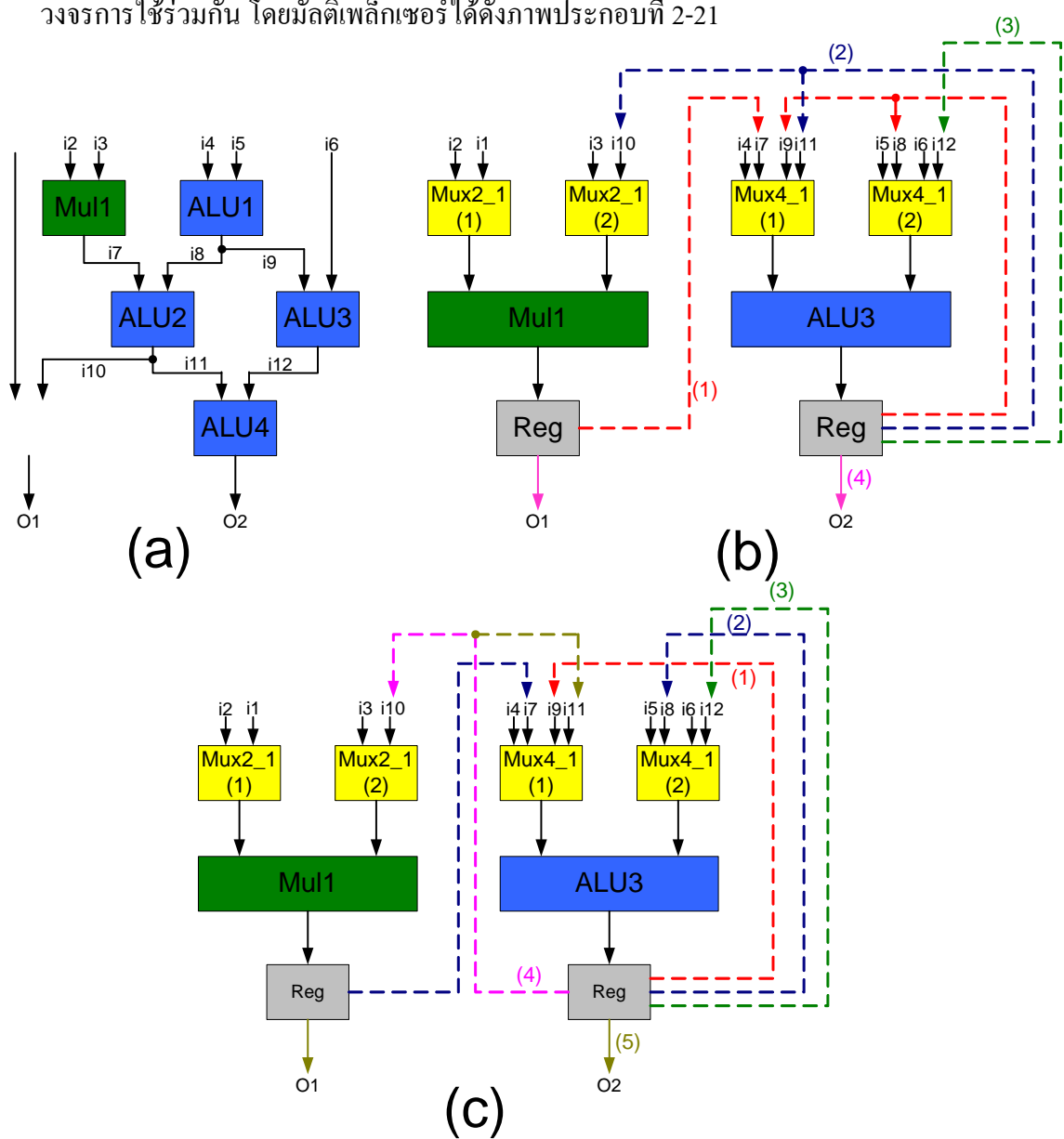
- (a) วงจรที่ไม่มีการใช้ทรัพยากรร่วมกันและเลือกแบ่งการทำงานออกเป็น 1 ช่วง
- (b) วงจรที่มีการใช้ทรัพยากรร่วมกันและเลือกแบ่งการทำงานออกเป็น 4 ช่วง
- (c) วงจรที่มีการใช้ทรัพยากรร่วมกันและเลือกแบ่งการทำงานออกเป็น 5 ช่วง

[EN-SHOU CHANG and DANIEL D. GAJSKI, 1996]

เมื่อพิจารณาภาพประกอบที่ 2-20 จะพบว่าในส่วน (a) นั้นวงจรจะมีการใช้ตัวคูณ 2 ตัวและตัวบวก 4 ตัว ซึ่งนับว่าเป็นการใช้ทรัพยากรที่มากและสิ้นเปลือง เมื่อเทียบกับวงจรในส่วน (b) และ (c) ซึ่งจะพบว่าในทั้ง 2 ส่วน จะมีการใช้ตัวคูณ และตัวบวก เพียงอย่างละ 1 ตัวเท่านั้นซึ่ง



เมื่อพิจารณาภาพรวมทั้ง 3 แบบที่มีการเปรียบเทียบนั้น ตามปกติเราจะต้องคำนึงถึงขนาดและความเร็วควบคู่กันไปในการใช้ทรัพยากรร่วม ซึ่งเมื่อพิจารณาในส่วน (a) จะพบว่า ในส่วน (a) ใช้เวลาในการประมวลผลน้อยที่สุดแต่ในทางตรงกันข้ามก็พบว่าใช้พื้นที่มากที่สุดเช่นกัน จึงได้มีการลดพื้นที่โดยการใช้อยู่ในส่วน ตัวคูณและตัวบวกร่วมกัน กล่าวคือในส่วน (b) และ (c) นั้นได้มีการนำมัลติเพล็กซ์เซอร์มาช่วยในการใช้ทรัพยากรร่วมโดยใช้ตัวมัลติเพล็กซ์เซอร์ในการสลับเปลี่ยนข้อมูลเมื่อพิจารณาจากช่วงเวลาที่ได้มีการแบ่งการทำงานจะพบว่า ในตัวคูณ 2 ตัวและในตัวบวก 4 ตัวนั้นไม่ได้มีการใช้ในช่วงเวลาเดียวกันจึงทำให้ตัวคูณใช้ร่วมกันได้เช่นเดียวกับตัวบวก ซึ่งสามารถวาดวงจรการใช้ร่วมกัน โดยมัลติเพล็กซ์เซอร์ได้ดังภาพประกอบที่ 2-21

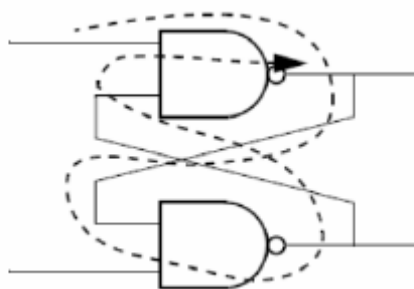


ภาพประกอบ 2-21 แผนภาพระดับเกตของวงจรที่มีการใช้ทรัพยากรร่วมและไม่ใช้ทรัพยากรร่วม

ข้อดีของการใช้เทคนิคการใช้ทรัพยากรร่วมช่วยในการออกแบบวงจรคือ จะทำให้วงจรมีขนาดเล็กลงเนื่องจากจะช่วยลดการสร้างตัวดำเนินการ เช่น ตัวคูณ, ตัวบวก และตัวเปรียบเทียบเป็นต้นซึ่งตัวดำเนินการเหล่านี้เมื่อมีการสร้างขึ้นใหม่จะพบว่าใช้ขนาดในการสร้างมาก ดังนั้นเมื่อมีการใช้เทคนิคการใช้ทรัพยากรร่วมมาช่วยก็จะส่งผลทำให้ ช่วยลดตัวดำเนินการที่มีการเรียกใช้ในจังหวะเวลาที่ต่างกันส่งผลทำให้วงจรมีขนาดเล็กลง แต่ถ้าในการออกแบบมีการใช้ทรัพยากรร่วมที่ไม่เหมาะสมจะส่งผลทำให้ขนาดของวงจรใหญ่ขึ้นมีความซับซ้อนของวงจรมากขึ้นและทำให้วงจรประมวลผลช้าลงรวมถึงมีพลังงานสูงขึ้นด้วย โดยข้อเสียดังกล่าวจะเกิดจากการที่มีการสร้างตัวมัลติเพล็กซ์เซอร์เพื่อการใช้ทรัพยากรที่มากขึ้น และเมื่อมีการเรียกใช้มากจะส่งผลทำให้การเชื่อมต่อภายในวงจรมากขึ้นจึงทำให้เกิดความซับซ้อน นอกจากนี้ยังก่อให้เกิดความล่าช้าในการประมวลผลเนื่องจากในการใช้ทรัพยากรร่วมแบบไม่เหมาะสมนั้นจะทำให้การประมวลผลช้าไปมากเปรียบเทียบได้ดังภาพประกอบที่ 2-18 ซึ่งเมื่อเทียบระหว่างวงจรในส่วน (a) และวงจรในส่วน (b) จะพบว่าเวลาที่ใช้ในการประมวลผลนั้นช้าลงเกือบ 2 เท่าเวลาในการประมวลผลส่วน (a) นั้นใช้เพียง 380 ns แต่ในส่วน (b) ใช้ถึง 600 ns ผลสืบเนื่องมาจากการใช้ทรัพยากรร่วมแบบไม่เหมาะสม แต่เมื่อมีการใช้ทรัพยากรร่วมแบบเหมาะสมจะพบได้ดังส่วน (c) โดยสังเกตได้ว่าในส่วน (c) เป็นส่วนที่แบ่งช่วงเวลางานมากที่สุดเมื่อเทียบกับส่วน (a) และ (b) แต่ผลลัพธ์รวมในวงจรกับมีเวลาที่ใช้ในการประมวลผลทั้งวงจรน้อยกว่าเนื่องจากส่วน (c) ได้มีการคำนึงถึงเส้นทางที่ยาวที่สุด (critical-path) โดยทำการแบ่งช่วงเวลาในการคูณ ออกเป็นการใช้ 2 ช่วงเวลาเพื่อทำการคูณควบคู่กับการบวกในการกระทำแบบนี้จะช่วยส่งผลทำให้เส้นทางที่ยาวที่สุดลดลง และไม่กระทบต่อการใช้ทรัพยากรร่วมในส่วนอื่นๆ ซึ่งเมื่อเปรียบเทียบจะพบว่าในวงจรส่วน (a) นั้นเส้นทางที่ยาวที่สุดยาวถึง 380 ns และในส่วน (b) ยาว 150 ns แต่ที่ส่วนนี้นั้นใช้เวลาในการประมวลผลช้ากว่าส่วน (a) เนื่องจากในส่วนนี้ได้มีการแบ่งช่วงเวลาออกเป็น 4 ช่วงซึ่งจะพบว่าในช่วงเวลาที่ 1 และ 3 นั้น เราเสียเวลาในการรอวงจรคูณทำงานโดยเปล่าประโยชน์ไป 70 ns ต่างกับในช่วง (c) ซึ่งมีการใช้ทรัพยากรร่วมอย่างเหมาะสมคือจะมีเวลาที่รอในเส้นทางที่ยาวที่สุดเพียง 5 ns เท่านั้นส่งผลทำให้วงจรนี้ใช้เวลาในการประมวลผลนานกว่าวงจรในส่วน (a) เล็กน้อย แต่สามารถลดตัวคูณลงได้ 1 ตัวและตัว บวกลงได้ 3 ตัวถือว่าคุ้มค่าเมื่อเทียบกับขนาดที่ลดลงมากกับเวลาที่เพิ่มขึ้นเพียงเล็กน้อย ปัญหาที่พบในด้านพลังงาน ซึ่งจะกล่าวถึงในหัวข้อถัดไปเกี่ยวกับการบริโภคพลังงานและเทคนิคที่ช่วยในการประหยัดพลังงาน

**2.8 ข้อควรคำนึงในการใช้เทคนิคการใช้ทรัพยากรร่วม**

เมื่อมีการใช้เทคนิคการใช้ทรัพยากรร่วมเพื่อลดความซับซ้อนส่วนฮาร์ดแวร์ ในการเขียนโปรแกรมระดับ RTL (Register Transfer Level) พบว่าสามารถเกิดปัญหา Combination feedback loop ขึ้นได้ง่ายมาก เช่น เวลาที่ใช้ในการเคลื่อนที่จากอินพุตไปยังเอาต์พุตของ เกตแนนด์ (NAND gate) ซึ่งสามารถแสดงปัญหา Combination feedback loop โดยเส้นประดัง ภาพประกอบที่ 2-22



ภาพประกอบ 2-22 ปัญหา Combination feedback loop [Shushan Qiao, 2007]

**2.8.1 วิเคราะห์ปัญหา Combination feedback loop ของการใช้ทรัพยากรร่วม**

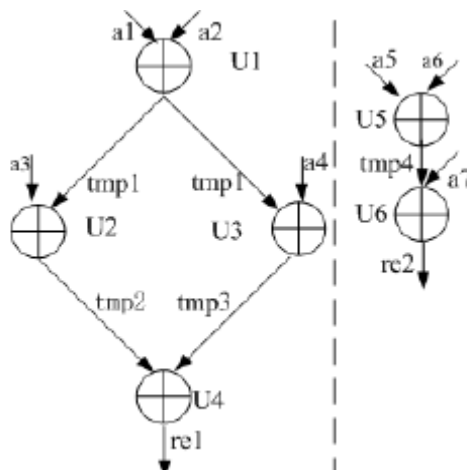
แบบต่างๆ

ในการวิเคราะห์ปัญหา Combination feedback loop นั้นเพื่อให้เห็นเห็นความกระจ่างจึงได้มีการยกตัวอย่างวงจร 1 วงจรโดยแสดงเป็น โปรแกรม และ DFG ของวงจรได้ดัง ภาพประกอบที่ 2-23

```

always@*
begin
  if (clock_cycle==1)
  begin
    tmp1=a1+a2;
    tmp2=a3+tmp1;
    tmp3=tmp1+a4;
    re1=tmp2+tmp3;
  end
  else if (clock_cycle==2)
  begin
    tmp4=a5+a6;
    re2=tmp4+a7;
  end
  .....
end
    
```

(a)



(b)

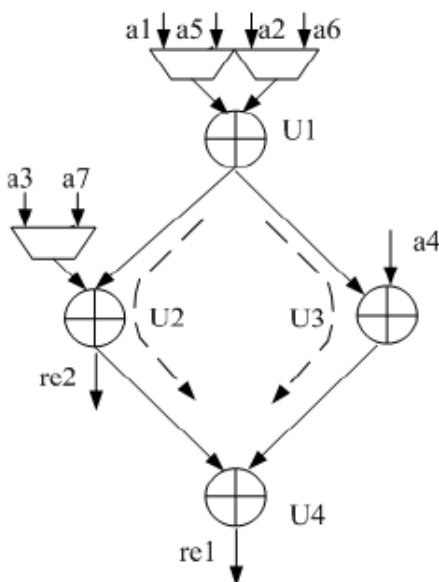
ภาพประกอบ 2-23 (a) โปรแกรมแสดงวงจรที่นำมาวิเคราะห์ (b) DFG แสดงวงจรที่นำมาวิเคราะห์

[Shushan Qiao, 2007]

โดยเมื่อนำวงจรที่ยกตัวอย่างดังภาพประกอบที่ 2-23 มาวิเคราะห์พบว่า ในวงจรนั้นมีการทำงาน 2 รอบสัญญาณนาฬิกา (Clock cycles) ซึ่งเมื่อพิจารณาในส่วนของ DFG ในภาพประกอบ 3-10 (b) จะพบว่า การทำงานแบ่งออกเป็น 2 รอบสัญญาณนาฬิกา แบ่งแยกโดยเส้นประ เราสามารถที่จะใช้เทคนิคการใช้ทรัพยากรร่วมแบ่งได้เป็น 3 รูปแบบโดยสามารถเขียนอธิบายถึงการทำงานและปัญหาของ Combination feedback loop ได้ดังต่อไปนี้

2.8.1.1 การวิเคราะห์ห้วงจรโดยใช้ทรัพยากรร่วมวิธีการที่ 1

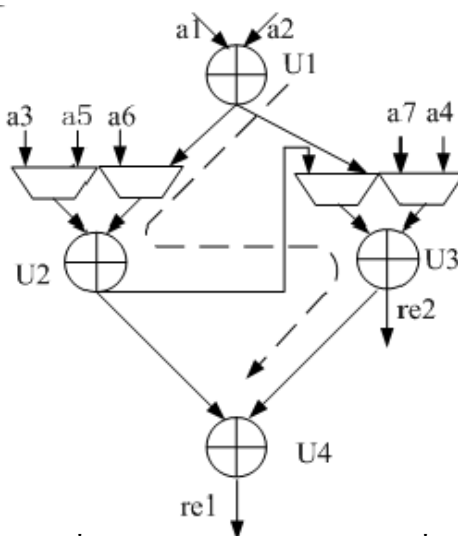
เมื่อทำการใช้ทรัพยากรร่วมในวิธีนี้จะพบว่าในวิธีที่จะกล่าวนี้ดีกว่า 2 วิธีที่กำลังจะอธิบายในหัวข้อไปเนื่องจากในกระบวนการนี้มีการทำงานที่ใช้เวลาปกติแต่ปัญหาในกระบวนการนี้คืออาจเกิดความสับสนในการเลือกเส้นทางได้ ซึ่งสามารถแสดงการใช้ทรัพยากรร่วมและเส้นทางในการดำเนินการได้ดังภาพประกอบที่ 2-24 โดยเส้นประจะแสดงถึงการเดินทางของข้อมูลในการประมวลผลเมื่อพิจารณาค่าหน่วยเวลา(Delay) จะได้  $Delay = MUX + U_1 + U_2 + U_4$



ภาพประกอบ 2-24 วงจรที่มีการใช้ทรัพยากรร่วมวิธีการที่ 1 [Shushan Qiao, 2007]

2.8.1.2 การวิเคราะห์ห้วงจรโดยใช้ทรัพยากรร่วมวิธีการที่ 2

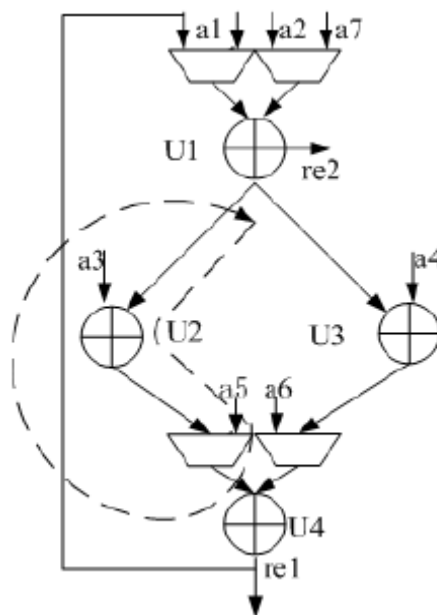
การใช้ทรัพยากรร่วมวิธีการนี้พบว่ามีการใช้เวลาในการประมวลผลนานเนื่องจากเมื่อพิจารณาที่เส้นทางวิกฤต (Critical path) จะพบว่าเมื่อพิจารณาเทียบกับในวิธีการแรกจะเห็นว่าข้อดีของวิธีการนี้คือ ผลลัพธ์ที่ได้มีความถูกต้องแน่นอนแต่ข้อเสียที่พบคือวงจรจะใช้เวลาอย่างมากในการประมวลผลสามารถเทียบกับวิธีการที่ 1 โดยคิดค่านำเวลาของวงจรนี้คือ  $Delay = U_1 + MUX + U_2 + MUX + U_3 + U_4$  โดยสามารถแสดงวิธีการนี้ได้ดังภาพประกอบที่ 2-25



ภาพประกอบ 2-25 วงจรที่มีการใช้ทรัพยากรร่วมวิธีการที่ 2 [Shushan Qiao, 2007]

2.8.1.3 การวิเคราะห์ห่วงจร โดยใช้ทรัพยากรร่วมวิธีการที่ 3

ในวิธีการนี้ถือว่าเป็นวิธีการที่ไม่ควรนำมาใช้มากที่สุดเนื่องจากพบแต่ข้อเสีย กล่าวคือเมื่อพิจารณาจากภาพประกอบที่ 2-26 จะพบว่าการทำงานก็เป็นไปอย่างช้ามากเมื่อมองที่เส้นทางวิกฤต นอกจากนั้นยังอาจเกิดความผิดพลาดของข้อมูลเนื่องจากเมื่อพิจารณาเส้นทางของข้อมูลพบว่ามีเส้นทางย้อนกลับจากเอาท์พุทไปยังอินพุทซึ่ง เมื่อพิจารณาค่าหน่วยเวลาของวิธีการนี้ จะได้  $Delay = MUX + U_1 + U_2 + MUX + U_4$



ภาพประกอบ 2-26 วงจรที่มีการใช้ทรัพยากรร่วมวิธีการที่ 3 [Shushan Qiao, 2007]

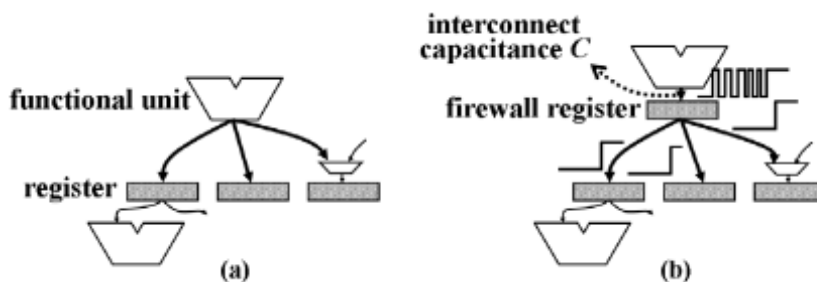
### 2.9 ปัจจัยที่เกี่ยวข้องกับการบริโภคพลังงานและเทคนิคการลดค่าพลังงาน

ประสิทธิภาพในการบริโภคพลังงานเริ่มก้าวขึ้นมาเป็นบทบาทสำคัญในกระบวนการออกแบบวงจรบน FPGA ในเทคโนโลยีแบบนาโน ซึ่งพิจารณาในกระบวนการทำงานพบว่าการจัดการสับเปลี่ยนเส้นทางการเชื่อมต่อกันและ การใช้ทรัพยากรอย่างคุ้มค่าเป็นปัจจัยในการออกแบบวงจรพลังงานต่ำ บน FPGA ซึ่งต้นกำเนิดของการบริโภคพลังงานในวงจรซิมอสคือการชาร์จประจุไฟฟ้า และการ คายประจุไฟฟ้า ของตัวเก็บประจุแต่ละตัว โดยการบริโภคพลังงานในตัวเก็บประจุแต่ละตัวนั้นอธิบายได้สามารถอธิบายได้ดังสมการที่ (2-3)

$$P = \alpha \cdot C_l \cdot V_{dd}^2 \cdot f \tag{2-3}$$

โดยที่  $\alpha$  คือค่าการเปลี่ยนระดับของสัญญาณ (Switching Activity) ซึ่งเป็นค่าเฉลี่ยของการเปลี่ยนแปลงสัญญาณจาก 0 เป็น 1 ภายใน 1 ช่วงวงจรสัญญาณนาฬิกา (Clock-cycle),  $C_l$  คือ โหลดตัวเก็บประจุที่ให้แก่อินพุต  $V_{dd}$  คือ ค่าแหล่งจ่ายแรงดันไฟฟ้า และ  $f$  เป็น ค่าความถี่สัญญาณนาฬิกา (clock-frequency) เมื่อพิจารณาถึงปัจจัยที่จะส่งผลทำให้ค่าการบริโภคพลังงานเพิ่มขึ้นนั้นมีอยู่ 2 ตัวแปรสามารถใช้เทคนิคเพื่อช่วยในการลดค่าพลังงานได้ คือ ค่า  $\alpha$  ซึ่งในค่านี้จะสัมพันธ์กับความซับซ้อนของวงจรซึ่งจะเกี่ยวพันโดยตรงกับการเชื่อมต่อกันภายในวงจร โดยเมื่อพิจารณาพลังงานของทั้งวงจรพบว่าในส่วนของการเชื่อมต่อภายในวงจรมันเป็นส่วนที่บริโภคพลังงานมากที่สุดจึงได้มีการใช้เทคนิคการสร้างรีจิสเตอร์คักข้อมูลเพื่อช่วยลดค่า  $\alpha$  ลงเนื่องจากตัวรีจิสเตอร์คักข้อมูลนี้จะคอยตรวจสอบผลก่อนที่จะทำการเปลี่ยนแปลงสัญญาณ ส่งผลทำให้ค่าการบริโภคพลังงานลดลงได้โดยสามารถแสดงการแทรกตัวรีจิสเตอร์คักข้อมูลลงในวงจรได้ดังภาพประกอบที่

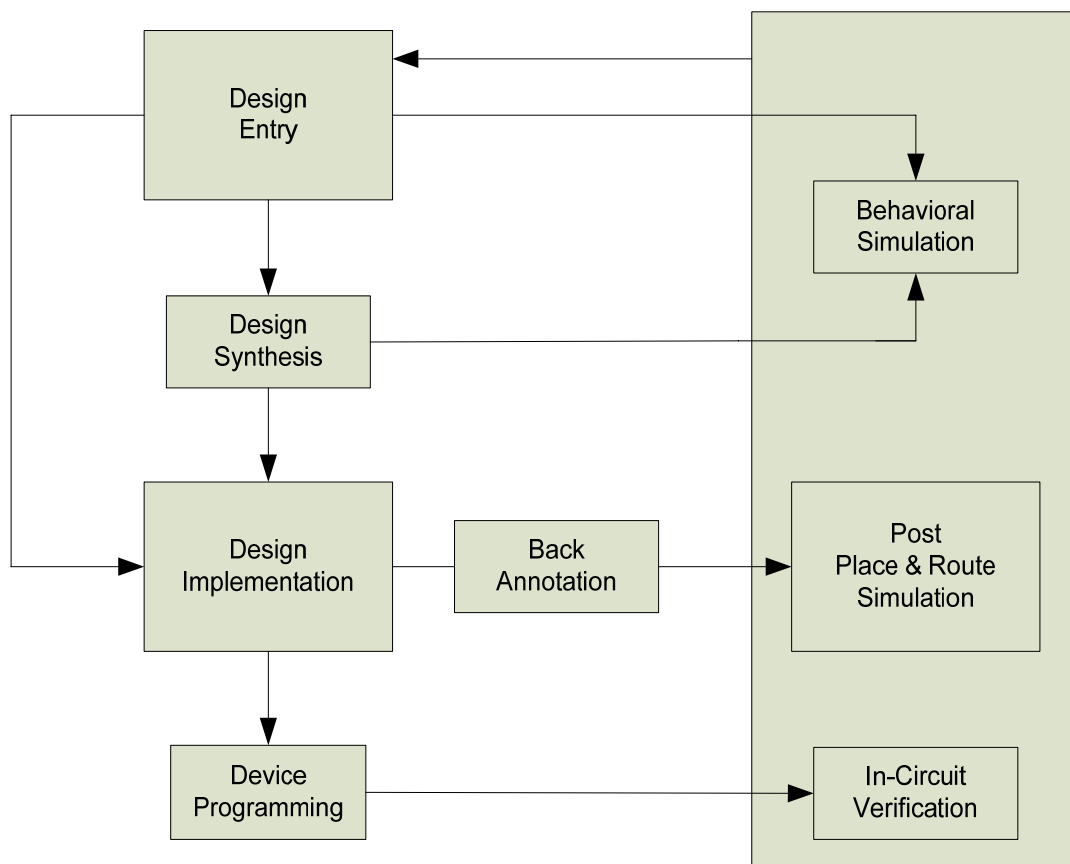
2-27



ภาพประกอบ 2-27 การแทรกรีจิสเตอร์คักข้อมูลลงในวงจร [Cheng-Tao, 2008]

## 2.10 เทคโนโลยี Field Programmable Gate Array (FPGA)

FPGA คือ วงจรรวมที่สามารถทำการโปรแกรมการเชื่อมต่อให้เกิดเป็นวงจรดิจิทัลได้ โดยภายในตัวชิพจะประกอบไปด้วย logic-gate ต่อกันจำนวนมากภายในลักษณะของอาร์เรย์ โดยที่ logic-gate สามารถทำการต่อเชื่อมกันได้ตามความต้องการ โดยขนาดความจุหรือจำนวน logic-gate ขึ้นอยู่กับตระกูลและรุ่นของชิพ FPGA ทำให้สามารถลดจำนวนของไอซีบนบอร์ดลงได้ การออกแบบและสร้างวงจรดิจิทัลบน FPGA สามารถทำได้หลายรูปแบบโดยมีขั้นตอนดังภาพประกอบที่ 2-28



ภาพประกอบ 2-28 ขั้นตอนการออกแบบวงจรดิจิทัลบน FPGA

จากภาพประกอบที่ 2-28 จะสามารถอธิบายการทำงานได้ดังนี้ ในขั้นแรกคือส่วนของการ Design Entry ในส่วนนี้กล่าวคือจะมีการเขียนโปรแกรมซึ่งในขั้นนี้สามารถตรวจสอบการทำงานได้ โดยการ Behavioral Simulation เพื่อตรวจสอบว่าสิ่งที่เราออกแบบมาสามารถใช้งานได้หรือไม่โดยที่ยังไม่เกี่ยวกับ Delay ซึ่งในส่วนถัดไปคือการสังเคราะห์วงจรและในส่วนถัดไปก็คือการ Implement ว่าสามารถใช้งานจริงได้หรือไม่โดยส่วนนี้สามารถตรวจสอบได้โดยการทำ Post Place-

route Simulation ซึ่งจะต่างจากในส่วนของ Behavioral Simulation คือในส่วนนี้ได้นำ Delay มาคิด ซึ่งจะเป็นการจำลองการทำงานที่เหมือนการนำไปลงอุปกรณ์จริง ในส่วนถัดไปคือการนำโปรแกรมไปโปรแกรมลงในอุปกรณ์ (Device Programming) และทดลองนำสัญญาณไปวัดได้โดยอาศัยเครื่องมือเช่น Oscilloscope



## บทที่ 3

### การออกแบบวงจร

#### 3.1 เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมบนวงจรฟิลเตอร์เบงก์

ในการออกแบบวงจรฟิลเตอร์เบงก์โดยใช้เทคนิคการใช้ทรัพยากรร่วมนั้นจากการศึกษาวงจรฟิลเตอร์เบงก์และการใช้ทรัพยากรร่วมจากบทที่ 2 เราสามารถเห็นความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของฟิลเตอร์แต่ละตัวภายในฟิลเตอร์เบงก์โดยสามารถเขียนเป็นกลุ่มของสมการได้ดังภาพประกอบที่ 3-1 และเมื่อพิจารณาจากความสัมพันธ์แล้วสามารถวาดวงจรในรูปแบบของแผนภาพกระแส (DFGs: Data Flow Graphs) ได้ดังภาพประกอบที่ 3-2

$$\begin{aligned} H1(z) : y_1(n) &= (x_0 + x_{48}) * h_0 + (x_{16} + x_{32}) * h_1 + x_{24} * h_2 \\ y_{1c}(n) &= x_{24} - y_1(n) \end{aligned}$$

$$\begin{aligned} H2(z) : y_2(n) &= (x_0 + x_{24}) * h_0 + (x_8 + x_{16}) * h_1 + x_{12} * h_2 \\ y_{2c}(n) &= x_{12} - y_2(n) \end{aligned}$$

$$\begin{aligned} H3(z) : y_3(n) &= (x_0 + x_{28}) * h_0 + (x_4 + x_{24}) * h_1 + (x_8 + x_{20}) * h_2 + (x_{12} + x_{16}) * h_3 + x_{14} * h_4 \\ y_{3c}(n) &= x_{14} - y_3(n) \end{aligned}$$

$$\begin{aligned} H4(z) : y_4(n) &= (x_0 + x_{12}) * h_0 + (x_4 + x_8) * h_1 + x_6 * h_2 \\ y_{4c}(n) &= x_6 - y_4(n) \end{aligned}$$

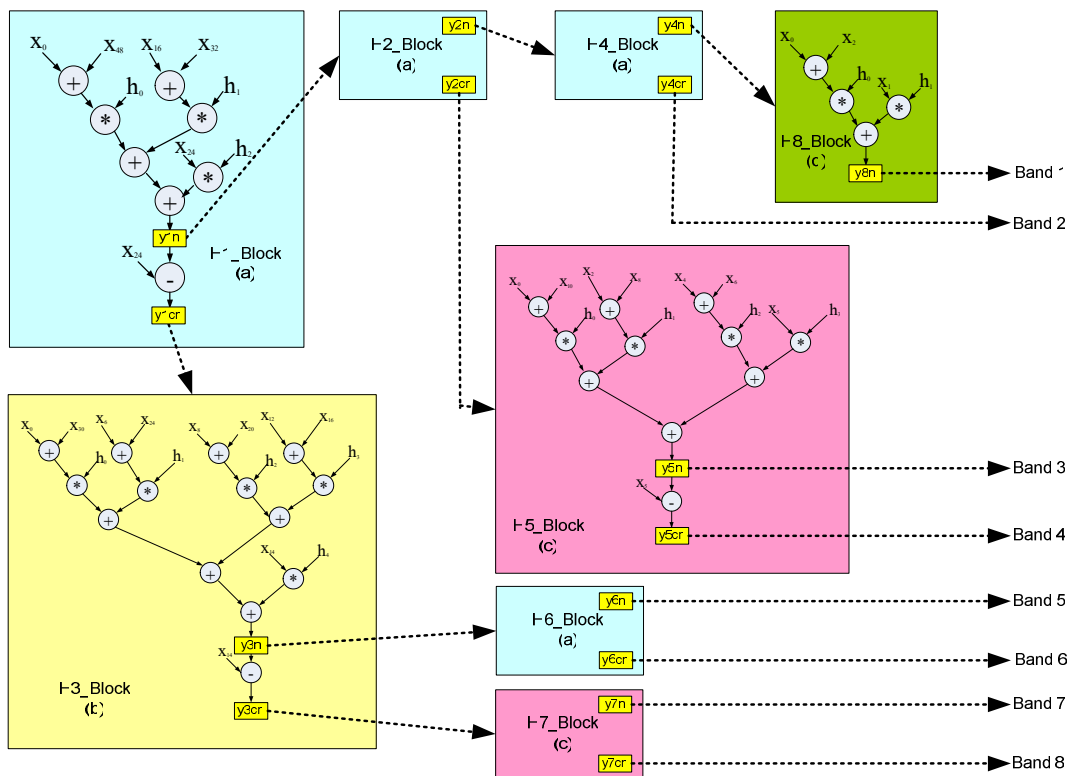
$$\begin{aligned} H5(z) : y_5(n) &= (x_0 + x_{10}) * h_0 + (x_2 + x_8) * h_1 + (x_4 + x_6) * h_2 + x_5 * h_3 \\ y_{5c}(n) &= x_5 - y_5(n) \end{aligned}$$

$$\begin{aligned} H6(z) : y_6(n) &= (x_0 + x_6) * h_0 + (x_2 + x_4) * h_1 + x_3 * h_2 \\ y_{6c}(n) &= x_3 - y_6(n) \end{aligned}$$

$$\begin{aligned} H7(z) : y_7(n) &= (x_0 + x_{30}) * h_0 + (x_6 + x_{24}) * h_1 + (x_{12} + x_{18}) * h_2 + x_{15} * h_3 \\ y_{7c}(n) &= x_{15} - y_7(n) \end{aligned}$$

$$H8(z) : y_8(n) = (x_0 + x_2) * h_0 + x_1 * h_1$$

ภาพประกอบ 3-1 ความสัมพันธ์ระหว่างอินพุตและเอาต์พุตของฟิลเตอร์เบงก์



ภาพประกอบ 3-2 DFGs ของวงจรฟิลเตอร์แบ่งกั 8 ช่วงความถี่

หลังจากได้มีการศึกษาวงจรฟิลเตอร์แบ่งกัแล้วจะพบว่าวงจรฟิลเตอร์แบ่งกันั้นมีขนาดใหญ่ทำให้มีการนำเทคนิคการใช้ทรัพยากรร่วมมาใช้เพื่อช่วยลดขนาดของวงจรให้มีขนาดเล็กลงโดยในเทคนิคการใช้ทรัพยากรร่วมนั้นสามารถแบ่งออกได้เป็น 2 แบบคือ

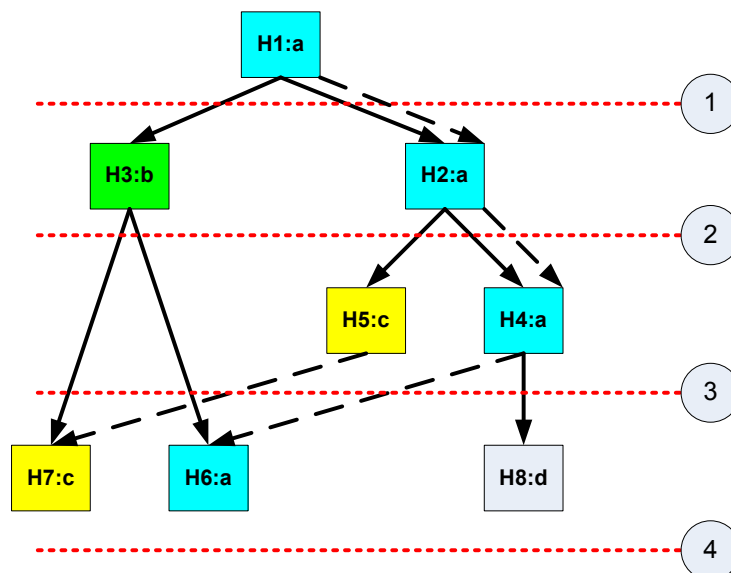
### 3.1.1 การใช้ทรัพยากรร่วมแบบลำดับชั้น (Hierarchical Resource-Sharing)

การใช้ทรัพยากรร่วมแบบลำดับชั้นได้มีการแบ่งออกเป็น 2 ส่วนคือ

#### 3.1.1.1 การใช้ทรัพยากรร่วมระหว่างบล็อก (Inter-Block Sharing)

คือการใช้ทรัพยากรร่วมกันระหว่างบล็อกในระดับบนสุดของ DFGs โดยที่การใช้ทรัพยากรร่วมกันระหว่างบล็อกนี้จะถูกนำมาพิจารณาเมื่อฟังก์ชันภายในของทั้ง 2 บล็อกนั้นเหมือนกันและนอกจากนี้หากบล็อก 2 บล็อกหรือหลายๆ บล็อกมีโครงสร้างของ DFGs ที่เหมือนกันนั้นก็ยังสามารถทำการออกแบบโดยวิธีนี้ได้เช่นเดียวกันซึ่งจากสมการวงจรดิจิทัลฟิลเตอร์แบ่งกัในภาพประกอบที่ 3-1 สามารถเขียน DFG การใช้ทรัพยากรร่วมระหว่างบล็อกได้ดัง

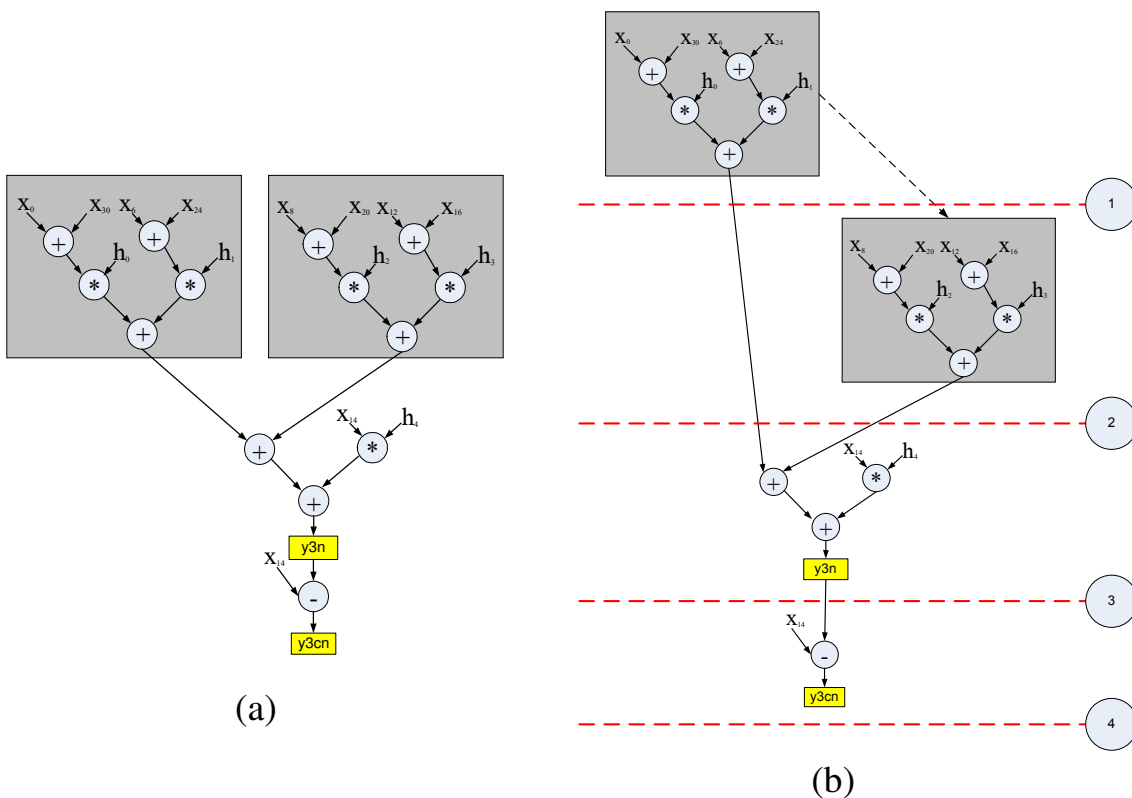
ภาพประกอบที่ 3-3 โดยสามารถแบ่งบล็อกที่เหมือนกันได้ 4 ชนิด ดังนี้ ชนิด a ประกอบด้วย H1 H2 H4 และ H6 ชนิด b ประกอบด้วย H3 ชนิด c ประกอบด้วย H5 และ H7 ชนิด d ประกอบด้วย H8 เพียงบล็อกเดียว (โดยบล็อกที่มีการใช้ทรัพยากรร่วมระหว่างบล็อกจะมีสีที่เหมือนกันคือมีการใช้ทรัพยากรร่วมกัน) ซึ่งเส้นลูกศรทึบ ( $\rightarrow$ ) หมายถึง บล็อกที่มาก่อน (Predecessor) และบล็อกที่มาหลัง (Successor) ซึ่งมีความสัมพันธ์ทางข้อมูลขึ้นต่อกัน (Data dependent) กล่าวคือ บล็อกที่มาหลังต้องรอข้อมูลที่สมบูรณ์จากบล็อกก่อนหน้า โดยเมื่อพิจารณาภาพประกอบที่ 3-3 สามารถอธิบายได้ว่าบล็อก H2 และ บล็อก H3 นั้นจะทำงานได้ต้องรอให้ บล็อก H1 ทำเสร็จสมบูรณ์แล้วเช่นเดียวกันกับ บล็อก H4 และ บล็อก H5 ต้องรอให้บล็อก H2 ทำงานเสร็จสมบูรณ์เช่นกัน ส่วนเส้นลูกศรประ ( $- \rightarrow$ ) หมายถึงบล็อกที่มาก่อนและบล็อกที่มาหลังมีความสัมพันธ์ทางทรัพยากรขึ้นต่อกัน (Resource dependent) กล่าวคือ ทั้งสองบล็อกใช้ทรัพยากรร่วมกัน ดังนั้นบล็อกที่มาหลังต้องรอจนกระทั่งบล็อกก่อนหน้าประมวลผลเสร็จสมบูรณ์ก่อน จึงเริ่มทำงานได้ยกตัวอย่างจากภาพประกอบที่ 3-3 จะได้ว่าเมื่อพิจารณาในบล็อก H7 นั้นผลลัพธ์ที่ได้จากบล็อก H5 จะเป็นค่าเอาต์พุตแล้วแต่เนื่องจากมีเส้นประซึ่งแสดงถึงว่าเมื่อบล็อก H5 ทำงานเสร็จสิ้นสมบูรณ์จะมีการนำทรัพยากรส่วนนั้นไปใช้ร่วมกับ H7 ซึ่ง บล็อก H7 เป็นบล็อกที่มาทีหลังบล็อก H5 จึงต้องรอให้บล็อก H5 ทำงานเสร็จสมบูรณ์ก่อนที่จะใช้ทรัพยากรร่วมกัน



