การออกแบบวงจรดิจิตอลฟิลเตอร์แบงก์โดยหลักการใช้ทรัพยากรดาต้าพาทร่วมกันแบบลำดับชั้นบน FPGAs
Design of a Digital Filter Bank Based on Hierarchical Data-Path Resource-Sharing on FPGAs

วิวัฒน์ บุญสูง (Wiwat Bunsung)[1]
ณัฏฐา จินดาเพ็ชร์ (Nattha Jindapetch)[2]*
พรชัย พฤกษ์ภัทรานนต์ (Pornchai Phukpattranont)[3]

บทคัดย่อ

บทความนี้นำเสนอระเบียบวิธีการออกแบบวงจรดิจิตอลขนาดใหญ่สำหรับวงจรดิจิตอลฟิลเตอร์แบงก์ภายใต้พื้นที่บน FPGA (Field Programmable Gate Array) ที่มีอยู่อย่างจำกัด การออกแบบแบ่งเป็นสองส่วนคือ ส่วนวงจรข้อมูล (Data-path part) สำหรับการประมวลผลสัญญาณข้อมูล และส่วนวงจรควบคุม (Control part) สำหรับควบคุมจังหวะการทำงานวงจรข้อมูล โดยส่วนวงจรข้อมูลถูกออกแบบโดยอาศัยหลักการใช้ทรัพยากรร่วมกันแบบลำดับชั้นเพื่อลดขนาดของวงจร เริ่มจากเซทของฟังก์ชันที่อธิบายพฤติกรรมของวงจรถูกแปลงให้อยู่ในรูปกราฟกระแสข้อมูล (Data Flow Graph ; DFG) ที่มีโครงสร้างเป็นลำดับชั้น จากนั้น DFG ที่เหมือนกันถูกจัดให้ใช้ทรัพยากรร่วมกันเป็นลำดับชั้นจากกลุ่มใหญ่ไปเล็ก โดยมีการคำนึงถึงเวลาที่ช้าลงหลังจากมีการใช้ทรัพยากรร่วมกัน ในส่วนวงจรควบคุม FSM (Finite State Machine) แบบ Moore machine ถูกใช้สำหรับควบคุมจังหวะการใช้ทรัพยากรร่วมกันของส่วนวงจรข้อมูลให้เป็นไปอย่างถูกต้อง วงจรที่ได้ออกแบบถูกทดสอบบนชิพ FPGA ของบริษัท Xilinx ตระกูล SPARTAN-3 เบอร์ XCS4000-5FG676 จากการจำลองแบบการทำงานวงจรสามารถทำงานได้ถูกต้องเมื่อเทียบกับผลการคำนวณจากโปรแกรม MATLAB และจากผลการสังเคราะห์วงจรที่ออกแบบใช้พื้นที่ลดลง 44% และมีความเร็วรอบการทำงานลดลง 30% เทียบกับวงจรที่ไม่มีการใช้ทรัพยากรร่วมกัน

คำสำคัญ: การใช้ทรัพยากรร่วมกัน, ดิจิตอลฟิลเตอร์แบงก์, เอฟพีจีเอ

ABSTRACT

This paper presents a large digital circuit design methodology for a digital filter bank circuit on an area limited FPGA (Field Programmable Gate Array). The design is divided into two parts: a data-path part for data processing and a control part for controlling data-path operations. The data-path part is designed by using a

[1]*นักศึกษาปริญญาโท ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์*
[2]*ผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์*
[3]*ผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์*
**corresponding author, e-mail: nattha.s@psu.ac.th*

hierarchical resource-sharing technique to reduce the circuit size. The set of functions describing the circuit behavior is translated into hierarchical data flow graphs (DFGs). Then the DFGs are hierarchically grouped into the same structure DFGs to share the same resources from large to small groups. The slower operation of the circuit after resource-sharing is of concern. In the control part, the FSM (Finite State Machine) in Moore machine format is used for controlling the correct sequence of the resource-sharing in the data-path part. The designed circuit was tested on a Xilinx FPGA SPARTAN-3 XCS4000-5FG676. The circuit worked correctly compared to the calculation results from MATLAB. From the synthesis results, the circuit size is 44% smaller, and the cycle time is 30% slower, compared to the original circuit without resource-sharing.

Keywords: resource-sharing, digital filter bank, FPGA (Field Programmable Gate Array)

บทนำ

วงจรดิจิตอลฟิลเตอร์แบงก์มีประโยชน์สำหรับการแยกสัญญาณออกเป็นหลายช่วงความถี่เพื่อใช้ในงานประยุกต์ที่บางช่วงความถี่มีความสำคัญกว่าช่วงความถี่อื่น เช่น งานด้านการประมวลสัญญาณเสียงในเครื่องช่วยฟังดิจิตอล (Nielsen et al., 1999 ; Lunner et al., 1991 ; bernardini et al., 2006) งานด้านการประมวลผลภาพดิจิตอล (Balasingham et al., 2008) เป็นต้น

การศึกษาและการออกแบบวงจรดิจิตอลในปัจจุบัน ได้มีนำวงจรรวมดิจิตอลชนิดหนึ่งที่เรียกว่าเอฟพีจีเอ (Field Programmable Gate Array ; FPGA) มาเป็นองค์ประกอบในการเรียนรู้ ทั้งนี้ เนื่องจากวงจรทั้งหมดที่ต้องการออกแบบจะถูกออกแบบบนโปรแกรมคอมพิวเตอร์ แล้วสามารถจำลองการทำงาน เพื่อวิเคราะห์ผลที่ได้ออกแบบ เมื่อถูกต้องแล้วก็ทำการโปรแกรมวงจรที่ได้ออกแบบไว้ลงบนชิพลงในชิพเพียงตัวเดียวที่สามารถใช้ทำงานจริงได้ทันที นอกจากนี้การพัฒนาระบบดิจิตอลบน FPGA ยังมีความยืดหยุ่นมากกว่าการพัฒนาบนไมโครคอนโทรลเลอร์และชิพประมวลผลสัญญาณดิจิตอล ระบบสามารถถูกออกแบบให้ทำงานแบบขนานเมื่อมีข้อจำกัดด้านความเร็ว สามารถถูกออกแบบให้มีขนาดเล็กเมื่อมีข้อจำกัดด้านพื้นที่ สามารถรวมระบบการทำงานร่วมฮาร์ดแวร์/ซอฟท์แวร์ไว้ในชิพเดียว เป็นต้น คุณสมบัติดังกล่าวจึงทำให้ FPGA เหมาะกับการออกแบบวงจรดิจิตอลที่มีขนาดใหญ่

จากคุณสมบัติของชิพวงจรรวม FPGA พบว่าเหมาะสมที่จะนำมาใช้ในการศึกษาและออกแบบวงจรรวมที่มีขนาดใหญ่อย่างวงจรดิจิตอลฟิลเตอร์แบงก์ซึ่งนิยมใช้ในการแยกสัญญาณ แต่ในการออกแบบวงจรดิจิตอลที่มีขนาดใหญ่นั้นจำเป็นต้องใช้ทรัพยากรภายในเอฟพีจีเอมาก และบ่อยครั้งที่ทรัพยากรดังกล่าวไม่เพียงพอ หากนำหลักการใช้ทรัพยากรร่วมกัน (Resource-sharing) ระหว่างโอเปอเรชันต่างๆ ซึ่งเป็นขั้นตอนหนึ่งในสังเคราะห์วงจรที่ระดับสูงที่ทำให้วงจรขนาดใหญ่สามารถถูกสร้างได้บนชิพที่มีพื้นที่จำกัดมาใช้ในการออกแบบ จะมีประโยชน์มากต่อการออกแบบ แต่หลังจากการใช้ทรัพยากรร่วมกันวงจรจะทำงานช้าลง และหากยิ่งมีการใช้ทรัพยากรร่วมกันมากขึ้นเท่าไหร่วงจรก็จะทำงานช้าลงมากยิ่งขึ้นเช่นกัน นอกจากนี้ยังส่งผลในการต่อสายสัญญาณ (Interconnection) ภายในชิพที่จะซับซ้อนยิ่งขึ้นด้วย มีงานวิจัยหลายงานที่นำเสนอระเบียบวิธี Resource-sharing ที่มีประสิทธิภาพ ได้แก่ ระเบียบวิธีเลี่ยงการเกิดเส้นทางย้อนกลับและเส้นทางที่ยาวเกิน (Qiao et al., 2007) ระเบียบวิธีการใช้ทรัพยากรร่วมกันระหว่างบล็อกของโอเปอเรชัน (Memik et al., 2003 ; Jaschke et al., 1999) ได้พิจารณาถึงความซับซ้อนเนื่องจากการเชื่อมต่อสายระหว่างโมดูล แต่ก็ไม่ได้พิจารณาการใช้ทรัพยากรร่วมกันภายในบล็อก