ภาพประกอบ 3-3 DFG แบบลำดับชั้นของวงจรฟิลเตอร์แบงก์ที่มีการใช้ทรัพยากรระหว่างบล็อก

### 3.1.1.2 การใช้ทรัพยากรร่วมภายในบล็อก (Intra-Block Sharing)

คือมีการใช้ทรัพยากรร่วมกันภายในบล็อกของตัวเองกล่าวคือพิจารณาจากสมการภายในบล็อกตัวเองว่ามีส่วนที่คล้ายคลึงกันเพื่อที่จะนำมาใช้ร่วมกันและบล็อกที่มีการใช้ทรัพยากรร่วมระหว่างบล็อกแล้วจะไม่สามารถนำมาใช้ทรัพยากรร่วมภายในบล็อกได้อีกดังนั้นในวงจรดิจิทัลฟิลเตอร์เบงก์แบบ 8 ช่วงความถี่จึงมีบล็อกที่มีการใช้ทรัพยากรร่วมภายในบล็อกเพียง 1 บล็อกคือ บล็อก H3 ซึ่งสามารถแสดง ได้ดังภาพประกอบที่ 3-4



ภาพประกอบ 3-4 DFG แบบลำดับชั้นของวงจรฟิลเตอร์เบงก์

(a) ก่อนใช้ทรัพยากรร่วมภายในบล็อก

(b) หลังใช้ทรัพยากรร่วมภายในบล็อก

### 3.1.2 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม (Global Resource-Sharing)

การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมจะต่างกับการใช้ทรัพยากรร่วมแบบลำดับชั้น ตรงที่จะมีการมองส่วนที่คล้ายคลึงกันทั้งวงจรรวมว่ามีส่วนใดที่เหมือนกันและสามารถนำมาใช้ร่วมกันได้โดยไม่จำเป็นต้องเหมือนกันทั้งบล็อก ซึ่งการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมจะส่งผลการใช้ทรัพยากรร่วมแบบนี้มีขนาดเล็กกว่าวิธีการมองแบบลำดับชั้น เพราะการ

ใช้ทรัพยากรร่วมแบบภาพรวมสามารถดึงส่วนที่คล้ายคลึงกันในบางส่วนมาทำการใช้ร่วมกันได้มากทำให้ลดปัญหาการที่จะต้องทำการสร้างวงจรขึ้นใหม่ แต่ปัญหาที่พบคือถ้าไม่มีการออกแบบให้มีการใช้ทรัพยากรร่วมที่เหมาะสมจะส่งผลเสียทำให้เกิดการใช้พลังงานมากและอาจจะมีการใช้พื้นที่มากเนื่องจากมีการสร้าง MUX มากขึ้น ดังนั้นในการออกแบบการใช้ทรัพยากรร่วมแบบการพิจารณาภาพรวมจะต้องมีการคำนึงถึงปัจจัยในการออกแบบซึ่งสามารถอธิบายโดยละเอียดในหัวข้อถัดไป

### 3.2 ปัจจัยในการออกแบบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม

ในการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมนั้น เมื่อทำการใช้เทคนิคนี้โดยไม่ได้มีการคำนึงถึงปัจจัยในการออกแบบอาจจะส่งผลเสียต่อวงจรมากกว่าผลดี กล่าวคือภายในวงจรอาจจะมีการทำงานที่ช้าลงเนื่องจาก การใช้ทรัพยากรร่วมที่ไม่เหมาะสม อาจจะทำให้เกิดปัญหาพื้นที่ของวงจรมีขนาดมากขึ้นแทนที่จะลดลงโดยเกิดจากตัว MUX ที่ต้องสร้างขึ้นเมื่อมีการใช้ทรัพยากรร่วม และ พลังงานอาจจะสูงขึ้นเนื่องจากการเปลี่ยนระดับของสัญญาณ ดังนั้นการที่จะนำเทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมมาใช้เพื่อให้ได้ผลที่ต้องการจะต้องคำนึงถึงปัจจัยในการออกแบบโดยมีปัจจัยที่พิจารณาอยู่ 4 ข้อดังนี้

#### 3.2.1 การพิจารณาค่าสัมประสิทธิ์

กล่าวคือ ถ้าเราจัดรูป สมการที่จะนำมาใช้ร่วมกัน โดยให้สมการนั้นมีค่าสัมประสิทธิ์เหมือนกันจะส่งผลทำให้ไม่ต้องมีการสร้าง MUX เพื่อนำมาสลับค่าสัมประสิทธิ์โดยค่าสัมประสิทธิ์ดังนั้นในปัจจุบันนี้สามารถช่วยลดปัญหาในด้านขนาดที่เกิดจากการสร้าง MUX โดยไม่จำเป็นออกได้โดยค่าสัมประสิทธิ์ของวงจรฟิลเตอร์เบงก์สามารถแสดงได้ดังภาพประกอบที่ 3-5

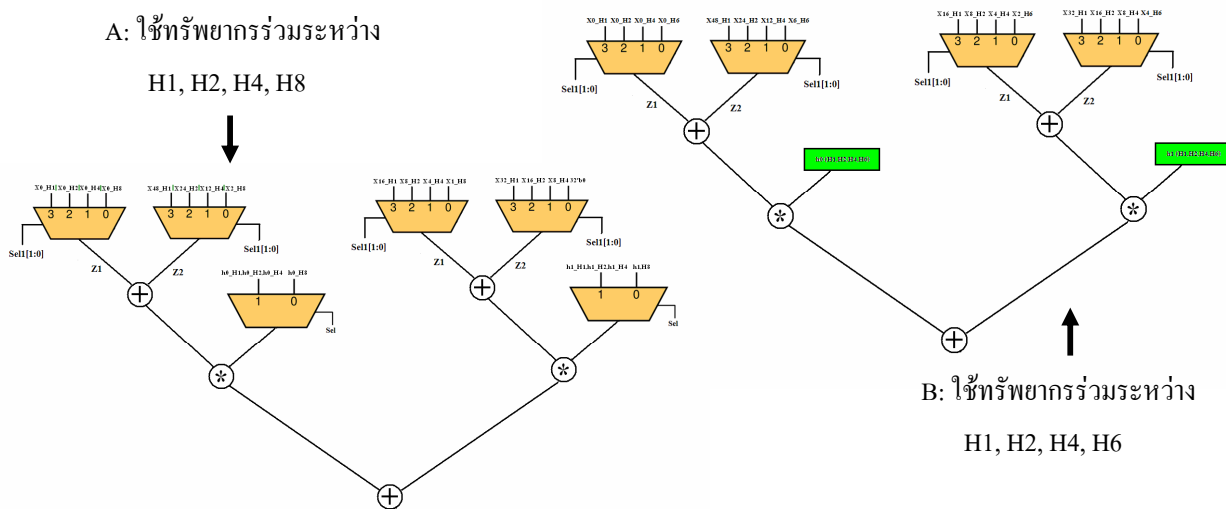
Filter	H(n)	Decimal Floating point value
<b>H1(z)</b>	h0 : h (0) = h (48)	-0.05062 4178 42547
	h1 : h (16) = h (32)	0.2950 5933 4702 99
	h2 : h (24)	0.5
<b>H2(z)</b>	h0 : h (0) = h (24)	-0.05062 4178 42547
	h1 : h (8) = h (16)	0.2950 5933 4702 99
	h2 : h (12)	0.5
<b>H3(z)</b>	h0 : h (0) = h (28)	-0.00373 7655 73262
	h1 : h (4) = h (24)	0.0205 6989 4870 10
	h2 : h (8) = h (20)	-0.07232 1904 7068 9
	h3 : h (12) = h (16)	0.3059 7047 3625 44
	h4 : h (14)	0.5
<b>H4(z)</b>	h0 : h (0) = h (12)	-0.05062 4178 42547
	h1 : h (4) = h (8)	0.2950 5933 4702 99
	h2 : h (6)	0.5

Filter	H(n)	Decimal Floating point value
<b>H5(z)</b>	h0 : h (0) = h (10)	0.0130 4920 5552 05
	h1 : h (2) = h (8)	-0.06387 1512 1040 5
	h3 : h (4) = h (6)	0.3016 1294 8075 61
	h2 : h (5)	0.5
<b>H6(z)</b>	h0 : h (0) = h (6)	-0.05062 4178 42547
	h1 : h (2) = h (4)	0.2950 5933 4702 99
	h2 : h (3)	0.5
<b>H7(z)</b>	h0 : h (0) = h (30)	0.0130 4920 5552 05
	h1 : h (6) = h (24)	-0.06387 1512 1040 5
	h3 : h (12) = h (18)	0.3016 1294 8075 61
	h2 : h (15)	0.5
	h4 : h (14)	0.5
<b>H8(z)</b>	h0 : h (0) = h (2)	0.2950 5933 4702 99
	h1 : h (1)	0.5

ภาพประกอบ 3-5 ค่าสัมประสิทธิ์ของวงจรถ่ายฟิลเตอร์แบ่งกึ่งทั้ง 8 บล็อก

3.2.2 พิจารณาความเหมือนของโครงสร้างสมการและความซับซ้อนของสมการ

กล่าวคือ ถ้าการใช้ทรัพยากรร่วมกันมีการใช้ทรัพยากรร่วมกันมากเกินไปจนเกิดความซับซ้อนจะส่งผลทำให้วงจรมันเกิดความผิดพลาดได้สูงและทำให้การคำนวณสัญญาณนาฬิกา มีโอกาสผิดพลาดได้สูงเช่นกันสามารถยกตัวอย่างเปรียบเทียบได้ดังภาพประกอบที่ 3-6

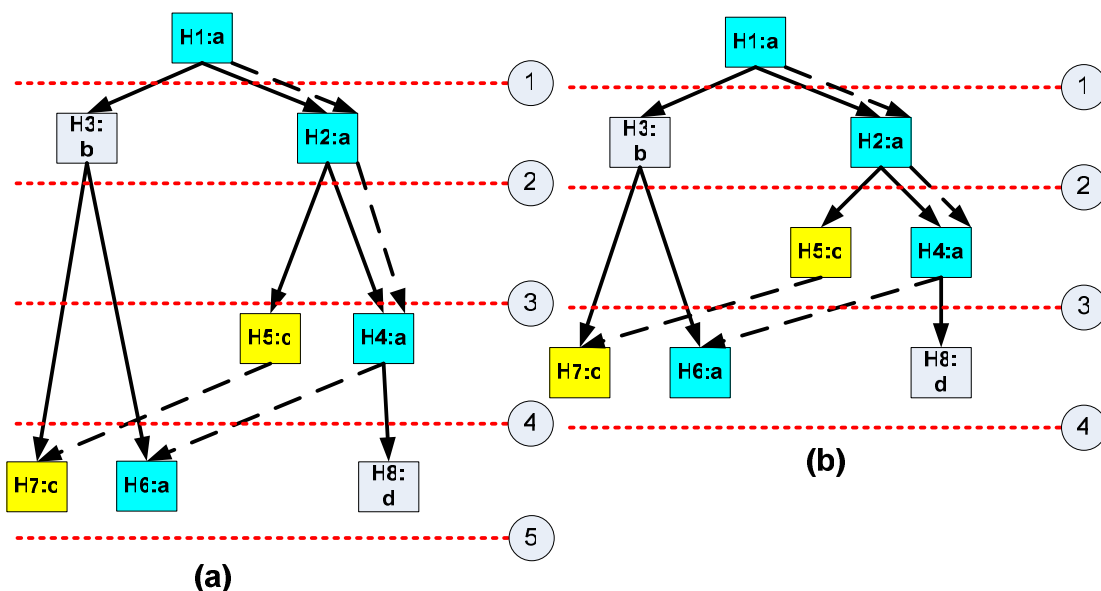


ภาพประกอบ 3-6 A: โครงสร้างการใช้ทรัพยากรร่วมแบบซับซ้อน

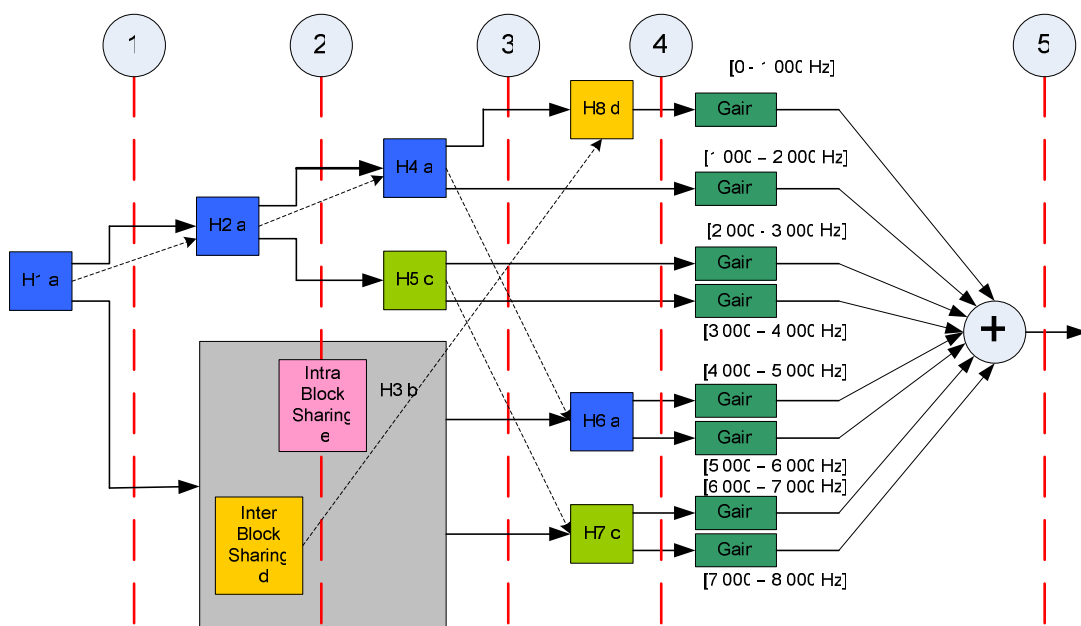
B: โครงสร้างการใช้ทรัพยากรร่วมแบบไม่ซับซ้อน

### 3.2.3 การวิเคราะห์ช่วงเวลา (Time Analysis)

กล่าวคือเราต้องวิเคราะห์เวลาเนื่องจากถ้าวงจรถูกใช้ร่วมกันมากเกินไป อาจจะทำให้วงจรทำงานช้าเกินไป ดังนั้นเราก็ไม่ควรที่จะใช้ทรัพยากรร่วมแบบนั้น นอกจากนี้การวิเคราะห์เวลาสามารถช่วยให้การใช้ทรัพยากรร่วมกันได้เพิ่มขึ้นโดยไม่กระทบกับเวลารวมของวงจรดัง DFG ในภาพประกอบที่ 3-7 โดยเมื่อพิจารณาภายในวงจร (a) มีการใช้ทรัพยากรร่วมที่ได้ผลลัพธ์เช่นเดียวกับในวงจร (b) แต่เมื่อมีการจัดรูปแบบการใช้ทรัพยากรร่วมภายในบล็อก H3 แตกต่างกันก็จะส่งผลทำให้วงจรที่ไม่มีการคำนึงถึงการวิเคราะห์ช่วงเวลาที่เหมาะสมทำงานช้ากว่าถึง 1 รอบสัญญาณพิก้า โดยในที่นี้ยกตัวอย่างให้ 1 รอบสัญญาณพิก้ามีค่าเท่ากับ 300 ns เมื่อพิจารณาในภาพประกอบที่ 3-7 ส่วน (a) จะใช้เวลามากถึง 1500 ns ซึ่งเทียบกับวงจร (b) นั้นเมื่อพิจารณาการแบ่งเวลาที่เหมาะสมจะพบว่าสามารถลดเวลาการทำงานได้ถึง 300 ns ซึ่งลดเวลาการทำงานได้ถึง 1 รอบสัญญาณพิก้า ดังนั้นหลักการพิจารณาการแบ่งช่วงเวลาให้เหมาะสม มีการพิจารณาโดยในขั้นแรกให้ทำการตรวจสอบว่าในการแบ่งช่วงเวลา แต่ละช่วงมีค่าเท่าใด และพิจารณาการใช้ทรัพยากรร่วมในรูปแบบใดที่มีการรอกการทำงานน้อยที่สุดดังในภาพประกอบที่ 3-7 จะพบว่าส่วน (a) มีการรอกการทำงานของบล็อก H3 ถึง 3 รอบการทำงาน แต่เมื่อมีการจัดรูปแบบใหม่ในส่วน (b) จะพบว่าผลที่ได้เหมือนกันแต่ต่างกันที่มีการรอกการทำงานเพียง 2 รอบ ทำให้พบว่าในส่วน (b) เป็นรูปแบบที่เหมาะสมเมื่อพิจารณาโดยวิเคราะห์ช่วงเวลาในการทำงาน



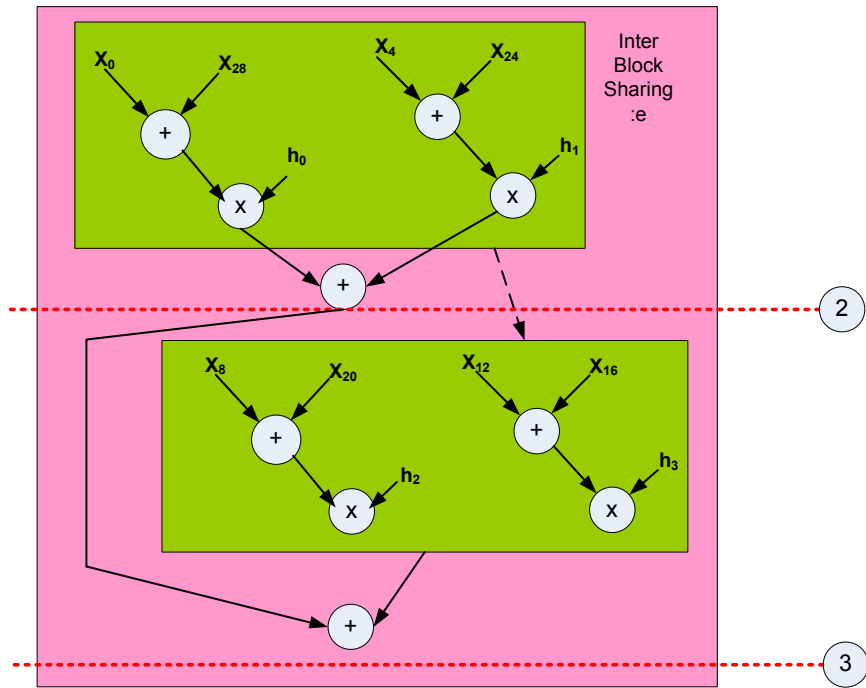
ภาพประกอบ 3-7 การเปรียบเทียบวงจรที่มีการใช้เทคนิคทรัพยากรร่วม โดยพิจารณาการแบ่งช่วงเวลา(a) แบบไม่คำนึงการแบ่งช่วงเวลา (b) แบบคำนึงการแบ่งช่วงเวลา



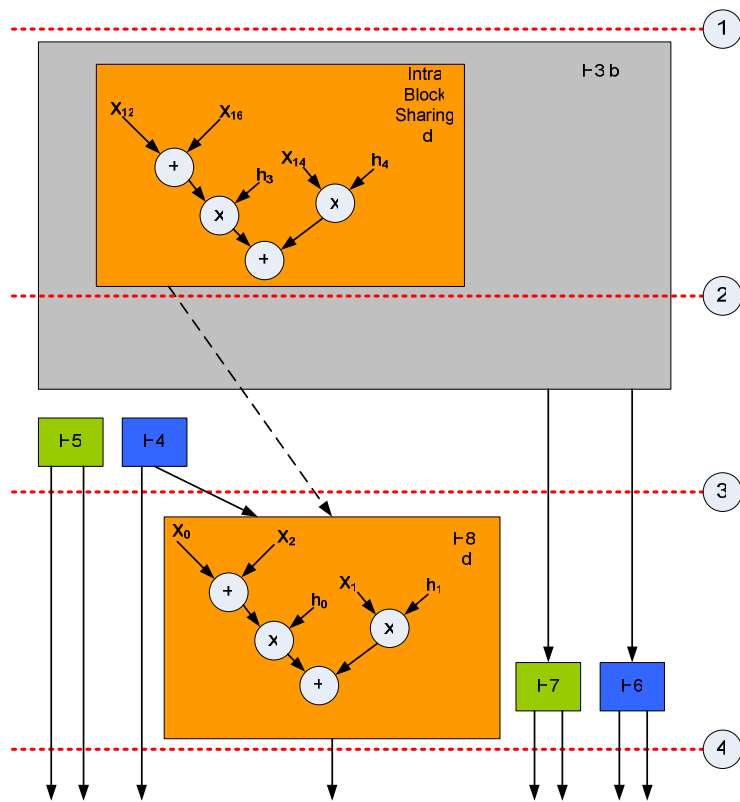
ภาพประกอบ 3-8 การแบ่งช่วงเวลาที่เหมาะสมในวงจรฟิลเตอร์เบงก์ที่ได้มีการออกแบบ

เมื่อพิจารณาวงจรที่มีการแบ่งช่วงเวลาที่มีการออกแบบ ในการแบ่งช่วงเวลาที่เหมาะสมสามารถแบ่งช่วงเวลาออกได้เป็น 5 ช่วง ดังภาพประกอบที่ 3-8 ซึ่งนับว่าเป็นการแบ่งช่วงเวลาที่เหมาะสมที่สุดเมื่อเทียบกับการแบ่งช่วงเวลาออกเป็น 4 หรือ 6 ช่วง เนื่องจากใช้หลักการแบ่งช่วงเวลาตามที่ได้กล่าวมาข้างต้น และสามารถอธิบายการใช้ทรัพยากรร่วมของส่วนบล็อก e และบล็อก d ได้โดยแผนภาพกระแสดังภาพประกอบที่ 3-9 และ 3-10 ตามลำดับ





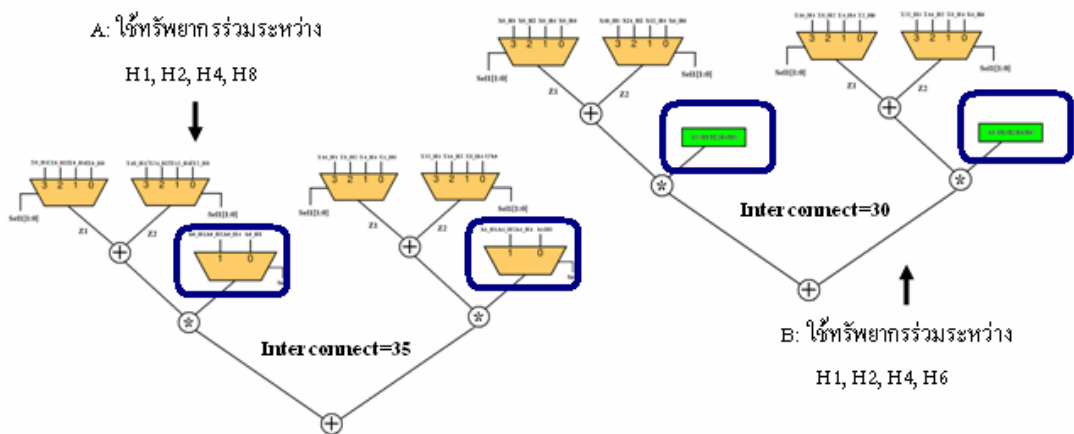
ภาพประกอบ 3-9 การใช้ทรัพยากรร่วมกันภายในบล็อกโครงสร้างบล็อกรูป e



ภาพประกอบ 3-10 การใช้ทรัพยากรร่วมกันระหว่างบล็อกโครงสร้างบล็อกรูป d

### 3.2.4 การพิจารณาสายการเชื่อมต่อภายในวงจร (Interconnect)

การพิจารณาสายเชื่อมต่อภายในวงจรมันสามารถที่จะเป็นตัวบ่งชี้ได้ว่าวงจรที่ได้มีการออกแบบนั้นซับซ้อนมากเพียงใด โดย ถ้าจำนวนสายการเชื่อมต่อภายในวงจรมากก็จะบ่งชี้ว่าวงจรที่ได้มีการออกแบบนั้นมีความซับซ้อนมาก อาจจะก่อให้เกิดความผิดพลาดในการประมวลผลข้อมูล และสิ่งที่เป็นผลเสียตามมาคือการทำวงจรมีการบริโภคพลังงานมากขึ้นเนื่องจากมี Switching Activity มากขึ้นโดยสามารถแก้ไขได้ตามที่ได้กล่าวมาในบทที่ 2 ได้จริง แต่สามารถแก้ไขได้บางส่วนเท่านั้น โดยถ้ามีการคำนึงการเชื่อมต่อภายในวงจวก่อนนั้นก็ช่วยลดปัญหาที่เกิดขึ้นโดยผลลัพธ์ที่ได้นั้น จะมีผลที่ดีว่าการแก้ไขโดยการใส่รีจิสเตอร์คักข้อมูลคั้งนั้นในการออกแบบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมนั้นจำเป็นที่จะต้องพิจารณาปัจจัยที่กล่าวมาทั้ง 4 ข้อ ถ้าไม่ทำการพิจารณาอาจจะพบปัญหา Combination feedback loop โดยได้อธิบายไว้ในบทที่ 2



ภาพประกอบ 3-11 การเปรียบเทียบการวัดค่าสายการเชื่อมต่อภายในวงจร โดยที่

- (A) การใช้ทรัพยากรร่วมแบบไม่เหมาะสมมีค่าสายการเชื่อมต่อ 35 สายสัญญาณ
- (B) การใช้ทรัพยากรร่วมแบบเหมาะสมมีค่าสายการเชื่อมต่อ 30 สายสัญญาณ

### 3.3 การออกแบบวงจรดิจิทัลเฟลเตอร์เบงก์โดยมีการใช้เทคนิคการใช้ทรัพยากรร่วมในรูปแบบต่างๆ

ในงานวิจัยนี้ได้มีการเปรียบเทียบการออกแบบวงจรดิจิทัลเฟลเตอร์เบงก์ที่มีการใช้ทรัพยากรร่วมในรูปแบบต่างๆกล่าวคือ ในขั้นแรกได้มีการนำการออกแบบวงจรดิจิทัลเฟลเตอร์เบงก์โดยใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้นมาเพื่อทำการเปรียบเทียบ [วิวัฒน์ บุญสูง,

2551] และแบบที่ใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมโดยคำนึงถึงปัจจัยในแบบต่างๆ กล่าวคือ ได้มีการเปรียบเทียบการออกแบบออกเป็น 4 รูปแบบโดยสามารถอธิบายการทำงานของในแต่ละรูปแบบได้ดังต่อไปนี้

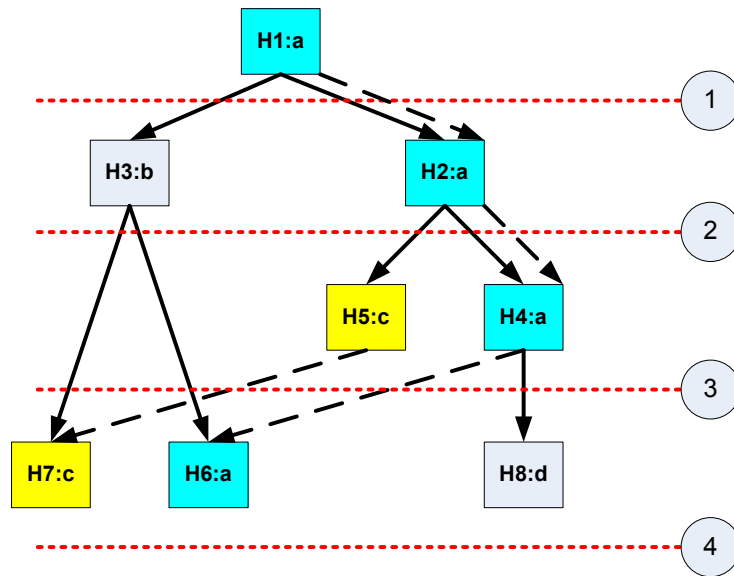
### 3.3.1 การใช้ทรัพยากรร่วมแบบลำดับขั้น

ในรูปแบบนี้ได้มีการกล่าวถึงในหัวข้อ 3.1.1 แล้ว โดยการใช้ทรัพยากรร่วมวิธีการนี้มีการแบ่งได้เป็น 2 แบบคือ การใช้ทรัพยากรร่วมระหว่างบล็อกรวม และการใช้ทรัพยากรร่วมภายในบล็อกรวม โดยสามารถอธิบายการใช้ทรัพยากรร่วมให้เห็นภาพชัดเจนได้ดังภาพประกอบที่ 3-12

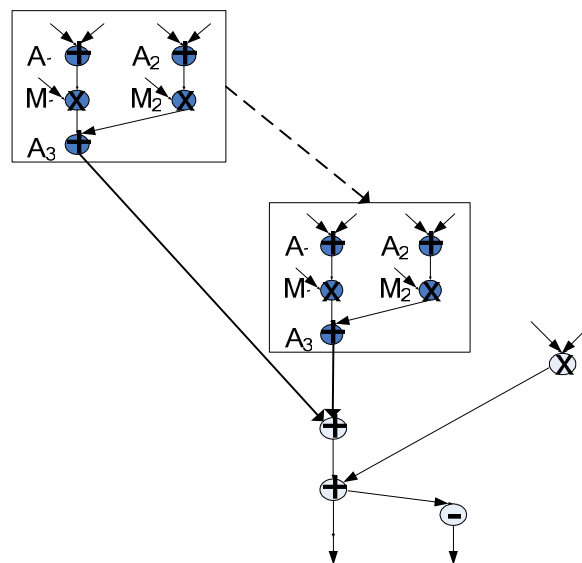
$$\begin{aligned}
 & \text{Circuit A} \\
 H1(z) : y_1(n) &= (x_0+x_{48}) * h_0 + (x_{16}+x_{32}) * h_1 + x_{24} * h_2 \\
 & y_{1c}(n) = x_{24} - y_1(n) \quad \text{A} \\
 H2(z) : y_2(n) &= (x_0+x_{24}) * h_0 + (x_8+x_{16}) * h_1 + x_{12} * h_2 \\
 & y_{2c}(n) = x_{12} - y_2(n) \quad \text{Circuit B} \\
 H3(z) : y_3(n) &= (x_0+x_{28}) * h_0 + (x_4+x_{24}) * h_1 + (x_8+x_{20}) * h_2 + (x_{12}+x_{16}) * h_3 + x_{14} * h_4 \\
 & y_{3c}(n) = x_{14} - y_3(n) \quad \text{A} \\
 H4(z) : y_4(n) &= (x_0+x_{12}) * h_0 + (x_4+x_8) * h_1 + x_6 * h_2 \\
 & y_{4c}(n) = x_6 - y_4(n) \quad \text{Circuit C} \\
 H5(z) : y_5(n) &= (x_0+x_{10}) * h_0 + (x_2+x_8) * h_1 + (x_4+x_6) * h_2 + x_5 * h_3 \\
 & y_{5c}(n) = x_5 - y_5(n) \quad \text{A} \\
 H6(z) : y_6(n) &= (x_0+x_6) * h_0 + (x_2+x_4) * h_1 + x_3 * h_2 \\
 & y_{6c}(n) = x_3 - y_6(n) \quad \text{C} \\
 H7(z) : y_7(n) &= (x_0+x_{30}) * h_0 + (x_6+x_{24}) * h_1 + (x_{12}+x_{18}) * h_2 + x_{15} * h_3 \\
 & y_{7c}(n) = x_{15} - y_7(n) \quad \text{Circuit D} \\
 H8(z) : y_8(n) &= (x_0+x_2) * h_0 + x_1 * h_1
 \end{aligned}$$

ภาพประกอบ 3-12 การใช้ทรัพยากรร่วมแบบลำดับขั้นของวงจรดิจิทัลเตอร์แบ่งกั

เมื่อพิจารณาจากภาพประกอบที่ 3-12 จะพบว่าในภาพประกอบนั้นจะแสดงเฉพาะในส่วนที่มีการใช้ทรัพยากรร่วมกันระหว่างบล็อกรวม โดยเมื่อนำมาวาดเป็นแผนภาพ DFG จะได้ดังภาพประกอบที่ 3-13 และเมื่อพิจารณาในบล็อกรวม H3 จะพบว่ามีการใช้ทรัพยากรร่วมภายในบล็อกรวมซึ่งสามารถแสดงแผนภาพ DFG ของส่วนที่มีการใช้ทรัพยากรร่วมภายในบล็อกรวมได้ดังภาพประกอบที่



ภาพประกอบ 3-13 แผนภาพกระแสของการใช้ทรัพยากรร่วมระหว่างบล็อก



ภาพประกอบ 3-14 แผนภาพกระแสของการใช้ทรัพยากรร่วมภายในบล็อก H3

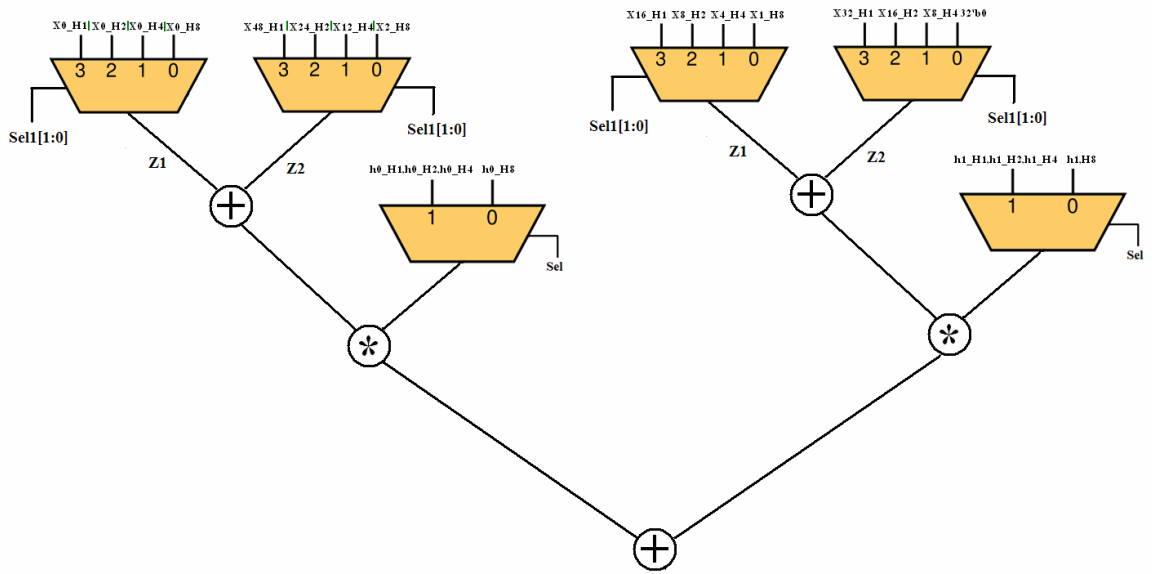
### 3.3.2 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1

ในการออกแบบวงจรดิจิทัลเตอร์แบบก้โดยใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบนี้นั้น อธิบายโดยหลักการทำงานคือพยายามหาส่วนที่เหมือนกันเป็นปัจจัยสำคัญ โดยที่เหลือไม่มีการคำนึงถึงปัจจัยอื่นซึ่งสามารถแสดงส่วนที่มีการใช้ทรัพยากรร่วมได้ดังภาพประกอบที่ 3-15 ซึ่งข้อดีของการออกแบบในลักษณะนี้คือมีการใช้ทรัพยากรร่วมกันมากส่งผลทำให้ขนาดวงจรเล็กลง แต่ถ้าคุณลัพท์แล้วจะพบว่าขนาดจะไม่เล็กลงเพราะมีการใช้ MUX มากสืบเนื่องมาจากส่วนที่ใช้ร่วมกันมีค่าสัมประสิทธิ์ที่ไม่เหมือนกันและยังพบว่าวงจรซับซ้อนมากเนื่องจากมีการใช้ทรัพยากรร่วมกันมาก

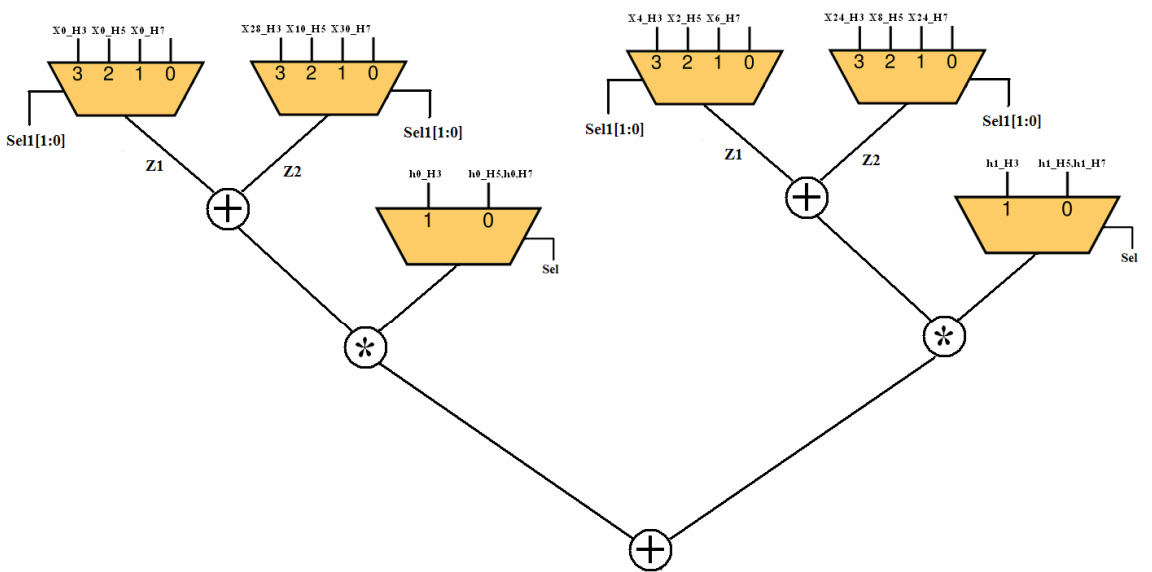
$$\begin{aligned}
 H1(z) : y_1(n) &= \underbrace{(x_0+x_{48}) * h_0 + (x_{16}+x_{32}) * h_1}_{\text{Circuit A1}} + \underbrace{x_{24} * h_2}_{\text{Circuit B1}} \\
 y_{1c}(n) &= x_{24} - y_1(n) \quad \text{A1} \quad \text{B1} \\
 H2(z) : y_2(n) &= \underbrace{(x_0+x_{24}) * h_0 + (x_8+x_{16}) * h_1}_{\text{A1}} + \underbrace{x_{12} * h_2}_{\text{B1}} \\
 y_{2c}(n) &= x_{12} - y_2(n) \\
 H3(z) : y_3(n) &= \underbrace{(x_0+x_{28}) * h_0 + (x_4+x_{24}) * h_1}_{\text{Circuit A2}} + \underbrace{(x_8+x_{20}) * h_2 + (x_{12}+x_{16}) * h_3}_{\text{Circuit A3}} + \underbrace{x_{14} * h_4}_{\text{Circuit B2}} \\
 y_{3c}(n) &= x_{14} - y_3(n) \quad \text{A1} \quad \text{B1} \\
 H4(z) : y_4(n) &= \underbrace{(x_0+x_{12}) * h_0 + (x_4+x_8) * h_1}_{\text{A2}} + \underbrace{x_6 * h_2}_{\text{B1}} \\
 y_{4c}(n) &= x_6 - y_4(n) \quad \text{A2} \quad \text{A3} \\
 H5(z) : y_5(n) &= \underbrace{(x_0+x_{10}) * h_0 + (x_2+x_8) * h_1}_{\text{A2}} + \underbrace{(x_4+x_6) * h_2 + x_5 * h_3}_{\text{A3}} \\
 y_{5c}(n) &= x_5 - y_5(n) \\
 H6(z) : y_6(n) &= \underbrace{(x_0+x_6) * h_0}_{\text{Circuit C1}} + \underbrace{(x_2+x_4) * h_1}_{\text{Circuit C1}} + \underbrace{x_3 * h_2}_{\text{B2}} \\
 y_{6c}(n) &= x_3 - y_6(n) \quad \text{A2} \quad \text{A3} \\
 H7(z) : y_7(n) &= \underbrace{(x_0+x_{30}) * h_0 + (x_6+x_{24}) * h_1}_{\text{A2}} + \underbrace{(x_{12}+x_{18}) * h_2 + x_{15} * h_3}_{\text{A3}} \\
 y_{7c}(n) &= x_{15} - y_7(n) \quad \text{A1} \\
 H8(z) : y_8(n) &= \underbrace{(x_0+x_2) * h_0 + x_1 * h_1}_{\text{A1}}
 \end{aligned}$$