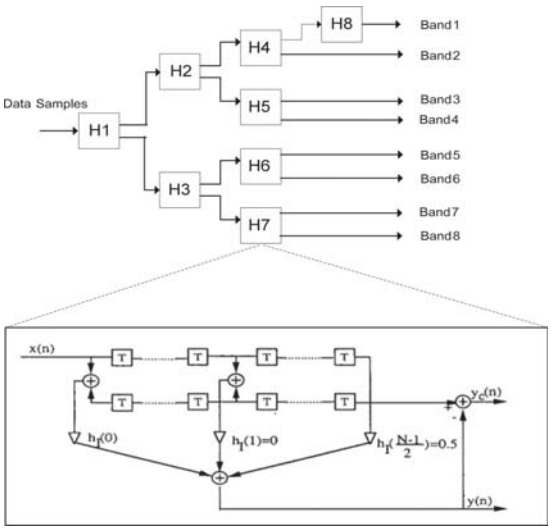
ในบทความนี้นำเสนอระเบียบวิธีการใช้ทรัพยากรร่วมกันแบบลำดับชั้นในการออกแบบวงจรดิจิตอลฟิลเตอร์แบงก์ภายใต้พื้นที่บน FPGA ที่มีอยู่อย่างจำกัด โดยมีการใช้ทรัพยากรร่วมกันระหว่างบล็อกและภายในบล็อกเดียวกัน โดยมีการคำนึงถึงเวลาที่ช้าลงและความซับซ้อนของสายเชื่อมต่อระหว่างโมดูลหลังจากมีการใช้ทรัพยากรร่วมกัน

วงจรดิจิตอลฟิลเตอร์แบงก์

วงจรดิจิตอลฟิลเตอร์แบงก์ (Digital filter bank) คือวงจรกรองสัญญาณเชิงเลขประเภทหนึ่ง มีการทำงานคือทำหน้าที่แยกสัญญาณด้านเข้าที่ถูกป้อนเข้ามาในวงจรเพื่อปรับปรุงคุณภาพของสัญญาณ โดยสัญญาณที่ถูกป้อนเข้ามาจะถูกแยกหรือกรองเป็นสัญญาณที่มีความถี่ต่างๆ กันไปซึ่งเป็นผลมาจากการทำงานของฟิลเตอร์ย่อยที่อยู่ภายในฟิลเตอร์แบงก์และสัญญาณที่ได้จากฟิลเตอร์ย่อยตัวแรกจะถูกป้อนเป็นอินพุตให้แก่ฟิลเตอร์ย่อยอื่นๆ ที่อยู่ภายในฟิลเตอร์แบงก์ไปตามลำดับจนครบ ซึ่งฟิลเตอร์แต่ละตัวเหล่านั้นก็จะทำหน้าที่แยกหรือกรองสัญญาณที่ได้รับเพื่อปรับปรุงคุณภาพของสัญญาณจากการประมวลผลของฟิลเตอร์แต่ละตัวเช่นกัน สุดท้ายก็จะได้สัญญาณด้านออกของฟิลเตอร์แบงก์

วงจรดิจิตอลฟิลเตอร์แบงก์ที่ใช้ในงานวิจัยนี้เป็นวงจรกรองดิจิตอลที่มีสัญญาณด้านเข้า 1 ช่อง ภายในประกอบไปด้วยบล็อกประมวลผล 8 บล็อก เพื่อทำการแยกสัญญาณที่เข้ามาออกเป็นสัญญาณด้านออก 8 ช่องสัญญาณ โดยแสดงได้ดังรูปที่ 1

จากรูปแสดงให้เห็นถึงการทำงานขององค์ประกอบพื้นฐานของวงจรกรองดิจิตอล โดยหากสังเกตจะพบว่าโครงสร้างภายในของฟิลเตอร์ย่อยนั้นมีโครงสร้างแบบเดียวกันกับโครงสร้างของวงจรกรองแบบ FIR ดังรูปขยายของบล็อก H7 ซึ่งโครงสร้างการทำงานนี้สามารถเขียนให้อยู่ในของฟังก์ชันทางคณิตศาสตร์ในรูปสมการผลต่างของระบบ (Difference Equation)



รูปที่ 1. การประมวลผลของฟิลเตอร์แต่ละตัวภายในฟิลเตอร์แบงก์

โดยการทำงานของวงจรฟิลเตอร์แบงก์นี้คือฟิลเตอร์ย่อยที่อยู่ภายในแต่ละบล็อกมีการทำงานที่เหมือนกันคือ รับสัญญาณอินพุตที่ด้านเข้าจากนั้นประมวลผลสัญญาณภายใต้สมการผลต่างโดยที่ฟิลเตอร์ย่อยแต่ละตัวจะมีสมการเฉพาะสำหรับฟิลเตอร์แต่ละบล็อก โดยการที่จะสร้างสมการผลต่างของฟิลเตอร์ได้นั้นจะต้องทราบค่าผลตอบสนองต่ออิมพัลส์ (h(n)) ซึ่งตารางที่ 1 แสดงให้เห็นความสัมพันธ์ของค่าตอบสนองต่ออิมพัลส์ของฟิลเตอร์แต่ละบล็อกเพื่อช่วยในการสร้างสมการผลต่างของฟิลเตอร์แต่ละบล็อกที่ใช้ประมวลผลสัญญาณภายในวงจรฟิลเตอร์แบงก์นั่นเอง รูปที่ 2 แสดงสมการผลต่างของฟิลเตอร์ย่อยแต่ละบล็อกที่ได้จากตารางที่ 1

ตารางที่ 1. ความสัมพันธ์ของค่าผลตอบสนองต่ออิมพัลส์ของฟิลเตอร์ย่อย

$H_1(z)$	$h_1(0)=h_1(48),\ h_1(16)=h_1(32),\ h_1(24)=0.5$
$H_2(z)$	$h_2(0)=h_2(24),\ h_2(8)=h_2(16),\ h_2(12)=0.5$
$H_3(z)$	$h_3(0)=h_3(28),\ h_3(4)=h_3(24),\ h_3(8)=h_3(20),\ h_3(12)=h_3(16),\ h_3(14)=0.5$
$H_4(z)$	$h_4(0)=h_4(12),\ h_4(4)=h_4(8),\ h_4(6)=0.5$
$H_5(z)$	$h_5(0)=h_5(10),\ h_5(2)=h_5(8),\ h_5(4)=h_5(6),\ h_5(5)=0.5$
$H_6(z)$	$h_6(0)=h_6(6),\ h_6(2)=h_6(4),\ h_6(5)=0.5$
$H_7(z)$	$h_7(0)=h_7(30),\ h_7(6)=h_7(24),\ h_7(12)=h_7(18),\ h_7(15)=0.5$
$H_8(z)$	$h_8(0)=h_8(2),\ h_8(1)=0.5$

$H_1(z)$: $y_1(n) = [x(n-0) + x(n-48)] * h_1(0) + [x(n-16) + x(n-32)] * h_1(1) + x(n-24) * h_1(2)$

$y_{1c}(n) = x(n-24) - y_1(n)$

$H_2(z)$: $y_2(n) = [x(n-0) + x(n-24)] * h_2(0) + [x(n-8) + x(n-16)] * h_2(1) + x(n-12) * h_2(2)$

$y_{2c}(n) = x(n-12) - y_2(n)$

$H_3(z)$: $y_3(n) = [x(n-0) + x(n-28)] * h_3(0) + [x(n-4) + x(n-24)] * h_3(1) +$
$[x(n-8) + x(n-20)] * h_3(2) + [x(n-12) + x(n-16)] * h_3(3) + x(n-14) * h_3(4)$

$y_{3c}(n) = x(n-14) - y_3(n)$

$H_4(z)$: $y_4(n) = [x(n-0) + x(n-12)] * h_4(0) + [x(n-4) + x(n-8)] * h_4(1) + x(n-6) * h_4(2)$

$y_{4c}(n) = x(n-6) - y_4(n)$

$H_5(z)$: $y_5(n) = [x(n-0) + x(n-10)] * h_5(0) + [x(n-2) + x(n-8)] * h_5(1) + [x(n-4) + x(n-6)] * h_5(2) +$
$x(n-5) * h_5(3)$

$y_{5c}(n) = x(n-5) - y_5(n)$

$H_6(z)$: $y_6(n) = [x(n-0) + x(n-6)] * h_6(0) + [x(n-2) + x(n-4)] * h_6(1) + x(n-3) * h_6(2)$