ภาพประกอบ 3-15 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1 ของวงจรฟิลเตอร์แบบก้

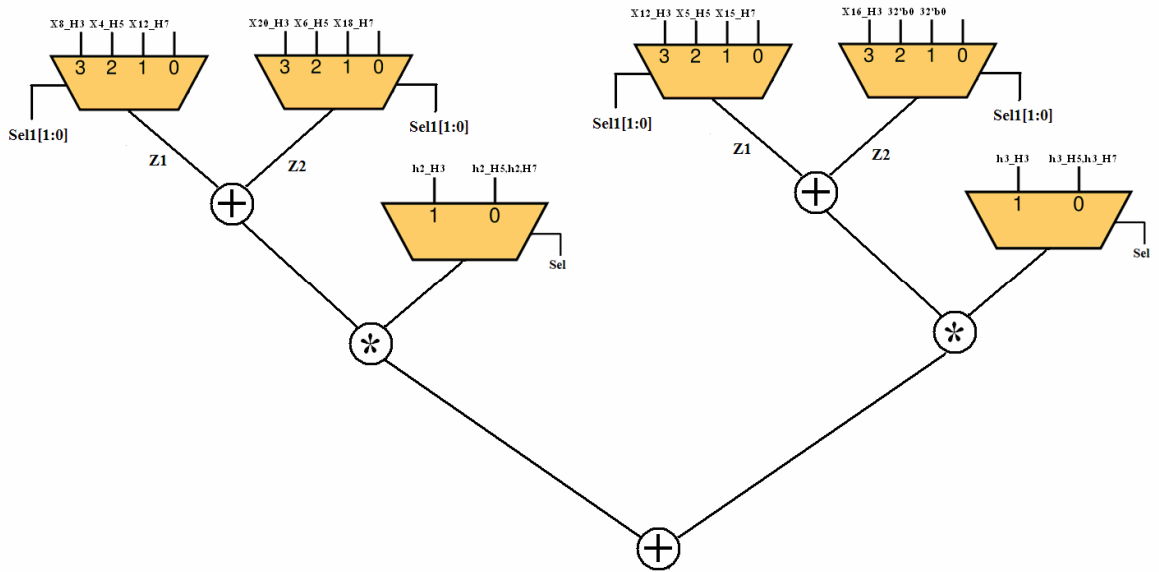
เมื่อลองนำการออกแบบดังภาพประกอบที่ 3-15 มาทำการวาดแผนภาพ DFG ระดับล่างนั้นจะได้วงจรแยกออกเป็นส่วนประกอบได้ดังภาพประกอบที่ 3-16, 3-17, 3-18, 3-19, 3-20 และ 3-21 ซึ่ง แสดงในส่วนที่มีการใช้ทรัพยากรร่วม A1, A2, A3, B1, B2 และ C1 ตามลำดับ ซึ่งเมื่อพิจารณาในแผนภาพ DFG ระดับล่างในส่วนต่างๆจะพบว่าการสร้าง MUX จำนวนมากเพื่อใช้ในการใช้ทรัพยากรร่วม ทำให้วิธีการนี้ไม่เหมาะสมในการที่จะนำมาออกแบบวงจรฟิลเตอร์แบบก้



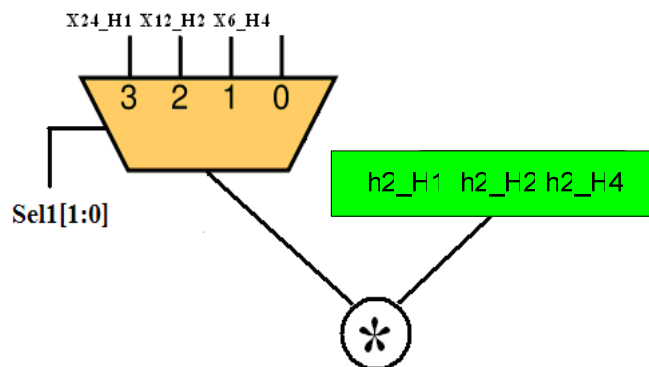
ภาพประกอบ 3-16 แผนภาพกระแสดระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน A1



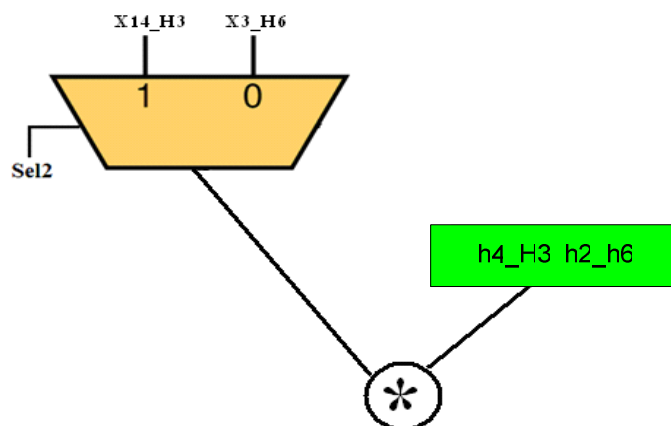
ภาพประกอบ 3-17 แผนภาพกระแสดระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน A2



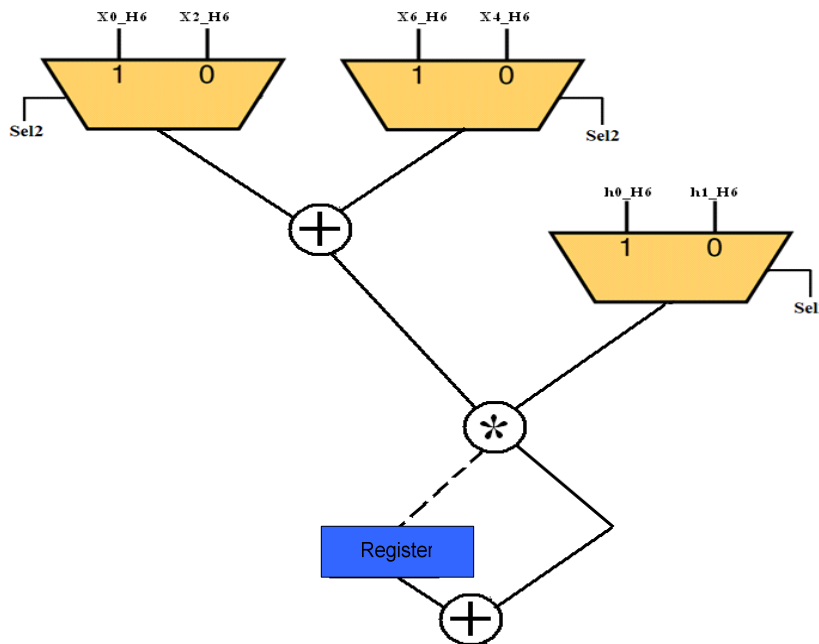
ภาพประกอบ 3-18 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน A3



ภาพประกอบ 3-19 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน B1



ภาพประกอบ 3-20 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน B2



ภาพประกอบ 3-21 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 1 ส่วน C1

### 3.3.3 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2

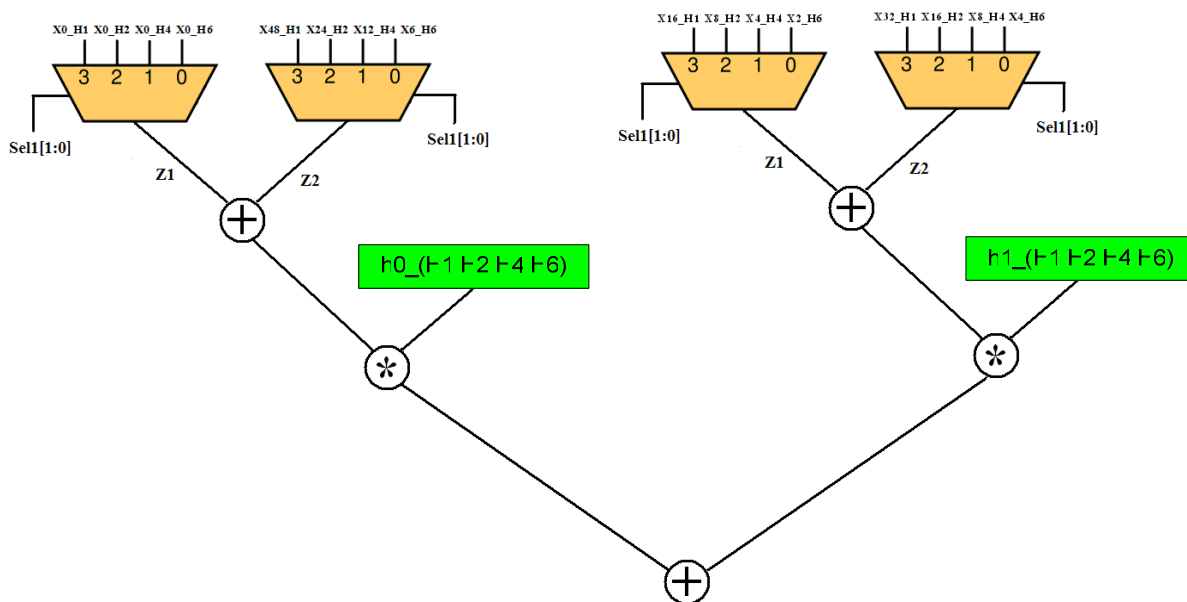
ในวิธีการนี้ได้มีการนำปัจจัยอื่นมาช่วยในการวิเคราะห์ด้วยนอกจากจะพิจารณาเรื่องความเหมือนของสมการแล้วในวิธีนี้ยังได้มีการพิจารณาค่าคงที่ในทีนี้ก็คือค่านัยสำคัญในวงจรฟิลเตอร์เบงกั้นนั่นเอง กล่าวคือ การที่ค่านึงถึงค่านี้นี้ด้วยเพื่อที่จะช่วยลดปัญหาการใช้ MUX ที่มากเกินไปโดยสามารถแสดงการใช้ทรัพยากรร่วมวิธีนี้ได้ดังภาพประกอบที่ 3-22

$$\begin{aligned}
 H1(z) : y_1(n) &= (x_0+x_{48}) * h_0 + (x_{16}+x_{32}) * h_1 + x_{24} * h_2 \\
 y_{1c}(n) &= x_{24} - y_1(n) \\
 H2(z) : y_2(n) &= (x_0+x_{24}) * h_0 + (x_8+x_{16}) * h_1 + x_{12} * h_2 \\
 y_{2c}(n) &= x_{12} - y_2(n) \\
 H3(z) : y_3(n) &= (x_0+x_{28}) * h_0 + (x_4+x_{24}) * h_1 + (x_8+x_{20}) * h_2 + (x_{12}+x_{16}) * h_3 + x_{14} * h_4 \\
 y_{3c}(n) &= x_{14} - y_3(n) \\
 H4(z) : y_4(n) &= (x_0+x_{12}) * h_0 + (x_4+x_8) * h_1 + x_6 * h_2 \\
 y_{4c}(n) &= x_6 - y_4(n) \\
 H5(z) : y_5(n) &= (x_0+x_{10}) * h_0 + (x_2+x_8) * h_1 + (x_4+x_6) * h_2 + x_5 * h_3 \\
 y_{5c}(n) &= x_5 - y_5(n) \\
 H6(z) : y_6(n) &= (x_0+x_6) * h_0 + (x_2+x_4) * h_1 + x_3 * h_2 \\
 y_{6c}(n) &= x_3 - y_6(n) \\
 H7(z) : y_7(n) &= (x_0+x_{30}) * h_0 + (x_6+x_{24}) * h_1 + (x_{12}+x_{18}) * h_2 + x_{15} * h_3 \\
 y_{7c}(n) &= x_{15} - y_7(n) \\
 H8(z) : y_8(n) &= (x_0+x_2) * h_0 + x_1 * h_1
 \end{aligned}$$

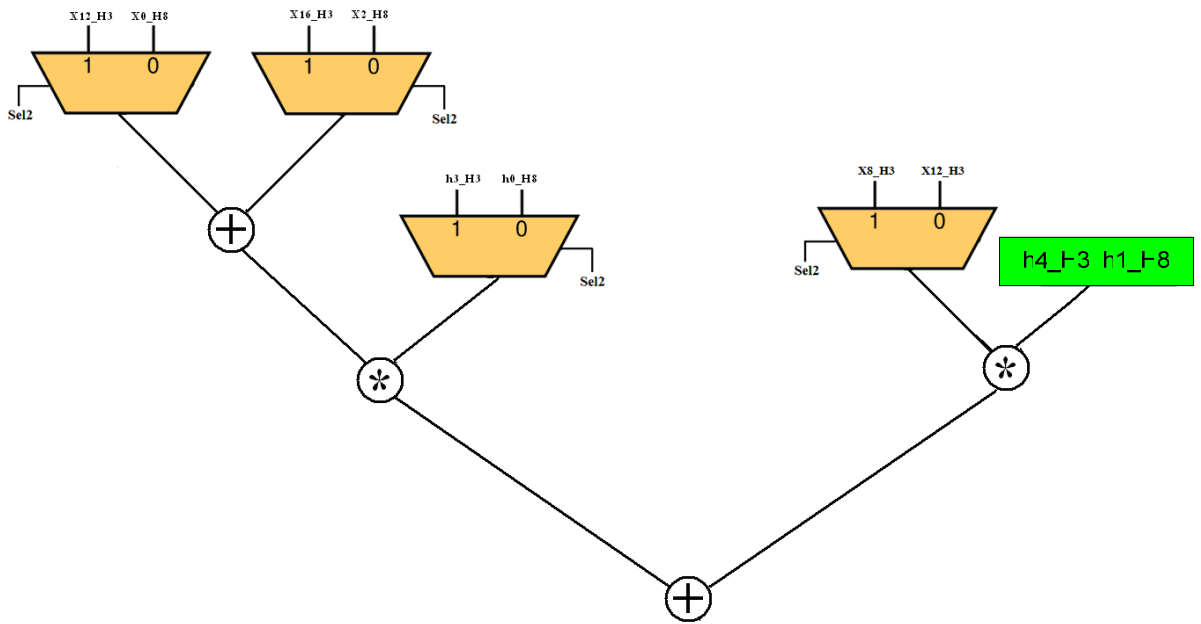
ภาพประกอบ 3-22 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2 ของวงจรฟิลเตอร์เบงกั้น



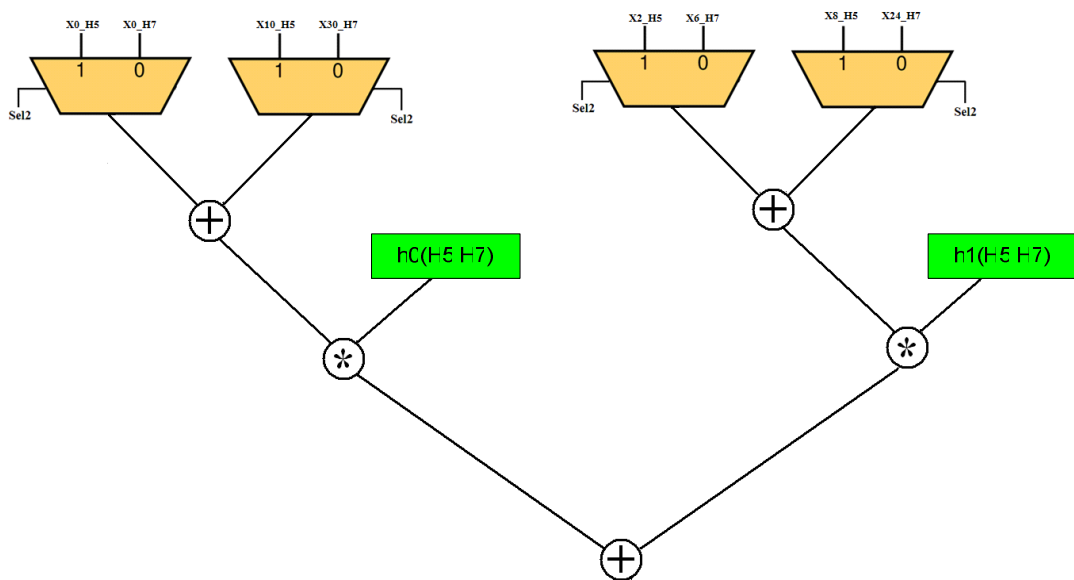
ข้อดีของการออกแบบลักษณะนี้คือได้มีการพิจารณาค่าสัมประสิทธิ์ก่อนที่จะมีการใช้ทรัพยากรร่วมทำให้การใช้ทรัพยากรร่วมวิธีนี้ลดปัญหาขนาดที่เพิ่มขึ้นเนื่องจากไม่ต้องสร้าง MUX เพื่อทำการสลับค่าสัมประสิทธิ์และยังพบว่าวงจรที่ได้มีการออกแบบนั้นซับซ้อนน้อยกว่าการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1 โดยแสดงในภาพประกอบที่ 3-15 แต่ข้อเสียของการใช้ทรัพยากรร่วมลักษณะนี้คือจะมีส่วนของวงจรที่มีชื่อว่า No\_Share ในภาพประกอบที่ 3-22 ที่ไม่สามารถใช้ร่วมกับกับวงจรอื่นได้ส่งผลทำให้ต้องมีการสร้างวงจรในส่วนนั้นเพิ่มจึงได้มีการออกแบบการใช้ทรัพยากรร่วมแบบที่เหมาะสมที่สุดในสุดท้ายที่จะได้มีการกล่าวต่อไป เมื่อ นำการออกแบบดังกล่าวประกอบที่ 3-22 มาทำการวาดแผนภาพกระแสระดับล่างนั้นจะได้วงจรแยกออกเป็นส่วนประกอบได้ดังภาพประกอบที่ 3-23, 3-24, 3-25, 3-26, 3-27 และ 3-28 ซึ่งแสดงในส่วนที่มีการใช้ทรัพยากรร่วม A1, A2, A3, A4, B1 และ B2 ตามลำดับเพื่อพิจารณาเปรียบเทียบกับ การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1 โดยเมื่อทำการพิจารณาแล้วพบว่าในการออกแบบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2 นั้นจะมีขนาดและความซับซ้อนลดลงเล็กน้อย จึงได้มีการออกแบบการใช้ทรัพยากรร่วมในรูปแบบที่ 3 เพื่อที่จะแก้ปัญหาในส่วน ของวงจร No\_Share โดยได้มีการอธิบายในหัวข้อถัดไป



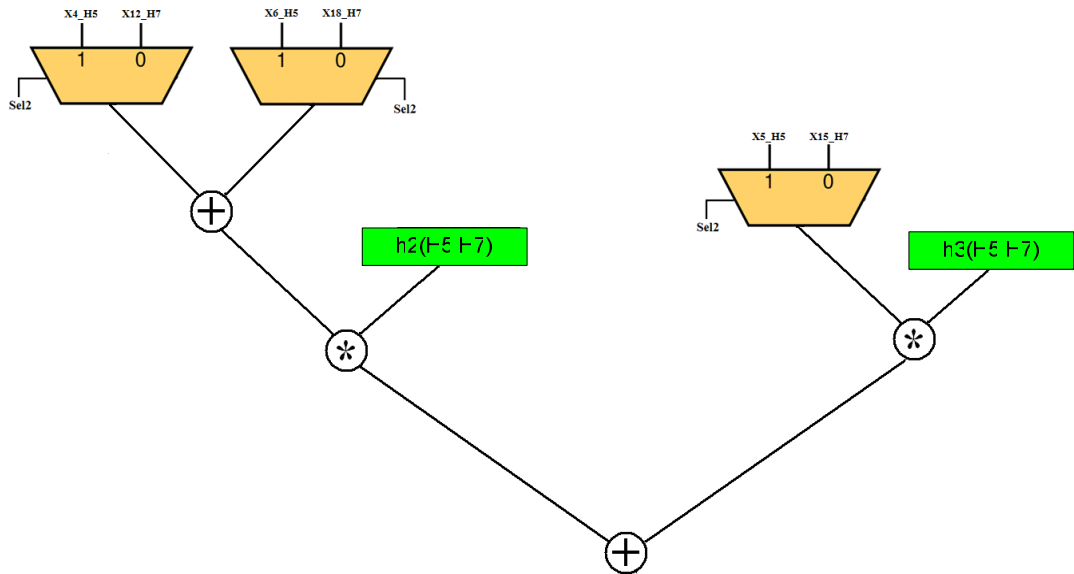
ภาพประกอบ 3-23 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A1



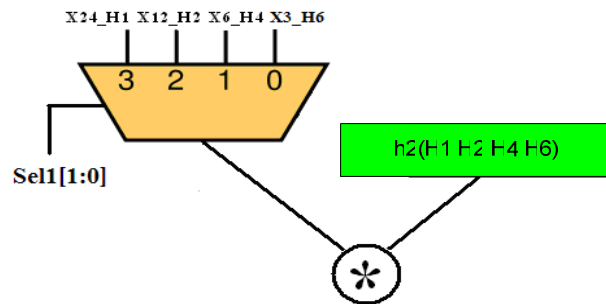
ภาพประกอบ 3-24 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A2



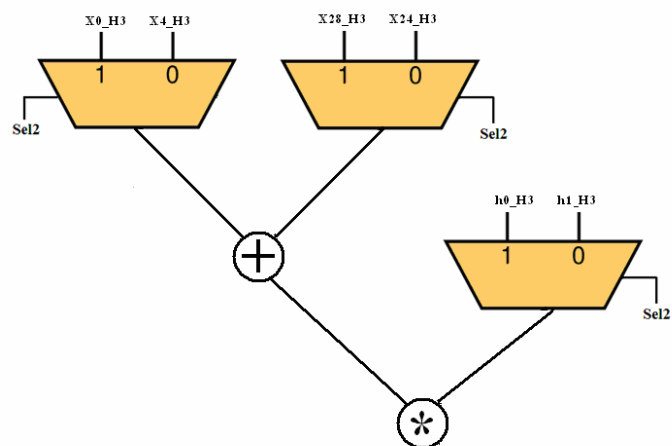
ภาพประกอบ 3-25 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A3



ภาพประกอบ 3-26 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วม รูปแบบที่ 2 ส่วน A4



ภาพประกอบ 3-27 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 2 ส่วน B1



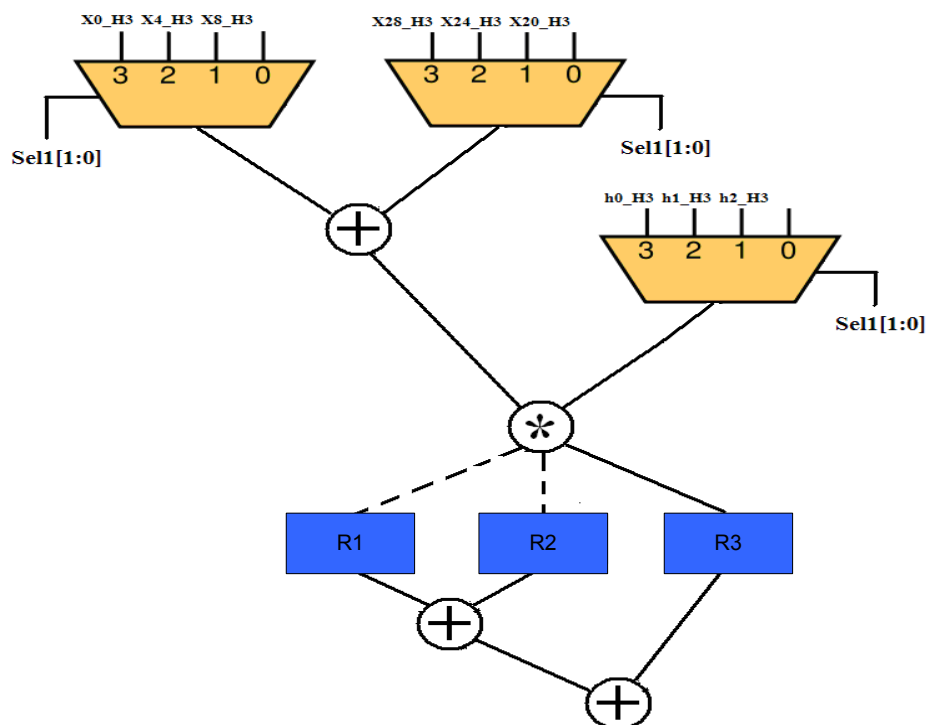
ภาพประกอบ 3-28 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 2 ส่วน B2

### 3.3.4 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3

ในวิธีการนี้ได้นำข้อบกพร่องในการออกแบบในรูปแบบ 2 ที่กล่าวมาข้างต้นมาทำการแก้ไขโดยได้มีการปรับเปลี่ยนรูปแบบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม ให้ส่วนที่ไม่สามารถใช้ร่วมกับส่วนอื่นนั้นสามารถใช้ร่วมกับส่วนใดส่วนหนึ่งที่มีความคล้ายคลึงกันได้ เพื่อลดขนาดวงจรที่จะต้องทำการสร้างขึ้นใหม่เพื่อส่วนนั้นส่วนเดียว จึงได้รูปแบบใหม่ดังภาพประกอบที่ 3-29 เมื่อพิจารณาเบื้องต้นอาจจะพบว่าการออกแบบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมดังภาพประกอบที่ 3-29 นั้นเป็นการออกแบบที่เหมาะสมที่สุด แต่เนื่องจากเมื่อมีการตรวจสอบเกี่ยวกับปัจจัยของการออกแบบในด้านของการเลือกช่วงเวลาการทำงานที่เหมาะสม จะพบว่าเมื่อมีการออกแบบตามภาพประกอบที่ 3-29 นั้น ส่วนที่จะพบปัญหาคือ ส่วนที่ B 2 เนื่องจากเมื่อตรวจสอบเวลาการทำงานนั้นจะพบว่าเราจะต้องเสียเวลาในการรอการทำงานของส่วน B2 ถึง 1 รอบสัญญาณนาฬิกา ถ้ามีการใช้ทรัพยากรร่วมดังภาพประกอบที่ 3-29 โดยสามารถเขียนแผนภาพกระแสดับล่างในส่วน B 2 เพื่อให้เห็นภาพที่ชัดเจนถึงเวลาในการทำงานโดยสามารถแสดงให้เห็นได้ดังภาพประกอบที่ 3-30 นอกจากนี้จะพบปัญหาเกี่ยวกับการทำงานช้าแล้วยังมีปัญหาคือ ความซับซ้อนอีกด้วยโดยในการออกแบบรูปแบบนี้จะมีความซับซ้อนมากกว่ารูปแบบที่ 2 แต่น้อยกว่ารูปแบบที่ 1 ดังนั้นจึงได้มีการออกแบบรูปแบบการใช้ทรัพยากรร่วมแบบที่ 4 ซึ่งคาดว่าจะเป็วิธีที่ดีที่สุดโดยพิจารณาจากปัจจัยทั้งหมด

$$\begin{aligned}
 H1(z) : y_1(n) &= \underbrace{(x_0+x_{48}) * h_0 + (x_{16}+x_{32}) * h_1}_{\text{Circuit A1}} + \underbrace{x_{24} * h_2}_{\text{Circuit B1}} \\
 y_{1c}(n) &= x_{24} - y_1(n) \quad \text{A1} \quad \text{B1} \\
 H2(z) : y_2(n) &= \underbrace{(x_0+x_{24}) * h_0 + (x_8+x_{16}) * h_1}_{\text{A1}} + \underbrace{x_{12} * h_2}_{\text{B1}} \\
 y_{2c}(n) &= x_{12} - y_2(n) \\
 H3(z) : y_3(n) &= \underbrace{(x_0+x_{28}) * h_0}_{\text{Circuit B2}} + \underbrace{(x_4+x_{24}) * h_1}_{\text{A1}} + \underbrace{(x_8+x_{20}) * h_2}_{\text{B1}} + \underbrace{(x_{12}+x_{16}) * h_3 + x_{14} * h_4}_{\text{Circuit A2}} \\
 y_{3c}(n) &= x_{14} - y_3(n) \quad \text{A1} \quad \text{B1} \\
 H4(z) : y_4(n) &= \underbrace{(x_0+x_{12}) * h_0 + (x_4+x_8) * h_1}_{\text{A1}} + \underbrace{x_6 * h_2}_{\text{B1}} \\
 y_{4c}(n) &= x_6 - y_4(n) \quad \text{Circuit A3} \quad \text{Circuit A4} \\
 H5(z) : y_5(n) &= \underbrace{(x_0+x_{10}) * h_0 + (x_2+x_8) * h_1}_{\text{A1}} + \underbrace{(x_4+x_6) * h_2 + x_5 * h_3}_{\text{B1}} \\
 y_{5c}(n) &= x_5 - y_5(n) \quad \text{A1} \quad \text{B1} \\
 H6(z) : y_6(n) &= \underbrace{(x_0+x_6) * h_0 + (x_2+x_4) * h_1}_{\text{A1}} + \underbrace{x_3 * h_2}_{\text{B1}} \\
 y_{6c}(n) &= x_3 - y_6(n) \quad \text{A3} \quad \text{A4} \\
 H7(z) : y_7(n) &= \underbrace{(x_0+x_{30}) * h_0 + (x_6+x_{24}) * h_1}_{\text{A1}} + \underbrace{(x_{12}+x_{18}) * h_2 + x_{15} * h_3}_{\text{A4}} \\
 y_{7c}(n) &= x_{15} - y_7(n) \quad \text{A2} \\
 H8(z) : y_8(n) &= \underbrace{(x_0+x_2) * h_0 + x_1 * h_1}_{\text{A1}}
 \end{aligned}$$

ภาพประกอบ 3-29 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3



ภาพประกอบ 3-30 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 3 ส่วน B2

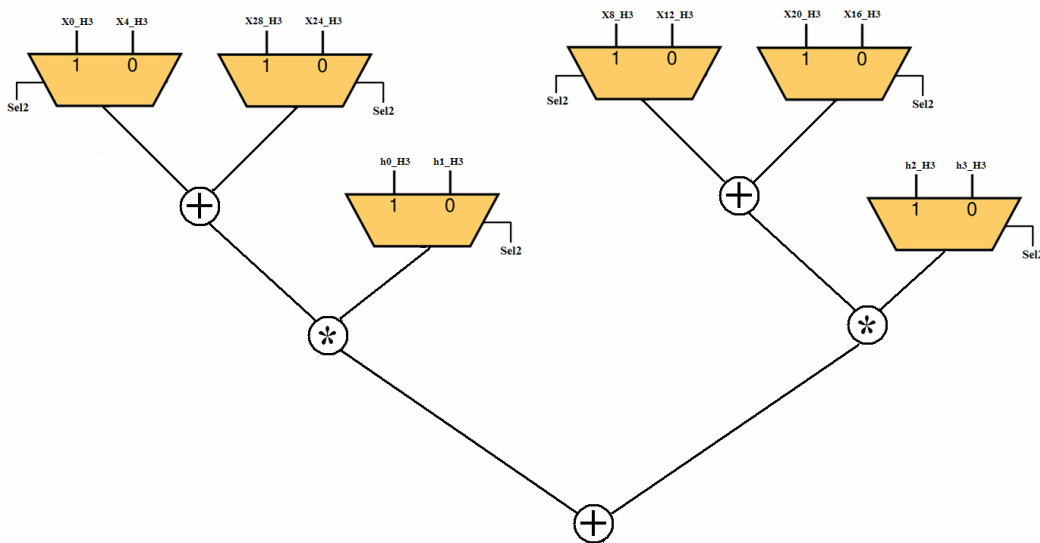
### 3.3.5 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4

ในการออกแบบรูปแบบนี้ได้มีการพยายามแก้ปัญหาที่เกิดจากรูปแบบที่มีการออกแบบก่อนหน้าทั้ง 4 รูปแบบโดย เมื่อพิจารณาจะพบว่าในรูปแบบนี้จะมีบางส่วนไม่สามารถใช้ทรัพยากรร่วมกับส่วนอื่นได้ คือ ส่วนของวงจร No\_Share แต่ถือว่ามีผลเสียน้อยกว่าการที่จะพยายามนำส่วนดังกล่าวไปใช้ทรัพยากรร่วมเนื่องจาก เมื่อพยายามนำส่วนดังกล่าวไปใช้ทรัพยากรร่วมกับส่วนที่คล้ายคลึงจะพบปัญหาในด้านของพื้นที่ที่มากขึ้นซึ่งเกิดจากขนาดของตัว MUX และขนาดของรีจิสเตอร์ที่ต้องสร้างมาเพื่อทำการเก็บค่า นอกจากนี้ยังมีปัญหาในส่วนของเวลาในการประมวลผลถือว่าช้ากว่ามากจึงได้มีการจัดรูปใหม่เพื่อแก้ปัญหาในส่วนนี้ ซึ่งมีการออกแบบส่วนที่ใช้ร่วมกันใหม่ได้ดังภาพประกอบที่ 3-31

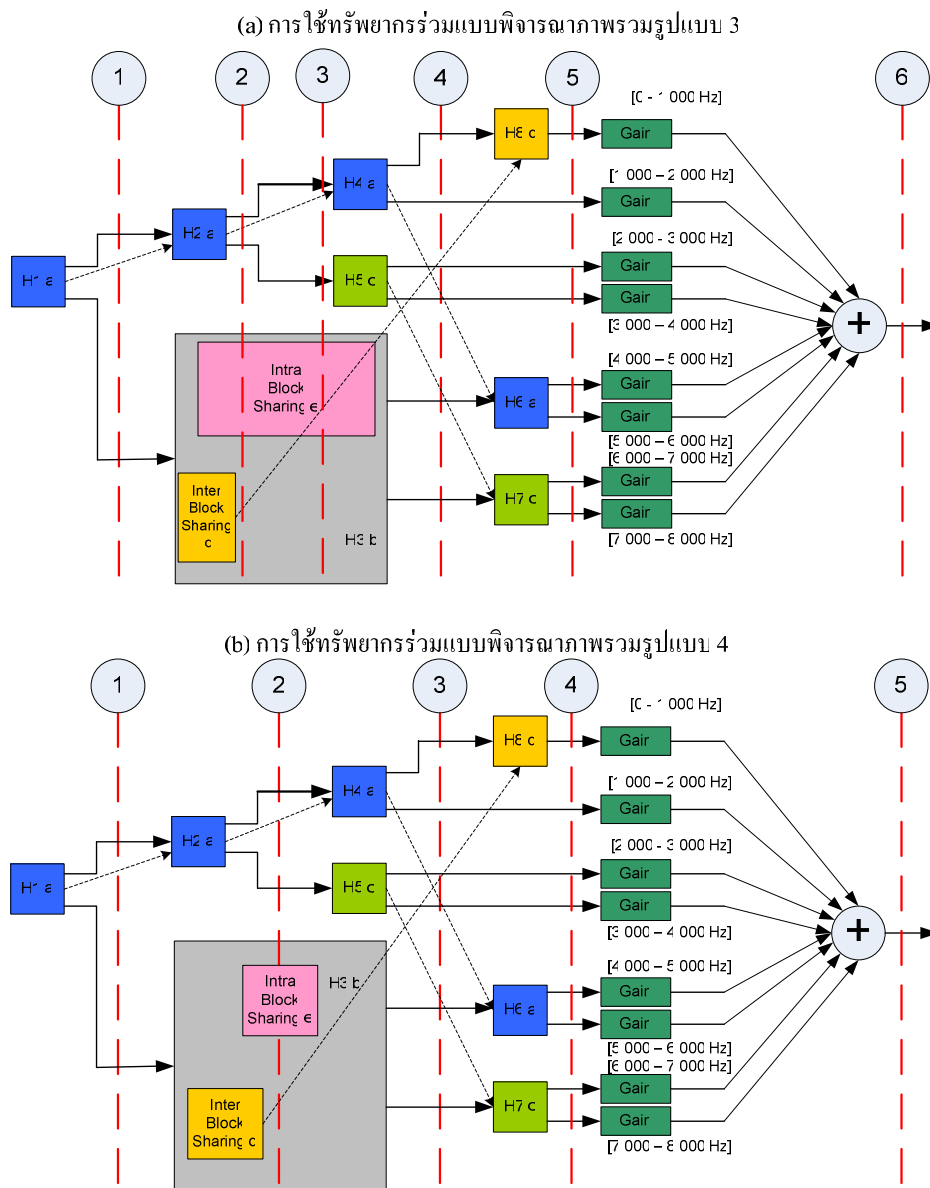
$$\begin{aligned}
 H1(z) : y_1(n) &= \text{Circuit A1} (x_0+x_{48}) * h_0 + (x_{16}+x_{32}) * h_1 + \text{Circuit B1} x_{24} * h_2 \\
 y_{1c}(n) &= x_{24} * y_1(n) \quad \text{A1} \quad \text{B1} \\
 H2(z) : y_2(n) &= (x_0+x_{24}) * h_0 + (x_8+x_{16}) * h_1 + x_{12} * h_2 \\
 y_{2c}(n) &= x_{12} * y_2(n) \\
 H3(z) : y_3(n) &= \text{Circuit C1} (x_0+x_{28}) * h_0 + (x_4+x_{24}) * h_1 + (x_8+x_{20}) * h_2 + (x_{12}+x_{16}) * h_3 + \text{Circuit B2} x_{14} * h_4 \\
 y_{3c}(n) &= x_{14} * y_3(n) \quad \text{A1} \quad \text{B1} \\
 H4(z) : y_4(n) &= (x_0+x_{12}) * h_0 + (x_4+x_8) * h_1 + x_6 * h_2 \\
 y_{4c}(n) &= x_6 * y_4(n) \quad \text{Circuit A2} \quad \text{Circuit A3} \\
 H5(z) : y_5(n) &= (x_0+x_{10}) * h_0 + (x_2+x_8) * h_1 + (x_4+x_6) * h_2 + x_5 * h_3 \\
 y_{5c}(n) &= x_5 * y_5(n) \quad \text{A1} \quad \text{B1} \\
 H6(z) : y_6(n) &= (x_0+x_6) * h_0 + (x_2+x_4) * h_1 + x_3 * h_2 \\
 y_{6c}(n) &= x_3 * y_6(n) \quad \text{A2} \quad \text{A3} \\
 H7(z) : y_7(n) &= (x_0+x_{30}) * h_0 + (x_6+x_{24}) * h_1 + (x_{12}+x_{18}) * h_2 + x_{15} * h_3 \\
 y_{7c}(n) &= x_{15} * y_7(n) \quad \text{B2} \\
 H8(z) : y_8(n) &= (x_0+x_2) * h_0 + x_1 * h_1 \quad \text{No\_Share}
 \end{aligned}$$

ภาพประกอบ 3-31 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4

เมื่อทำการพิจารณาภาพประกอบ 3-31 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมในรูปแบบที่ 4 นั้นมีความคล้ายคลึงกับในรูปแบบที่ 2 โดยสามารถแสดงแผนภาพ DFG ระดับล่างของส่วนที่ต่างคือวงจร C1 ได้ดังภาพประกอบที่ 3-32 โดยที่มีตัวที่ไม่สามารถจะใช้ร่วมกับตัวใดได้เช่นเดียวกัน แต่รูปแบบที่ 4 นั้นดีกว่ารูปแบบที่ 2 ในด้านของเวลาและขนาด เนื่องจากถ้ามีการใช้ทรัพยากรร่วมในรูปแบบที่ 2 จะพบว่าเส้นทางวิกฤตในการทำงานของ H3 จะยาวมากและเสียเวลาในการรอคอยดังนั้นจึงได้มีการเลือกรูปแบบที่ 4 โดยสามารถแสดงการเปรียบเทียบช่วงเวลาที่ใช้ในการทำงานระหว่างรูปแบบที่ 3 และรูปแบบที่ 4 ได้ดังภาพประกอบที่ 3-33 จะพบว่าการทำงานของระบบในรูปแบบที่ 3 จะช้ากว่าการทำงานในรูปแบบที่ 4 เป็นเวลา 1 รอบสัญญาณนาฬิกา