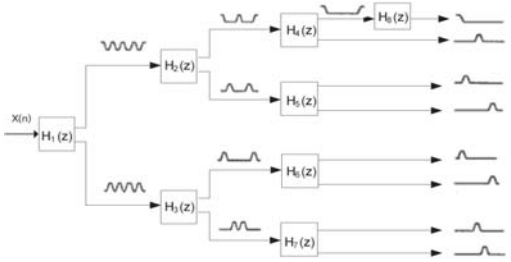
$y_{6c}(n) = x(n-3) - y_6(n)$

$H_7(z)$: $y_7(n) = [x(n-0) + x(n-30)] * h_7(0) + [x(n-8) + x(n-24)] * h_7(1) + [x(n-12) + x(n-18)] * h_7(2)$
+
$x(n-15) * h_7(3)$

$y_{7c}(n) = x(n-15) - y_7(n)$

$H_8(z)$: $y_8(n) = [x(n-0) + x(n-2)] * h_8(0) + x(n-1) * h_8(1)$

รูปที่ 2. สมการผลต่างของฟิลเตอร์ภายในฟิลเตอร์แบงก์



รูปที่ 3. สัญญาณอินพุตและสัญญาณเอาท์พุท ที่ฟิลเตอร์ย่อยแต่ละบล็อก

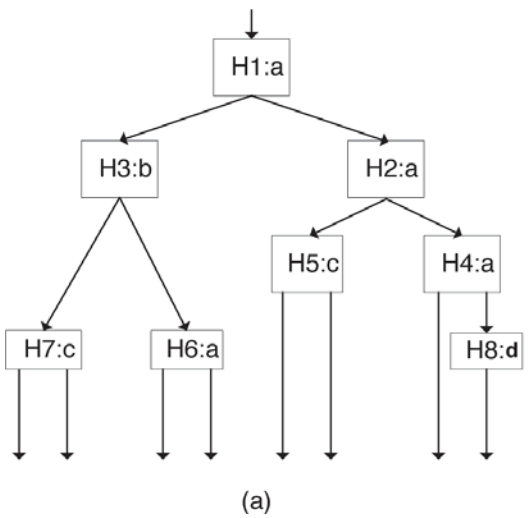
การทำงานของฟิลเตอร์แบงก์จะเริ่มจากสัญญาณอินพุต x(n) ผ่านเข้ามาที่ฟิลเตอร์ $H_1(z)$ จากนั้นสัญญาณจะถูกประมวลผลแล้วส่งค่าเอาท์พุตออกมาเป็นสองค่าคือ $y_1(n)$ กับ $y_{1c}(n)$ โดยที่เอาท์พุตสองค่านี้หากสังเกตจากรูปที่ 3 จะเห็นว่าเป็นค่าที่ตรงข้ามกันซึ่งกันและกันนั่นเอง จากรูปเมื่อได้ค่าเอาท์พุตสองค่าดังที่กล่าวมาแล้ว ค่าเอาท์พุตนี้จะถูกส่งไปประมวลผลต่อด้วยฟิลเตอร์ในลำดับถัดไป ($H_2(z)$ และ $H_3(z)$)

ลำดับถัดไปฟิลเตอร์ย่อย $H_2(z)$ จะรับค่าอินพุตซึ่งคือค่าเอาท์พุต $y_1(n)$ ของฟิลเตอร์ $H_1(z)$ มาป้อนให้ฟิลเตอร์ $H_2(z)$ ซึ่งทำการแยกสัญญาณออกมาได้ค่าเอาท์พุตเป็น $y_2(n)$ และ $y_{2c}(n)$ จากนั้นค่าสัญญาณ $y_2(n)$ จะถูกป้อนเป็นอินพุตให้กับฟิลเตอร์ $H_4(z)$ ซึ่งจากฟิลเตอร์ $H_4(z)$ นั้นสัญญาณอินพุตจะถูกประมวลผลได้ค่าสัญญาณเอาท์พุตเป็น $y_4(n)$ กับ $y_{4c}(n)$ ค่าเอาท์พุต $y_4(n)$ จะเป็นค่าอินพุตให้แก่ $H_8(z)$ จนสุดท้ายจะได้ค่า $y_8(n)$ ส่วนค่า $y_{8c}(n)$ จะไม่นำมาใช้ และสำหรับสัญญาณ $y_{2c}(n)$ จะป้อนไปยังฟิลเตอร์ $H_5(z)$ ได้ค่าเอาท์พุตเป็น $y_5(n)$ กับ $y_{5c}(n)$ เมื่อทำการประมวลผลเสร็จแล้วจะได้ค่าสัญญาณทั้งหมด 4 ค่า

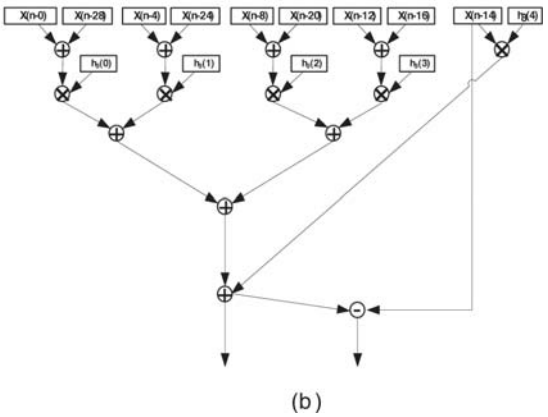
ส่วนด้านของฟิลเตอร์ย่อย $H_3(z)$ มีการทำงานที่เป็นไปในลักษณะเดียวกันกับ ฟิลเตอร์ย่อย $H_2(z)$ โดยที่ค่าอินพุตเริ่มต้นคือค่า $y_{1c}(n)$ และเมื่อทำงานเสร็จแล้วนั่นจะได้ค่าเอาท์พุตอีก 4 ค่า ซึ่งเมื่อไปรวมกับเอาท์พุตก่อนหน้านี้ก็จะได้ค่าสัญญาณเอาท์พุตทั้งหมด 8 ค่านั่นเอง

Data Flow Graph (DFG)

กราฟกระแสข้อมูล หรือ Data Flow Graph (DFG) นิยมใช้ในการแทนโครงสร้างของดิจิตอลฟิลเตอร์ ซึ่งง่ายต่อการสังเคราะห์วงจรที่ระดับสูง (High-level synthesis) เพื่อให้ได้วงจรที่เหมาะสมที่สุด จากโครงสร้างของฟิลเตอร์แบงก์ในรูปที่ 1 และสมการผลต่างของฟิลเตอร์แต่ละบล็อกในรูปที่ 2 สามารถถูกแสดงอยู่ในรูปของ DFG ได้ดังรูปที่ 4 โดยรูปที่ 4(a) แสดง DFG ลำดับบน โดยแต่ละบล็อกแทนวงจรฟิลเตอร์แต่ละบล็อกตามชื่อที่ระบุ และรูปที่ 4(b) แสดง DFG ลำดับล่าง ซึ่งแสดงโครงสร้างภายในของบล็อก H3 เป็นตัวอย่าง การคูณ การบวก การลบ ในสมการผลต่างถูกแทนด้วยโอเปอเรชันคูณ บวก และลบตามลำดับดังแสดงในวงกลม ส่วนโอเปอแรนด์ หรือสัญญาณและค่าสัมประสิทธิ์ถูกแทนด้วยกล่องสี่เหลี่ยม ลูกศรแสดงการไหลของข้อมูล โดยในการสร้างวงจรจริงทรัพยากรที่ต้องใช้ในการสร้างบล็อก H3 คือ วงจรคูณ 5 ตัว วงจรบวก 8 ตัว วงจรลบ 1 ตัว และชิฟท์รีจิสเตอร์ขนาด 29 สเตจสำหรับเก็บค่าสัญญาณจาก x(n-0) - x(n-28) และรีจิสเตอร์ 5 ตัวสำหรับเก็บค่าสัมประสิทธิ์



(a)



รูปที่ 4. DFG ของวงจรฟิลเตอร์แบงก์ (a) DFG ลำดับบน และ (b) DFG ลำดับล่าง (block H3)