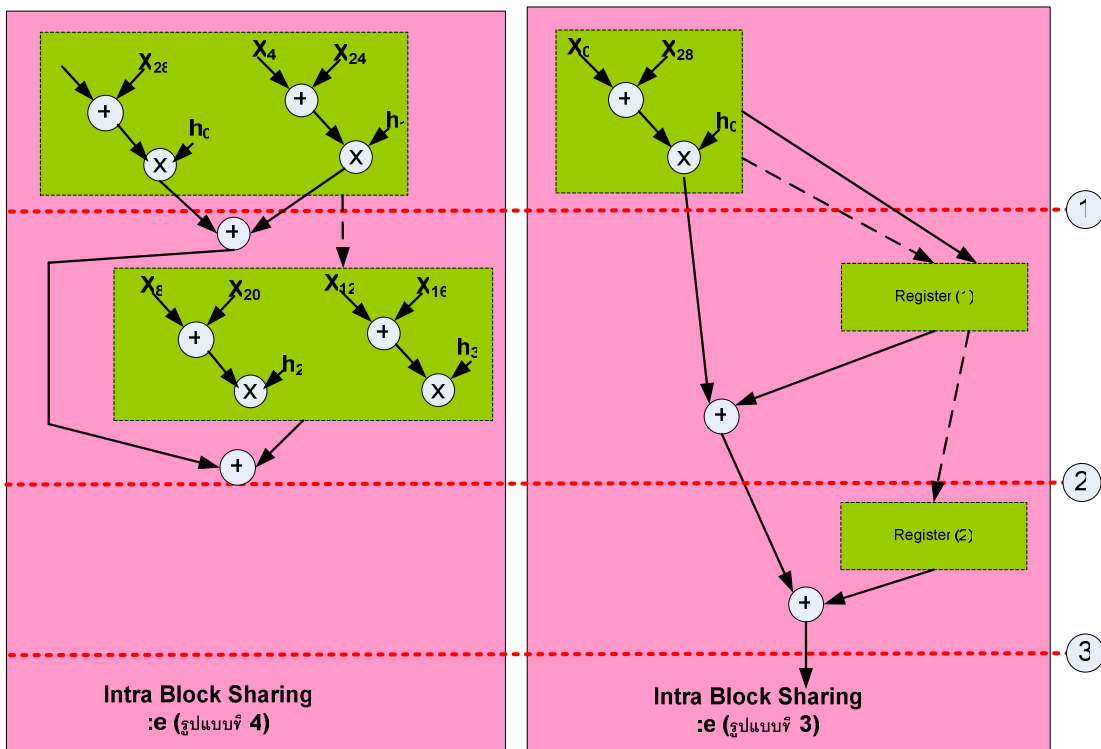


ภาพประกอบ 3-28 แผนภาพกระแสระดับล่างในการใช้ทรัพยากรร่วมรูปแบบที่ 2 ส่วน C1



ภาพประกอบ 3-33 การเปรียบเทียบช่วงเวลาที่ใช้ระหว่างรูปแบบที่ 3 และรูปแบบที่ 4

เมื่อพิจารณาจากภาพประกอบที่ 3-33 จะพบว่าส่วนของวงจรที่ทำให้รูปแบบที่ 3 และรูปแบบที่ 4 มีการแบ่งรอบสัญญาณพิกัดต่างกันคือส่วนของบล็อก e โดยสามารถวาดแผนภาพ D F G เพื่อแสดงความชัดเจนในการใช้ทรัพยากรร่วมโดยมีการแบ่งช่วงเวลาที่ต่างกันได้ดังภาพประกอบที่ 3-34 โดยในรูปแบบที่ 3 ภายในส่วน Intra Block Sharing จะใช้เวลาในการทำงานถึง 3 ช่วงเวลา แต่ในรูปแบบที่ 4 ใช้เพียง 2 ช่วงเวลาในการประมวลผล



ภาพประกอบ 3-34 แผนภาพกระแสเปรียบเทียบการแบ่งช่วงเวลาของบล็อก e ระหว่างรูปแบบที่ 3 และรูปแบบที่ 4

เมื่อทำการสังเคราะห์วงจรทั้ง 5 รูปแบบด้วยโปรแกรม Xilinx 10.1 พบว่าการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4 ดีที่สุดโดยสามารถแสดงผลการเปรียบเทียบได้ดังตารางที่ 3-1

รูปแบบ	Estimated Area (Slices/FFs)	Inter - Connects	Clock Period (ns)	Latency (ns)
1. การใช้ทรัพยากรร่วมแบบลำดับขั้น	11171/3812	174	174.03	870.15
2. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1	9436/3282	216	170.24	1021.44
3. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2	9248/2919	179	157.93	789.65
4. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3	8272/2818	189	160.87	1126.09
5. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4	8486/2789	179	156.58	782.9

ตาราง 3-1 การเปรียบเทียบผลการสังเคราะห์วงจรของการออกแบบการใช้ทรัพยากรร่วม

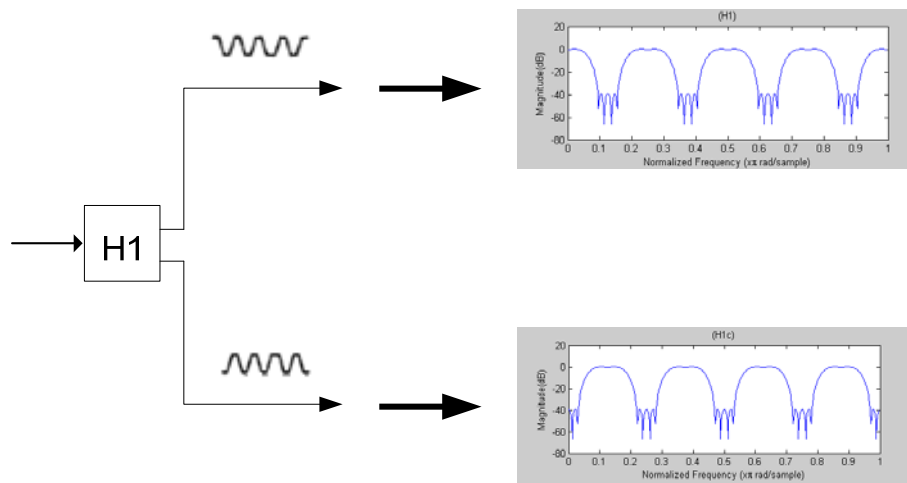


## บทที่ 4

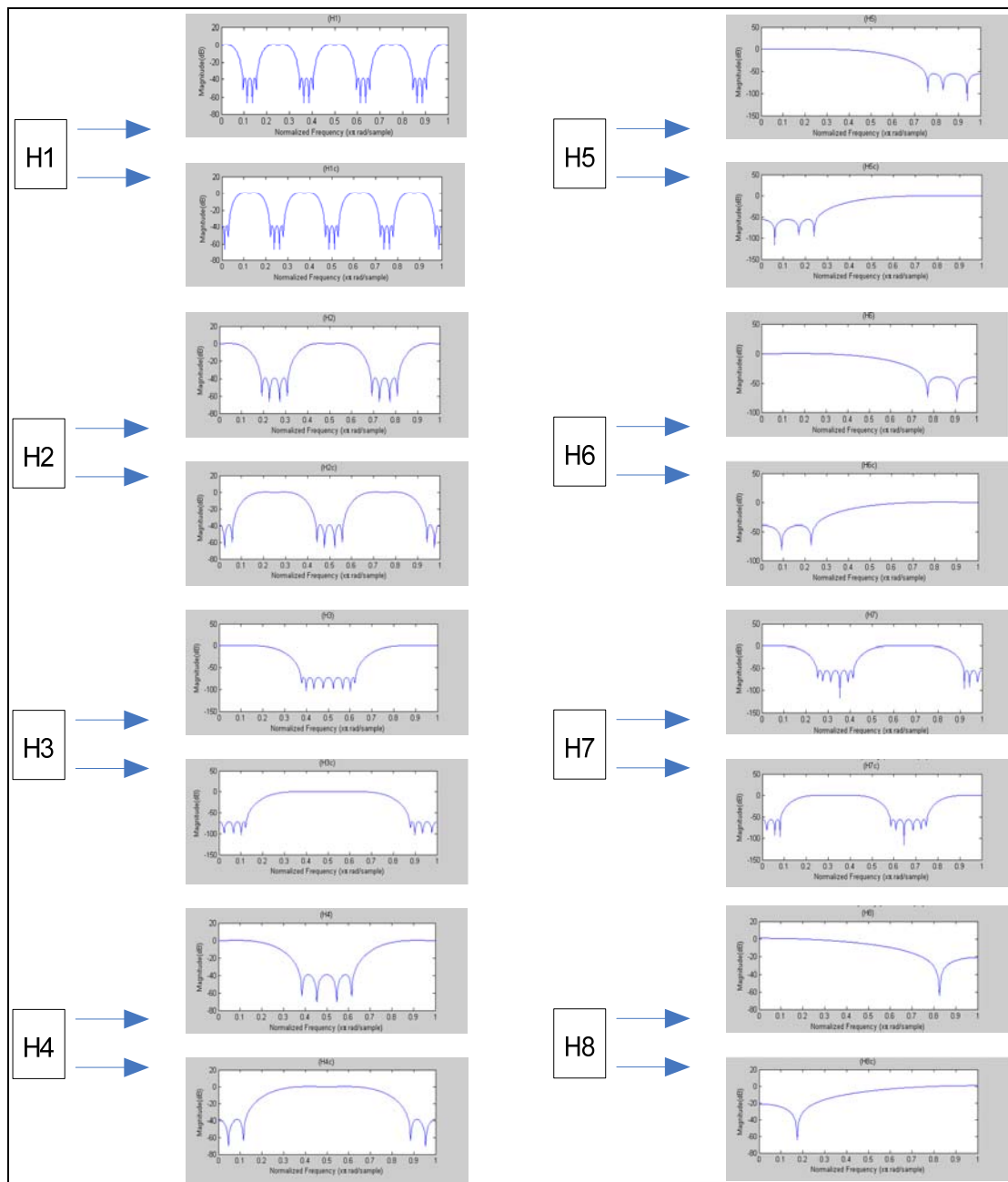
### ผลการวิจัย

#### 4.1 ผลการออกแบบวงจรฟิลเตอร์เบงก์ [วิวัฒน์ บุญสูง, 2551]

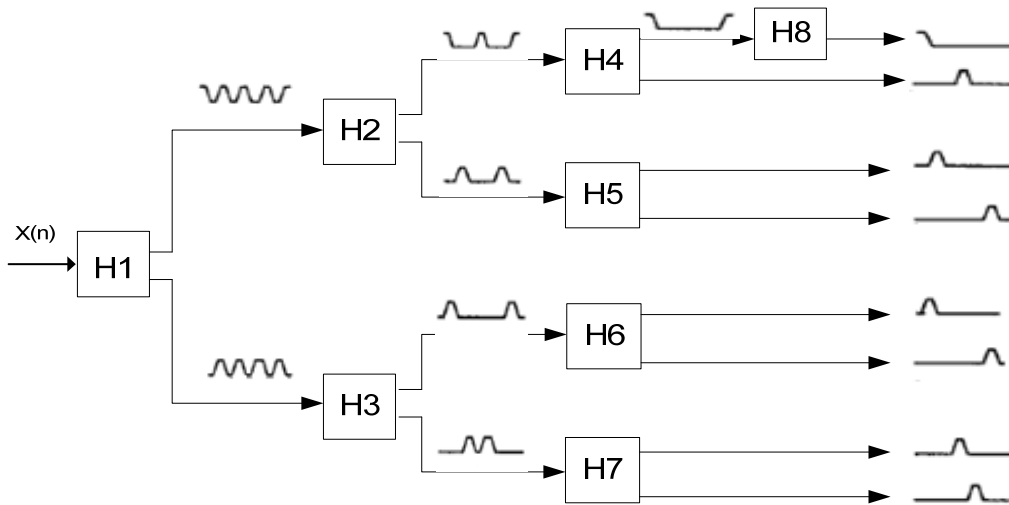
เมื่อพิจารณาความสัมพันธ์ที่ได้มีการอธิบายในภาพประกอบที่ 3-1 นั้นมาทำการเขียนโปรแกรมบนโปรแกรม MATLAB โดยเริ่มต้นจะทำการเปรียบเทียบรูปสัญญาณที่ควรได้จาก  $H_1(z)$  กับกราฟแสดงรูปสัญญาณที่ได้จากการออกแบบบนโปรแกรม MATLAB โดยจะแสดงไว้ตามภาพประกอบ 4-1 โดยผลลัพธ์ที่ได้จะมีเอาต์พุต 2 สายสัญญาณกล่าวคือจะเป็นเอาต์พุตในส่วน  $y_1(n)$  และ  $y_{1c}(n)$  ซึ่งสัญญาณ  $y_1(n)$  นั้น จะเป็นอินพุตให้กับบล็อก H2 ต่อไปส่วนในสายสัญญาณ  $y_{1c}(n)$  ก็จะเป็นสัญญาณอินพุตให้กับบล็อก H3 เช่นกัน และในภาพประกอบที่ 4-2 แสดงถึงผลลัพธ์ที่ได้ ออกมาจาก บล็อกแต่ละบล็อกภายในวงจรดิจิทัลฟิลเตอร์เบงก์โดยลำดับของสายสัญญาณนั้นสามารถอธิบายให้เห็นได้ดังภาพประกอบที่ 4-3



ภาพประกอบ 4-1 สัญญาณที่ออกจาก  $H_1(z)$  ตามเอกสารอ้างอิงกับที่ได้จากการออกแบบ

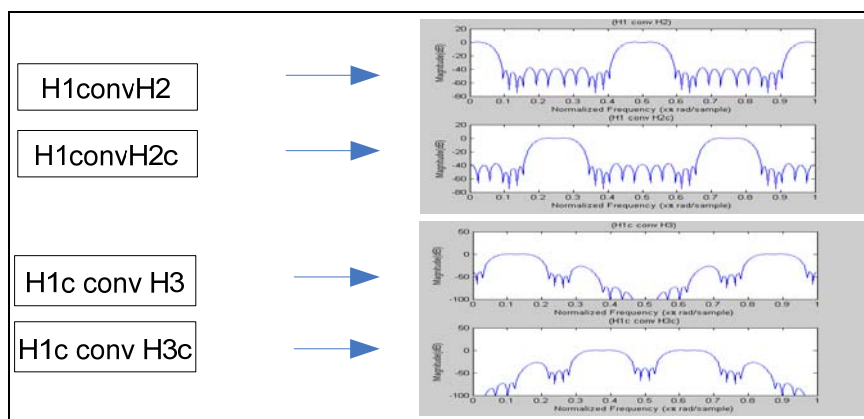


ภาพประกอบ 4-2 ผลลัพธ์สัญญาณภายในของแต่ละบล็อกภายในวงจรฟิลเตอร์เบงก์

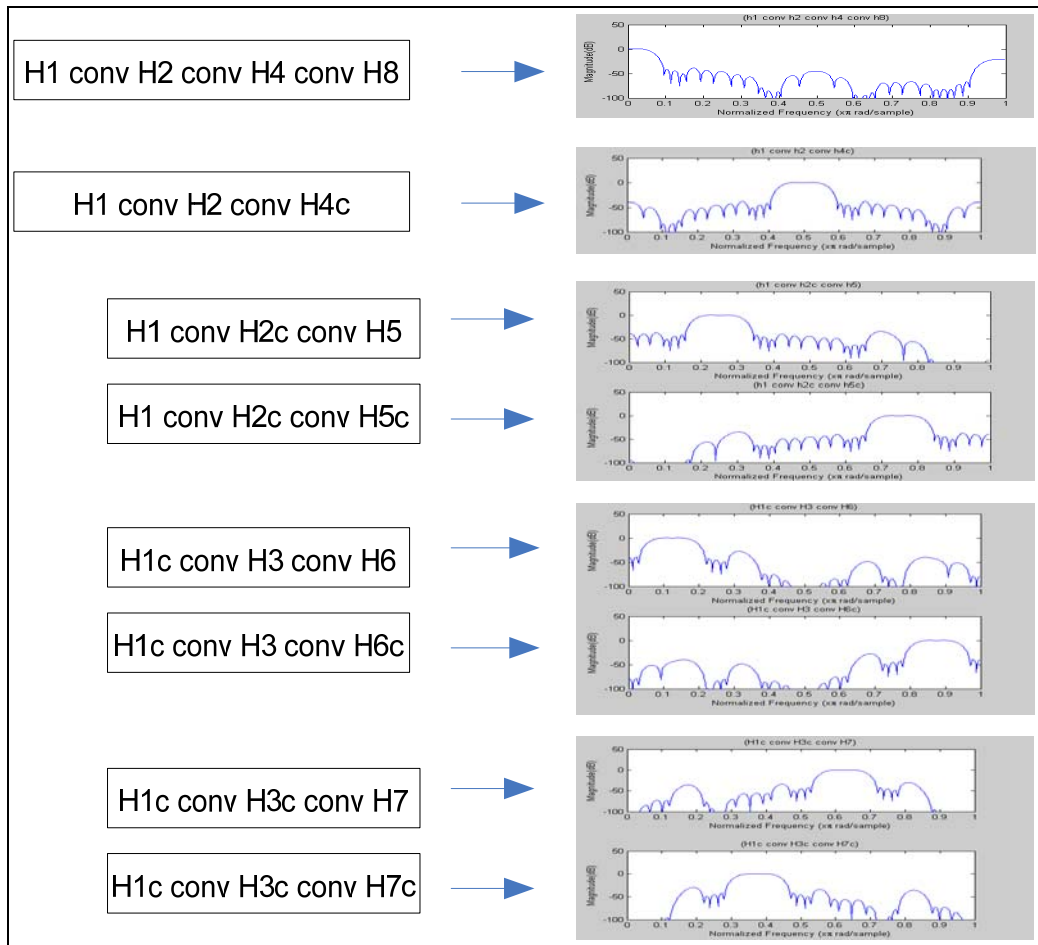


ภาพประกอบ 4-3 สัญญาณเอาต์พุตของฟิลเตอร์แบงก์

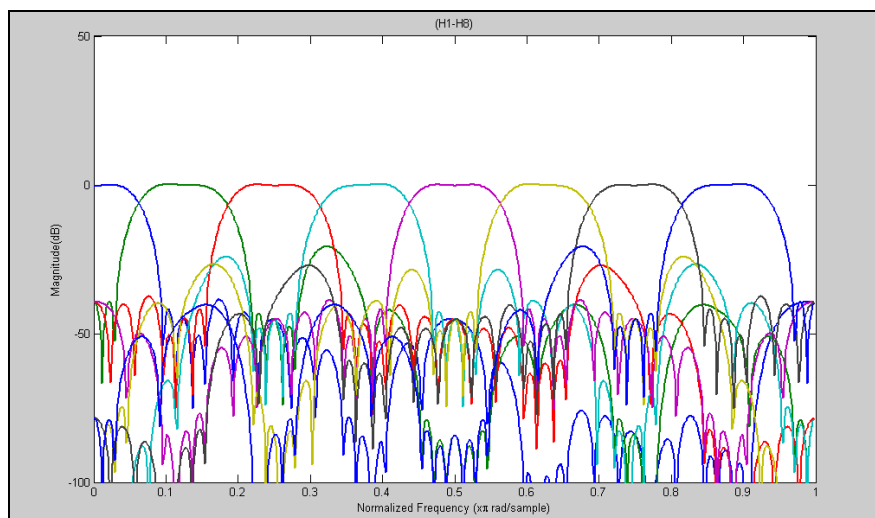
สังเกตสัญญาณด้านออกของฟิลเตอร์ H2 และ H3 จะเกิดจากการที่สัญญาณซึ่งออกจาก H1 ทั้งสองค่าผ่านเข้าไปประมวลผลในฟิลเตอร์ H2 และ H3 ตามลำดับ จากนั้นก็เขียนคำสั่งในโปรแกรม MATLAB เพื่อทำการตรวจสอบโดยจะเป็นการคอนโวลูชันสัญญาณเอาต์พุตจาก H1 กับสัญญาณภายใน H2 และ H3 จะได้สัญญาณของเอาต์พุตดังแสดงด้วยภาพประกอบ 4-4 จากหลักการเดียวกันนี้จะทำการตรวจสอบสัญญาณไปจนได้สัญญาณเอาต์พุตทั้งหมดของฟิลเตอร์แบงก์ ซึ่งหลังจากทำการเขียนคำสั่งและประมวลผลเรียบร้อยแล้ว



ภาพประกอบ 4-4 สัญญาณจากเอาต์พุตของ H1 คอนโวลูชันกับสัญญาณภายใน H2 และ H3



ภาพประกอบ 4-5 สัญญาณเอาต์พุตจากการทำการคอนโวลูชัน

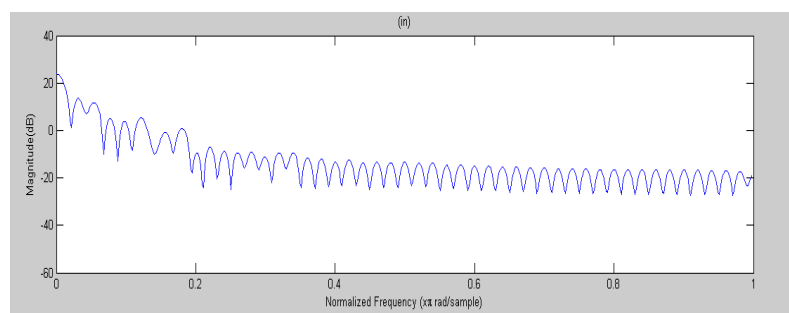


ภาพประกอบ 4-6 กราฟของเอาต์พุตเมื่อรวมกันทั้ง 8 แถบของฟิลเตอร์แบงก์

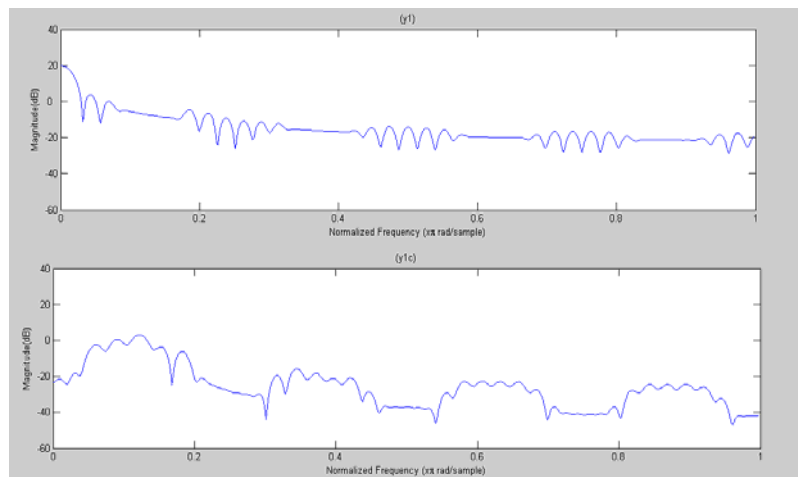
#### 4.2 ผลการวิจัยของวงจรดิจิทัลฟิลเตอร์แบบกึ่งด้วยโปรแกรม MATLAB

หลังจากที่ได้ค่า  $h(n)$  และทราบสมการประมวลผลของวงจรฟิลเตอร์แบบกึ่งแล้วนั้น ขั้นตอนต่อไปก็จะเป็นการทดสอบวงจรด้วยโปรแกรม MATLAB โดยจะเป็นการป้อนค่าอินพุตให้แก่วงจรฟิลเตอร์แบบกึ่งและค่าเอาต์พุตที่ได้จากการทำงานของวงจรจะถูกนำไปใช้ตรวจสอบเปรียบเทียบกับเอาต์พุตที่ได้จากวงจรฟิลเตอร์แบบกึ่งที่ออกแบบบนโปรแกรม Xilinx ต่อไป

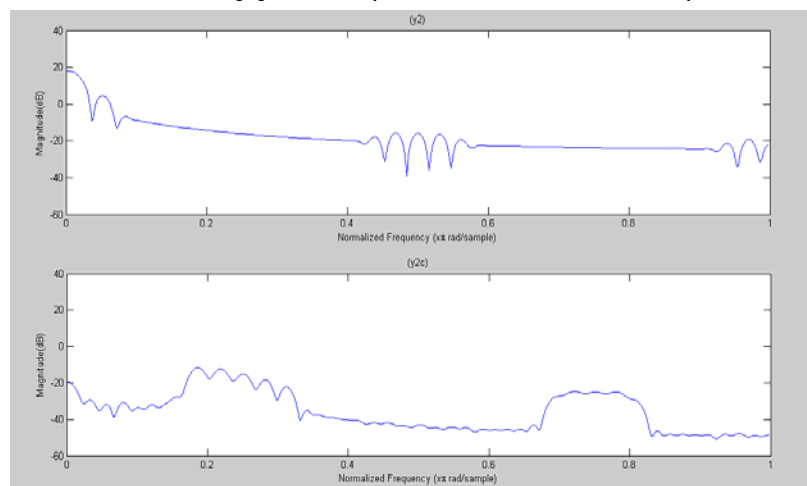
โดยในงานวิจัยนี้จะใช้อินพุตเป็นสัญญาณเสียงเพลงซึ่งจะถูกเก็บค่าไว้ในอาร์เรย์เป็นลำดับใน MATLAB ทั้งนี้สัญญาณเสียงเพลงดังกล่าวจะมีจำนวนอินพุตที่มากลำดับแต่ในการทดสอบจะขอยกอินพุต 100 ลำดับแรกมาใช้ทดสอบเพื่อเปรียบเทียบผลในงานวิจัยนี้โดยในภาพประกอบ 4-7 จะแสดงลักษณะของสัญญาณเสียงเพลงที่นำมาใช้จำนวน 100 ลำดับแรกเพื่อป้อนเป็นอินพุตให้วงจรฟิลเตอร์แบบกึ่งเพื่อประมวลผลหาค่าเอาต์พุตและแสดงสัญญาณเสียงเอาต์พุตที่ได้และในภาพประกอบที่ 4-8, 4-9, 4-10, 4-11, 4-12, 4-13, 4-14 และ 4-15 แสดงถึงสัญญาณเอาต์พุตที่ออกจากบล็อก H1, H2, H3, H4, H5, H6, H7 และบล็อก H8 ตามลำดับ ส่วนในภาพประกอบที่ 4-16 คือสัญญาณเอาต์พุตที่มีการรวมสายสัญญาณเอาต์พุตในแต่ละบล็อกเข้าด้วยกันซึ่งกล่าวได้ว่าภาพประกอบที่ 4-16 นั้นเป็นสัญญาณเอาต์พุตของวงจรดิจิทัลฟิลเตอร์แบบกึ่งแต่ในที่นี้ได้มีการพัฒนาเป็นต้นแบบเครื่องช่วยฟังซึ่งจะมีขั้นตอนที่แตกต่างกันคือสัญญาณเอาต์พุตที่แยกออกมาในแต่ละช่วงความถี่ของฟิลเตอร์แบบกึ่งจะต้องมีการนำค่านั้นมาทำการคูณกับค่าถ่วงน้ำหนักในแต่ละช่วงความถี่ซึ่งได้กล่าวไว้ข้างต้นในบทที่ 2 ดังภาพประกอบที่ 2-12 ซึ่งจะได้กราฟสัญญาณต้นแบบเอาต์พุตของเครื่องช่วยฟังดิจิทัลประมวลผลโดยโปรแกรม MATLAB ดังภาพประกอบที่ 4-17, 4-18, 4-19, 4-20, 4-21, 4-22, 4-23 แสดงถึงค่าเอาต์พุตที่ได้หลังจากผ่านการคูณค่าถ่วงน้ำหนักตั้งแต่ช่วงสัญญาณที่ 1 ถึงช่วงสัญญาณที่ 8 คือ  $y_8, y_{4c}, y_5, y_{5c}, y_6, y_{6c}, y_7, y_{7c}$  ตามลำดับ และในภาพประกอบที่ 4-24 จะแสดงถึงเอาต์พุตของต้นแบบเครื่องช่วยฟังที่มีการรวมสัญญาณทั้ง 8 ช่วงเข้าด้วยกันแล้ว



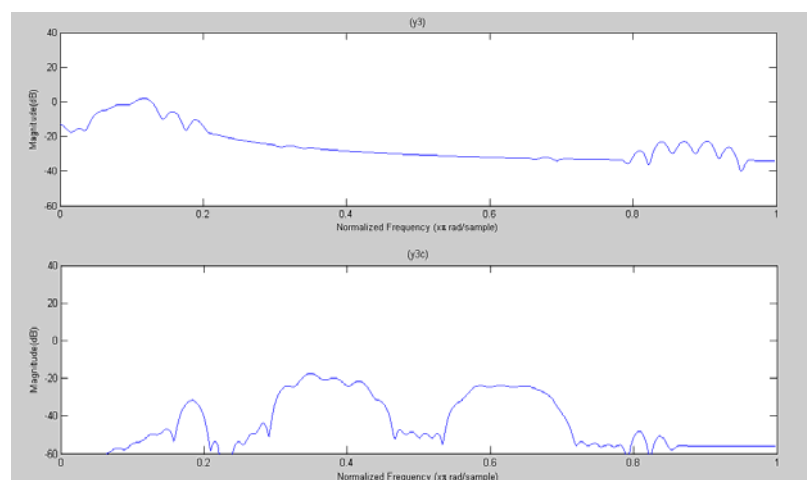
ภาพประกอบ 4-7 กราฟสัญญาณของอินพุตที่ใช้ทำการประมวลผลบน MATLAB



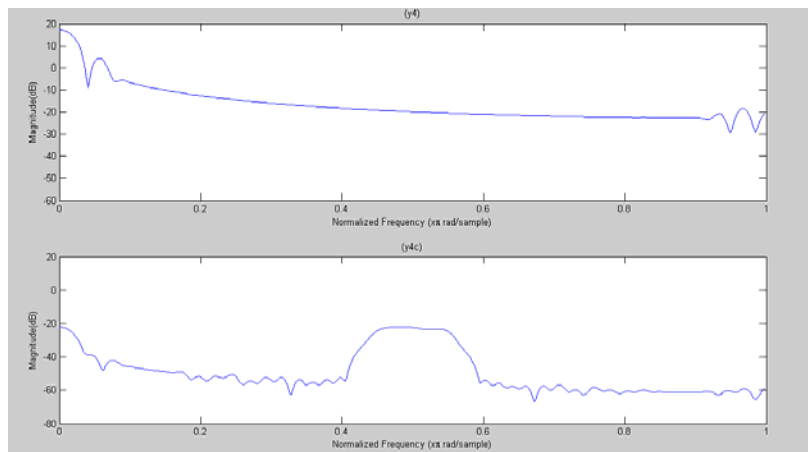
ภาพประกอบ 4-8 กราฟสัญญาณเอาต์พุต  $y_1$  และ  $y_{1c}$  ที่ได้จากอินพุตผ่านฟิลเตอร์ H1



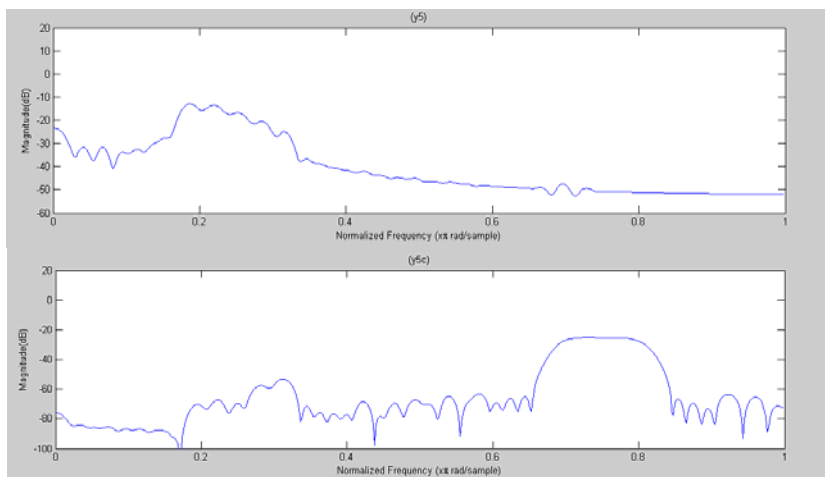
ภาพประกอบ 4-9 กราฟสัญญาณ  $y_2$  และ  $y_{2c}$  ที่ได้จากเอาต์พุต  $y_1$  ผ่านฟิลเตอร์ H2



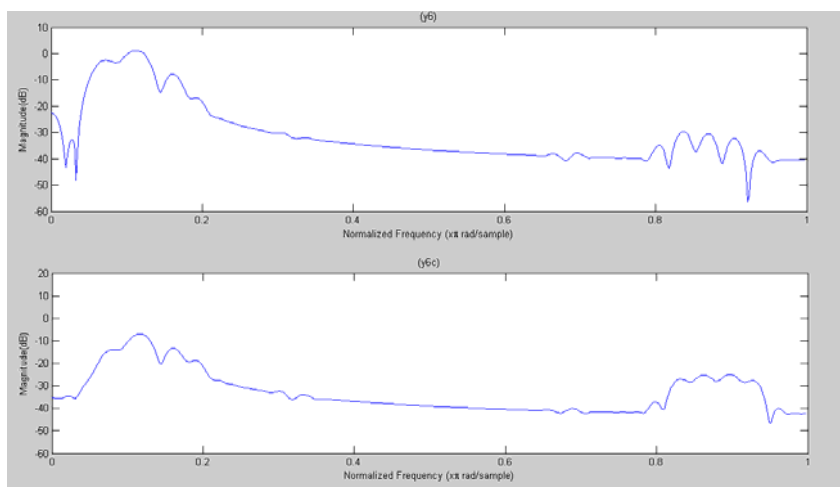
ภาพประกอบ 4-10 กราฟสัญญาณ  $y_3$  และ  $y_{3c}$  ที่ได้จากเอาต์พุต  $y_{1c}$  ผ่านฟิลเตอร์ H3



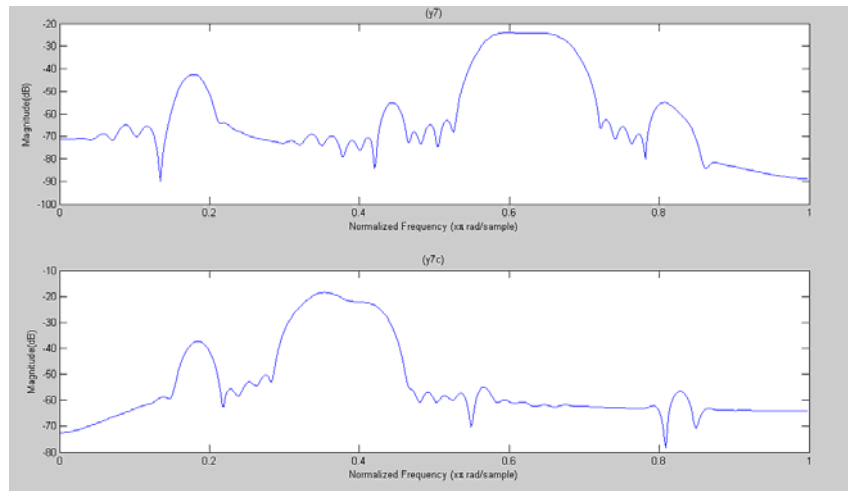
ภาพประกอบ 4-11 กราฟสัญญาณ  $y_4$  และ  $y_{4c}$  ที่ได้จากเอาต์พุต  $y_2$  ผ่านฟิลเตอร์  $H_4$



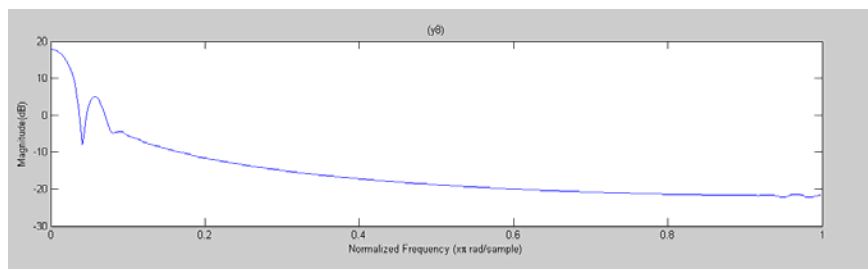
ภาพประกอบ 4-12 กราฟสัญญาณ  $y_5$  และ  $y_{5c}$  ที่ได้จากเอาต์พุต  $y_{2c}$  ผ่านฟิลเตอร์  $H_5$



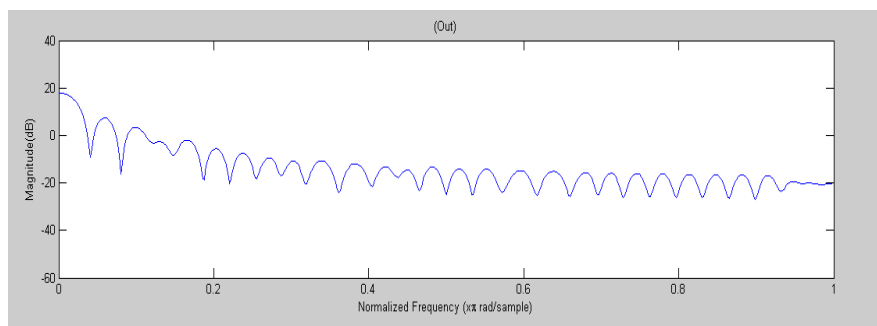
ภาพประกอบ 4-13 กราฟสัญญาณ  $y_6$  และ  $y_{6c}$  ที่ได้จากเอาต์พุต  $y_3$  ผ่านฟิลเตอร์  $H_6$



ภาพประกอบ 4-14 กราฟสัญญาณ  $y7$  และ  $y7c$  ที่ได้จากเอาต์พุต  $y3c$  ผ่านฟิลเตอร์ H7

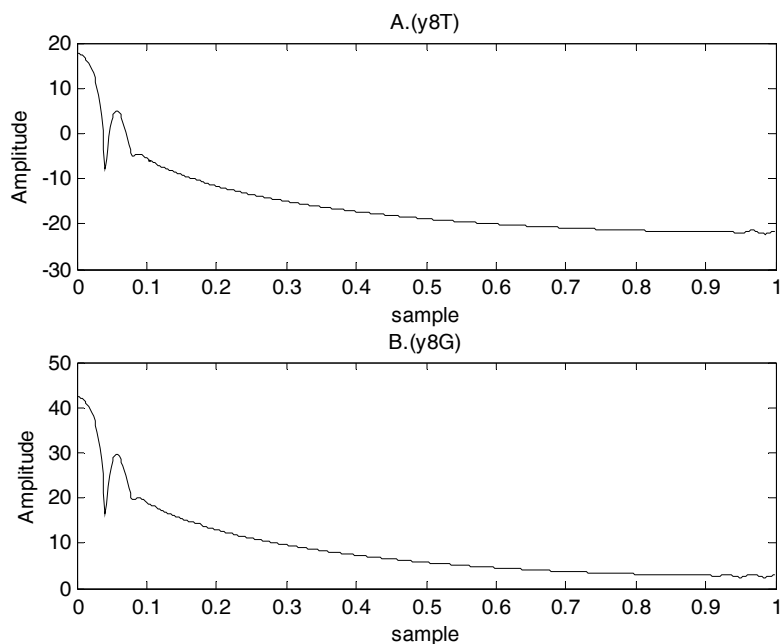


ภาพประกอบ 4-15 กราฟสัญญาณ  $y8$  ที่ได้จากเอาต์พุต  $y4$  ผ่านฟิลเตอร์ H8

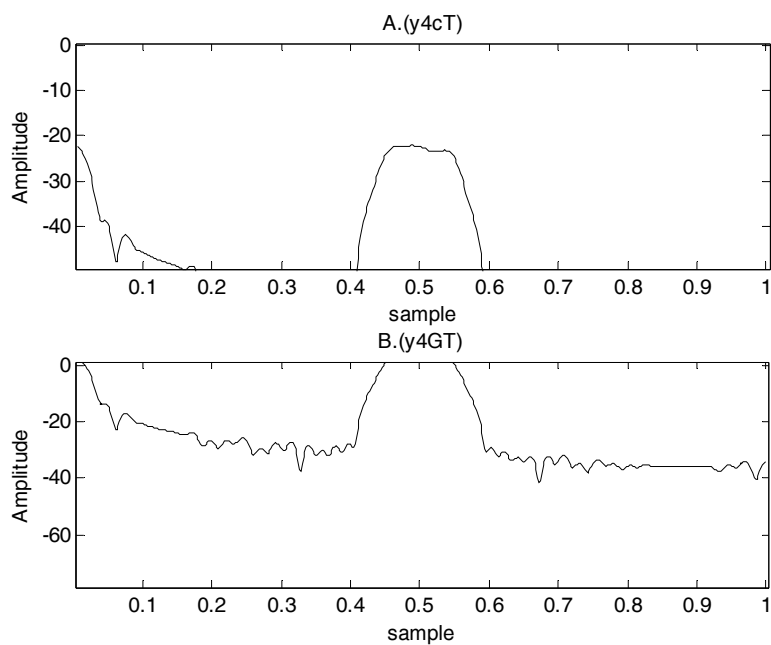


ภาพประกอบ 4-16 กราฟสัญญาณของเอาต์พุตที่ได้จากการประมวลผลบน MATLAB

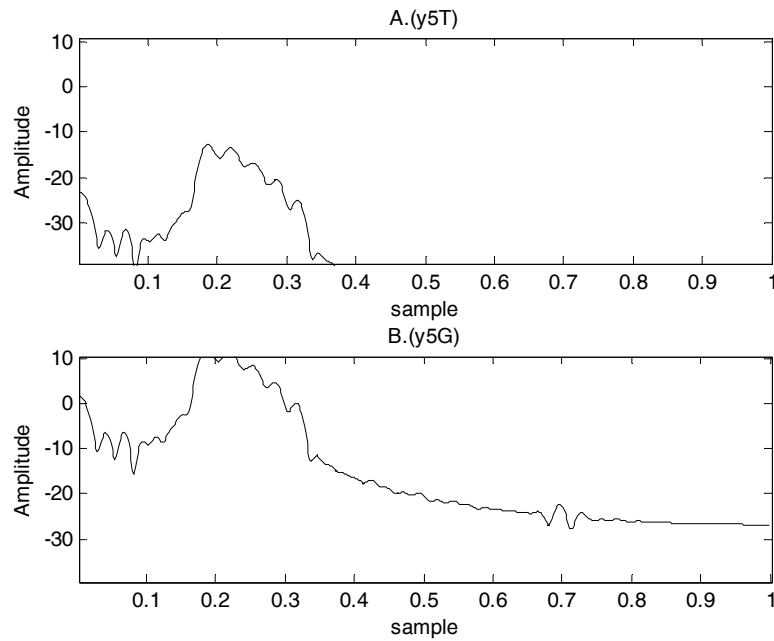




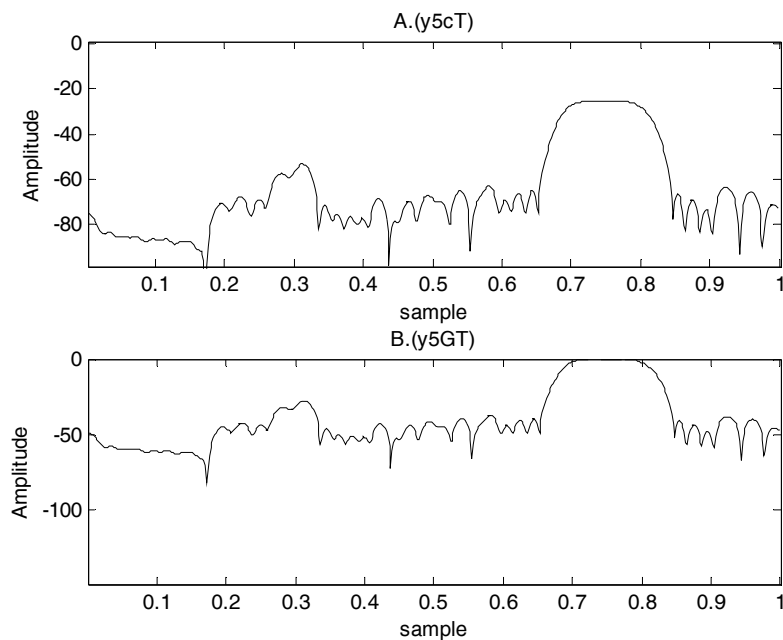
ภาพประกอบ 4-17 กราฟสัญญาณของเอาท์พุทของช่วงความถี่ช่วงแรกโดยที่  
 A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่าถ่วงน้ำหนัก)