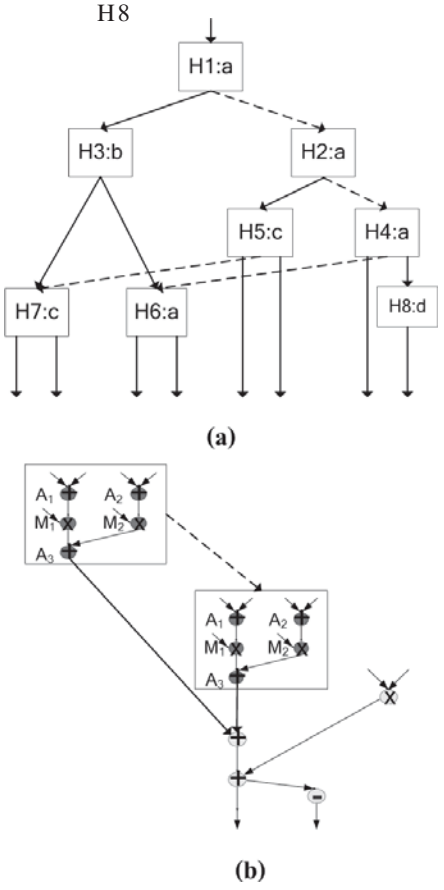
การใช้ทรัพยากรร่วมกันแบบลำดับชั้น

การออกแบบวงจรขนาดใหญ่บนชิพที่มีพื้นที่จำกัดดังเช่น FPGA มีความจำเป็นที่จะต้องจัดการกับทรัพยากรที่ต้องใช้ในการออกแบบวงจรให้เหมาะสมกับขนาดของ FPGA งานวิจัยนี้จึงนำเสนอระเบียบวิธีการใช้ทรัพยากรร่วมกันแบบลำดับชั้นในการออกแบบวงจรฟิลเตอร์แบงก์ซึ่งเป็นวงจรที่มีขนาดใหญ่ ทรัพยากรที่ต้องใช้ก็คือตัวบวก ตัวคูณ และ รีจิสเตอร์ ซึ่งเป็นองค์ประกอบพื้นฐานของวงจรฟิลเตอร์นั่นเอง โดยที่ระเบียบวิธีที่นำเสนอในการออกแบบวงจรดังที่กล่าวไปข้างต้นนั้นสามารถแบ่งย่อยออกเป็น 2 วิธี คือ การใช้ทรัพยากรร่วมระหว่างบล็อก (Inter-Block Sharing) และ การใช้ทรัพยากรร่วมภายในบล็อก (Intra-Block Sharing) ดังรายละเอียดต่อไปนี้

1. การใช้ทรัพยากรร่วมระหว่างบล็อก (Inter-Block Sharing)

การใช้ทรัพยากรร่วมระหว่างบล็อก (Inter-Block Sharing) คือการใช้ทรัพยากรร่วมกันระหว่างบล็อกในระดับบนสุดของ DFG โดยที่การใช้ทรัพยากรร่วมกันระหว่างบล็อกนี้จะถูกนำมาพิจารณาเมื่อฟังก์ชันภายในของทั้ง 2 บล็อกนั้นเหมือนกันและนอกจากนี้หากบล็อก 2 บล็อกหรือหลายๆ บล็อกมีโครงสร้างของ DFG ที่เหมือนกันนั้นก็สามารถใช้ทรัพยากรในบล็อกร่วมกันได้ เมื่อนำโครงสร้างของฟิลเตอร์ย่อยแต่ละตัวภายในฟิลเตอร์แบงก์รูปที่ 1 มาพิจารณานั้น พบว่าสามารถทำการจัดกลุ่มโครงสร้างของฟิลเตอร์ภายในฟิลเตอร์แบงก์ตามลักษณะฟังก์ชันภายในที่เหมือนกันได้เป็น 4 แบบ ดังนี้

- โครงสร้างแบบ a จะสอดคล้องกับบล็อก H1, H2, H4 และ H6
- โครงสร้างแบบ b จะสอดคล้องกับบล็อก H3
- โครงสร้างแบบ c จะสอดคล้องกับบล็อก H5 และ H7
- โครงสร้างแบบ d จะสอดคล้องกับ Block H8



รูปที่ 5. ฟิลเตอร์แบงก์ที่ออกแบบด้วยการใช้ทรัพยากรร่วมกันระหว่างบล็อก (a) และการใช้ทรัพยากรร่วมกันภายในบล็อก (b)

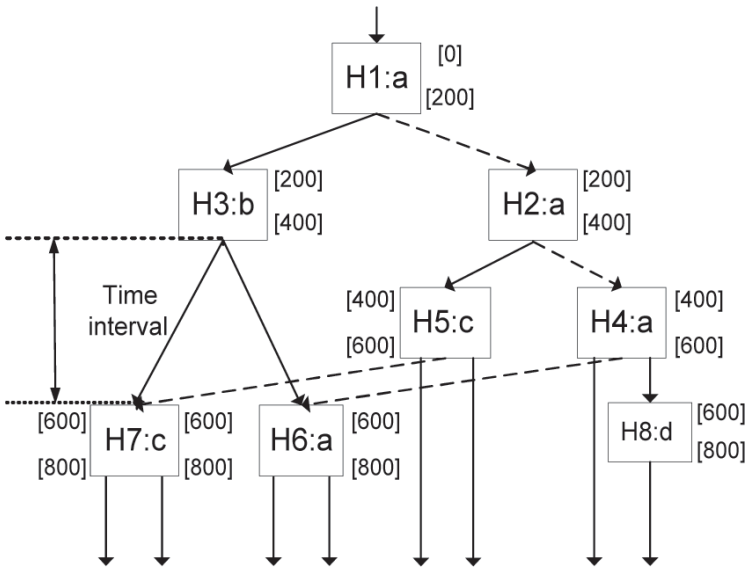
จากนิยามหรือความหมายดังกล่าวเมื่อนำมาพิจารณาใช้กับวงจรฟิลเตอร์แบงก์นั้นพบว่ากลุ่มโครงสร้างฟิลเตอร์ที่สามารถออกแบบโดยเลือกใช้ระเบียบวิธีการใช้ทรัพยากรร่วมกันระหว่างบล็อกก็คือ โครงสร้างแบบ a และโครงสร้างแบบ c เพราะมีฟังก์ชันการทำงานภายในที่เหมือนกัน ดังนั้นสามารถเขียนวงจรที่ใช้หลักการใช้ทรัพยากรร่วมกันระหว่างบล็อกในการออกแบบด้วยรูปที่ 5 โดยเส้นประที่แสดงในภาพประกอบนั้นเป็นการแสดงว่าบล็อกที่อยู่ระหว่างเส้นประนั้นถูกออกแบบด้วยการใช้ทรัพยากรร่วมกันระหว่างบล็อก ตัวอย่างเช่นบล็อก H2 และ H4 ใช้ทรัพยากรที่มีโครงสร้างแบบ a ร่วมกัน โดยการประมวลผลของ H4 จะเริ่มได้ก็ต่อเมื่อการประมวลผลบล็อก H2 ใช้งาน a เสร็จสมบูรณ์ก่อน

2. การใช้ทรัพยากรร่วมภายในบล็อก (Intra-Block Sharing)

การใช้ทรัพยากรร่วมภายในบล็อก (Intra-Block Sharing) คือการใช้ทรัพยากรร่วมกันภายในบล็อกทั้งนี้เพื่อหลีกเลี่ยงความซับซ้อนของการเชื่อมต่อสัญญาณภายในชิพและบล็อกที่มีการใช้ทรัพยากรแบบ Inter-Block Sharing ไปแล้วจะไม่นำมาพิจารณาการออกแบบด้วยระเบียบวิธี Intra-Block Sharing อีก ดังนั้นสำหรับวงจรฟิลเตอร์แบงก์ภายในงานวิจัยนี้จะมีเพียงบล็อกโครงสร้างแบบ b และแบบ d เท่านั้นที่สามารถจะนำมาพิจารณาการทำ Intra-Block Sharing ได้

พิจารณารูปที่ 5(b) ซึ่งคล้ายคลึงกับวิธีของ Inter-Block Sharing และเมื่อพิจารณาโครงสร้างของ H3 พบว่าจะมีวงจรบวก 3 ตัว คือ A1, A2 และ A3, วงจรคูณ 2 ตัวคือ M1 และ M2 ที่จะถูกนำมาออกแบบโดยการใช้ทรัพยากรร่วมภายในบล็อก

วิธีการออกแบบด้วยระเบียบวิธีดังกล่าวนี้นอกจากการพิจารณาโครงสร้างภายในแล้วนั้นจะต้องพิจารณาเวลาที่จะต้องสอดคล้องกับการเริ่มทำงานของวงจรรวมส่วนอัดไปด้วย ดังแสดงในรูปที่ 6 พบว่าเมื่อฟิลเตอร์ H3 ทำงานเสร็จแล้วนั้นเอาท์พุตที่ได้จะถูกนำไปประมวลผลในขั้นตอนต่อไปจะต้องรอให้ฟิลเตอร์ H4 และ H5 ทำงานเสร็จก่อนเพราะฟิลเตอร์ H6 และ H7 นั้นได้ถูกออกแบบด้วยระเบียบวิธีการใช้ทรัพยากรร่วมกันระหว่างบล็อก ดังนั้นจะมีเวลาว่างเกิดขึ้นจึงสามารถพิจารณาการใช้ทรัพยากรร่วมกันสำหรับโครงสร้างแบบ b ได้และสำหรับโครงสร้างแบบ d นั้นจะไม่มีการพิจารณาการใช้ทรัพยากรร่วมกัน เพราะโครงสร้างภายในที่ไม่ซับซ้อน



รูปที่ 6. เวลาที่นำมาพิจารณาสำหรับการออกแบบการใช้ทรัพยากรร่วมกันภายในบล็อกสำหรับวงจรดิจิตอลฟิลเตอร์แบงก์

การออกแบบวงจร

ระเบียบวิธีในการออกแบบที่กล่าวไปข้างต้นทั้งสองแบบจะนำไปใช้เมื่อทำการออกแบบวงจรฟิลเตอร์แบงก์ด้วยโปรแกรม Xilinx เพื่อยืนยันว่าด้วยวิธีดังกล่าวนี้สามารถลดจำนวนทรัพยากรที่ต้องใช้ภายใน FPGA ได้ โดยเริ่มต้นนั้นจะทำการออกแบบวงจรฟิลเตอร์แบงก์ด้วยวิธีทั่วไปแต่ทั้งนี้ในการออกแบบจะคำนึงถึงการใช้ทรัพยากรให้ประหยัดที่สุดและวงจรสามารถทำงานได้ตรงกับวงจรที่ออกแบบด้วยโปรแกรม MATLAB ด้วย

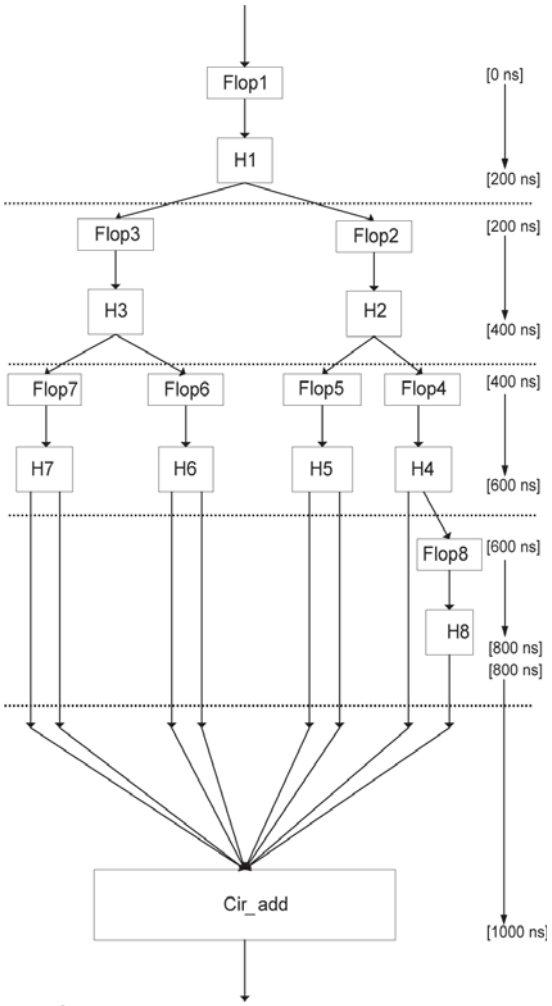
จากรูปที่ 7 สามารถอธิบายลำดับการประมวลผลได้ดังนี้คือเริ่มต้นด้วยการที่มีข้อมูลอินพุตป้อนไปยัง Flop1 ซึ่งเป็นรีจิสเตอร์พักข้อมูลที่มีสัญญาณควบคุมการทำงานคือ enable เมื่อสัญญาณ enable มีค่าเป็น 0 ข้อมูลก็ยังคงถูกพักไว้จนกระทั่งสัญญาณ enable มีค่าเป็น 1 ข้อมูลที่รับเข้ามาใหม่จะถูกส่งออกไปยังส่วนถัดไป ซึ่งในที่นี้คือบล็อกฟิลเตอร์ H1 สำหรับภายใน Flop1 นั้นเป็นวงจรชิฟท์รีจิสเตอร์ (Shift register) ขนาด 32 บิต 49 สเตจ

$$H_1(z):\ y_1(n) = [x(n-0) + x(n-48)] * h_1(0) + [x(n-16) + x(n-32)] * h_1(1) + x(n-24) * h_1(2) \quad (1)$$

เมื่อพิจารณาสมการการประมวลผลของ H1 ดังสมการที่ (1) สมการการทำงานแสดงให้เห็นว่าจะมีการใช้ข้อมูลอินพุตที่ตำแหน่งเวลาก่อนหน้าไป 48 ตำแหน่งด้วย ในการออกแบบจึงออกแบบให้ภายใน Flop1 นั้นบรรจุ flip flop ไว้จำนวน 49 ตัวและจะดึงข้อมูลจากรีจิสเตอร์ในสเตจที่ n-0, n-16, n-24 n-32 และ n-48 ไปใช้ในการคำนวณหาค่าเอาท์พุตของฟิลเตอร์ H1 ด้วยวิธีเดียวกันนี้การออกแบบ Flop ตัวอื่นๆ ในวงจรก็จะนำสมการของฟิลเตอร์ตำแหน่งที่อยู่หลัง Flop นั้นๆ มาพิจารณาเช่น Flop2 ก็จะมี ชิฟท์รีจิสเตอร์ขนาด 25 สเตจ และ Flop3 มี flip flop ชิฟท์รีจิสเตอร์ขนาด 29 สเตจ เป็นต้น จากนั้นในส่วนบล็อกที่เป็น H1 H2 จนถึง H8 ก็คือฟิลเตอร์ตัวต่างๆนั่นเองซึ่งบล็อกเหล่านี้ภายในก็จะประกอบไปด้วยตัวบวก ตัวคูณและตัวลบ จำนวนขององค์ประกอบแต่ละตัวสามารถทราบได้จากสมการของฟิลเตอร์ตัวนั้นๆ เช่น H1 จะประกอบด้วยตัวบวก 4 ตัว ตัวคูณ 3 ตัว และตัวลบ 1 ตัว เป็นต้น ในลำดับสุดท้ายของวงจรก็จะเป็นการนำสัญญาณเอาท์พุตทั้ง 8 ค่าที่ได้มารวมกันอีกครั้งหนึ่งเพื่อเป็นค่าเอาท์พุตของวงจรฟิลเตอร์แบงก์

จากรูปที่ 7 การทำงานของวงจรฟิลเตอร์แบงก์แยกได้เป็น 4 ขั้นเวลา (Time step) จาก 0 -800 นาโนวินาที โดยแต่ละขั้นเวลามีค่าเท่ากับ 200 นาโนวินาทีเป็นเวลาที่ได้จากการสังเคราะห์วงจรด้วยโปรแกรม Xilinx กล่าวคือโปรแกรมจะสังเคราะห์วงจรแล้ววิเคราะห์หาเส้นทางที่ยาวที่สุดระหว่างรีจิสเตอร์ถึงรีจิสเตอร์ เช่น Flop1 - Flop2 หรือ Flop2 - Flop4 เป็นต้น เส้นทางดังกล่าวประกอบด้วยเวลาของวงจรเลขคณิต (เช่น คูณ บวก ลบ ฯลฯ) เวลาของรีจิสเตอร์ซึ่งมีทั้ง write time setup time และยังรวมเวลาของสายเชื่อมต่อด้วย (net delay) ด้วย เส้นทางที่ยาวที่สุดจะถูกเลือกใช้เป็นคาบเวลาของสัญญาณนาฬิกา เพื่อยืนยันว่าโอเปอเรชันในทุก Time step สามารถทำงานได้ถูกต้อง จริงๆ แล้วเวลาได้จากโปรแกรมคือ 189.160 นาโนวินาที แต่ต้องเลือกใช้ 200 นาโนวินาทีเพราะเป็นเวลาที่ DCM (Digital Clock Manager) ภายใน FPGA สามารถสร้างได้