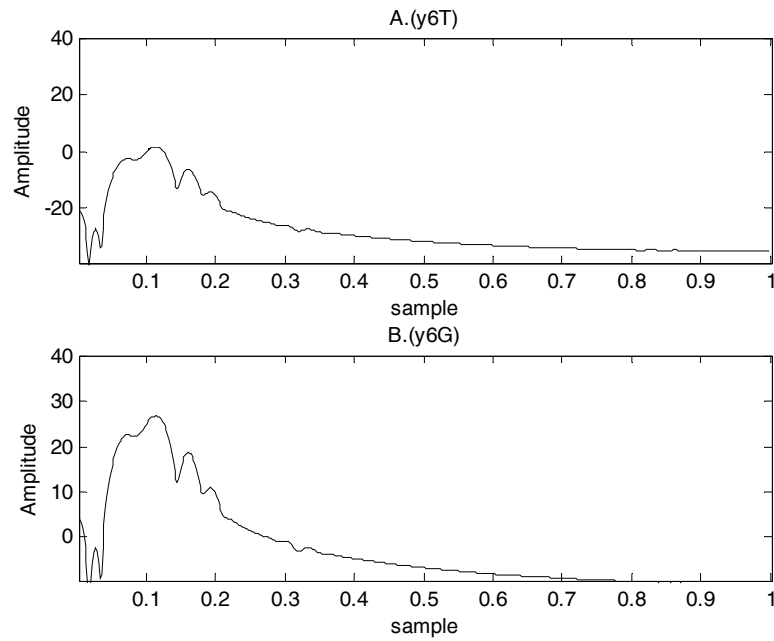
ภาพประกอบ 4-18 กราฟสัญญาณของเอาท์พุทของช่วงความถี่ช่วงที่ 2 โดยที่  
 A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



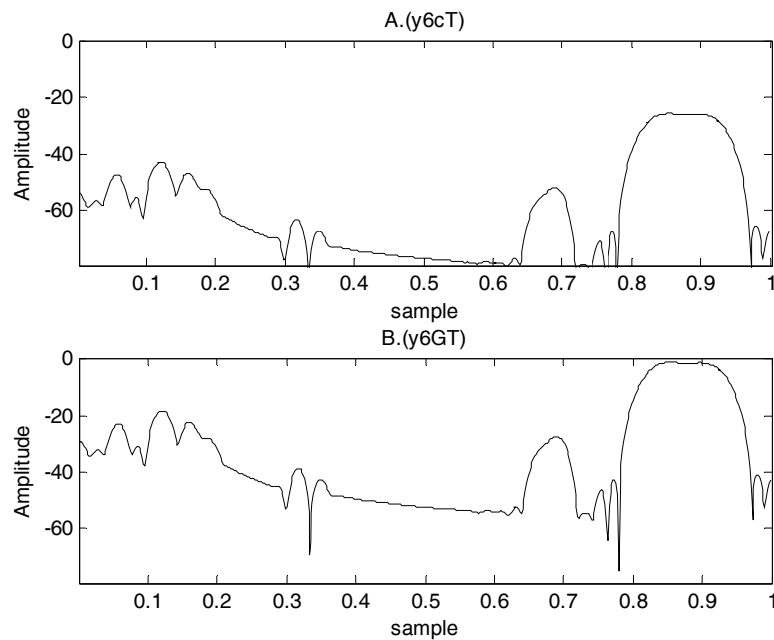
ภาพประกอบ 4-19 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่3 โดยที่  
A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



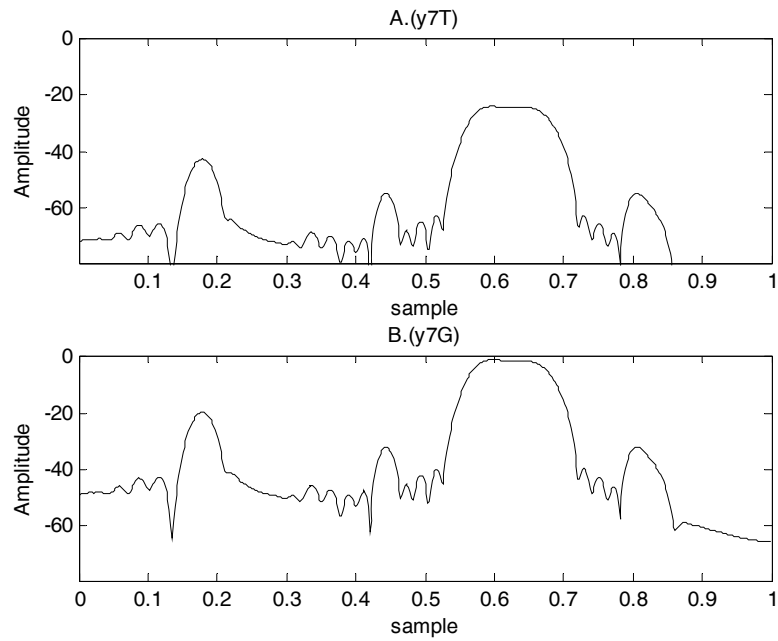
ภาพประกอบ 4-20 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่4 โดยที่  
A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



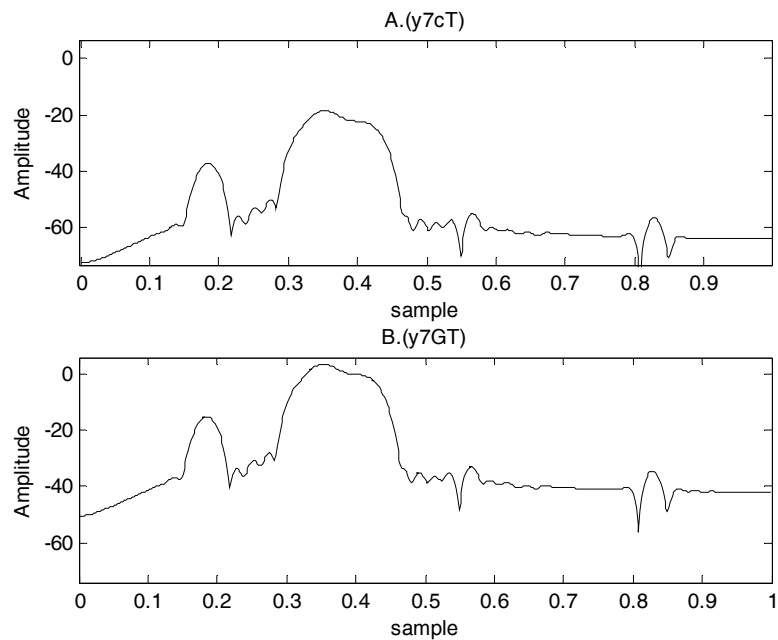
ภาพประกอบ 4-21 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 1 โดยที่  
A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



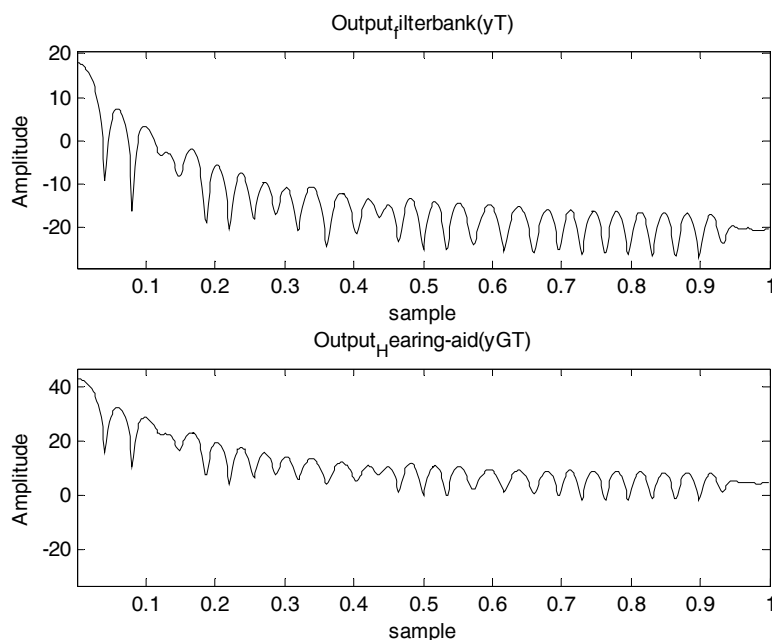
ภาพประกอบ 4-22 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 6 โดยที่  
A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



ภาพประกอบ 4-23 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 7 โดยที่  
A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



ภาพประกอบ 4-24 กราฟสัญญาณของเอาต์พุตของช่วงความถี่ช่วงที่ 8 โดยที่  
A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)



ภาพประกอบ 4-25 กราฟสัญญาณของเอาต์พุตรวมทุกช่วงความถี่โดยที่

A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)

#### 4.3 ผลการวิจัยของวงจรดิจิทัลฟิลเตอร์แบ่งกัด้วยโปรแกรม Xilinx

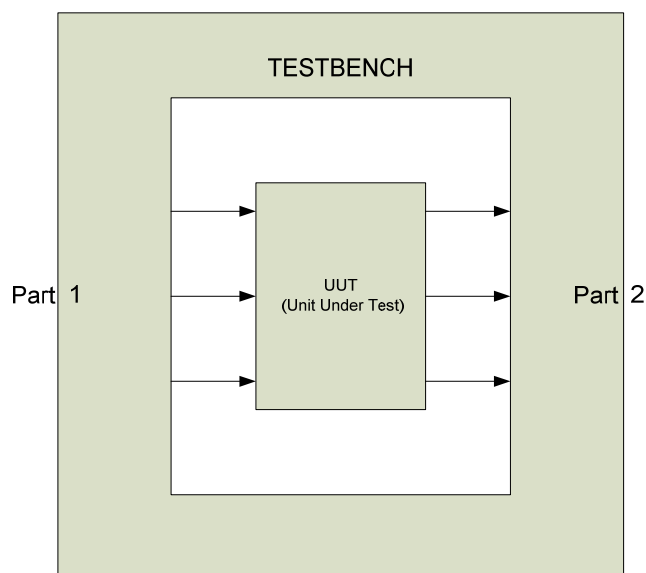
เมื่อผ่านการทดสอบการทำงานของวงจรฟิลเตอร์แบ่งกับนโปรแกรม MATLAB มาแล้วนั้นในลำดับถัดไปก็คือการออกแบบและทดสอบการทำงานของฟิลเตอร์แบ่งกับน FPGAs หรือหมายถึงการออกแบบและทดสอบบนโปรแกรม Xilinx นั้นเอง โดยที่ลำดับการออกแบบจะเป็นลำดับดังนี้คือ เริ่มต้นจากการออกแบบวงจรในรูปแบบธรรมดาคือการออกแบบวงจรแบบเดียวกันกับที่ออกแบบบน MATLAB จากนั้นจะออกแบบวงจรโดยจะใช้ระเบียบวิธีที่นำเสนอในงานวิจัยนี้คือการออกแบบโดยใช้ทรัพยากรร่วมแบบลำดับขั้นและการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมซึ่งในงานวิจัยนี้มีการเปรียบเทียบกัน 3 รูปแบบกล่าวคือในรูปแบบแรกเป็นการใช้ทรัพยากรร่วมแบบลำดับขั้น ในรูปแบบที่สองเป็นการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม และในรูปแบบสุดท้ายเป็นการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมร่วมกับการลดจำนวนบิต

งานวิจัยนี้จะออกแบบวงจรและทดสอบบนชิพ FPGA ของบริษัท Xilinx ตระกูล Spartan3f4000fg900-4 ในการออกแบบจะเป็นการเขียนคำสั่งเพื่อบรรยายวงจรด้วยภาษา Verilog และใช้โปรแกรม Xilinx ISE 10.1 ในการสังเคราะห์ (Synthesis) เพื่อดูผลจากการสังเคราะห์วงจรทั้งด้านการใช้ทรัพยากรบน FPGA และความเร็วในการทำงานของวงจรจากนั้นนำวงจรที่ได้ทำการจำลองการทำงาน (Simulation) เพื่อตรวจสอบผลการทำงานของวงจรโดยจะมีการเปรียบเทียบ

ระหว่างการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและเทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น

โดยจะทำการออกแบบวงจรมาจนถึงขั้นตอนการจำลองการทำงานระดับฐานเวลาจริง (Timing Simulation) แล้วทำการทดสอบวงจรด้วยการเขียน Testbench ซึ่งเป็นการสร้างระบบการทดสอบ (Test environment) ของวงจรที่ต้องการทดสอบการทำงาน

Testbench สามารถเขียนขึ้นจากภาษา Verilog โดยหน้าที่ของ Testbench คือการทำหน้าที่สร้างสัญญาณกระตุ้นให้กับวงจรที่ต้องการทดสอบการทำงานและดูผลจากการทดสอบได้



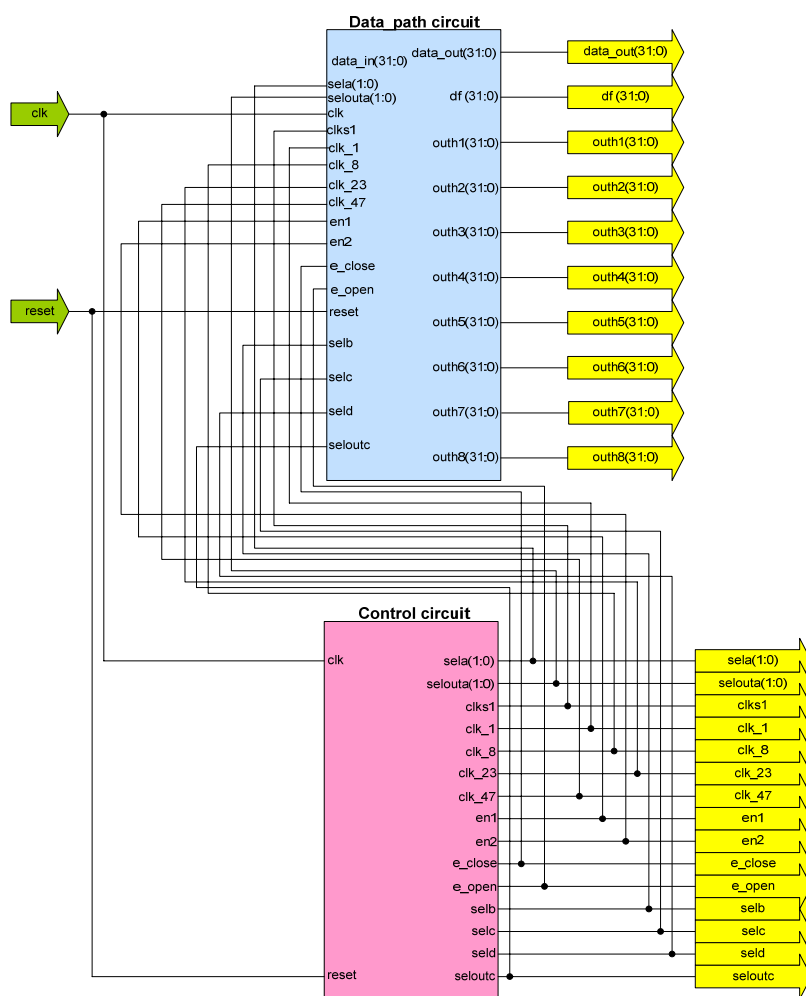
ภาพประกอบ 4-26 ระบบการทดสอบ (Testbench)

จากภาพประกอบ 4-26 จะเห็นได้ว่าในระบบทดสอบ (Testbench) จะถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนที่ 1 ทำหน้าที่สร้างสัญญาณกระตุ้นให้กับสัญญาณอินพุตของวงจรที่อยู่ในระบบหรือที่เรียกว่า UUT: Unit Under Test และส่วนที่ 2 จะเป็นส่วนของการตรวจสอบสัญญาณเอาต์พุตหลังจากการกระตุ้นสัญญาณให้กับสัญญาณอินพุตของวงจรเรียบร้อยแล้ว

### 4.3.1 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบลำดับชั้น

#### (Hierarchal Resource Sharing Technique)

ในการออกแบบวงจรรูปแบบแรกสำหรับงานวิจัยนี้เป็นการออกแบบวงจรดิจิทัลฟิลเตอร์แบงก์ด้วยเทคนิคการใช้ทรัพยากรร่วมกันแบบลำดับชั้นเมื่อทำการสังเคราะห์วงจรจะได้แผนผัง RTL (RTL Schematic) ของวงจรดิจิทัลฟิลเตอร์ดังกล่าวประกอบที่ 4-27 นอกจากนั้นยังสามารถแสดงพื้นที่ที่ใช้ในการออกแบบวงจรได้โดยพิจารณาจาก Device Utilization Summary ดังภาพประกอบที่ 4-28 ภาพประกอบที่ 4-29, 4-30 และ 4-31 แสดงถึง Timing Summary เพื่อใช้ตรวจสอบเวลาที่ใช้ในการประมวลผล, ผลการจำลองการทำงานของวงจร และ กราฟเปรียบเทียบเอาต์พุตที่ได้จากวงจรฟิลเตอร์แบงก์บน MATLAB กับ เอาต์พุตของวงจรฟิลเตอร์แบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมกันแบบลำดับชั้น



ภาพประกอบ 4-27 RTL Schematic ของวงจรฟิลเตอร์แบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมกันแบบลำดับชั้น

Device utilization summary:

-----

Selected Device : 3s4000fg900-4

Number of Slices:	13212	out of	27648	47%
Number of Slice Flip Flops:	5851	out of	55296	10%
Number of 4 input LUTs:	24832	out of	55296	44%
Number of bonded IOBs:	369	out of	633	58%
Number of MULT18X18s:	10	out of	96	10%
Number of GCLKs:	5	out of	8	62%

ภาพประกอบ 4-28 แสดง Device Utilization Summary ของวงจรฟิลเตอร์แบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น

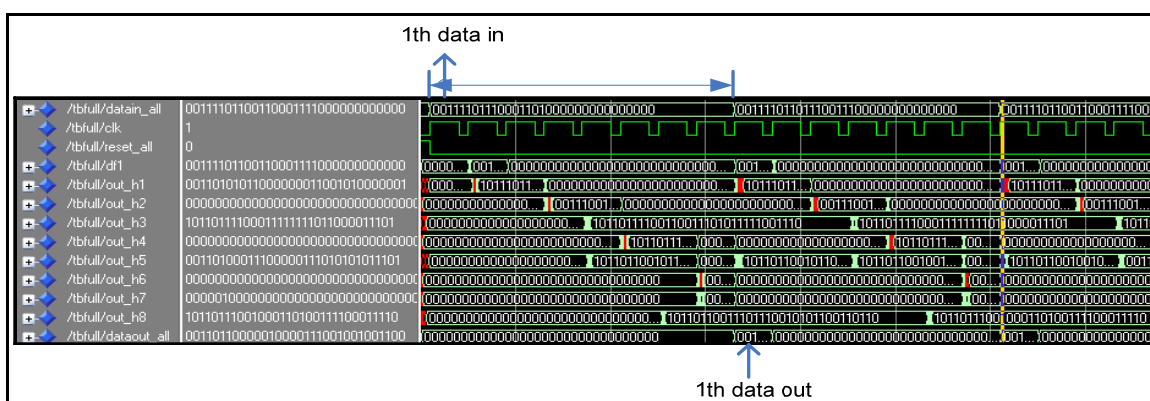
Timing Summary:

-----

Speed Grade: -4

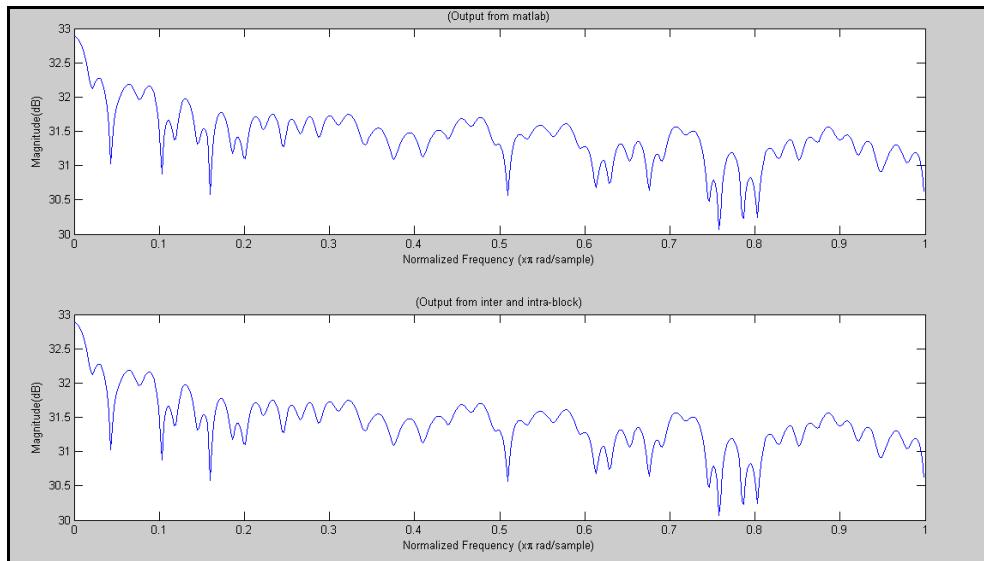
Minimum period: 299.464ns (Maximum Frequency: 3.339MHz)  
 Minimum input arrival time before clock: 11.048ns  
 Maximum output required time after clock: 156.147ns  
 Maximum combinational path delay: 17.366ns

ภาพประกอบ 4-29 แสดง Timing Summary ของวงจรวจรฟิลเตอร์แบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น



ภาพประกอบ 4-30 แสดงผลการจำลองการทำงานของวงจรวจรฟิลเตอร์แบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น

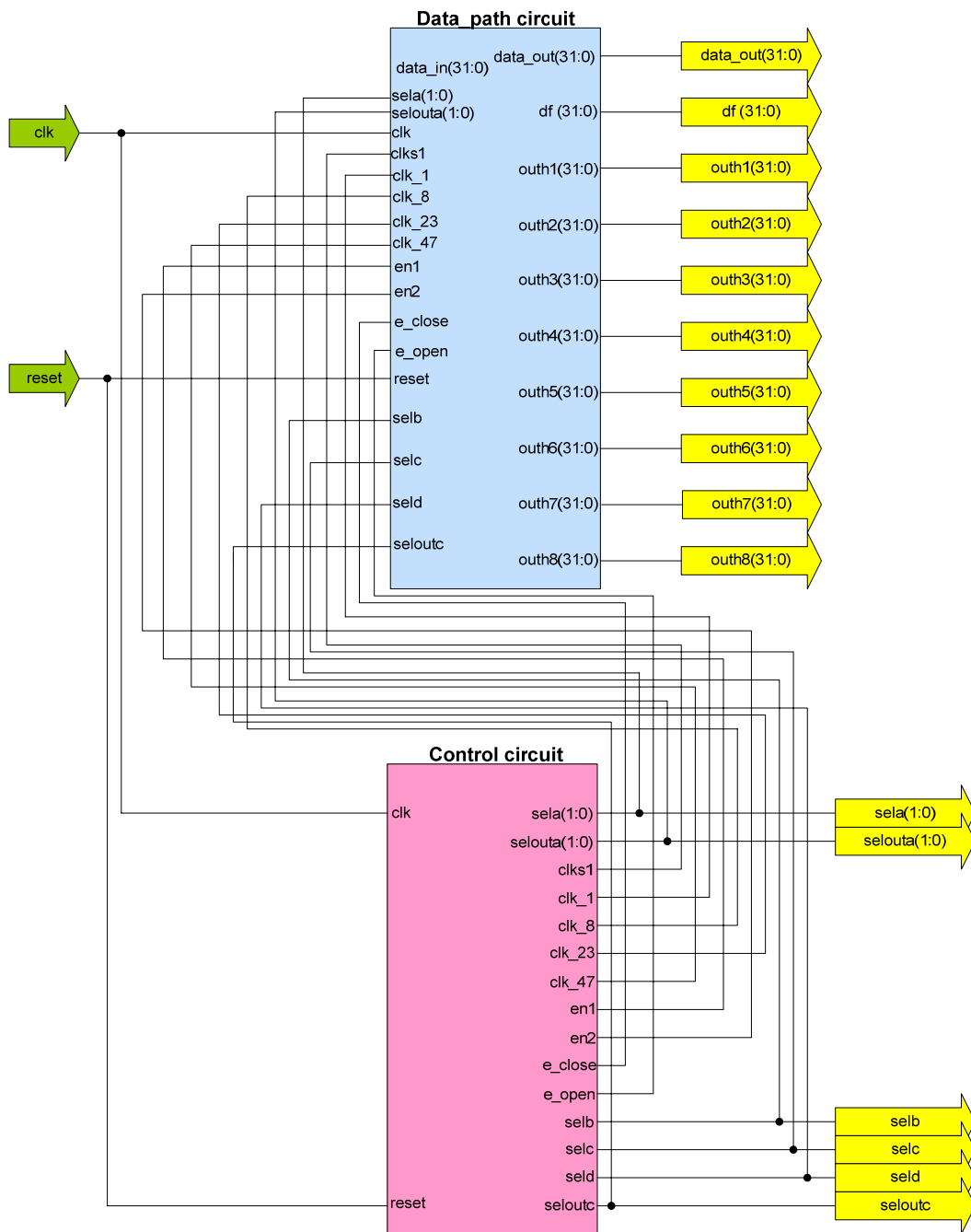




ภาพประกอบ 4-31 กราฟเปรียบเทียบเอาต์พุตที่ได้จากวงจรฟิลเตอร์เบงก์บน MATLAB กับ เอาต์พุตวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น

#### 4.3.2 วงจรฟิลเตอร์เบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม (Global Resource Sharing Technique)

จากนั้นการออกแบบวงจรในขั้นตอนต่อไปก็จะนำระเบียบวิธีที่นำเสนอคือการ ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวมมาใช้ในการออกแบบ เพื่อดูพฤติกรรมการทำงานของ วงจรว่าทำงานได้ผลเอาต์พุตเหมือนกับวงจรที่ไม่มีการใช้ทรัพยากรร่วมกันหรือไม่ โดยเมื่อทำการ ตั้งเคราะห์วงจรจะได้แผนผัง RTL (RTL Schematic) ของวงจรดิจิทัลฟิลเตอร์ดังภาพประกอบ ที่ 4-32 นอกจากนั้นยังสามารถแสดงพื้นที่ที่ใช้ในการออกแบบวงจรได้โดยพิจารณาจาก Device Utilization Summary ดังภาพประกอบที่ 4-33 ภาพประกอบที่ 4-34, 4-35 และ 4-36 แสดงถึง Timing Summary เพื่อใช้ตรวจสอบเวลาที่ใช้ในการประมวลผล, ผลการจำลองการทำงานของวงจร และ กราฟเปรียบเทียบเอาต์พุตที่ได้จากวงจรฟิลเตอร์เบงก์บน MATLAB กับ เอาต์พุตวงจร ฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม



ภาพประกอบ 4-32 RTL Schematicของวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบ  
พิจารณาภาพรวม

```
=====
Device utilization summary:
-----
```

```
Selected Device : 3s4000fg900-4
```

```
Number of Slices:                9446 out of 27648   34%
Number of Slice Flip Flops:      3081 out of 55296    5%
Number of 4 input LUTs:         17571 out of 55296   31%
    Number used as logic:        17191
    Number used as Shift registers: 380
Number of IOs:                   402
Number of bonded IOBs:          362 out of 633     57%
Number of MULT18X18s:           38 out of 96       39%
Number of GCLKs:                 5 out of 8         62%
```

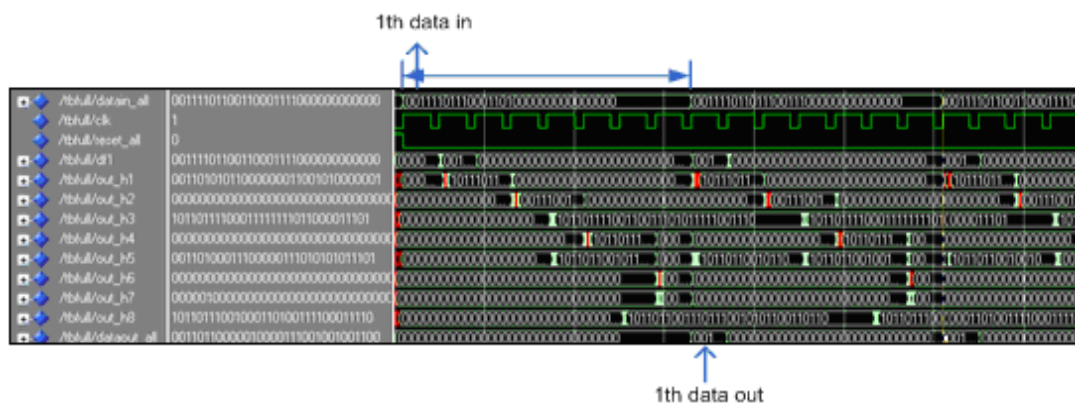
ภาพประกอบ 4-33 แสดง Device Utilization Summary ของวงจรรีไซมิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม

```
Timing Summary:
-----
```

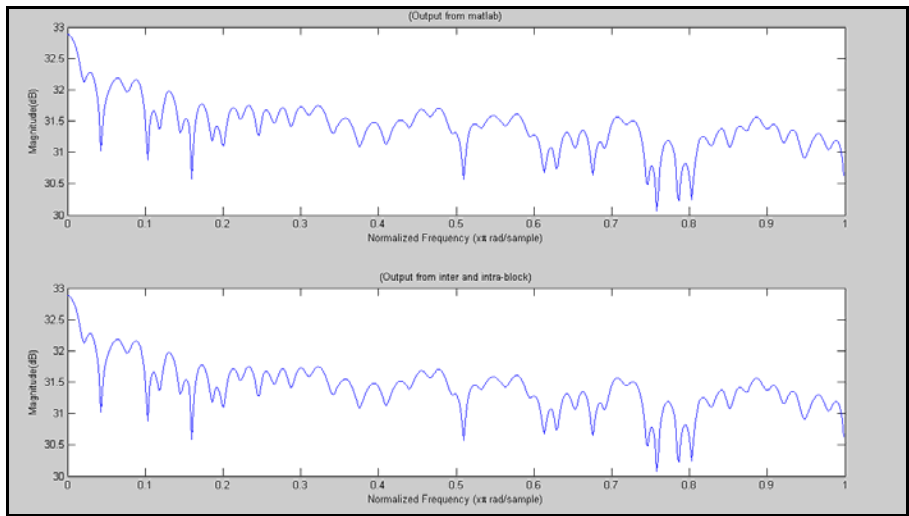
```
Speed Grade: -4
```

```
Minimum period: 603.458ns (Maximum Frequency: 1.657MHz)
Minimum input arrival time before clock: 9.885ns
Maximum output required time after clock: 201.735ns
Maximum combinational path delay: 11.299ns
```

ภาพประกอบ 4-34 แสดง Timing Summary ของวงจรรีไซมิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม



ภาพประกอบ 4-35 แสดงผลการจำลองการทำงานของวงจรรีไซมิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น



ภาพประกอบ 4-36 กราฟเปรียบเทียบเอาต์พุตที่ได้จาก เอาต์พุตทวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้น กับ เอาต์พุตทวงจรฟิลเตอร์เบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม

**4.3.3 วงจรฟิลเตอร์เบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม (Global Resource Sharing Technique) ร่วมกับการลดบิตข้อมูล**

ในส่วนนี้จะนำวงจรที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมที่มีการเปรียบเทียบผลว่าถูกต้องแล้วมาทำการลดบิตข้อมูลจาก 32 บิต ให้เหลือ 26 บิต โดยเกณฑ์ในการพิจารณาการลดบิตข้อมูลจาก 32 บิต ให้เหลือ 26 บิต เนื่องจากการลดบิตลงจาก 32 บิต ให้เหลือ 26 บิต จะมีผลในการลดจำนวนตัวคูณ MUL18x18 ลงซึ่ง ภายในวงจรคูณ 1 วงจรสามารถลดจำนวนตัวคูณ MUL18x18 จาก 4 ตัวลดเหลือเพียง 1 ตัวในที่นี้บิตที่ทำการลดไปจะมีการลดเฉพาะในส่วนของ Mantissa ซึ่งเป็นส่วนที่ใช้ในการคูณเท่านั้นโดยบิตที่ทำการลดไปจะไม่ส่งผลกับข้อมูล โดยสามารถแสดงรูปแบบของบิตข้อมูล 32 บิต และ 26 บิตได้ดังภาพประกอบที่ 4-37 และ 4-38 โดยในรูปแบบข้อมูลจะประกอบไปด้วย 3 ส่วนคือ ส่วน sign, exponent และส่วน mantissa

<b>Sign bit(1 bit)</b>	<b>Exponent(8 bit)</b>	<b>Mantissa(24 bit)</b> <b>Mantissa(24 bit)= 1.Fraction(23 bit)</b>
------------------------	------------------------	--

ภาพประกอบที่ 4-37 รูปแบบของชุดข้อมูล 32 บิต

<b>Sign bit(1 bit)</b>	<b>Exponent(8 bit)</b>	<b>Mantissa(18 bit)</b> <b>Mantissa(18 bit)= 1.Fraction(17 bit)</b>
------------------------	------------------------	--

ภาพประกอบที่ 4-38 รูปแบบของชุดข้อมูล 26 บิต

โดยในแต่ละส่วนทำงานดังนี้

4.3.3.1. ค่า Sign จะนำค่า S ของทั้งสองตัวมาทำการ Exclusive-Or (XOR) กัน คือ ถ้าค่า  $S_1$  เหมือนกันกับค่า  $S_2$  จะได้ค่า  $S_3$  เป็น 0 (Zero) ดังนั้นค่า Sign ของผลลัพธ์จะเป็นค่าบวก แต่ถ้าค่า  $S_1$  ต่างกันกับค่า  $S_2$  จะได้ค่า  $S_3$  เป็น 1 ดังนั้นค่า Sign ของผลลัพธ์จะเป็นค่าลบ นั่นก็คือ  $S_3 = S_1 \oplus S_2$  สามารถแสดงตารางความจริง ดังตารางที่ 4-1

$S_1$	$S_2$	$S_3$
0	0	0
0	1	1
1	0	1
1	1	0

ตารางที่ 4-1 แสดงตารางความจริง (Truth Table) ของการ Exclusive-Or ค่า Sign

4.3.3.2. ค่าของเอ็กซ์โปเนนต์ (Exponent) จะนำค่าเลขยกกำลังมาบวกกัน และลบออกด้วย 127 ซึ่งเป็นค่าไบอัสของแบบ Single Precision เพราะในการคูณจำนวนทศนิยมนั้น ค่าของเลขยกกำลังจะเกินมา 127 จาก  $E_1 = E_1^{true} + 127$  บวกกับ  $E_2 = E_2^{true} + 127$  ดังนั้นจึงต้องนำมาลบออก 127 จึงจะเป็นค่า Exponent ที่ถูกต้อง

4.3.3.3. ค่าแมนทิสซา (Mantissa) หรือ ซิกนิฟิแคนด์ (Significant) ก็จะเป็นการคูณกัน โดยการเอาค่า  $M_1 \times M_2$  คูณกันธรรมดาได้ค่าเป็น 48 บิต โดยของที่ไว้สำหรับ Fraction 46 บิต และจงสำหรับหน้าจุดทศนิยม 2 บิต ซึ่งโอกาสที่จะเกิดขึ้นมี 3 กรณี คือ 01.F, 10.F และ 11.F ตามลำดับ

เมื่อพิจารณาจากการทำงานงานทั้ง 3 ส่วนพบว่าการทำงานที่จะลดขนาดของบิตข้อมูล โดยให้ข้อมูลมีความถูกต้องมากที่สุดและสามารถลดขนาดของตัวคูณ MUL 18x18 ลงได้จึงมีการใช้รูปแบบข้อมูล 26 บิต แต่ในที่นี้จะมีการเปรียบเทียบผลชุดข้อมูลเพิ่มอีก 1 ชุดขนาดข้อมูลเพื่อแสดงให้เห็นว่าถึงแม้จะมีการลดขนาดของบิตข้อมูลลงก็ไม่ได้ส่งผลต่อขนาดตัวคูณ MUL 18x18 ลงโดยข้อมูลที่จะนำมาเปรียบเทียบจะมีการใช้จำนวนชุดข้อมูลขนาด 16 บิต มาร่วมเปรียบเทียบด้วยโดยสามารถแสดงรูปแบบของชุดข้อมูลได้ดังภาพประกอบที่ 4-39

Sign bit(1 bit)	Exponent(8 bit)	Mantissa(8 bit)
		Mantissa(8bit)= 1.Fraction(7 bit)

ภาพประกอบที่ 4-39 รูปแบบของชุดข้อมูล 16 บิต

โดยในที่นี้จะทำการเปรียบเทียบขนาดวงจรและจำนวนตัวคูณ MUL18x18 ในการสร้างวงจรคูณเลขทศนิยม 1 วงจรซึ่งสามารถแสดงผลการสังเคราะห์วงจรได้ดังภาพประกอบที่ 4-40, 4-41 และ 4-42 โดยจะแสดงถึงผลการสังเคราะห์วงจรคูณเลขทศนิยมของจำนวนชุดข้อมูล 32 บิต, 26 บิต และ 16 บิตตามลำดับ

```
Device utilization summary:
-----

Selected Device : 3s400tq144-4

Number of Slices:                80 out of 3584    2%
Number of 4 input LUTs:          147 out of 7168    2%
Number of IOs:                   96
Number of bonded IOBs:           96 out of 97      98%
Number of MULT18X18s:            4 out of 16      25%
```

ภาพประกอบที่ 4-40 ผลการสังเคราะห์วงจรคูณเลขทศนิยมของจำนวนชุดข้อมูล 32 บิต

```
Device utilization summary:
-----

Selected Device : 3s400tq144-4

Number of Slices:                38 out of 3584    1%
Number of 4 input LUTs:          67 out of 7168    0%
Number of IOs:                   78
Number of bonded IOBs:           76 out of 97      78%
Number of MULT18X18s:            1 out of 16      6%
```

ภาพประกอบที่ 4-41 ผลการสังเคราะห์วงจรคูณเลขทศนิยมของจำนวนชุดข้อมูล 26 บิต

```
Device utilization summary:
-----

Selected Device : 3s400tq144-4

Number of Slices:                35 out of 3584    0%
Number of 4 input LUTs:          62 out of 7168    0%
Number of bonded IOBs:           46 out of 97      47%
Number of MULT18X18s:            1 out of 16      6%
```

ภาพประกอบที่ 4-42 ผลการสังเคราะห์วงจรคูณเลขทศนิยมของจำนวนชุดข้อมูล 16 บิต

เมื่อพิจารณาพบว่าถึงจะมีการลดขนาดบิตลงไปถึง 16 บิต พบว่าขนาดที่ได้รับความนิยม 1 วงจรกลับมีขนาดใกล้เคียงกันเนื่องจากในวงจรชุดข้อมูล 26 บิต หรือ 32 บิตต่างใช้ ตัวคูณ MUL18x18 1 ตัว เท่ากันในการสร้างวงจรคูณเลขทศนิยม 1 วงจรจึงทำให้มีการเลือกใช้ชุดข้อมูล 26 บิต แทนที่จะใช้ ชุดข้อมูล 16 บิตเนื่องจาก ในการลดจำนวนบิต ข้อมูลลงจาก 32 บิตให้เหลือ 26 บิตจะมีความผิดพลาดของข้อมูลน้อยกว่าการลดให้เหลือ 16 บิต