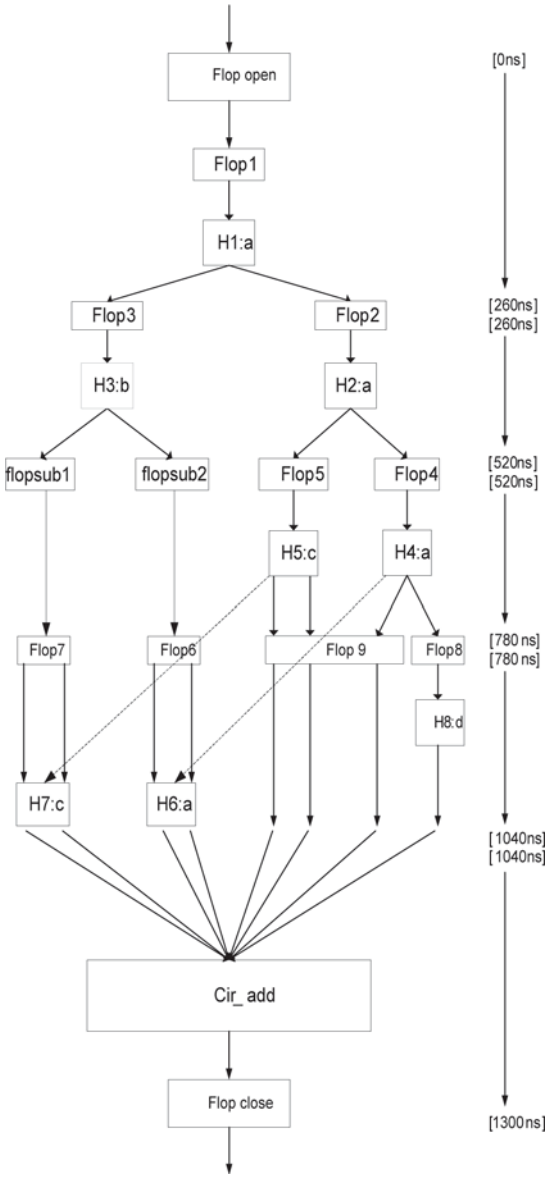
การออกแบบครั้งแรกทำการออกแบบโดยไม่มีการใช้ทรัพยากรร่วมกันดังนั้นการออกแบบวงจรลำดับต่อมาจะออกแบบโดยมีการใช้การใช้ทรัพยากรร่วมกันโดยใช้ระเบียบวิธีการใช้ทรัพยากรร่วมกันระหว่างบล็อกพิจารณาในการออกแบบเพียงวิธีเดียวก่อนซึ่งเมื่อออกแบบวงจรฟิลเตอร์ด้วยวิธีดังที่กล่าวไปนั้นวงจรที่ได้สามารถแสดงด้วยรูปที่ 8 ซึ่งตอนนี้คาบเวลาเพิ่มขึ้นเป็น 260 นาโนวินาที เนื่องมาจากมีการเพิ่มเวลาของมัลติเพล็กเซอร์ (Multiplexer) ที่ใช้สำหรับการเลือกสัญญาณป้อนให้กับบล็อกหรือทรัพยากรที่ถูกใช้ร่วมกัน



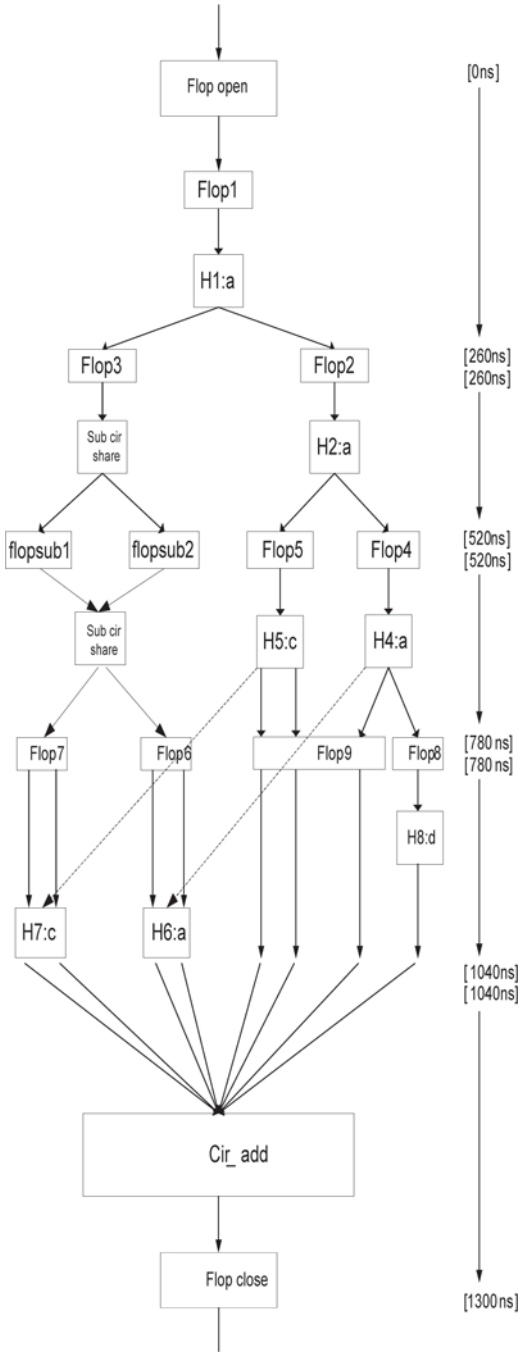
รูปที่ 7. วงจรฟิลเตอร์แบงก์ที่ออกแบบโดยไม่มีการใช้ทรัพยากรร่วมกัน

ต่อจากนั้นในลำดับสุดท้ายก็จะเป็นการนำระเบียบวิธีการใช้ทรัพยากรร่วมกันภายในบล็อกและระหว่างบล็อกมาใช้ในการออกแบบทั้งคู่ซึ่งแสดงวงจรหลังจากออกแบบด้วยรูปที่ 9 ในรูปรีจิสเตอร์ Flop open และ Flop close มีไว้สำหรับเก็บข้อมูลอินพุทและเอาท์พุทตามลำดับ บล็อก Cir_add มีไว้สำหรับรวมสัญญาณที่ผ่านการแยกและคูณน้ำหนักในแต่ละแถบความถี่แล้ว

จากรูปที่ 8 และ 9 สังเกตเห็นว่าเวลาในการทำงานของวงจรฟิลเตอร์แบงก์ยังคงเท่ากัน ทั้งนี้เนื่องจากหลักการการใช้ทรัพยากรร่วมกันภายในบล็อกที่นำเสนอได้คำนึงถึง Time interval ดังรายละเอียดในหัวข้อที่แล้วว่าสามารถทำการใช้ทรัพยากรร่วมกันภายในบล็อกได้หรือไม่



รูปที่ 8. วงจรที่ออกแบบด้วยระเบียบวิธีการใช้ทรัพยากรร่วมระหว่างบล็อก

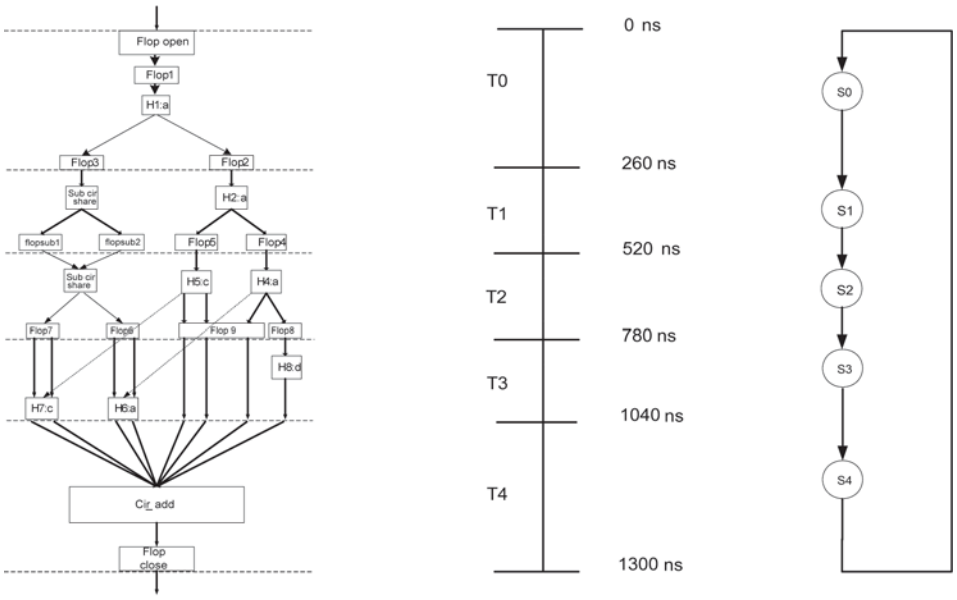


รูปที่ 9. วงจรที่ออกแบบด้วยระเบียบวิธีการใช้ทรัพยากรร่วมระหว่างบล็อกและภายในบล็อก

วงจรควบคุม

วงจรควบคุมแบบ FSM (Finite State Machine) ถูกนำมาใช้ในการควบคุมการทำงานของวงจรดาต้าพาทของวงจรดิจิตอลฟิลเตอร์แบงก์ โดยที่เลือกรูปแบบวงจรควบคุมแบบ Moore Machine ซึ่งเป็นรูปแบบการเขียนสเตตไดอะแกรมแบบค่าของเอาท์พุตในแต่ละสเตตจะขึ้นอยู่กับค่าของสเตตปัจจุบัน (Current State) เท่านั้นมาใช้ในการออกแบบควบคุมวงจรดาต้าพาทสำหรับงานวิจัยนี้ ทั้งนี้ในการออกแบบการทำงานของวงจรควบคุมจะต้องพิจารณาเวลาการทำงานของวงจรดาต้าพาทเป็นหลัก ดังนั้นค่าสัญญาณนาฬิกาที่เลือกใช้เพื่อออกแบบวงจรควบคุมจะต้องสัมพันธ์กับจังหวะเวลาการทำงานของวงจรภายในวงจรดาต้าพาทด้วย

ซึ่งสำหรับงานวิจัยนี้จะนำวงจรดาต้าพาทที่ได้ออกแบบไว้ข้างต้นมาใช้ในการพิจารณาหาค่าสัญญาณนาฬิกาที่เหมาะสมเพื่อการออกแบบวงจรควบคุม ตามวิธีที่แนะนำในหัวข้อก่อนหน้านี้ ซึ่งเริ่มต้นจากพิจารณาลำดับการทำงานของวงจรดาต้าพาทที่ต้องการออกแบบวงจรควบคุมแล้วทำการสร้างลำดับการทำงานของวงจรหรือเรียกว่าสเตปควบคุม (Control Step) ซึ่งทำหน้าที่ควบคุมแต่ละ Time step ของดาต้าพาท และจากสเตปควบคุมสามารถเขียน State ของ FSM ได้ในขั้นตอนสุดท้ายของการออกแบบ โดยรูปที่ 10 จะเป็นการแสดงกระบวนการสร้าง state ของ FSM ที่จะควบคุมวงจรแต่ละแบบ ตามลำดับ ในแต่ละสเตตวงจรควบคุมต้องสร้างสัญญาณควบคุมเช่น สัญญาณเลือกของมัลติ-เพล็กเซอร์เพื่อเลือกสัญญาณอินพุทที่ต้องการให้แต่ละโอเปอเรชัน และสัญญาณอิเนเบิลไปยังรีจิสเตอร์ที่ทำหน้าบันทึกผลลัพธ์จากโอเปอเรชันในแต่ละ Time step ด้วย



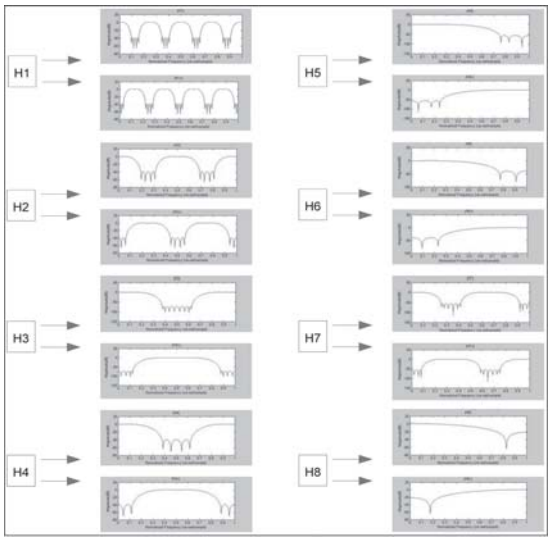
รูปที่ 10. วงจรควบคุมของวงจรดิจิตอลฟิลเตอร์แบงก์ ประกอบด้วย (a) คือ DFG ของวงจรคาต้าพาทพร้อมทั้งเวลาทำงาน (b) ขั้นเวลาในการออกแบบวงจรควบคุมและ (c) คือ วงจรควบคุมแบบ FSM

ผลการทดลอง

จากหลักการออกแบบที่นำเสนอดังหัวข้อที่ผ่านมา เมื่อนำมาออกแบบจริงด้วยโปรแกรม MATLAB และ Xilinx ได้ผลการทดลองดังต่อไปนี้

1. ผลการออกแบบฟิลเตอร์เพื่อหาค่าผลตอบสนองต่ออิมพัลส์ หรือ h(n)

การตรวจสอบดูค่า h(n) ของฟิลเตอร์แต่ละตัวทำได้โดยการนำสมการผลต่างสำหรับใช้หาค่า h(n) ของวงจรฟิลเตอร์แบงก์ดังรูปที่ 2 ไปออกแบบในโปรแกรม MATLAB ทั้งนี้ในระหว่างกระบวนการหาค่า h(n) นั้นจะยึดหลักการตรวจสอบค่าของ h(n) โดยการให้โปรแกรมแสดงกราฟของลักษณะสัญญาณที่ได้จากการออกแบบสำหรับฟิลเตอร์แต่ละตัว รวมไปถึงจะทำการตรวจความถูกต้องของตำแหน่งค่า h(n) ตามตารางที่ 1 ไปด้วย รูปที่ 11 แสดงสัญญาณที่ออกจากฟิลเตอร์ทั้งหมดในฟิลเตอร์แบงก์ ซึ่งถูกต้องตามหลักการในรูปที่ 3



รูปที่ 11. สัญญาณที่ออกจากฟิลเตอร์ทั้งหมดในฟิลเตอร์แบงก์

เมื่อตรวจสอบความถูกต้องของแถบสัญญาณทั้งหมดแล้ว ขั้นตอนต่อไปก็คือการตรวจสอบค่าของสัมประสิทธิ์ h(n) ของฟิลเตอร์ทั้งหมดซึ่งจะถูกเก็บไว้

ในอาร์เรย์ของโปรแกรม MATLAB ซึ่งในการตรวจสอบนั้นจะต้องตรวจสอบตำแหน่งที่ค่า h(n) ถูกเก็บไว้ในอาร์เรย์เปรียบเทียบกับตำแหน่งที่เขียนไว้ในตารางที่ 1 พบว่าเมื่อทำการออกแบบฟิลเตอร์ครบทั้งหมด 8 ตัว และทำการประมวลผลเพื่อหาค่าสัมประสิทธิ์ จากนั้นอ่านค่า h(n) ของฟิลเตอร์จากโปรแกรม MATLAB เมื่อนำค่าที่ได้มาเขียนตารางแสดงผลจะได้ตารางแสดงผลค่า h(n) ของฟิลเตอร์ทั้งหมดดังตารางที่ 2 ทั้งนี้ค่า h(n) ที่ได้นั้นจะแสดงให้เห็นเป็นเลขทศนิยมฐานสิบแต่การออกแบบวงจรฟิลเตอร์บน FPGA หรือบนโปรแกรม Xilinx สำหรับงานวิจัยนี้จะเป็นการประมวลผลแบบเลขทศนิยมตามมาตรฐาน IEEE 754 ขนาด 32 บิต จึงต้องทำการแปลงค่า h(n) ที่ได้นี้ก่อนจะนำไปใช้ในวงจรบนโปรแกรม Xilinx

ตารางที่ 2. FIR Filter Bank Coefficient

Filter	H(n)	Decimal Floating Point
$H_1(z)$	$h_1(0) = h_1(48)$ $h_1(16) = h_1(32)$ $h_1(24)$	-0.05062417842547 0.29505933470299 0.5
$H_2(z)$	$h_2(0) = h_2(24)$ $h_2(8) = h_2(16)$ $h_2(12)$	-0.05062417842547 0.29505933470299 0.5
$H_3(z)$	$h_3(0) = h_3(28)$ $h_3(4) = h_3(24)$ $h_3(8) = h_3(20)$ $h_3(12) = h_3(16)$ $h_3(14)$	-0.00373765573262 0.02056989487010 -0.07232190470689 0.30537047362544 0.5
$H_4(z)$	$h_4(0) = h_4(12)$ $h_4(4) = h_4(8)$ $H_4(6)$	-0.05062417842547 0.29505933470299 0.5
$H_5(z)$	$h_5(0) = h_5(10)$ $h_5(2) = h_5(8)$ $h_5(4) = h_5(6)$ $h_5(5)$	0.01304920555205 -0.06387151210405 0.30161294807561 0.5
$H_6(z)$	$h_6(0) = h_6(6)$ $h_1(2) = h_1(4)$ $h_6(3)$	-0.05062417842547 0.29505933470299 0.5
$H_7(z)$	$h_7(0) = h_7(30)$ $h_7(6) = h_7(24)$ $h_7(12) = h_7(18)$ $h_7(15)$	0.01304920555205 -0.06387151210405 0.30161294807561 0.5
$H_8(z)$	$h_1(0) = h_1(2)$ $h_8(1)$	0.29289321881345 0.5