โดยมีการเปรียบเทียบผลลัพธ์ที่ได้จากการคูณจำนวนทศนิยม 2 จำนวนแบบสุ่มทั้งหมด 30 ชุดข้อมูล เพื่อใช้ในการหาค่าความคลาดเคลื่อนมาตรฐานของการพยากรณ์ (MSE: Mean Square Error) โดยมีการเปรียบเทียบระหว่างผลความคลาดเคลื่อนของผลลัพธ์ 26 บิตและ 16 บิตเมื่อเปรียบเทียบกับผลลัพธ์ 32 บิตโดยคำนวณค่าจากสมการที่(4-1)

$$MSE = \frac{\sum_{i=1}^N \sqrt{(X - X_i)^2}}{N} \quad (4-1)$$

โดยที่ N คือจำนวนชุดข้อมูล X คือข้อมูลตั้งต้นและ  $X_i$  คือข้อมูลที่นำมาเปรียบเทียบกับผลที่ได้ระหว่างการหาค่าความคลาดเคลื่อนมาตรฐานของการพยากรณ์คือ

$$MSE_{32compare26} = 0.4573$$

$$MSE_{32compare16} = 1.1142$$

เมื่อพิจารณาจากค่าความคลาดเคลื่อนมาตรฐานแล้วจึงเลือกใช้การลดข้อมูลจากจำนวน 32 บิตเป็น 26 บิต จากเหตุผลข้างต้นและ เมื่อพิจารณารูปแบบของข้อมูลในส่วนนี้จะแสดงเฉพาะขนาดของวงจรที่ใช้ในการออกแบบ เนื่องจากผลลัพธ์จะเหมือนกับในหัวข้อ 4.3.2 โดยเมื่อทำการสังเคราะห์วงจรจะได้พื้นที่ที่ใช้ในการออกแบบวงจรโดยพิจารณาจาก Device Utilization Summary ดังภาพประกอบที่ 4-43 และภาพประกอบที่ 4-44 แสดงถึง Timing Summary เพื่อใช้ตรวจสอบเวลาที่ใช้ในการประมวลผล

```
Device utilization summary:
-----
Selected Device : 3s4000fg900-4

Number of Slices:                7396 out of 27648    26%
Number of Slice Flip Flops:      2334 out of 55296    4%
Number of 4 input LUTs:         13863 out of 55296    25%
  Number used as logic:          13499
  Number used as Shift registers: 364
Number of IOs:                   330
Number of bonded IOBs:           296 out of 633    46%
Number of MULT18X18s:            8 out of 96    8%
Number of GCLKs:                 5 out of 8    62%
```

ภาพประกอบ 4-43 แสดง Device Utilization Summary ของวงจรฟิลเตอร์แบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต

Timing Summary:

-----  
Speed Grade: -4

Minimum period: 553.164ns (Maximum Frequency: 1.808MHz)  
Minimum input arrival time before clock: 9.507ns  
Maximum output required time after clock: 185.418ns  
Maximum combinational path delay: 10.875ns

ภาพประกอบ 4-44 แสดง Timing Summary ของวงจรวจรฟิวด์แบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต

จากผลลัพธ์ที่ได้จากการสังเคราะห์วงจรทั้ง 3 รูปแบบที่ได้กล่าวมาข้างต้นนั้นในภาพประกอบที่ 4-28, 4-29, 4-33, 4-34, 4-43 และ 4-44 นั้น เราจะนำผลลัพธ์ที่ได้มาทำการเปรียบเทียบกับค่าที่ตั้งค่าในโปรแกรม Xilinx โดยให้มีการตั้งค่า จุดมุ่งหมายในการสร้างวงจร (Optimization Goal) คือ พื้นที่ และทำการกำหนดค่ามิให้มีการใช้ทรัพยากรร่วมที่กระทำด้วยโปรแกรม Xilinx โดยอัตโนมัติออกทำให้สามารถแสดงผลที่ได้ทั้ง 3 รูปแบบได้ดังภาพประกอบที่ 4-45, 4-46, 4-47, 4-48, 4-49 และ 4-50 ซึ่งจะแสดงถึง Device Utilization Summary และ Timing Summary ของวงจรมีการใช้ทรัพยากรร่วมแบบลำดับขั้น, วงจรการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม และการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมร่วมกับการลดขนาดบิตตามลำดับ

Device utilization summary:

-----  
Selected Device : 3s4000fg900-4

Number of Slices:	14096	out of	27648	50%
Number of Slice Flip Flops:	5912	out of	55296	10%
Number of 4 input LUTs:	26727	out of	55296	48%
Number of bonded IOBs:	369	out of	633	58%
Number of MULT18X18s:	48	out of	96	50%
Number of GCLKs:	5	out of	8	62%

ภาพประกอบ 4-45 แสดง Device Utilization Summary ของวงจรวจรฟิวด์แบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้นที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx

-----  
Speed Grade: -4

Minimum period: 298.395ns (Maximum Frequency: 3.351MHz)  
Minimum input arrival time before clock: 6.078ns  
Maximum output required time after clock: 160.721ns  
Maximum combinational path delay: 12.396ns

ภาพประกอบ 4-46 แสดง Timing Summary ของวงจรวจรฟิวด์แบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับขั้นที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx



```
Device utilization summary:
-----
```

```
Selected Device : 3s4000fg900-4
```

Number of Slices:	9796	out of	27648	35%
Number of Slice Flip Flops:	5550	out of	55296	10%
Number of 4 input LUTs:	17464	out of	55296	31%
Number of bonded IOBs:	362	out of	633	57%
IOB Flip Flops: 40				
Number of MULT18X18s:	9	out of	96	9%
Number of GCLKs:	5	out of	8	62%

ภาพประกอบ 4-47 แสดง Device Utilization Summary ของวงจรถ่ายทอดแบบที่ มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx

```
Timing Summary:
-----
```

```
Speed Grade: -4
```

```
Minimum period: 1027.069ns (Maximum Frequency: 0.974MHz)
Minimum input arrival time before clock: 11.071ns
Maximum output required time after clock: 338.372ns
Maximum combinational path delay: 17.179ns
```

ภาพประกอบ 4-48 แสดง Timing Summary ของวงจรถ่ายทอดแบบที่ มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx

```
Device utilization summary:
-----
```

```
Selected Device : 3s4000fg900-4
```

Number of Slices:	7342	out of	27648	26%
Number of Slice Flip Flops:	3887	out of	55296	7%
Number of 4 input LUTs:	13044	out of	55296	23%
Number of bonded IOBs:	296	out of	633	46%
IOB Flip Flops: 34				
Number of MULT18X18s:	7	out of	96	7%
Number of GCLKs:	5	out of	8	62%

ภาพประกอบ 4-49 แสดง Device Utilization Summary ของวงจรถ่ายทอดแบบที่ มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิตที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx

## Timing Summary:

-----  
Speed Grade: -4

Minimum period: 714.782ns (Maximum Frequency: 1.399MHz)  
 Minimum input arrival time before clock: 11.334ns  
 Maximum output required time after clock: 236.099ns  
 Maximum combinational path delay: 17.443ns

ภาพประกอบ 4-50 แสดง Timing Summary ของวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิตที่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx

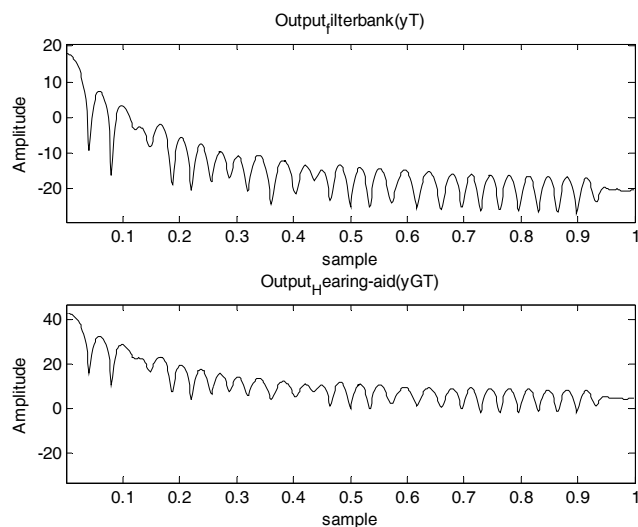
เมื่อพิจารณาจากผลลัพธ์ที่ได้จากการสังเคราะห์วงจรในแต่ละวงจรสามารถที่จะเปรียบเทียบผลการสังเคราะห์วงจรในรูปแบบที่ไม่มีการตั้งค่า Optimization Goal และ resource sharing ใหม่ภายในโปรแกรม Xilinx เทียบกับวงจรที่มีการตั้งค่าเริ่มต้นมาตรฐานได้ดังตารางที่ 4-2 พบว่าผลการสังเคราะห์มีความแตกต่างกันเล็กน้อยในด้านขนาด แต่ในด้านความเร็วจะพบว่าการนำโปรแกรมที่มีการตั้งค่าเริ่มต้นมาตรฐานนั้นจะเร็วกว่ามาก ในที่นี้จึงได้เลือกนำโปรแกรมที่มีการตั้งค่าเริ่มต้นมาตรฐานเพื่อใช้ในการเปรียบเทียบผลลัพธ์

รูปแบบ	Estimated Area A.(Slices/FFs)	Estimated Area B.(Slices/FFs)	Clock A. Period (ns)	Clock B. Period (ns)
1. การใช้ทรัพยากรร่วมแบบลำดับขั้น	13212/5851	14096/5912	299.46	298.395
2. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	9446/3081	9796/5550	603.45	1027.069
3. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต	7396/2334	7342/3887	553.164	714.782

ตารางที่ 4-2 เปรียบเทียบผลการสังเคราะห์วงจรในรูปแบบที่ไม่มีการตั้งค่า Optimization Goal และ Resource Sharing ใหม่ภายในโปรแกรม Xilinx (B) เทียบกับวงจรที่มีการตั้งค่าเริ่มต้นมาตรฐาน (A)

เมื่อพิจารณาพบว่าผลการจำลองการทำงานของวงจรวงจรฟิลเตอร์เบงก์ที่มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและใช้เทคนิคการลดบิตข้อมูล ได้ผลลัพธ์เหมือนกันซึ่งมาจากการแสดงไว้ก่อนหน้านี้ในภาพประกอบที่ 4-33 และ 4-34 ดังนั้นเมื่อนำผลที่ได้ซึ่งเป็นวงจรดิจิทัลฟิลเตอร์เบงก์มาทำการใส่ค่ากำลังขยายในแต่ละช่วงความถี่ และทำการสังเคราะห์วงจรพบว่า ผลที่ได้เมื่อวาดกราฟเปรียบเทียบกับวงจรฟิลเตอร์เบงก์ที่ไม่ได้ทำการคูณค่ากำลังขยาย

จะได้ผลลัพธ์ดังภาพประกอบที่ 4-51



ภาพประกอบ 4-51 กราฟสัญญาณของเอาต์พุตโดยที่

A. วงจรดิจิทัลฟิลเตอร์ B. ต้นแบบเครื่องช่วยฟัง (วงจรดิจิทัลฟิลเตอร์ที่มีการคูณค่ากำลังขยาย)

#### 4.4 ผลการวิจัยการบริโภคพลังงานของวงจรดิจิทัลฟิลเตอร์แบบกึ่งด้วยโปรแกรม XPower

ในการวัดพลังงานจะใช้โปรแกรมที่มีชื่อว่า XPower ที่บรรจุมาในโปรแกรม Xilinx เพื่อทำการวัดค่าพลังงานที่ใช้ไป โดยการใช้สามารถศึกษาได้จากภาคผนวก ก. ในที่นี้จึงได้มีการแสดงผลลัพธ์ที่มีการเปรียบเทียบโดยสามารถแยกการเปรียบเทียบได้เป็น 3 วงจรดังนี้ กล่าวคือ

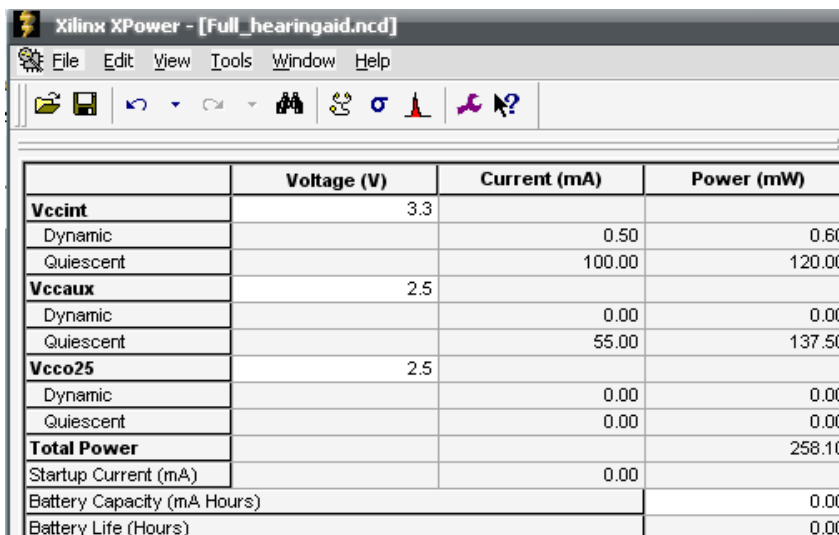
##### 4.4.1 วงจรฟิลเตอร์แบบกึ่งที่ใช้ทรัพยากรร่วมกันแบบลำดับชั้น

มีผลดังภาพประกอบที่ 4-52

Xilinx XPower - [Full_interintra.ncd]			
File Edit View Tools Window Help			
[Toolbar icons]			
	Voltage (V)	Current (mA)	Power (mW)
<b>Vccint</b>	3.3		
Dynamic		3.06	10.09
Quiescent		100.00	330.00
<b>Vccaux</b>	2.5		
Dynamic		0.00	0.00
Quiescent		55.00	137.50
<b>Vcco25</b>	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
<b>Total Power</b>			477.59
Startup Current (mA)		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

ภาพประกอบที่4-52 พลังงานที่ถูกใช้ในวงจรดิจิทัลฟิลเตอร์แบบกึ่งแบบที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบลำดับชั้น

#### 4.4.2 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวม มีผลดังภาพประกอบที่4-53

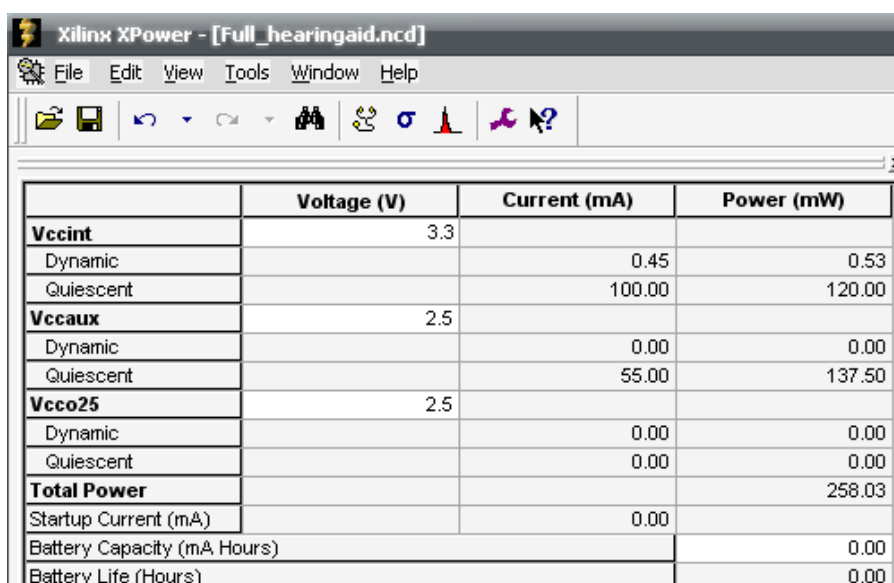


	Voltage (V)	Current (mA)	Power (mW)
<b>Vccint</b>	3.3		
Dynamic		0.50	0.60
Quiescent		100.00	120.00
<b>Vccaux</b>	2.5		
Dynamic		0.00	0.00
Quiescent		55.00	137.50
<b>Vcco25</b>	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
<b>Total Power</b>			258.10
Startup Current (mA)		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

ภาพประกอบที่4-53 พลังงานที่ถูกใช้ในวงจรดิจิทัลฟิลเตอร์แบงก์แบบที่ใช้เทคนิคการใช้  
ทรัพยากรร่วมแบบพิจารณาภาพรวม

#### 4.4.3 วงจรฟิลเตอร์แบงก์ที่ใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวมและการลด จำนวนบิต

##### มีผลดังภาพประกอบที่4-54



	Voltage (V)	Current (mA)	Power (mW)
<b>Vccint</b>	3.3		
Dynamic		0.45	0.53
Quiescent		100.00	120.00
<b>Vccaux</b>	2.5		
Dynamic		0.00	0.00
Quiescent		55.00	137.50
<b>Vcco25</b>	2.5		
Dynamic		0.00	0.00
Quiescent		0.00	0.00
<b>Total Power</b>			258.03
Startup Current (mA)		0.00	
Battery Capacity (mA Hours)			0.00
Battery Life (Hours)			0.00

ภาพประกอบที่4-54 พลังงานที่ถูกใช้ในวงจรดิจิทัลฟิลเตอร์แบงก์แบบที่ใช้เทคนิคการใช้  
ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต

จากการวิเคราะห์ผลพลังงานที่ใช้จะพบว่าพลังงานที่ถูกใช้วงจรฟิลเตอร์แบงก์ที่ใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมนั้นจะมีอัตราการใช้พลังงานต่ำกว่าเนื่องจากเมื่อพิจารณาสมการที่ (2-3) ในบทที่ 2 จะพบว่าพลังงานที่ลดลงเนื่องจากเมื่อพิจารณาจะพบว่า มี 2 ปัจจัยในการบริโภคพลังงาน ได้แก่ค่าการเปลี่ยนระดับของสัญญาณลดลงเนื่องจากการใช้เทคนิคใส่ตัวรีจิสเตอร์คักข้อมูลลงในวงจรและค่าความถี่ลดลงซึ่งพิจารณาจากผลที่ได้จากการสังเคราะห์วงจรระหว่างภาพประกอบที่ 4-28 และภาพประกอบที่ 4-44 จะเห็นว่าความถี่ลดลงจาก 3.34 MHz เป็น 1.8 MHz จึงเป็นผลทำให้พลังงานที่มีการใช้ภายในวงจรลดลง

เมื่อพิจารณาผลลัพธ์ที่ได้จากวงจรฟิลเตอร์แบงก์ทั้ง 3 แบบสามารถเปรียบเทียบให้ง่ายต่อความเข้าใจได้ดังตารางที่ 4-3 เพื่อแสดงการเปรียบเทียบขนาดพื้นที่ของวงจรรวมทั้งพลังงานที่ถูกบริโภคในแต่ละวงจรด้วย

รูปแบบ	Estimated Area (Slices/FFs)	Clock Period (ns)	Latency (ns)	Estimated Power(mW)
1. การใช้ทรัพยากรร่วมแบบลำดับขั้น	13212/5851	299.46	2096.22	477.59
2. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม	9446/3081	603.45	3017.25	258.10
3. การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมและการลดจำนวนบิต	7396/2334	553.164	2765.82	258.03

ตารางที่ 4-3 เปรียบเทียบขนาดพื้นที่ของวงจรรวมทั้งพลังงานที่ถูกบริโภคในแต่ละวงจร

## บทที่ 5

### บทสรุปและข้อเสนอแนะ

#### 5.1 บทสรุป

งานวิจัยนี้เป็นการนำเสนอระเบียบวิธีในการออกแบบวงจรดิจิทัลฟิลเตอร์เบงก์ ซึ่งเป็นส่วนประกอบที่สำคัญของต้นแบบเครื่องช่วยฟัง โดยงานวิจัยนี้จะมีเป้าหมายให้ขนาดวงจรมีขนาดเล็กลงและมีการใช้พลังงานต่ำลง เมื่อเทียบกับวงจรต้นแบบใช้ซึ่งภายในวงจรต้นแบบเครื่องช่วยฟัง โดยใช้วงจรดิจิทัลฟิลเตอร์เบงก์ชนิด 8 ช่วงความถี่ทำหน้าที่เป็นวงจรที่ช่วยในการปรับปรุงเกี่ยวกับคุณภาพเสียงที่ใช้ในงานเกี่ยวกับเครื่องช่วยฟัง ทั้งนี้ที่เลือกใช้วงจรชนิดนี้เนื่องจากเมื่อศึกษาและหาข้อมูลอัลกอริทึมของวงจรพบว่าวงจรดิจิทัลฟิลเตอร์ที่กล่าวไปเป็นวงจรที่มีขนาดใหญ่และหากแนวทางการออกแบบดังกล่าวได้ผลตรงกับที่ต้องการแล้วนั้นในอนาคตอาจพัฒนาการออกแบบวงจรที่มีขนาดใหญ่ด้วยแนวทางที่นำเสนอนี้เพื่อลดขนาดของวงจรและพลังงานให้มีค่าลดลง โดยในงานวิจัยนี้ได้มีการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมมาทำการช่วยในการลดขนาดของวงจร ทั้งนี้ทั้งนั้นได้มีการใช้เทคนิคการใช้ทรัพยากรร่วมอย่างเหมาะสมสำหรับวงจรต้นแบบเครื่องช่วยฟังนั้นประกอบไปด้วย 2 ส่วนใหญ่ๆ คือ ส่วนวงจรข้อมูล (Data-path part) สำหรับการประมวลผลสัญญาณข้อมูล และส่วนวงจรควบคุม (Control part) สำหรับควบคุมจังหวะการทำงานวงจรข้อมูล ซึ่งในส่วนวงจรข้อมูลก็คืออัลกอริทึมการทำงานของวงจรฟิลเตอร์เบงก์ซึ่งภายในวงจรรวมจะมีวงจรฟิลเตอร์ย่อยอีก 8 ตัวโดยที่ภายในฟิลเตอร์แต่ละตัวจะเป็นสมการของการบวก การคูณและการลบค่าของสัญญาณที่ผ่านเข้ามาในวงจรมันเองซึ่งมีสมการทั้งหมด 15 สมการประกอบด้วยตัวบวกจำนวน 38 ตัว ตัวคูณ 27 ตัวและตัวลบ 7 ตัว ซึ่งตรงนี้พบว่าหากสามารถลดปริมาณหน่วยของตัวคำนวณที่กล่าวไปลงได้จะเป็นการลดขนาดของวงจรที่ออกแบบด้วย ดังนั้นจึงนำวงจรรวมมาออกแบบโดยอาศัยหลักการใช้ทรัพยากรร่วมกันแบบพิจารณาภาพรวมเพื่อลดขนาดของวงจรโดยมีการคำนึงถึงเวลาการทำงานที่ช้าลงของวงจรด้วย

ส่วนวงจรควบคุม FSM (Finite State Machine) แบบ Moore machine ถูกใช้สำหรับควบคุมจังหวะการใช้ทรัพยากรร่วมกันของส่วนวงจรข้อมูลให้เป็นไปอย่างถูกต้อง วงจรที่ได้ออกแบบถูกทดสอบบนชิพ FPGA ของบริษัท Xilinx ตระกูล Spartan3f4000fg900-4

ดังนั้นเริ่มจากเซตของฟังก์ชันหรือสมการที่อธิบายพฤติกรรมของวงจรถูกแปลงให้อยู่ในแผนภาพ DFG (Data Flow Graph) ที่มีโครงสร้างเป็นลำดับชั้น จากนั้นพิจารณา DFG เป็นส่วนๆ พยายามหาส่วนที่เหมือนกันเพื่อที่จะนำมาใช้ร่วมกันซึ่งในเทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมนั้นเมื่อทำการเปรียบเทียบเทคนิคดังกล่าวกับเทคนิคการใช้ทรัพยากรร่วมแบบลำดับชั้นพบว่าการใช้เทคนิคการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมอย่างเหมาะสมจะช่วยลดขนาดของวงจรได้ส่วนหนึ่ง ถ้าวรวมกับการใช้เทคนิคการลดบิตข้อมูลจาก 32 บิต ให้เหลือ 26 บิต จะได้ว่าวงจรจะมีขนาดเล็กลงมากและ การบริโภคพลังงานของวงจรก็มีการลดลงเช่นกันเนื่องจากการลดบิตข้อมูลจาก 32 บิต เหลือ 26 บิต โดยพิจารณาภายในวงจรคูณ 1 วงจรพบว่า ตัวคูณ MUL 18x18 นั้น มีค่าลดลงจาก 4 ตัวเหลือเพียง 1 ตัว วิเคราะห์การทำงานได้ว่าในงานวิจัยนี้ใช้แนวคิดดังกล่าวในการออกแบบได้และทำการสังเคราะห์วงจรพบว่าปริมาณการใช้เนื้อที่บน FPGAs ลดลงจากวงจรที่ใช้ทรัพยากรร่วมแบบลำดับชั้นถึง 44 % ใช้เวลาเพิ่มขึ้น 0.67 us และการบริโภคพลังงานลดลง 46%

การตรวจสอบว่าการลดการใช้ทรัพยากรในการออกแบบก็ทำการป้อนค่าสัญญาณซึ่งถูกแปลงเป็นชุดของตัวเลขที่เป็นเลขฐานสองขนาด 32 บิตเป็นอินพุตให้แก่วงจรทุกรูปแบบที่ทำการทดลองบน FPGAs โดยจะเทียบความถูกต้องกับค่าเอาต์พุตที่ได้จากการออกแบบการทำงานของวงจรฟิลเตอร์แบบคัสด้วยโปรแกรม MATLAB พบว่าวงจรที่ออกแบบบน FPGAs ในทุกๆกรณีที่มีการเปรียบเทียบ เมื่อประมวลผลเรียบร้อยแล้วจะให้ค่าเอาต์พุตที่เหมือนกันกับวงจรที่ประมวลผลด้วย MATLAB

## 5.2 ข้อเสนอแนะ

1. ในส่วนของการหาค่ากำลังขยาย(Gain) เพื่อที่จะนำมาคูณนั้น มีกระบวนการคิดที่ซับซ้อน เนื่องจากต้องมีการเทียบเสียงจากผู้ป่วยจริงและต้องมีการลองกับผู้ป่วยจริงดังนั้นควรที่จะมีการทำวิจัยเฉพาะในด้านนี้ โดยเป็นความรู้ทางแพทย์ส่วนมากจึงควรจะมีการวิจัยร่วมกันกับคณะแพทยศาสตร์เพื่อหากระบวนการวิธีในการคำนวณหาค่ากำลังขยายเพื่อที่จะนำมาใช้จริงต่อไป แต่ในงานวิจัยนี้ได้ยกตัวอย่างค่าถ่วงน้ำหนักจากข้อมูลที่มีการศึกษาเท่านั้น จึงทำให้วงจรเครื่องช่วยฟังต้นแบบไม่สมบูรณ์เพราะไม่มีการทดลองกับผู้ป่วยจริง

2. ในส่วนของการบริโภคพลังงานเป็นด้านที่ซับซ้อนมากเนื่องจากต้องมีพื้นฐานเกี่ยวกับด้านพลังงานและค่าต่างๆ โดยในที่นี้ได้มีการอ้างอิงจาก โปรแกรม XPower ที่อยู่ภายในโปรแกรม Xilinx 10.1i เท่านั้น ถ้าต้องการให้มีการบริโภคพลังงานลดลงควรที่จะมีการแยกในส่วนของพลังงานไปทำการวิเคราะห์เฉพาะเพื่อให้ได้ผลลัพธ์ที่ดีที่สุด

3. เนื่องจากโปรแกรม Xilinx 10.1i ไม่รองรับการทำงานแบบใช้สัญญาณนาฬิกาหลายสัญญาณมารวมกันทำให้เกิดข้อจำกัดทางความเร็วในการประมวลผล ควรจะมีงานวิจัยที่ทดลองทำบนโปรแกรมที่รองรับการทำงานหลายสัญญาณนาฬิกาเพื่อที่จะพัฒนาในด้านความเร็วต่อไป

4. เนื่องจากการทำงานของโปรแกรม Xilinx ISE 10.1i นั้น เมื่อมีการใช้งานพร้อมกันกับโปรแกรมอื่น อาจเกิดกรณีที่ตัวโปรแกรมใช้ทรัพยากรของหน่วยความจำในเครื่อง PC มากเกินไปจนทำให้เครื่องเกิดอาการแฮงค์ (Hang) หรือ เกิดรีสตาร์ทเครื่องขึ้นมาใหม่ ส่งผลทำให้ไฟล์ข้อมูลอาจเสียหาย หรือ สูญหายขึ้นได้ ดังนั้นเพื่อความปลอดภัยและความสะดวกในการทำงานของโปรแกรมนี้อ ควรตรวจสอบในส่วน of หน่วยความจำ RAM (Random Access Memory) ของเครื่องคอมพิวเตอร์นั้นก่อน โดยควรให้มีจำนวนสูงไว้ จะได้ไม่เกิดปัญหาดังกล่าวข้างต้น



## บรรณานุกรม

- ชำนาญ ปัญญาใส และ วัชรกร หนูทอง. 2547. ภาษา VHDL สำหรับการออกแบบวงจรดิจิทัล. กรุงเทพฯ: ซีเอ็ดยูเคชั่น.
- วิวัฒน์ บุญสูง. 2551. การออกแบบวงจรดิจิทัลฟิลเตอร์แบงก์โดยหลักการใช้ทรัพยากรดาต้าพาธร่วม. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์.
- วุฒิ วิริยะสม. 2550. การออกแบบหน่วยประมวลผลคณิตศาสตร์ความเร็วสูงสำหรับวงจรกรองปรับตัวบน FPGAs. วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิต สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์.
- สุภาวดี ประคุณหังสิตและสมยศ คุณจักร 2544. ตำรา โสต ศอ นาสิกวิทยา บทที่ 11 เครื่องช่วยฟังและอุปกรณ์ช่วยฟัง: 159-168.
- พรชัย ภาวรงค์ศักดิ์. 2543. การประมวลผลสัญญาณดิจิทัลเบื้องต้น. (หนังสือนี้แจกฟรีผู้อ่านสามารถหาหนังสือนี้อ่านได้ทางอินเทอร์เน็ตที่ <http://www.ee.mut.ac.th/home/pornchai>).
- Erkan Onat, M.A.Sc. Thesis, *DSP algorithms for digital hearing instruments*, University of Windsor, 2000.
- Taewhan Kim, Noritake Yonezawa and Jane W.S. Liu. "A scheduling algorithm for conditional resource sharing". 1994: IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume 13, Issue 4, 425-438.
- Chang E., Gajski D.D., Narayan S., "An optimal clock period selection method based on slack minimization criteria". ACM Transactions on Design Automation of Electronic Systems, Volume 1, Issue 3 (July 1996), 352 – 370.
- S.O. Memik, G. Memik, R. Jafari, E. Kursun., "Global resource sharing for synthesis of control data flow graphs on FPGAs". Proceedings of Design Automation Conference 2003, pp. 604-609, 2-6 June 2003.
- C. Jaschke, R. Laur., "Resource constrained modulo scheduling with global resource sharing". Proceedings of 11th International Symposium on System Synthesis 1998, pp. 60-65, 2-4 Dec. 1998.

- C. Jaschke, F. Beckmann, R. Laur, "Time constrained module scheduling with global resource sharing". Proceedings of Design, Automation and Test in Europe Conference and Exhibition 1999, pp. 210-216, 9-12 March 1999.
- L.S. Nielsen and J. Sparso, "Designing asynchronous circuits for low power : an IFIR filter bank for digital hearing aid". Proceedings of the IEEE vol. 87(2), February 1999, pp: 268-281.
- T. Lunner, J. Hellgren, "A digital filterbank hearing-aid design, implementation and evaluation". International Conference on Acoustics, Speech, and Signal Processing, ICASSP-91, 14-17 April 1991, vol.5, 3661-3664.
- Y. Lin, P.P. Vaidyanathan, "Application of DFT filter banks and cosine modulated filter banks in filtering". IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS94), 5-8 Dec, 1994, 254 - 259.
- R. Bernardini, R. Rinaldo, "Oversampled filter banks from extended perfect reconstruction filter banks". IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 54(7), July 2006, 2625 - 2635.
- B. Dautrich, L. Rabiner, T. Martin, "On the use of filter bank features for isolated word recognition". IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '83. Vol.8, April 1983, 1061 - 1064.
- R. Bregovic, T. Saramaki, "An efficient approach for designing nearly perfect-reconstruction low-delay cosine-modulated filter banks". IEEE International Symposium on Circuits and Systems, ISCAS 2002. Vol.1, 26-29 May 2002, I-825 - I-828.
- T. Uto, M. Okuda, M. Ikehara, S. Takahashi, "Image coding using wavelets based on two-channel linear phase orthogonal IIR filter banks". International Conference on Image Processing, ICIP 99. Vol. 2, 24-28 Oct. 1999, 265 - 268.
- C. Kwen-Siong, G. Bah-Hwee and C. Joseph S. "A Low Energy Asynchronous FFT/IFFT Processor for Hearing Aid Application", IEEE International Symposium on Circuits and Systems 2005, pp.751-754.
- Sunitha SL, Dr V. Udaya shankara "DWT-LMS Speech Enhancement Techniques for Performance Enhancement of Digital Hearing Aid" , ICSCI , Jan 2005.

- A.B. Hamida, M. samet, N. Lakhoua, M. Drira, J. Mouine “Time-Frequency Speech Processing Strategy Based on and Adjustable Algorithms Dedicated to the Cochlear Prostheses”, ICN’99, December 14-16 1998, pp.245-248.
- Cheng-Tao, Jason Chong, Zhiru Zhang, Shih-Chieh Chang “Behavioral Synthesis with Activating Unused Flip-Flops for Reducing Glitch Power in FPGA”, Proceeding of the 2008 Asia and South pacific Design Automation Conference, Seoul Korea.
- Ruzica Jevtic, Carlos Carreras, and Gabriel Caffarena “Switching Activity Model for Power Estimation in FPGA Multipliers”, ARC 2007, LNCS 4419, pp. 201-213, 2007.
- Yongsam Kim, Jack Xin, Yingyong Qi “A study of hearing aid gain functions base on a nonlinear nonlocal feedforward cochlea model”, Hearing Research, Vol.215, pp. 84-96, 2006.
- Shushan Qiao, Yong Hei, Bin Wu, Yumei Zhou “A Technique to Avoid Combination Feedback Loop and Long Critical Path in Resource Sharing”, 7th International Conference on ASIC China, 2007, Page: 1170-1173.
- G. De Micheli, “Synthesis and Optimization of Digital Circuits”, McGraw-Hill, 1994.
- <http://en.wikipedia.org/wiki/Filterbank>
- Xilinx Corp. <http://www.xilinx.com>.

ภาคผนวก ก

บทความทางวิชาการที่นำเสนอใน

**The 32<sup>nd</sup> Electrical Engineering Conference (EECON 32)**

**28-30 ตุลาคม 2551 จ.ปราจีนบุรี**

## การออกแบบดิจิทัลฟิลเตอร์แบงก์โดยใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมบน FPGA A Digital Filter Bank Design using Global Resource Sharing Techniques on an FPGA

เกรียงไกร ทัศนวิภาส<sup>1</sup> ณัฐฐา จินดาเพ็ชร<sup>1</sup> พรชัย พฤกษ์ภัทรานนท์ และวรรณรัช สันติอมรทัต<sup>2</sup>

<sup>1</sup>ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

<sup>2</sup>ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์

ต.คอหงส์ อ.หาดใหญ่ จ.สงขลา 90112 โทรศัพท์ : 0-7428-7045

E-mail: coppermind@hotmail.com nattha.s@psu.ac.th pornchai.p@psu.ac.th wannarat@coe.psu.ac.th

### บทคัดย่อ

บทความนี้นำเสนอการออกแบบดิจิทัลฟิลเตอร์แบงก์โดยใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมบน FPGA โดยที่วงจรฟิลเตอร์แบงก์คือวงจรกรองสัญญาณประเภทหนึ่ง ที่ทำหน้าที่แยกสัญญาณที่ถูกป้อนเข้ามาเพื่อทำการปรับปรุงคุณภาพของสัญญาณแล้วส่งออกมาและเทคนิคการใช้ทรัพยากรร่วมก็ช่วยให้ขนาดวงจรเล็กลงมาก โดยการใช้ทรัพยากรร่วมนั้นได้มีการแบ่งเป็น 2 วิธีคือ การใช้ทรัพยากรร่วมแบบลำดับชั้น และ การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม โดยในงานวิจัยนี้ได้มีการเปรียบเทียบวงจรฟิลเตอร์แบงก์ที่มีการใช้ทรัพยากรร่วมแบบต่างกัน และการออกแบบในการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม ในแบบต่างๆ โดยผลที่ได้พบว่าวงจรฟิลเตอร์แบงก์ที่มีการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมมีขนาด เล็กลงว่าวงจรดิจิทัลฟิลเตอร์แบงก์ที่มีการใช้ทรัพยากรร่วมแบบลำดับชั้นถึง 24 %

**คำสำคัญ:** ดิจิทัลฟิลเตอร์แบงก์, การใช้ทรัพยากรร่วมแบบลำดับชั้น, การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม

### Abstract

This paper presents a design of digital filter bank by using global resource sharing techniques on an FPGA. The filter bank circuit is a digital filter that can decompose the input signal for signal quality improvements. Resource sharing techniques can reduce the circuit area. The resource sharing techniques are divided into two methods: hierarchical resource sharing techniques and global resource sharing techniques. In this research, the digital filter bank has been designed by using these sharing methods in several ways. The experimental results show that the digital filter bank using the global resource sharing technique was about 24% smaller than the digital filter bank using the hierarchical resource sharing technique.

**Keywords:** Digital Filter Bank, Hierarchical Resource sharing technique, Global Resource sharing technique

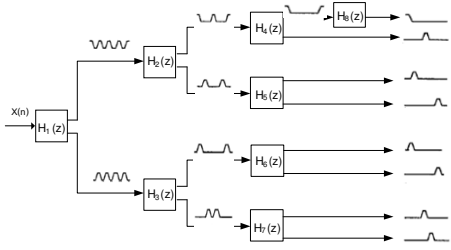
### 1. บทนำ

ในการทำงานแบบฟิลเตอร์เมื่อเปรียบเทียบกระบวนการที่ทำแบบต่างๆจะพบว่า กระบวนการฟิลเตอร์แบงก์ เป็นกระบวนการที่ใช้ค่าตัวคูณ (ค่าสัมประสิทธิ์) น้อยที่สุดเมื่อเทียบกับกระบวนการ แบบอื่นๆเช่น FFT (Fast Fourier Transform) [4] หรือ DCT (Discrete Cosine Transform) ซึ่งข้อดีของการใช้ตัวคูณคือจะส่งผลทำให้วงจรที่มีการออกแบบมีขนาดที่เล็กลง ในที่นี้ได้มีการศึกษาวงจร เครื่องช่วยฟังที่มีเอาต์พุต 8 ช่องสัญญาณพบว่าเมื่อใช้กระบวนการแบบฟิลเตอร์แบงก์ ใช้ตัวคูณเพียง 27 ตัวเมื่อเทียบกับกระบวนการ FFT ซึ่งใช้ตัวคูณมากถึง 554 ตัว ในที่นี้จึงมีการเลือกศึกษาวงจรฟิลเตอร์แบงก์

นอกจากการใช้กระบวนการฟิลเตอร์แบงก์แล้วยังมีการใช้ทรัพยากรร่วมกันควบคู่ เพื่อให้วงจรที่ออกแบบมีขนาดเล็กลงอีกด้วย โดยการใช้ทรัพยากรร่วมต้องคำนึงถึงหลายส่วนเป็นองค์ประกอบเกี่ยวกับการออกแบบการใช้ทรัพยากรร่วมกล่าวคือ ต้องพิจารณาถึง ตัวคูณที่เหมือนกันเพื่อลด Multiplexer ความเหมือนกันของโครงสร้างสมการ สัญญาณนาฬิกา ว่ามีความสอดคล้องหรือเปล่าและเมื่อใช้ทรัพยากรร่วมแล้วส่งผลให้วงจรทำงานช้าลงหรือไม่ ดังนั้นจึงได้มีการเปรียบเทียบการใช้ทรัพยากรร่วมในแบบต่างๆ เพื่อ แสดงว่าแบบใดที่ส่งผลให้วงจรมีความเหมาะสมที่สุด

### 2. วงจรฟิลเตอร์แบงก์

วงจรฟิลเตอร์แบงก์คือวงจรกรองสัญญาณประเภทหนึ่ง ที่ทำหน้าที่แยกสัญญาณที่ถูกป้อนเข้ามาเพื่อทำการปรับปรุงคุณภาพของสัญญาณแล้วส่งออกมา โดยวงจรฟิลเตอร์แบงก์ที่ได้มีการออกแบบ เป็นวงจรฟิลเตอร์แบงก์ที่มีความถี่ 8 ช่วงความถี่ ซึ่งการทำงานจะมีสัญญาณอินพุต(สัญญาณขาเข้า) 1สัญญาณ และภายในวงจรฟิลเตอร์แบงก์ จะมีการแยกสัญญาณออกเป็น 8 ช่วงความถี่ เพื่อนำมาปรับคุณภาพและส่งออกทางเอาต์พุต (สัญญาณขาออก) จะพบว่าสัญญาณขาออกมี 8 สายสัญญาณ และในแต่ละสัญญาณจะมีความถี่ที่แตกต่างกัน [2] โดยแสดงได้ดังรูปที่ 1



รูปที่ 1 วงจรฟิลเตอร์เบงก์แบบเอาท์พุท 8 ช่วงความถี่

ความสัมพันธ์ระหว่างฟิลเตอร์แต่ละตัวภายในฟิลเตอร์เบงก์นั้นสามารถเขียนแสดงได้เป็นสมการได้ดังรูปที่ 2 และ DFGs (Data Flow Graphs) ได้ดังรูปที่ 3 (บล็อกที่มีโครงสร้างเหมือนกันก็จะมีสัญลักษณ์เหมือนกัน)

F-1(z)  $y_1(n) = (x_c + x_{14})h_{1c} + (x_{16} + x_{32})h_{11} + x_{24}h_2$   
 $y_{1c}(n) = x_{24} - y_1(n)$

F-2(z)  $y_2(n) = (x_c + x_{24})h_{2c} + (x_2 + x_{10})h_1 + x_{12}h_2$   
 $y_{2c}(n) = x_{12} - y_2(n)$

F-3(z)  $y_3(n) = (x_c + x_{24})h_{3c} + (x_4 + x_{20})h_{11} + (x_6 + x_{22})h_{12} + (x_{12} + x_{16})h_{2c} + x_{14}h_4$   
 $y_{3c}(n) = x_{14} - y_3(n)$

F-4(z)  $y_4(n) = (x_c + x_{12})h_{4c} + (x_4 + x_8)h_1 + x_4^2h_2$   
 $y_{4c}(n) = x_4 - y_4(n)$

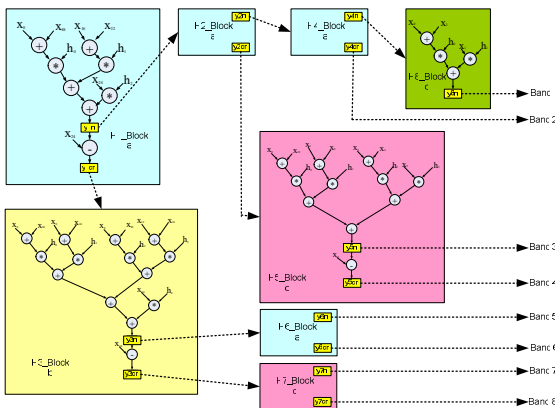
F-5(z)  $y_5(n) = (x_c + x_{10})h_{5c} + (x_2 + x_6)h_1 + (x_4 + x_8)h_2 + x_4^2h_2$   
 $y_{5c}(n) = x_4 - y_5(n)$

F-6(z)  $y_6(n) = (x_c + x_6)h_{6c} + (x_2 + x_4)h_1 + x_2^2h_2$   
 $y_{6c}(n) = x_2 - y_6(n)$

F-7(z)  $y_7(n) = (x_c + x_{30})h_{7c} + (x_4 + x_{20})h_{11} + (x_{12} + x_{16})h_{12} + x_{14}h_4$   
 $y_{7c}(n) = x_{14} - y_7(n)$

F-8(z)  $y_8(n) = (x_c + x_2)h_{8c} + x_4h_4$

รูปที่ 2 ความสัมพันธ์ระหว่างอินพุทและเอาท์พุทของฟิลเตอร์เบงก์



รูปที่ 3 DFGs วงจรฟิลเตอร์เบงก์ 8 ช่วงความถี่

3. วิธีการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวม[3]

การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมจะต่างกับการใช้ทรัพยากรร่วมแบบลำดับขั้น[1] ตรงที่จะมีการมองส่วนที่คล้ายคลึงกันทั้งวงจรว่ามีส่วนใดที่เหมือนกันและสามารถนำมาใช้ร่วมกันได้โดยไม่ต้องเหมือนกันทั้ง block ซึ่งการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมจะส่งผลการใช้ทรัพยากรร่วมแบบนี้มีขนาดเล็กกว่าวิธีการมองแบบลำดับขั้น เพราะการใช้ทรัพยากรร่วมแบบภาพรวมสามารถดึงส่วนที่คล้ายคลึงกัน ในบางส่วนมาทำการใช้ร่วมกันได้มากทำให้ลดปัญหาการที่จะต้องทำการสร้างวงจรขึ้นใหม่ แต่ปัญหาที่พบคือถ้าไม่มีการออกแบบให้มีการใช้ทรัพยากรร่วมที่เหมาะสมจะส่งผลเสียทำให้เกิดการใช้พลังงานมากและอาจจะมีการใช้พื้นที่มากเนื่องจากการสร้าง

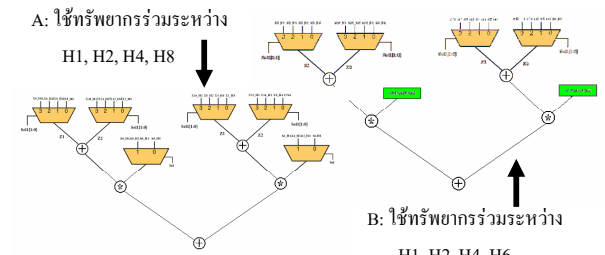
MUX (Multiplexer) มากขึ้น ดังนั้นในการออกแบบการใช้ทรัพยากรร่วมแบบการพิจารณาภาพรวมจะต้องมีการคำนึงถึงปัจจัยในการออกแบบดังนี้

1. พิจารณาค่าสัมประสิทธิ์ กล่าวคือ ถ้าเราจัดรูป สมการที่จะนำมาใช้ร่วมกันโดยให้สมการนั้นมีค่าสัมประสิทธิ์เหมือนกันจะส่งผลทำให้ไม่ต้องการสร้าง MUX เพื่อนำมาสลับค่าสัมประสิทธิ์โดยค่าสัมประสิทธิ์สามารถแสดงได้ดังรูปที่ 4

Filter	H(n)	Decimal Floating point value	Filter	H(n)	Decimal Floating point value
H1(z)	h0 : h (0) = h (42)	-0.05042 4178 42547	H5(z)	h0 : h (0) = h (3 0)	0.0139 4920 552 05
	h1 : h (4) = h (32)	0.2959 5933 4702 99		h1 : h (2) = h (20)	-0.06327 1512 1040 5
	h2 : h (24)	0.5		h2 : h (6) = h (6)	0.3014 1294 8075 61
H2(z)	h0 : h (0) = h (2 4)	-0.05042 4178 42547	H6(z)	h0 : h (0) = h (6)	-0.05042 4178 42547
	h1 : h (2) = h (4)	0.2959 5933 4702 99		h1 : h (2) = h (6)	0.2959 5933 4702 99
	h2 : h (12)	0.5		h2 : h (2)	0.5
H3(z)	h0 : h (0) = h (2 0)	-0.00373 7655 7326 2	H7(z)	h0 : h (0) = h (3 0)	0.0139 4920 552 05
	h1 : h (4) = h (2 4)	0.0285 6859 4870 10		h1 : h (6) = h (2 4)	-0.06327 1512 1040 5
	h2 : h (2) = h (2 0)	-0.07232 1894 7062 9		h2 : h (12) = h (12)	0.3014 1294 8075 61
	h3 : h (12) = h (14)	0.3059 7947 3625 44		h2 : h (15)	0.5
H4(z)	h0 : h (0) = h (3 2)	-0.05042 4178 42547	H8(z)	h0 : h (0) = h (2)	0.2959 5933 4702 99
	h1 : h (4) = h (0)	0.2959 5933 4702 99		h1 : h (1)	0.5
	h2 : h (6)	0.5			

รูปที่ 4 ค่าสัมประสิทธิ์ของวงจรฟิลเตอร์เบงก์ทั้ง 8 บล็อก

2. พิจารณาความเหมือนของโครงสร้างสมการและความซับซ้อนของสมการ กล่าวคือ ถ้าการใช้ทรัพยากรร่วมนั้นมีการใช้ทรัพยากรร่วมมากเกินไปจนเกิดความซับซ้อนจะส่งผลทำให้วงจรนั้นเกิดความผิดพลาดได้สูงและทำให้การคำนวณสัญญาณามีโอกาสผิดพลาดได้สูงเช่นกันสามารถยกตัวอย่างเปรียบเทียบได้ดังรูปที่ 5

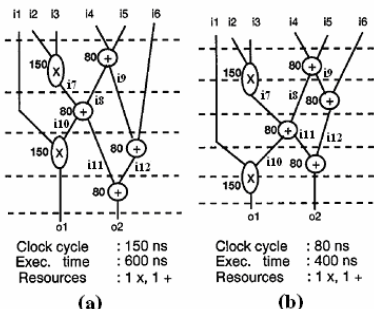


รูปที่ 5 A: โครงสร้างการใช้ทรัพยากรร่วมแบบซับซ้อน

B: โครงสร้างการใช้ทรัพยากรร่วมแบบไม่ซับซ้อน

3. การวิเคราะห์เวลา (Timing Analysis) การวิเคราะห์เวลาสามารถช่วยในการใช้ทรัพยากรร่วมกันได้เพิ่มขึ้นโดยไม่กระทบกับเวลารวมของวงจร DFG ในรูปที่ 6 โดยเมื่อพิจารณาจะพบว่าในวงจร (a) จะพบว่ามีการมีการใช้ทรัพยากรร่วมที่ได้ผลลัพธ์ เช่นเดียวกับในวงจร (b) โดยผลคือสามารถใช้งานประมวลผลเหลือตัวคูณ 1 ตัวและตัวบวก 1 ตัวเช่นเดียวกันแต่เมื่อพิจารณาถึงการแบ่งช่วงเวลาที่เหมาะสมจะพบว่าเมื่อไม่คำนึงช่วงเวลาในวงจร (a) จะใช้เวลามากถึง 600 ns ซึ่งเมื่อเทียบกับวงจร (b) นั้นเมื่อพิจารณาการแบ่งเวลาที่เหมาะสมจะพบว่าสามารถลดเวลาการทำงานได้ถึง 200 ns ซึ่งลดเวลาการทำงานได้เกือบเท่าตัว ดังนั้นหลักการพิจารณาการแบ่งช่วงเวลาให้เหมาะสม มีการพิจารณาโดยในขั้นแรกให้ทำการตรวจสอบว่าในการแบ่งช่วงเวลาแต่ละช่วงมีค่าเท่าใดและนำมาเปรียบเทียบกับตัวดำเนินการว่าในการทำงาน 1 ครั้ง ต้องมีการรอตัวดำเนินการทำงานโดยเฉลี่ยโดยไม่มีการใช้ทรัพยากรร่วมนานมากน้อยเพียงใด ถ้ามีการรอมากจนเกินไปควรที่จะพิจารณาการใช้ทรัพยากรร่วมใหม่ เนื่องจาก เวลา

ในการดำเนินการ ที่ช้าเกินไปก็จะส่งผลทำให้วงจรประมวลผลไม่ทัน ถึงแม้ว่าจะลดพื้นที่ได้มากกว่าก็ตาม



รูปที่ 6 การเปรียบเทียบวงจรที่มีการใช้เทคนิคทรัพยากรร่วม โดยพิจารณาการแบ่งช่วงเวลา[5]

(a) แบบไม่คำนึงการแบ่งช่วงเวลา (b) แบบคำนึงการแบ่งช่วงเวลา

4. การทดลอง

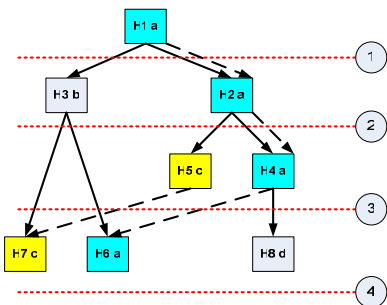
ในการทดลองได้มีการเปรียบเทียบกันทั้งหมด 5 รูปแบบโดยมีการออกแบบวงจรฟิลเตอร์เบงก์ 8 ช่วงความถี่และมีการเปรียบเทียบระหว่างการใช้ทรัพยากรร่วมแบบลำดับขั้นและการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมในแบบต่างๆของวงจรฟิลเตอร์เบงก์ โดยสามารถที่จะแบ่งออกได้เป็น 5 รูปแบบ ดังต่อไปนี้

4.1 การใช้ทรัพยากรร่วมแบบลำดับขั้น [1]

การใช้ทรัพยากรร่วมแบบลำดับขั้นได้มีการออกเป็น 2 ส่วนคือ

4.1.1 การใช้ทรัพยากรร่วมระหว่างบล็อก (Inter-Block Sharing)

คือการใช้ทรัพยากรร่วมกันระหว่างบล็อกในระดับบนสุดของ DFGs โดยที่การใช้ทรัพยากรร่วมกันระหว่างบล็อกนี้จะถูกนำมาพิจารณาเมื่อฟังก์ชันภายในของทั้ง 2 บล็อกนั้นเหมือนกันและนอกจากนี้หากบล็อก 2 บล็อกหรือหลายๆ บล็อกมีโครงสร้างของ DFGs ที่เหมือนกันนั้นก็ยังสามารถทำการออกแบบโดยวิธีนี้ได้เช่นเดียวกันซึ่งจากสมการวงจรดิจิทัลฟิลเตอร์เบงก์ในรูปที่ 3 สามารถเขียน DFG การใช้ทรัพยากรร่วมระหว่างบล็อกได้ดังรูปที่ 7 (โดยบล็อกที่มีการใช้ทรัพยากรร่วมระหว่างบล็อกจะมีสีที่เหมือนกันคือมีการใช้ทรัพยากรร่วมกัน)

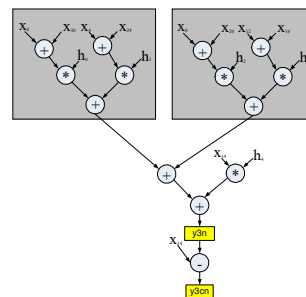


รูปที่ 7 DFG วงจรดิจิทัลฟิลเตอร์เบงก์ที่มีการใช้ทรัพยากรระหว่างบล็อก

4.1.2 การใช้ทรัพยากรร่วมภายในบล็อก (Intra-Block Sharing)

คือมีการใช้ทรัพยากรร่วมกันภายในบล็อกของตัวเองกล่าวคือดูสมการภายในบล็อกตัวเองว่ามีส่วนที่คล้ายคลึงกันเพื่อที่จะนำมาใช้ร่วมกันและบล็อกที่มีการใช้ทรัพยากรร่วมระหว่างบล็อกแล้วจะไม่

สามารถนำมาใช้ทรัพยากรร่วมภายในบล็อกได้อีกดังนั้นในวงจรดิจิทัลฟิลเตอร์เบงก์แบบ 8 ช่วงความถี่จึงมีบล็อกที่มีการใช้ทรัพยากรร่วมภายในบล็อกเพียง 1 บล็อกคือ บล็อก H3 ซึ่งสามารถแสดงได้ดังรูปที่ 8



รูปที่ 8 DFG ของวงจรดิจิทัลฟิลเตอร์เบงก์ที่มีการใช้ทรัพยากรภายในบล็อก

4.2 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1

การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมนั้นต้องคำนึงถึงปัจจัยดังที่ได้กล่าวไว้ในหัวข้อที่ 3 แล้ว ดังนั้นในการใช้ทรัพยากรร่วมแบบที่ 1 สามารถอธิบายได้ชัดเจนโดยรูปที่ 9

$$\begin{aligned}
 H1(z) : y_1(n) &= (x_1 + x_{10})^2 h_1 + (x_{11} + x_{12})^2 h_1 + x_{13}^2 h_1 \\
 y_1(n) &= x_{12} y_1(n) \quad A1 \quad B1 \\
 H2(z) : y_2(n) &= (x_1 + x_{10})^2 h_1 + (x_2 + x_{11})^2 h_1 + x_{12}^2 h_1 \\
 y_2(n) &= x_{12} y_2(n) \quad A2 \\
 H3(z) : y_3(n) &= (x_1 + x_{10})^2 h_1 + (x_2 + x_{11})^2 h_1 + (x_3 + x_{12})^2 h_1 + x_{13}^2 h_1 \\
 y_3(n) &= x_{12} y_3(n) \quad A3 \quad B1 \quad B2 \\
 H4(z) : y_4(n) &= (x_1 + x_{10})^2 h_1 + (x_2 + x_{11})^2 h_1 + x_{12}^2 h_1 \\
 y_4(n) &= x_{12} y_4(n) \quad A4 \\
 H5(z) : y_5(n) &= (x_1 + x_{10})^2 h_1 + (x_2 + x_{11})^2 h_1 + (x_3 + x_{12})^2 h_1 + x_{13}^2 h_1 \\
 y_5(n) &= x_{12} y_5(n) \quad A5 \quad B1 \quad B2 \\
 H6(z) : y_6(n) &= (x_1 + x_{10})^2 h_1 + (x_2 + x_{11})^2 h_1 + x_{12}^2 h_1 \\
 y_6(n) &= x_{12} y_6(n) \quad A6 \\
 H7(z) : y_7(n) &= (x_1 + x_{10})^2 h_1 + (x_2 + x_{11})^2 h_1 + (x_3 + x_{12})^2 h_1 + x_{13}^2 h_1 \\
 y_7(n) &= x_{12} y_7(n) \quad A7 \quad A4 \\
 H8(z) : y_8(n) &= (x_1 + x_{10})^2 h_1 + x_{11}^2 h_1
 \end{aligned}$$

รูปที่ 9 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 1 ของวงจรฟิลเตอร์เบงก์

ข้อดีของการออกแบบในลักษณะนี้คือมีการใช้ทรัพยากรร่วมกันมากที่สุดผลทำให้ขนาดวงจรเล็กลง แต่ถ้าดูผลลัพธ์แล้วจะพบว่าขนาดจะไม่เล็กลงเพราะมีการใช้ MUX มากขึ้นเนื่องมาจากส่วนที่ใช้ร่วมกันมีค่าสัมประสิทธิ์ที่ไม่เหมือนกันและยังพบว่าวงจรซับซ้อนมากเนื่องจากมีการใช้ทรัพยากรร่วมกันมาก

4.3 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2

การใช้ทรัพยากรร่วมแบบที่ 2 สามารถอธิบายได้ชัดเจนโดยรูปที่ 10

$$\begin{aligned}
 H1(z) : y_1(r) &= (x_1 + x_{10})^2 F_1 + (x_{11} + x_{12})^2 F_1 + x_{13}^2 F_1 \\
 y_1(r) &= x_{12} y_1(r) \quad A1 \quad B1 \\
 H2(z) : y_2(r) &= (x_1 + x_{10})^2 F_1 + (x_2 + x_{11})^2 F_1 + x_{12}^2 F_1 \\
 y_2(r) &= x_{12} y_2(r) \quad A2 \quad B1 \\
 H3(z) : y_3(r) &= (x_1 + x_{10})^2 F_1 + (x_2 + x_{11})^2 F_1 + (x_3 + x_{12})^2 F_1 + x_{13}^2 F_1 \\
 y_3(r) &= x_{12} y_3(r) \quad A3 \quad B1 \quad B2 \\
 H4(z) : y_4(r) &= (x_1 + x_{10})^2 F_1 + (x_2 + x_{11})^2 F_1 + x_{12}^2 F_1 \\
 y_4(r) &= x_{12} y_4(r) \quad A4 \\
 H5(z) : y_5(r) &= (x_1 + x_{10})^2 F_1 + (x_2 + x_{11})^2 F_1 + (x_3 + x_{12})^2 F_1 + x_{13}^2 F_1 \\
 y_5(r) &= x_{12} y_5(r) \quad A5 \quad B1 \quad B2 \\
 H6(z) : y_6(r) &= (x_1 + x_{10})^2 F_1 + (x_2 + x_{11})^2 F_1 + x_{12}^2 F_1 \\
 y_6(r) &= x_{12} y_6(r) \quad A6 \\
 H7(z) : y_7(r) &= (x_1 + x_{10})^2 F_1 + (x_2 + x_{11})^2 F_1 + (x_3 + x_{12})^2 F_1 + x_{13}^2 F_1 \\
 y_7(r) &= x_{12} y_7(r) \quad A7 \quad A4 \\
 H8(z) : y_8(r) &= (x_1 + x_{10})^2 F_1 + x_{11}^2 F_1
 \end{aligned}$$

รูปที่ 10 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 2 ของวงจรฟิลเตอร์เบงก์

ข้อดีของการออกแบบลักษณะนี้คือได้มีการพิจารณาค่าสัมประสิทธิ์ก่อนที่จะมีการใช้ทรัพยากรร่วมทำให้การใช้ทรัพยากรร่วมวิธี

นี้คือปัญหาขนาดที่เพิ่มขึ้นเนื่องจากไม่ต้องสร้าง MUX เพื่อทำการสลับค่าสัมประสิทธิ์ ข้อเสียของการใช้ทรัพยากรร่วมลักษณะนี้คือจะมีส่วนของวงจรที่มีชื่อว่า No\_Share และมีการสร้างรีจิสเตอร์ในการเก็บค่าทำให้มีขนาดเพิ่มมากขึ้น ในรูปที่ 10 ที่ไม่สามารถแชร์กับวงจรอื่นได้ส่งผลทำให้ต้องมีการสร้างวงจรในส่วนนั้นเพิ่มจึงได้มีการออกแบบการใช้ทรัพยากรร่วมแบบที่เหมาะสมกว่าในรูปแบบที่ 3

4.4 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3

การใช้ทรัพยากรร่วมแบบที่ 3 สามารถอธิบายได้โดยรูปที่ 11

$$\begin{aligned}
 H1(z) \quad y_1(z) &= (x_1+x_{1d})^2 F_1 + (x_{1c}+x_{3c})^2 F_1 + x_{2c}^2 F_2 \\
 y_1(z) &= x_{2c} y_1(z) \\
 H2(z) \quad y_2(z) &= (x_1+x_{2c})^2 F_1 + (x_1+x_{1c})^2 F_1 + x_{2c}^2 F_2 \\
 y_2(z) &= x_{1c} y_2(z) \\
 H3(z) \quad y_3(z) &= (x_1+x_{2c})^2 F_1 + (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_2 + (x_{1c}+x_{1d})^2 F_2 + x_{1d}^2 F_2 \\
 y_3(z) &= x_{1c} y_3(z) \\
 H4(z) \quad y_4(z) &= (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_1 + x_{2c}^2 F_2 \\
 y_4(z) &= x_{1c} y_4(z) \\
 H5(z) \quad y_5(z) &= (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_1 + (x_1+x_{2c})^2 F_2 + x_{2c}^2 F_2 \\
 y_5(z) &= x_{1c} y_5(z) \\
 H6(z) \quad y_6(z) &= (x_1+x_{2c})^2 F_1 + (x_1+x_{1c})^2 F_1 + x_{2c}^2 F_2 \\
 y_6(z) &= x_{1c} y_6(z) \\
 H7(z) \quad y_7(z) &= (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_1 + (x_{1c}+x_{1d})^2 F_2 + x_{1d}^2 F_2 \\
 y_7(z) &= x_{1c} y_7(z) \\
 H8(z) \quad y_8(z) &= (x_1+x_{2c})^2 F_1 + x_{1c}^2 F_1
 \end{aligned}$$

รูปที่ 11 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 3 ของวงจรฟิลเตอร์แบงก์

การใช้ทรัพยากรร่วมรูปแบบที่ 3 นี้เป็นรูปแบบที่มีการแก้ข้อเสียของรูปแบบที่ 2 โดยเมื่อพิจารณาวงจรจะพบว่าได้มีการจัดโครงสร้างและหาส่วนที่เหมือนกันได้ทั้งวงจรกล่าวคือสามารถใช้ทรัพยากรร่วมกันในส่วนที่เหมือนกันทั้งวงจร โดยไม่มีส่วนที่ต้องทำการสร้างใหม่ทำให้ประหยัดพื้นที่ลงแต่เมื่อพิจารณาถึงความเร็วทำให้พบว่าในรูปแบบที่ 3 นี้จะมีการทำงานช้ากว่ารูปแบบที่ 2 อยู่ 1 รอบสัญญาณนาฬิกาและยังไม่มีการแก้ปัญหาในส่วนของรีจิสเตอร์ที่เพิ่มขึ้นในการเก็บค่าออกจากนั้นเมื่อพิจารณาแล้วจึงพบว่ามีการเพิ่มขึ้นอีกในส่วนวงจร B2 ทำให้รูปแบบที่ 3 นี้ไม่เหมาะสมทั้งในด้าน ขนาด ความเร็ว ความซับซ้อน จึงได้มีการออกแบบรูปแบบใหม่เพื่อให้เหมาะสมที่สุดจึงได้ดังรูปแบบที่ 4

4.5 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4

ในการใช้ทรัพยากรร่วมรูปแบบนี้จะมีความคล้ายคลึงกับการออกแบบรูปแบบที่ 2 แต่จะพบว่ารูปแบบนี้มีข้อดีในด้านเส้นทางวิกฤตจะมีเส้นทางที่สั้นกว่าส่งผลทำให้วงจรทำงานเร็วกว่าซึ่งเมื่อเปรียบเทียบความเหมาะสมทั้งในด้านขนาดของวงจรและความเร็วในการประมวลผลจึงได้มีการเลือกการออกแบบรูปแบบนี้

$$\begin{aligned}
 H1(z) \quad y_1(z) &= (x_1+x_{1d})^2 F_1 + (x_{1c}+x_{3c})^2 F_1 + x_{2c}^2 F_2 \\
 y_1(z) &= x_{2c} y_1(z) \\
 H2(z) \quad y_2(z) &= (x_1+x_{2c})^2 F_1 + (x_1+x_{1c})^2 F_1 + x_{2c}^2 F_2 \\
 y_2(z) &= x_{1c} y_2(z) \\
 H3(z) \quad y_3(z) &= (x_1+x_{2c})^2 F_1 + (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_2 + (x_{1c}+x_{1d})^2 F_2 + x_{1d}^2 F_2 \\
 y_3(z) &= x_{1c} y_3(z) \\
 H4(z) \quad y_4(z) &= (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_1 + x_{2c}^2 F_2 \\
 y_4(z) &= x_{1c} y_4(z) \\
 H5(z) \quad y_5(z) &= (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_1 + (x_1+x_{2c})^2 F_2 + x_{2c}^2 F_2 \\
 y_5(z) &= x_{1c} y_5(z) \\
 H6(z) \quad y_6(z) &= (x_1+x_{2c})^2 F_1 + (x_1+x_{1c})^2 F_1 + x_{2c}^2 F_2 \\
 y_6(z) &= x_{1c} y_6(z) \\
 H7(z) \quad y_7(z) &= (x_1+x_{1c})^2 F_1 + (x_1+x_{2c})^2 F_1 + (x_{1c}+x_{1d})^2 F_2 + x_{1d}^2 F_2 \\
 y_7(z) &= x_{1c} y_7(z) \\
 H8(z) \quad y_8(z) &= (x_1+x_{2c})^2 F_1 + x_{1c}^2 F_1
 \end{aligned}$$

รูปที่ 12 การใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมรูปแบบที่ 4 ของวงจรฟิลเตอร์แบงก์

สามารถเปรียบเทียบคุณสมบัติเพื่อแสดงว่ารูปแบบที่ 4.5 เหมาะสมที่สุดได้ดังตารางที่ 1

รูปแบบ	Estimated Area (Slices/FFs)	Inter - Connects	Clock Period (ns)	Latency (ns)
1. รูปแบบที่ 4.1 *	11171/3812	174	174.03	870.15
2. รูปแบบที่ 4.2 *	9436/3282	216	170.24	1021.44
3. รูปแบบที่ 4.3 *	9248/2919	179	157.93	789.65
4. รูปแบบที่ 4.4 *	8272/2818	189	160.87	1126.09
5. รูปแบบที่ 4.5 *	8486/2789	179	156.58	782.9

\* = สังเคราะห์วงจรด้วยโปรแกรม Xilinx 10.1

ตาราง 1 เปรียบเทียบคุณสมบัติในการใช้ทรัพยากรร่วมรูปแบบต่างๆ

5. สรุป

ในงานวิจัยนี้ได้มีการเปรียบเทียบการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมใน 4 รูปแบบ กับการใช้ทรัพยากรร่วมแบบลำดับชั้นจากการทดลองพบว่าการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมสามารถที่จะลดขนาดของวงจรลดได้มากกว่าการใช้ทรัพยากรร่วมแบบลำดับชั้น ทั้งนี้ทั้งนั้นก็ต้องออกแบบการใช้ทรัพยากรร่วมให้เหมาะสมเมื่อทำการเปรียบเทียบขนาดของวงจรฟิลเตอร์แบงก์ 8 ช่วงความถี่ พบว่าเมื่อมีการใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมจะสร้างวงจรที่มีขนาดเล็กลงกว่าการใช้ทรัพยากรร่วมแบบลำดับชั้นถึง 24 % และมีการทำงานที่เร็วขึ้น

เอกสารอ้างอิง

- [1] วิวัฒน์ บุญสูง, การออกแบบวงจรดิจิทัลฟิลเตอร์แบงก์โดยหลักการใช้ทรัพยากรค่าตัวพาท่วมกัน, วิทยานิพนธ์วิศวกรรมศาสตรมหาบัณฑิตสาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์มหาวิทยาลัยสงขลานครินทร์, 2551.
- [2] T. Lunner, J. Hellgren, "A digital filterbank hearing aid-design, implementation and evaluation", International Conference on Acoustics, Speech, and Signal Processing, ICASSP-91, 14-17 April 1991 Page(s), 3661- 3664 vol.5.
- [3] Seda O. Memik, G. Memik, R. Jafari, E. Kursun, "Global Resource Sharing for Synthesis of Control Data Flow Graphs on FPGAs", Annual ACM IEEE Design Automation Conference, 2003, Page(s): 604 – 609.
- [4] Kwen-Siong Chong, Bah-Hwee Gwee and Joseph S. Chang, "A Low Energy Asynchronous FFT/IFFT Processor for Hearing Aid Application", IEEE International Symposium on Circuits and Systems 2005, pp.751-754.
- [5] EN-Shou Chang, Daniel D. Gajski and Sanjiv Narayan, "An Optimal Clock Period Selection Method Based on Slack Minimization Criteria", ACM Transactions on Automation of Electronic Systems Vol. 1 No. 3, July 1996, Pages 352-370.



**ภาคผนวก ข**

**คู่มือการใช้โปรแกรม XPower**

**XPower Tutorial FPGA Design**

**XPower(v1.3)**

**15 กรกฎาคม 2545**

# คู่มือการใช้โปรแกรม XPower

ในการใช้ XPower เพื่อประมาณการใช้พลังงาน เราต้องอิมพลีเมนต์ของเราเข้ากับอุปกรณ์ FPGA ก่อนซึ่งสามารถทำได้โดยรัน "Implement design" process บน Xilinx ISE tools. ในหัวข้อนี้จะกล่าวถึงเรื่องต่างๆดังนี้

- “Getting Started”
- “Opening the Design in XPower”
- “Setting Voltage and Temperature Values in the Summary View”
- “Setting Activity Rates”
- “Loading Simulation Data”
- “Changing from Types to Physical View”
- “Generating the Power Report”
- “Saving the Power Report”
- “Saving Your Design Settings”
- “Closing the Design”
- “Opening the Design with a Saved Settings File”

โดยสามารถอธิบายหัวข้อต่างๆ ที่กล่าวมาข้างต้นได้ดังต่อไปนี้

## Getting Started

ในบทนี้เป็นการอธิบายอย่างคร่าวๆว่าโปรแกรม XPower คำนวณพลังงานได้อย่างไร

การคำนวณพลังงานโดยโปรแกรม XPower

โปรแกรม Xpower คำนวณพลังงานโดยอาศัยการเฝ้าดูการใช้พลังงานในวงจร CMOS ระหว่างการเปลี่ยนแปลงระดับของสัญญาณซึ่งความแม่นยำของข้อมูลจากการเปลี่ยนแปลงระดับของสัญญาณจะมีผลต่อความแม่นยำของการประมาณการใช้พลังงานเช่นกัน

XPower จะคำนวณพลังงานออกมาเป็นผลรวมของพลังงานที่ใช้โดยส่วนต่างๆในการออกแบบซึ่งสามารถแสดงได้โดยสมการต่อไปนี้

$$P = C * V^2 * E * F / 1000$$

โดยที่

P= พลังงานหน่วยเป็นมิลลิวัตต์ (mW)

C= ตัวเก็บประจุหน่วยเป็นฟารัด (F)

V= แรงดันประจุไฟฟ้าหน่วยเป็นโวลต์ (V)

E= การเปลี่ยนแปลงค่าระดับของสัญญาณต่อรอบสัญญาณนาฬิกา(Activity Rate)

F= ความถี่สัญญาณหน่วยเป็นเฮิรตซ์(Hz)

### Activity Rates

คืออัตราซึ่งสายสัญญาณหรือส่วนของตรรกะทำการสลับสับเปลี่ยนสำหรับการคำนวณ และแสดงพลังงานพลวัต (Dynamic power) จะอยู่ในรูปของความถี่ activity rate อาจจะสัมพันธ์กับสัญญาณนาฬิกาเนื่องจากสายสัญญาณหรือส่วนของตรรกะอาจมีการสลับสับเปลี่ยนเป็นเปอร์เซ็นต์ของความถี่ของสัญญาณนาฬิกาโดยความหมายของไฟล์ชนิดต่างๆมีดังนี้

NCD: FPGA design files ซึ่งแสดง design topology และการใช้ physical resource. ไฟล์นี้ จะถูกสร้างระหว่างการ implementation.

CXT: CPLD design files ซึ่งแสดง design topology และการใช้ physical resource. ไฟล์นี้ จะถูกสร้างระหว่างการทำ FIT

PCF: design files ซึ่งแสดงความถี่ clock และ voltage. ไฟล์นี้ จะถูกสร้างระหว่างการ Implementation

VCD: Design file ซึ่งแสดงรายละเอียดของ Activity rate สำหรับสายสัญญาณทั้งหมดไฟล์นี้จะถูกสร้างระหว่างการทำ Timing simulation โดยใช้ M T I

XML: ไฟล์การกำหนดค่าต่างๆของผู้ใช้โปรแกรม Xpower และโปรแกรม XPower จะสร้างไฟล์นี้ขึ้นเมื่อเราทำการบันทึกข้อมูลการตั้งค่าพลังงาน

PWR: ไฟล์รายงานการทำงานของโปรแกรม XPower โดยที่ XPower จะสร้างไฟล์นี้ขึ้นเมื่อเราสร้าง power report ขึ้น

### Opening the Design in XPower

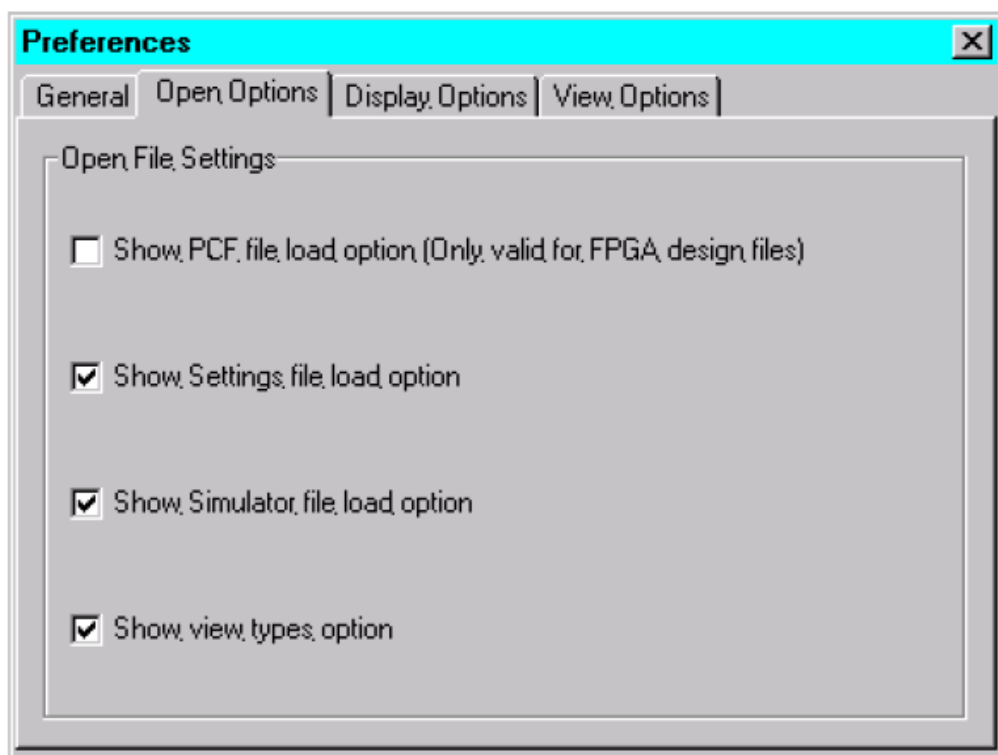
สำหรับการออกแบบที่เราจะใช้ในไฟล์นี้ชื่อ counter.ncd มี 2 วิธีในการโหลดการออกแบบเข้าสู่โปรแกรม XPower วิธีแรกคือรัน "Analyze Power (XPower)" process จาก ISE และ วิธีที่สองคือใช้โปรแกรม Xpower เป็น stand-alone operation และทำการ โหลดไฟล์ที่จำเป็นด้วยตัวเอง

ในวิธีที่ใช้ ISE ให้ทำตามขั้นตอนต่อไปนี้:

1. เปิด ISE Project Navigator โดยเลือกที่ Start->Program->Xilinx ISE 4  
->Project Navigator.
2. เลือกที่ Open Project... ใน Project Navigator. Browse ไปยังตำแหน่งที่เก็บ project และทำการเลือกไฟล์ counter.npl แล้วกดเปิด
3. เลือก Project source ของเรา (counter.vhd) ภายในหน้าต่าง Source ปัจจุบันซึ่งทำการขยาย Implement Design tree โดยคลิกที่เครื่องหมาย "+" จากนั้นคลิกเครื่องหมาย "+" ที่ติดกับ Place & Route เราจะเห็น process ที่ชื่อ Analyze Power (XPower). ดับเบิลคลิกที่ Process นี้ (คือคลิกขวาแล้วเลือก run) จะเป็นการดำเนินการ Process ทั้งหมดที่จำเป็นต้องใช้เพื่อสร้างไฟล์ .NCD และเมื่อไฟล์ .NCD ถูกสร้างขึ้นแล้วในส่วน Project Navigator จะเรียก XPower และโหลด NCD โดยอัตโนมัติโดยจะมีการโหลดไฟล์ PCF file ด้วยถ้าหากมี จากนั้นเราต้องโหลดไฟล์ Simulation (ในที่นี้คือไฟล์ counter.vcd) โดยการเลือกที่ Open Simulation File... จากเมนูของ XPower และ browse ไปยังไฟล์ .VCD ที่ถูกต้อง

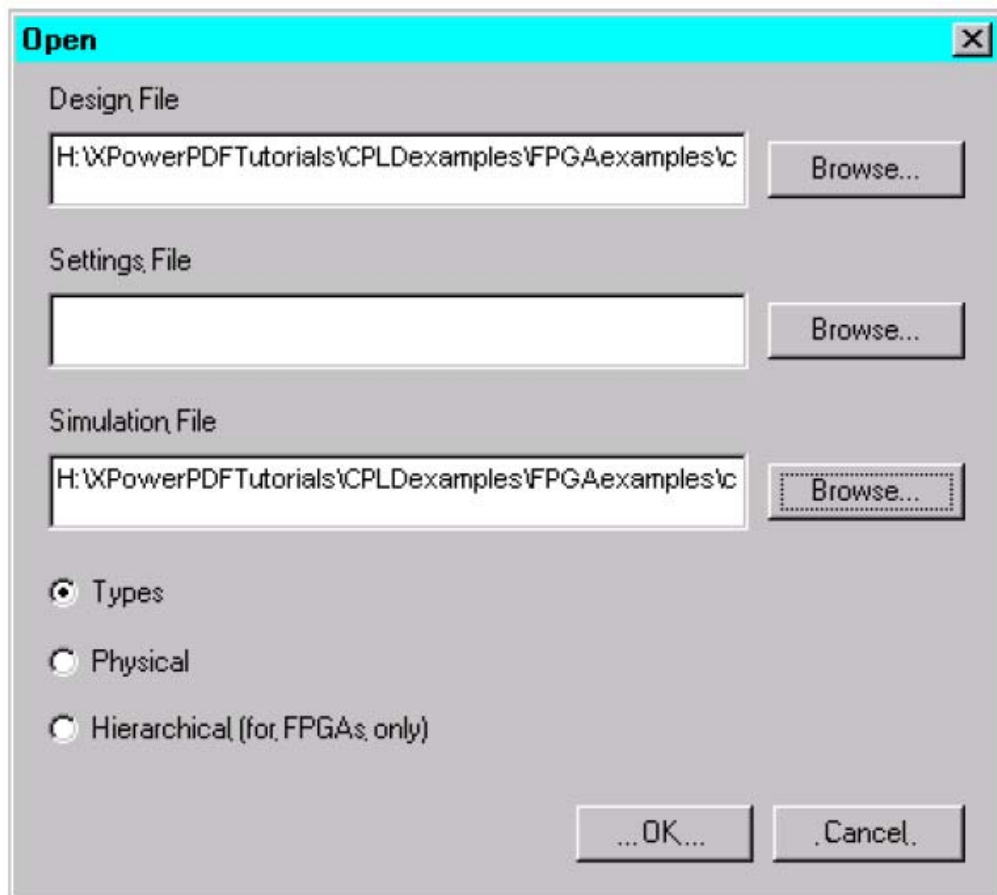
ส่วนกรณีที่ใช้ XPower เป็น stand-alone tool โดยที่ไฟล์ .CXT จะมียู่ใน design folder เราสามารถเข้าถึงโปรแกรม XPower ได้ทั้งทาง Command Prompt หรือโดยคลิกไฟล์ exe ที่อยู่ใน ISE directory เมื่อเราเปิดโปรแกรม Xpower ขึ้นมาแล้ว ให้ทำตามต่อไปนี้

1. เลือกคำสั่ง Preference จากเมนู Edit และคลิกที่ส่วน Open Options เลือกตัวเลือกที่มีทั้งหมด (Physical Constraints File Option, Settings file load option, Simulator file load option and view types option) คลิก OK.