2. ผลการวิจัยของวงจรดิจิตอลฟิลเตอร์แบงค์ด้วยโปรแกรม Xilinx

เมื่อผ่านการทดสอบการทำงานของวงจรฟิลเตอร์แบงก์บนโปรแกรม MATLAB มาแล้วนั้นในลำดับถัดไปก็คือการออกแบบและทดสอบการทำงานของฟิลเตอร์แบงก์บน FPGAs หรือหมายถึงการออกแบบและทดสอบบนโปรแกรม Xilinx นั่นเอง โดยที่ลำดับการออกแบบจะเป็นลำดับดังนี้คือ เริ่มต้นจากการออกแบบวงจรในรูปแบบธรรมดาคือการออกแบบวงจรแบบเดียวกันกับที่ออกแบบบน MATLAB จากนั้นจะออกแบบวงจรโดยจะใช้ระเบียบวิธีที่นำเสนอในงานวิจัยนี้คือการออกแบบโดยใช้ทรัพยากรคาต้าพาทร่วมกันแบบลำดับชั้น ซึ่งจะออกแบบโดยใช้ระเบียบวิธีการใช้ทรัพยากรร่วมกันระหว่างบล็อกเท่านั้นก่อน ในลำดับสุดท้ายจึงจะใช้ระเบียบวิธีการใช้ทรัพยากรร่วมกันระหว่างบล็อกและการใช้ทรัพยากรร่วมภายในบล็อกมาทำการออกแบบ

งานวิจัยนี้จะออกแบบวงจรและทดสอบบนชิพ FPGA ของบริษัท Xilinx ตระกูล SPARTAN-3 เบอร์ XC3S4000-5FG676 ในการออกแบบจะเป็นการเขียนคำสั่งเพื่อบรรยายวงจรด้วยภาษา VHDL และใช้โปรแกรม Xilinx ISE 8.1i ในการสังเคราะห์ (Synthesis) เพื่อดูผลจากการสังเคราะห์วงจรทั้งด้านการใช้ทรัพยากรบน FPGA และความเร็วในการทำงานของวงจรจากนั้นนำวงจรที่ได้ทำการจำลองการทำงาน (Simulation) เพื่อตรวจสอบผลการทำงานของวงจรแบบต่างๆ ทั้ง 3 แบบที่กล่าวไปข้างต้นเทียบกับคำตอบจากโปรแกรม MATLAB

จากผลการสังเคราะห์และผลการจำลองการทำงานของวงจรสามารถถูกแสดงเป็นตารางเปรียบเทียบกันเพื่อวิเคราะห์ผลการใช้ทรัพยากรของวงจรในรูปแบบต่างๆ ที่ได้ทำการออกแบบโดยแสดงด้วยตารางที่ 3 และการใช้เวลาในการประมวลแสดงดังตารางที่ 4 สังเกตเห็นว่า หลักการใช้ทรัพยากรร่วมกันระหว่างบล็อกช่วยให้ขนาดของวงจรเล็กลง 38% ในขณะที่ความเร็วช้าลง 30% แต่เมื่อนำทั้งหลักการใช้ทรัพยากรร่วมกันระหว่างบล็อกและภายในบล็อกมาใช้พบว่าวงจรมีขนาดเล็กลง 44% ในขณะที่ความเร็วช้าลง 30% เท่าเดิม เหตุผลที่เวลาที่ใช้จากวงจรที่ออกแบบให้ใช้ทรัพยากรร่วมกันระหว่างบล็อกเพียงอย่างเดียว จึงมีค่าเท่ากับเวลาที่ได้จากวงจรที่ออกแบบให้ใช้ทรัพยากรร่วมกันทั้งระหว่างบล็อกและภายในบล็อก คือ ระเบียบวิธีที่นำเสนอได้พิจารณาก่อนแล้วว่ามี Time interval เพียงพอสำหรับการใช้ทรัพยากรร่วมกันภายในบล็อกโดยที่ไม่กระทบกับเวลาการทำงานของวงจรเดิม

ตารางที่ 3. เปรียบเทียบการใช้ทรัพยากรในการออกแบบวงจรฟิลเตอร์แบงก์ทั้ง 3 แบบบน FPGAs SPARTAN-3 เบอร์ XC3S4000FG676-5

วงจรฟิลเตอร์แบงก์	ปริมาณการใช้ LUTs	เทียบกับวงจรแบบไม่มีการใช้ทรัพยากรร่วมกัน
แบบที่ไม่มีการใช้ทรัพยากรร่วมกัน	43476 out of 55296	(เป็นวงจรต้นแบบ)
แบบที่มีการใช้ทรัพยากรร่วมกันระหว่างบล็อก	26591 out of 55296	ลดลง 38%
แบบที่มีการใช้ทรัพยากรร่วมกันระหว่างบล็อกและภายในบล็อก	24285 out of 55296	ลดลง 44%

ตารางที่ 4. เปรียบเทียบการใช้เวลาในการคำนวณผลของวงจรฟิลเตอร์แบงก์ทั้ง 3 แบบบน FPGAs SPARTAN-3 เบอร์ XC3S4000FG676-5

วงจรฟิลเตอร์แบงก์	เวลาที่ใช้ใน 1 รอบของการคำนวณ 1 เอาท์พุต	เทียบกับวงจรแบบไม่มีการใช้ทรัพยากรร่วมกัน
แบบที่ไม่มีการใช้ทรัพยากรร่วมกัน	1000 ns	(เป็นวงจรต้นแบบ)
แบบที่มีการใช้ทรัพยากรร่วมกันระหว่างบล็อก	1300 ns	ลดลง 30%
แบบที่มีการใช้ทรัพยากรร่วมกันระหว่างบล็อก	1300 ns	ลดลง 30%

สรุป

บทความนี้ได้นำเสนอการออกแบบวงจรดิจิตอลฟิลเตอร์แบงก์บนเอฟพีจีเอที่มีพื้นที่จำกัด โดยทำการวิเคราะห์หากลุ่มของฟังก์ชันที่เหมือนกันของวงจรดาต้าพาท เพื่อนำมาออกแบบวงจรดาต้าพาทที่มีการใช้ทรัพยากรร่วมกันเป็นรูปลำดับขั้น โดยมีการคำนึงถึงเวลาที่ช้าลงหลังจากมีการใช้ทรัพยากรร่วมกัน จากนั้นออกแบบวงจรควบคุมและการคำนวณหาสัญญาณนาฬิกาที่เหมาะสมที่สุด ผลการทดลองแสดงให้เห็นการใช้เนื้อที่บน FPGA และการใช้เวลาในการประมวลผลของวงจรดาต้าพาททั้งแบบที่ไม่มีการใช้ทรัพยากรร่วมกัน แบบใช้ทรัพยากรร่วมกันระหว่างบล็อกและแบบที่มีการใช้ทรัพยากรร่วมกันระหว่างบล็อกและภายในบล็อก พบว่าการออกแบบวงจรฟิลเตอร์

แบงก์บน FPGAs ด้วยระเบียบวิธีการใช้ทรัพยากรในการประมวลผลร่วมกันนั้นสามารถลดปริมาณการใช้ทรัพยากรในการออกแบบได้เป็นปริมาณมากแม้ว่าจะต้องชดเชยด้วยเวลาการทำงานที่มากขึ้นแต่ผลจากการประมวลผลของวงจรก็ไม่ได้ผิดเพี้ยนไปจากวงจรที่ประมวลผลด้วยโปรแกรม MATLAB เลย ดังนั้นระเบียบวิธีการใช้ทรัพยากรร่วมกันในการออกแบบวงจรฟิลเตอร์แบงก์ในงานวิจัยนี้จึงเป็นอีกแนวทางที่น่าสนใจเพื่อใช้ในการออกแบบวงจรดิจิตอลขนาดใหญ่

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนจากบัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์

เอกสารอ้างอิง

Memik S.O., Memik G., Jafari R., Kursun E. 2003. Global resource sharing for synthesis of control data flow graphs on FPGAs. **Proceedings of Design Automation Conference 2003**, pp. 604-609.

Jaschke C., Beckmann F., Laur R. 1999. Time constrained module scheduling with global resource sharing. **Proceedings of Design, Automation and Test in Europe Conference and Exhibition 1999**, pp. 210-216.

Qiao S., Hei Y., Wu B., and Zhou Y. 2007. A technique to avoid combination feedback loop and long critical path in resource sharing. **7th International Conference on ASICON**. pp. 1170 - 1173.

Nielsen L.S. and Sparso J. 1999. 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.

Lunner T. and Hellgren J. 1991. A digital filterbank hearing-aid design, implementation and evaluation. **International Conference on Acoustics, Speech, and Signal Processing**, ICASSP-91, vol.5, pp.3661-3664.

Balasingham I., Ramstad T. 2008. Are theWavelet Transforms the Best Filter Banks for Image Compression?. **Journal on Image and Video Processing**. Vol. 8(2).

Bernardini R., Rinaldo R. 2006. Oversampled filter banks from extended perfect reconstruction filter banks. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, Vol. 54(7), pp.2625 - 2635.

Xilinx Corp. http://www.xilinx.com.