2. เลือกคำสั่ง Open File จากเมนู dialog box จะปรากฏขึ้นให้คลิกปุ่ม Browse ที่อยู่ติดกับ "Design File" data entry line. หน้าต่าง File จะปรากฏขึ้นทำการดับเบิลคลิกบนแฟ้มข้อมูลที่เราบันทึก design file ไว้และเลือกไฟล์ counter.ncd. คลิก Open. ชื่อเต็มของ Path จะปรากฏบน "Design File" data entry line ไฟล์ PCF ที่ชื่อ counter.pcf จะปรากฏขึ้นในส่วน Physical Constraints File โดยอัตโนมัติ

3. ไฟล์ .VCD (Value Change Dump) (counters.vcd) อาจจะปรากฏหรือไม่ปรากฏบนแถบ Simulation File data entry ซึ่งถ้าหากมันไม่ปรากฏ คลิกปุ่ม Browse และเลือก counter.vcd จากแฟ้มข้อมูลที่เราบันทึกไฟล์ design ที่เราดาวน์โหลดมา สำหรับจุดประสงค์ของ tutorial นี้ไฟล์ .VCD นี้ได้ถูกสร้างขึ้นสำหรับ design นี้แล้ว เมื่อเราโหลดไฟล์นี้ด้วยไฟล์ .CXT.ค่ามาตรฐานของ activity rates ที่แสดงไว้ในโปรแกรม XPower จะเป็นผลลัพธ์ของ simulation data

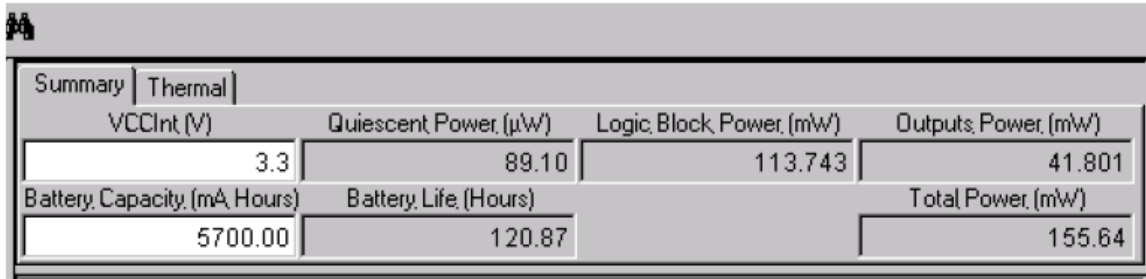


4. ใน ตัวเลือก "View Types" เราจะสามารถเลือกได้ระหว่างจะดู Types view (element ที่ถูกจัดกลุ่มไว้ด้วยกันตาม type), Physical view (element ที่ถูกจัดกลุ่มโดย CLBs) และ Hierarchical view (ส่วนประกอบจะถูกจัดกลุ่มไว้ตาม Hierarchy) โดยที่ Types จะเป็นตัวเลือก view โดยดีฟอลต์ในโปรแกรม Xpower เลือก Types เมื่อเปิด design ขึ้นมาแล้วและเมื่อ Frequency rate ถูกตั้งค่าเราสามารถเลือกดู option อื่นๆได้เพื่อวิเคราะห์ activity rate

5. คลิก OK. design จะเปิดในแบบ Types view. เป็นอันพร้อมสำหรับการเริ่มต้น

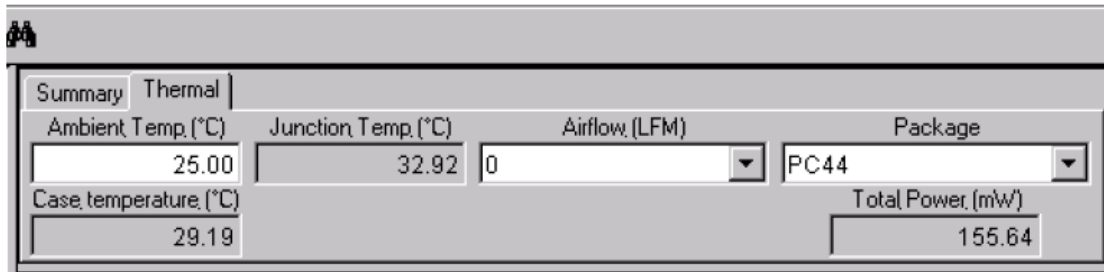
## Setting Voltage and Temperature Values in the Summary View

Summary view ที่อยู่ด้านบนของหน้าจอจะมีอยู่ 2 ส่วนคือ Summary tab และ Thermal tab ในส่วน Summary tab จะแสดง  $V_{CCInt}$ , ค่าพลังงานที่ใช้ที่ต่างกัน, ตัวเก็บประจุ แบตเตอรี่, อายุแบตเตอรี่, และพลังงานทั้งหมด



VCCInt (V)	Quiescent Power (µW)	Logic Block Power (mW)	Outputs Power (mW)
3.3	89.10	113.743	41.801
Battery Capacity (mA Hours)	Battery Life (Hours)	Total Power (mW)	
5700.00	120.87	155.64	

เมื่อคลิกที่ส่วน Thermal ใน summary view จะเป็นการแสดง ambient temperature, Junction temperature, Airflow, ชนิดของ package ที่เราเลือกไว้ Case temperature และ พลังงานทั้งหมดที่ใช้โดย design นั้นๆ



Ambient Temp. (°C)	Junction Temp. (°C)	Airflow (LFM)	Package
25.00	32.92	0	PC44
Case temperature (°C)	Total Power (mW)		
29.19	155.64		

1. ใน Summary tab คลิกภายใน data entry line ได้  $V_{CCInt}$  ใน summary view ที่อยู่ด้านบนของหน้าจอ โดยที่ค่าแรงดันมาตรฐานสำหรับการออกแบบคือ 2.5V และช่วงค่าแรงดันที่ยอมรับได้คือ 2.38 - 2.63 V. ถ้าหากเราเปลี่ยนค่าโวลต์อยู่นอกเหนือช่วงนี้ ค่าเตือนต่อไปนี้จะปรากฏขึ้นใน History Bar ด้านล่างของหน้าจอ

**WARNING: Power - VccInt <3.0> not in recommended range [2.38...2.73] V.**

Highlight 2.5, เปลี่ยนค่าเป็น 2.38 และกดปุ่ม return เพื่อทำการบันทึกค่าการใช้พลังงานที่ลดลงจากนั้นเพิ่มค่า  $V_{CCInt}$  เป็น 2.63 และบันทึกค่าพลังงานรวมที่เพิ่มขึ้น

2. ทำเช่นเดียวกันกับค่า Ambient Temperature. อุณหภูมิตั้งต้นของโปรแกรม Xpower เป็น 25 องศาเซลเซียสเสมอแต่ช่วงของอุณหภูมิที่ยอมรับได้คือ -40 องศาถึง 125 องศาเซลเซียส ลองเปลี่ยนค่าให้อยู่ในช่วงนี้และสังเกตดูว่ามีผลกระทบต่อ Junction Temperature

3. Thermal Tab จะให้เราสามารถเปลี่ยน airflow และชนิดของ package ได้โดยที่การเพิ่ม airflow จะลด case และ junction temperature ซึ่งการเปลี่ยน package type จะกระทบต่ออุณหภูมิเช่นกัน. Package type ISE ที่ถูกเลือกใน design นี้คือ cs44 เมื่อเปลี่ยนเป็น fg256 จะเป็นการลดทั้ง case temperature และ junction temperature และถ้าเปลี่ยนเป็น cs240 จะเป็นการเพิ่มในอัตราที่สูงขึ้นไปอีก

4. จากนั้น กลับไปยัง Summary tab, การเพิ่มความจุแบตเตอรี่จะเป็นการเพิ่มอายุแบตเตอรี่ ขณะที่การลดความจุก็จะลดอายุแบตเตอรี่เช่นกัน

### Setting Activity Rates

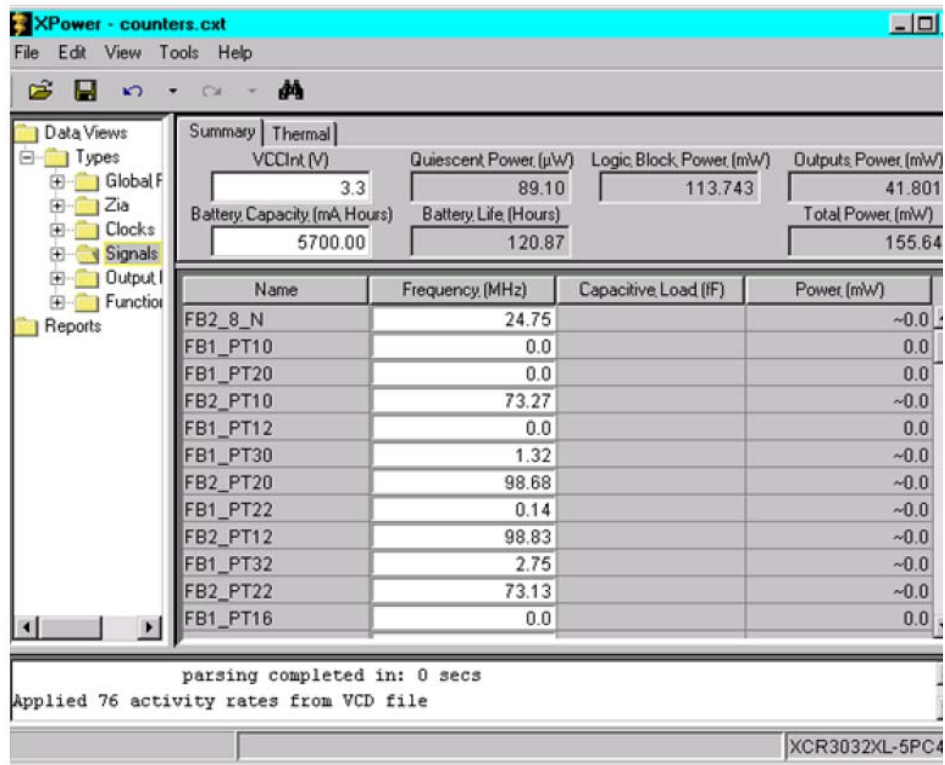
โปรแกรม Xpower จะให้เราสามารถเปลี่ยน activity rates information บน Clock, Signal, Logic และ Output ที่ต่างกันได้ทำให้เราสามารถเปลี่ยนความถี่เพื่อตรวจสอบผลกระทบต่อการใช้พลังงานโดยโปรแกรม Xilinx ได้แนะนำว่าควรจะต้องตั้งค่าตามลำดับที่ถูกต้องเพื่อให้ได้ผลที่ดีที่สุดดังนี้

- Set or verify voltage and ambient temperature
- Set or verify clock frequencies
- Set or verify input frequencies
- Set or verify output loading
- Set global default activity to estimated value
- Set any specific or signal groups to estimated values

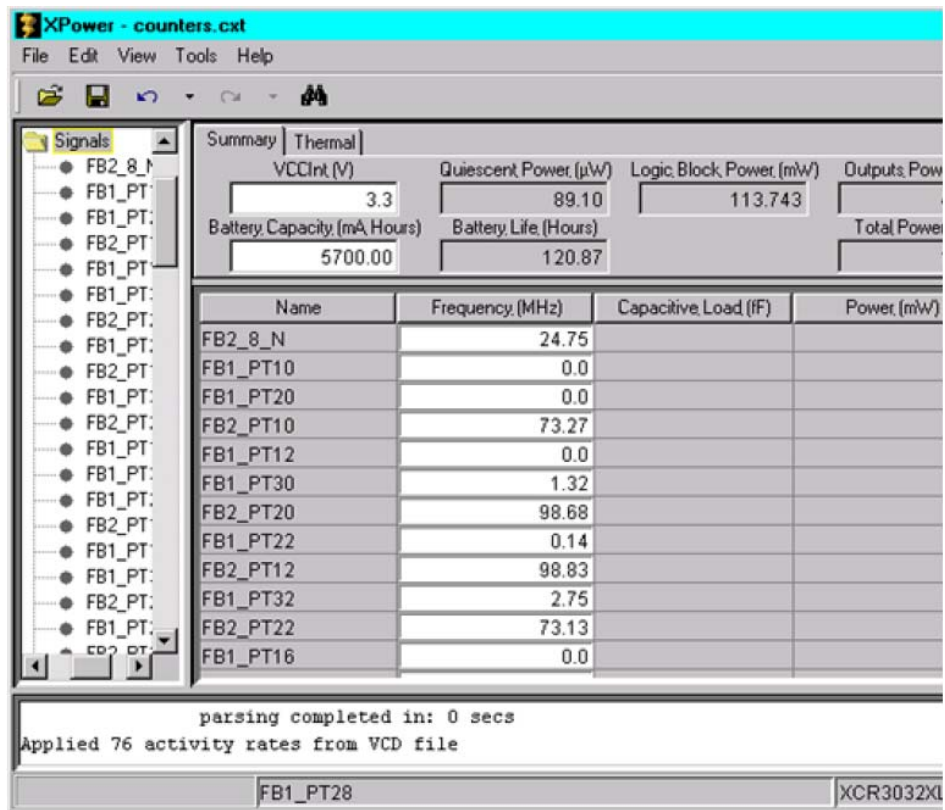
ในการจะเข้าถึงส่วนประกอบ เหล่านี้ ทำได้โดย

1. ใน Explorer window ดับเบิ้ลคลิกเพิ่มข้อมูล Data view จากนั้นดับเบิ้ลคลิกที่เพิ่มข้อมูล Types เราจะเห็น 4 เพิ่มข้อมูลภายใน Types ได้แก่: Signals, Clocks, Logic และ Output





2. คลิกเครื่องหมาย "+" เพื่อขยายจากนั้นคลิกบน โพลเดอร์เพื่อแสดงชื่อของ Signal ใน main window



3. สังเกตดูค่าพลังงานที่แสดงไว้ใน summary view ด้านบนของหน้าจอจะมีการแสดงค่าการใช้พลังงานของ Quiescent power, Signal Power, Inputs Power, IOs Power, Logic Blocks Power, Clocks Power, and Outputs Power ควบคู่ไปกับค่ามาตรฐานของการใช้พลังงานรวมด้วย

4. ค่าเหล่านี้บางค่าจะถูกตั้งค่าโดย Simulation file ของเรา แต่ก็ควรตั้งค่า activity rate ทั้งหมดด้วยตัวเองหรือทำการ set global default activity rate

5. ในการกำหนด Activity rate ด้วยตัวเอง, คลิกใน frequency box ในสัญญาณใดๆ ที่มีค่ามาตรฐานพลังงานเป็น 0.0 ในใส่ค่า activity rate โดยสามารถเป็นได้ทั้งรูปของเปอร์เซ็นต์ของสัญญาณนาฬิกา หรือความถี่สัมบูรณ์

6. ในการตั้งค่า ค่ามาตรฐานความถี่ด้วยตนเองนอกเหนือจาก 0.0 Mhz. เลือกแถบ Tools บน toolbar และคลิกคำสั่ง "Set Default Activity" Dialog box จะปรากฏขึ้นให้เราสามารถตั้งค่าความถี่มาตรฐานได้

7. เราสามารถพิมพ์เลขจำนวนใดๆลงไปก็ได้ จากนั้นคลิก OK หรือ Cancel.

8. ทำเช่นเดียวกันนี้กับเพิ่มข้อมูล Clocks และ Outputs โดยคลิกบนเพิ่มข้อมูล เพื่อแสดงรายละเอียดใน Main window จากนั้นจึงทำการเปลี่ยนแปลงค่าต่างๆ

9. เมื่อเราเปลี่ยนความถี่แล้ว เราสามารถดูชนิดของ Capacitance element โดยขยาย โพลเดอร์ Clock, Logic และ Output เมื่อคลิกบนชื่อเพิ่มข้อมูลจะเป็นการแสดงส่วนประกอบทั้งหมดใน Main window

### **Loading Simulation Data**

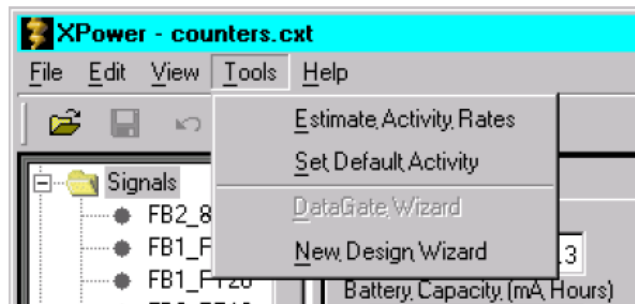
โปรแกรม XPower ได้ถูกพัฒนามาโดยมีความสามารถในการรับไฟล์ .VCD (Value Change Dump) จาก MTI simulators โดยสามารถโหลด simulation data file สู่ โปรแกรม XPower ด้วยตัวเองได้ดังนี้

1. เลือกคำสั่ง Open Simulation File... ในแถบ File ใน XPower.
2. Browse ไปยังตำแหน่งที่เก็บไฟล์ counter.vcd เลือกและคลิก Open.

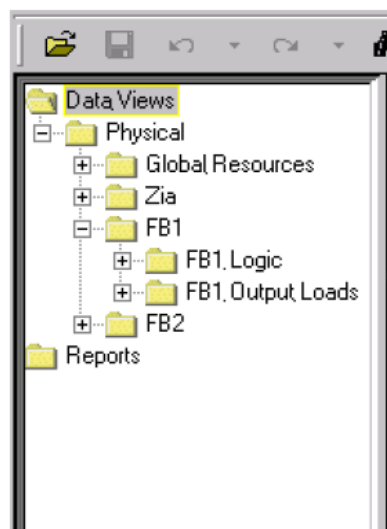
### Changing from Types to Physical View

มันไม่ใช่สิ่งจำเป็นในการเปลี่ยน View types สำหรับการคำนวณพลังงาน. อย่างไรก็ตาม ถ้าหากเราพบวิธีที่ง่ายกว่าในการดูสัญญาณโดย function block แทนที่จะใช้ Type เราสามารถเปลี่ยนเป็น Physical view ขณะที่เรากำลังตั้งค่า activity rate เราไม่ต้องปิดการออกแบบและเปิดใหม่ โดย

1. จากเมนู View ใน toolbar ด้านบน คลิก Data views และเลือก Physical จาก menu ที่ปรากฏขึ้นมา



2. ใน Explorer bar ดับเบิลคลิก Data view และจากนั้นใน Physical จะปรากฏเพิ่มข้อมูลย่อยชื่อ CLBs ทำการขยายโดยคลิกเครื่องหมาย "+" และเราจะเห็นเพิ่มข้อมูลย่อยทั้ง 4 สำหรับแต่ละ logic block โดยที่ Physical view นี้จะแสดงสายสัญญาณที่อยู่ใน logic block ทั้งหมด



## Generating the Power Report

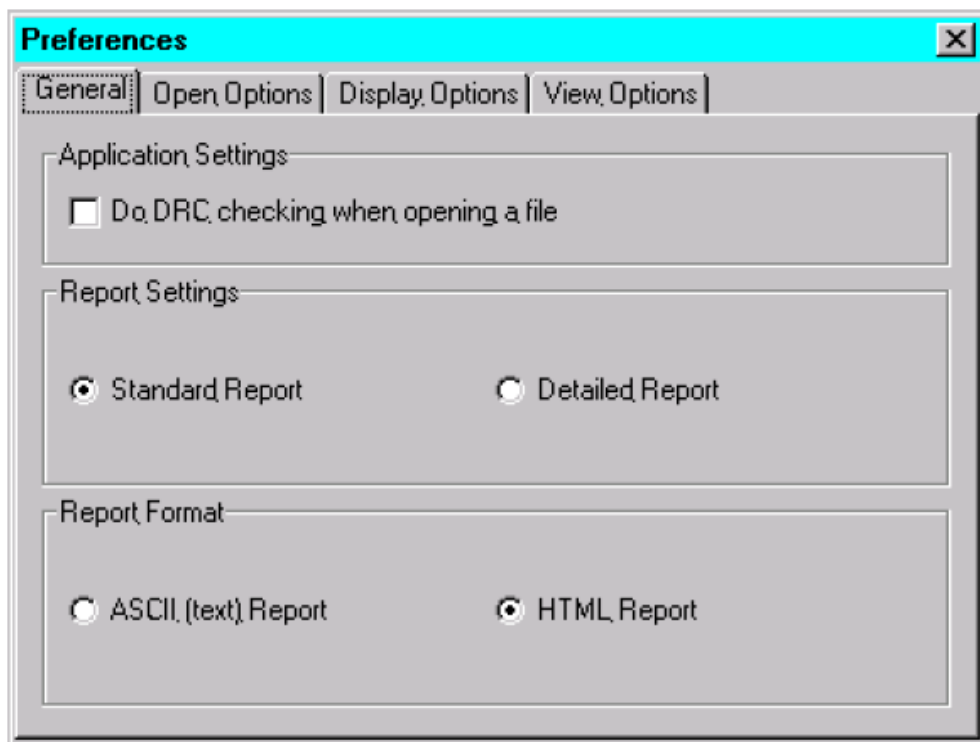
เมื่อเราเสร็จสิ้นการอัปเดต Frequency rate และปรับ voltage และ ambient temperature แล้ว เราจะพร้อมสำหรับการ generate power report แล้ว โดยจะเป็นได้ทั้ง HTML หรือ text ซึ่งทั้งสองจะถูกแสดงไว้ใน Main windows ของโปรแกรม Xpower

เราสามารถเลือกการตั้งค่าการแสดงผลของพลังงานได้ 2 แบบ : Standard report และ detailed report ดังนี้

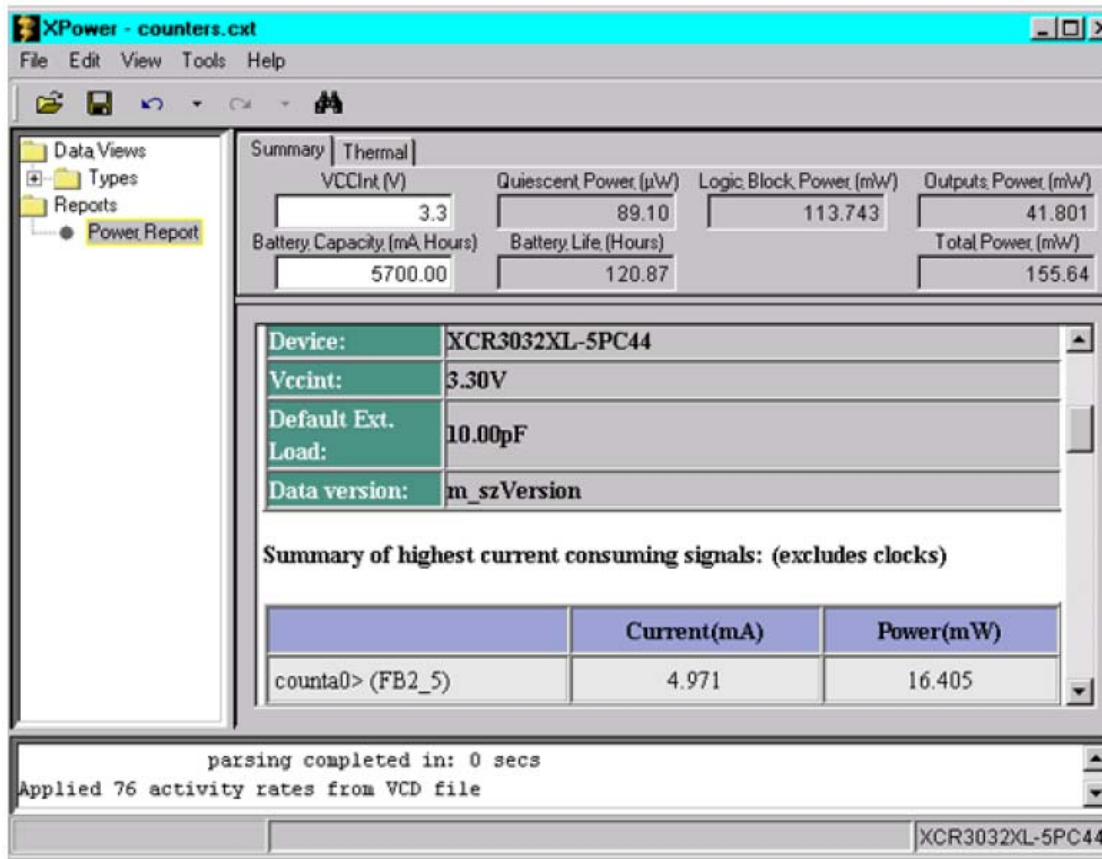
### 1. Standard report

สามารถทำได้โดย

1.1 เลือก Preference จากเมนู Edit และถ้าหากมันไม่ปรากฏขึ้นอัตโนมัติ ให้คลิก General tab ที่อยู่บนหัว dialog box ภายใน Report Settings เลือก "Standard Report", และภายใน Report Format เลือก "HTML Report" คลิก OK



## 1.2 ใน Explorer Bar ดับเบิลคลิกที่ Report จากนั้นคลิกที่ Power Report



1.3 Power Report จะปรากฏใน Main window และประกอบไปด้วย 4 ส่วนคือ

1.3.1 แสดงชื่อ, รุ่นของอุปกรณ์, ชื่อ package,  $V_{CCInt}$ , data version

1.3.2 แสดง Power summaries ของทั้ง design, รวมทั้ง พลังงานรวมทั้งหมดที่ใช้ แต่มีการแตกออกเป็นการใช้พลังงานโดย Net, Logic, Output, และ Quiescent Power

1.3.3 Thermal summary ซึ่งประกอบด้วย junction temperature, Ambient temperature และ  $\Theta_{JA}$  โดยทั้งหมดเป็นค่าโดยประมาณ

1.3.4 สุดท้ายจะแสดงวันและเวลาที่การ Analysis เสร็จสิ้น

## 2. Detailed report

มีวิธีการดังนี้

2.1 ทำตามขั้นตอน 1 ข้างต้น แต่เลือก "Detailed Report" แทน "Standard Report"

2.2.ทำตามขั้นตอนที่ 2 ข้างต้น แล้ว Detailed report จะปรากฏใน Main windows หลังจากเราสร้าง Power report แล้ว เรามีทางเลือกในการทำ 3 สิ่งต่อไปนี้ คือ

2.2.1 Saving the Power Report

2.2.2 Saving Your Design Settings

2.2.3 Closing the Design

### Saving the Power Report

เมื่อเราเปิด Power report คู ไฟล์ .pwr จะถูกสร้างขึ้น และจะถูกเซฟโดยอัตโนมัติไว้ใน directory เดียวกับไฟล์ design ของเรา. โดยจะถูกบันทึกทับของเดิม ทุกๆครั้งที่เราสร้าง Power report ขึ้นใหม่

### Saving Your Design Settings

เมื่อเราเสร็จสิ้นการกำหนด activity rate, voltage temperature และแบตเตอรี่ ของเราเองแล้ว จากนั้นจึงสร้าง power report แล้วเราจะมีทางเลือกในการบันทึกค่าเหล่านี้ลงในการออกแบบและการการตั้งค่าบนโปรแกรม XPower ซึ่งเราสามารถทำได้โดย 2 วิธีต่อไปนี้

1. คลิกสัญลักษณ์ Save บน Toolbar โดยจะบันทึกเป็นไฟล์ชื่อ counter\_xpwr.xml
2. ถ้าหากต้องการสร้างไฟล์ Setting ขึ้นมากกว่า 1 ไฟล์เพื่อใช้งาน ให้เลือก Save Settings As... จากเมนู File จากนั้นจึงเปลี่ยนชื่อตามต้องการ

### Closing the Design

1. เมื่อเสร็จขั้นตอนทุกอย่างแล้ว เลือกคำสั่ง Close จากเมนู File
2. ถ้าหากเรายังไม่ทำการบันทึก Design setting จะปรากฏ dialog box ขึ้น เพื่อให้เราทำการบันทึกไฟล์ก่อนทำการปิดทั้งหมด

### Opening the Design with a Saved Settings File

เมื่อเราบันทึกการตั้งค่าการออกแบบเป็นไฟล์ .XML และปิด design แล้ว เราเปิดใหม่โดยใช้ setting เดิมได้โดยทำได้ดังนี้

1. จากเมนู Edit เลือก Preferences และคลิกในส่วน Open Option ควรแน่ใจว่า Option ทั้งหมดได้ถูกเลือกแล้ว
2. เลือก Open file จากเมนู File ซึ่ง dialog box จะปรากฏขึ้นซึ่งให้เราสามารถ Browse ไฟล์ design, ไฟล์ Setting(.xml) และไฟล์ simulation (.VCD)
3. คลิกปุ่ม Browse ภายใน dialog box ดับเบิลคลิก Example และเลือกไฟล์ counter.ncd คลิก OK
4. จากนั้นคลิก Browse ภายใน dialog box ดับเบิลคลิก Example และเลือกไฟล์ counter\_xpwr.xml คลิก OK
5. คลิก OK ที่ด้านล่างของ dialog box จากนั้น design จะถูกเปิดขึ้นพร้อมกับค่า setting เดิมจากไฟล์ที่เราบันทึกไว้

**ภาคผนวก ค**

**คู่มือแนะนำการเขียนโปรแกรม  
ตรวจสอบผลลัพธ์ภายในโปรแกรม  
Xilinx 10.1**



## คู่มือการเขียนโปรแกรมตรวจสอบผลลัพธ์

ในงานวิจัยนี้ได้มีการโปรแกรมตรวจสอบโดยใช้โปรแกรม Xilinx เช่นเดียวกับงานวิจัยเกี่ยวกับงานวิจัยก่อนหน้า [วิวัฒน์ บุญสูง, 2551] โดยจะมีการตัดค่าจากไฟล์เสียงที่ได้นำมาทำการเขียนโปรแกรมเพื่อทำการตรวจสอบค่าผลที่ได้โดยในที่นี้ได้มีการนำไฟล์ก่อนหน้ามาใช้แต่มีแปลงบางส่วนที่ต่างกันไปซึ่งในงานวิจัยก่อนหน้าใช้ภาษา VHDL ในการพัฒนา ในที่นี้จึงได้ทำการแก้ไขโปรแกรมตรวจสอบผลลัพธ์โดยใช้ภาษา VHDL เนื่องจากสามารถที่จะนำมาใช้ร่วมกันได้ในส่วนไฟล์ตรวจสอบข้อมูลโดยจะนำเสนอในส่วนที่สำคัญๆ ดังนี้

### 1. ประกาศไลบรารีเพื่อที่จะทำการเรียกใช้การอ่าน/เขียนไฟล์

โดยในเบื้องต้นต้องทำการเขียนไลบรารีเพื่อให้มีมาตรฐานในการเขียนอ่านไฟล์ได้ โดยจะต้องมีการประกาศไลบรารี 2 ตัว ดังภาพต่อไปนี้

```
USE IEEE.STD_LOGIC_TEXTIO.ALL;
USE STD.TEXTIO.ALL;
```

### 2. ส่วนที่ทำการอ่าน/เขียนไฟล์ลงในไฟล์นามสกุล .txt

สามารถอธิบายการทำงานคือ โปรแกรมในบรรทัดแรกจะเป็นคำสั่งให้ทำการอ่านไฟล์ที่มีชื่อว่า infull.txt เพื่อที่จะอ่านค่าที่เราได้ทำการเขียนไว้ในไฟล์ infull.txt โดยในที่นี้ก็จะคือค่าของบิตเสียงที่ได้มีการตัดไว้เพื่อจะนำมาทดสอบนั่นเองและในบรรทัดที่ 3 จะเป็นการสั่งให้มีการเขียนผลลัพธ์ที่ได้ลงในไฟล์ที่มีชื่อว่า outfull.txt ซึ่งเมื่อผลลัพธ์ที่ได้ออกมาเราจะนำไฟล์ดังกล่าวมาทำการวาดกราฟเพื่อเปรียบเทียบผลที่ได้ในโปรแกรม MATLAB ต่อไปโดยสามารถแสดงโปรแกรมได้ดังรูปภาพต่อไปนี้

```
file fpi: text open read_mode is "infull.txt";
variable lni: line;
file fpo: text open write_mode is "outfull.txt";
variable lno: line;
variable datain, dataout: std_logic_vector(31 downto 0);
variable i: integer :=25;
```

### 3. การประกาศตัวแปรที่สำคัญที่ใช้ในการเก็บค่าอินพุตและเอาต์พุต

เมื่อพิจารณาจากรูปภาพในหัวข้อที่ 1 จะเห็นว่าในบรรทัดที่ 2 และ 4 ได้มีการสร้างตัวแปร Ini และ Ino โดยประกาศเป็น variable ซึ่งกล่าวคือเป็นตัวแปรที่สามารถเปลี่ยนแปลงค่าได้เพื่อจะนำตัวแปรดังกล่าวมาทำการเก็บชุดข้อมูลเป็นแถวของอินพุตและเอาต์พุตตามลำดับส่วนในตัวแปร datain และ dataout ที่ได้มีการเขียนในโปรแกรมบรรทัดที่ 5 ได้มีไว้สำหรับใช้เก็บค่าแต่ละบิตในแต่ละแถวที่ได้จากการอ่านไฟล์ infull.txและนำไปเขียนลงในไฟล์ outfull.txt ตามลำดับ

#### 4. การอ่านไฟล์และเขียนไฟล์ในแต่ละแถว

เนื่องจากในการอ่านค่าและเขียนค่าจะเป็นการทำงานที่แถวของข้อมูลโดยสามารถแสดงตัวข้อมูลในไฟล์ infull.txt ได้ดังนี้

```

infull.txt - Notepad
File Edit Format View Help
00111101110001101000000000 ← STEP 1
00111101101110011100000000 ●
00111101100110001111000000 ●
00111101011011110000000000 ●
00111101000000011000000000 ●
00111010111110000000000000 ●
10111101000010101110000000 ●
10111101100000110010000000 ●
10111101110000000101000000 ●
10111101110100100010000000 ●
10111100000010111000000000 ●
10111110000011010110100000 ●
10111110000101110001100000 ●
10111110001111111000000000 ●
10111110001011001111100000 ●
10111110001100100111000000 ●
10111110001100110101100000 ●
10111110001010010110000000 ●
10111110000111100111000000 ●
10111110000011110001000000 ●
10111110000001000011000000 ●
10111101111001011101000000 ●
10111101110100110101000000 ●
10111101110010000001000000 ← STEP r

```

ซึ่งการทำงานจะเริ่มตั้งแต่บรรทัดแรกลงมาเรื่อยๆจนถึงบรรทัดสุดท้าย โดยสามารถอธิบายการทำงานในแต่ละบรรทัดของไฟล์อินพุตตลอดจนเขียนไฟล์ผลลัพธ์กลับในไฟล์เอาต์พุตได้ดังโปรแกรมต่อไปนี้

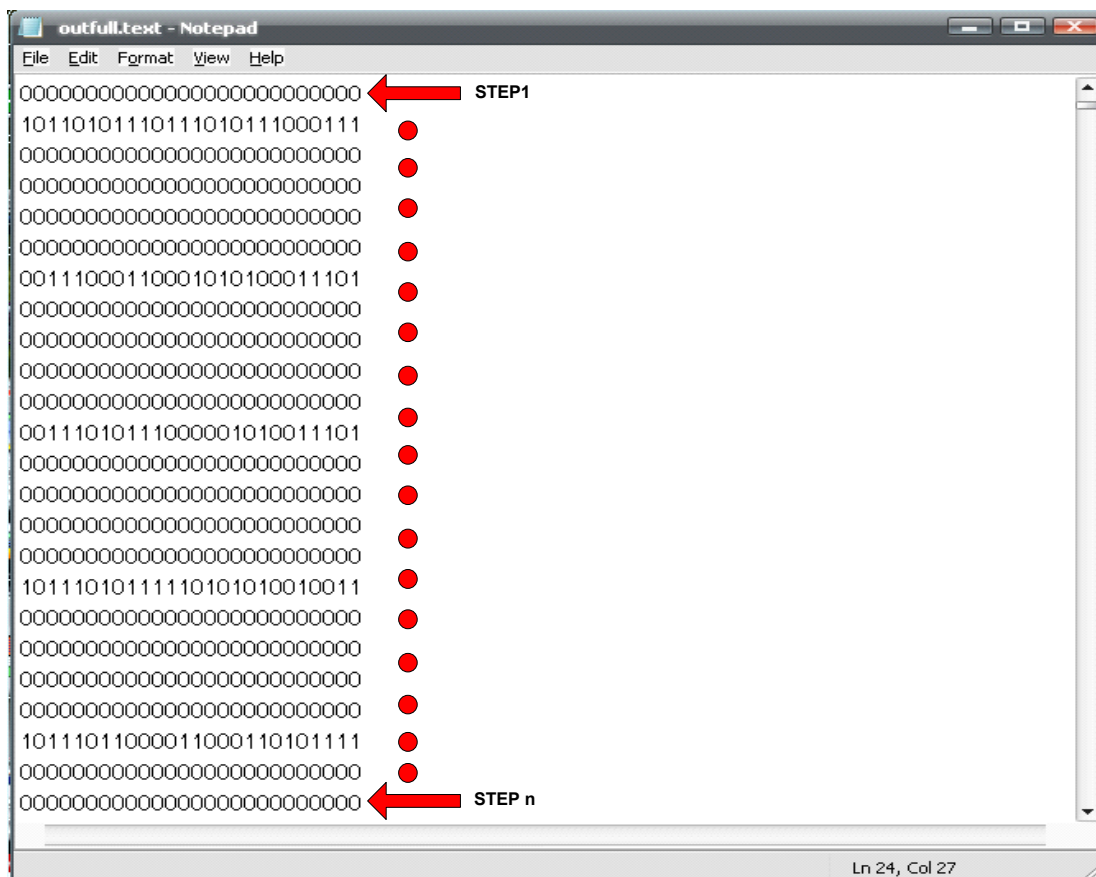
```

WAIT FOR 2800 ns;
readline( fpi, lni );
read( lni, datain );
datain_all <= datain;
WAIT FOR 600 ns;
dataout := datain_all;
write( lno, dataout );
writeline( fpo, lno );

```

สามารถอธิบายการทำงานได้คือในโปรแกรมบรรทัดแรกจะเป็นการบอกให้รอเท่ากับช่วงเวลางานของโปรแกรมในทีนี้ได้มีการกำหนดค่าไว้ที่ 2800 ns ซึ่งเป็นช่วงเวลาในการประมวลผลของ วงจรดิจิทัลฟิวเตอร์แบบกึ่งแบบพิจารณาภาพรวมและใช้เทคนิคการลดบิตข้อมูลส่วนในบรรทัดที่ 2 เป็นการอ่านไฟล์จากไฟล์ infull.txt ในแถวแรกที่ได้มีการแสดงดังลูกศรสีแดงเพื่อจะนำค่านั้นมาเก็บไว้ในตัวแปร lni หลังจากนั้นทำการอ่านค่าในแถวแรกนี้ทีละค่าเพื่อที่จะ

ถ่ายข้อมูลลงในตัวแปร `datain` และในบรรทัดที่ 4 เป็นการนำค่าเอาต์พุตที่ได้จากการอ่านไฟล์ไปทำการใส่ตัวแปร `datain_all` ซึ่งเป็นตัวแปรอินพุทของวงจรดิจิทัลเฟลเตอร์แบงก์เพื่อที่จะให้ข้อมูลในชุดแรก ได้มีการประมวลผลเพื่อให้ได้ผลลัพธ์เอาต์พุตออกมา ในบรรทัดที่ 5 เป็นการรอเพื่อป้องกันข้อมูลที่ค้างในส่วนเอาต์พุตซึ่งในที่นี่จะมีการตั้งค่าการรอด้วยเวลา 1 ช่วงสัญญาณนาฬิกา ซึ่งหลังจากที่ได้เอาต์พุตที่ได้จากการประมวลผลเอาต์พุตในจะอยู่ในตัวแปรที่มีชื่อว่า `dataout_all` โดยในบรรทัดที่ 6 จะเป็นการนำผลลัพธ์ที่ได้ใน `dataout_all` ไปเก็บในตัวแปรที่มีชื่อว่า `dataout` เพื่อเตรียมพร้อมที่จะทำการบันทึกข้อมูลลงในไฟล์ `outfull.txt` โดยจะมีการบันทึกค่าที่ละค่าจนครบแถวของชุดข้อมูลโดยจะเก็บในตัวแปร `Ino` ดังโปรแกรมบรรทัดที่ 7 และในบรรทัดสุดท้ายจะเป็นการนำผลลัพธ์ในชุดข้อมูลแรกบันทึกลงในไฟล์ `outfull.txt` ดังนั้นเมื่อต้องการที่จะทำการตรวจสอบผลลัพธ์ด้วยจำนวนชุดข้อมูล 50 ชุดก็จะมีการนำส่วนของโปรแกรมนี้อีก 50 ตัวเพื่อจะบันทึกผลลัพธ์ที่ได้โดยผลลัพธ์ที่ได้สามารถแสดงได้ดังรูปภาพต่อไปนี้



## ประวัติผู้เขียน

ชื่อ สกุล นายเกรียงไกร ทักษณวิภาส

รหัสประจำตัวนักศึกษา 5010120002

## วุฒิการศึกษา

วุฒิ	ชื่อสถาบัน	ปีที่สำเร็จการศึกษา
วิศวกรรมศาสตรบัณฑิต (วิศวกรรมคอมพิวเตอร์)	มหาวิทยาลัยสงขลานครินทร์	2550

## ทุนการศึกษา (ที่ได้รับในระหว่างการศึกษา)

ทุนผู้ช่วยวิจัย คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ ประจำปีการศึกษา 2550

## การตีพิมพ์เผยแพร่ผลงาน

เกรียงไกร ทักษณวิภาส, ณัฐฐา จินดาเพชร , พรชัย พฤกษ์ภัทรานนท์ และ วรณรัช สันติอมรทัต, “การออกแบบดิจิทัลฟิลเตอร์เบงค์โดยใช้ทรัพยากรร่วมแบบพิจารณาภาพรวมบน FPGAs”, การประชุมวิชาการทางวิศวกรรมไฟฟ้าครั้งที่32, ปราณบุรี, 28-30 ตุลาคม 2551, หน้า 1063-1066