วิศวกรรมสาร มข. ปีที่ 34 ฉบับที่ 3 (361 - 373) พฤษภาคม – มิถุนายน 2550
KKU Engineering Journal Vol. 34 No .3 (361 - 373) May – June 2007

การออกแบบวงจรกรองปรับตัว ADALINE โดยหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด *

วุฒิ วิริยะสม[1] ณัฏฐา จินดาเพ็ชร์[2] และ พรชัย พฤกษ์ภัทรานนท์ [2]

[1] นักศึกษาปริญญาโท ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ 90112

[2] ผู้ช่วยศาสตราจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์ 90112

Email: nattha.s@psu.ac.th

บทคัดย่อ

บทความนี้นำเสนอระเบียบวิธีการออกแบบวงจรกรองปรับตัว ADALINE สำหรับงานที่มีข้อจำกัดของพื้นที่โดยอาศัยหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด มีการนำหลักการใช้ทรัพยากรร่วมกันระหว่างโอเปอเรชันต่าง ๆมาช่วยแก้ปัญหาดังกล่าว ซึ่งวงจรประมวลผลทศนิยมทางคณิตศาสตร์ที่ใช้ถูกออกแบบให้เป็นไปป์ไลน์ละเอียดเพื่อเป็นการเพิ่มความเร็วการทำงานของวงจรรวม โดยในการออกแบบต้องมีวงจรควบคุมในรูปของ FSM (Finite State Machine) ที่เหมาะสม จากการทดลองวิเคราะห์เปรียบเทียบประสิทธิภาพของหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดพบว่าวงจรที่ได้ทำงานได้เร็วสมเหตุสมผลภายใต้ทรัพยากรที่จำกัดเมื่อเทียบกับวงจรที่ไม่มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดและวงจรที่มีการใช้ทรัพยากรร่วมกันแต่ไม่เป็นไปป์ไลน์ละเอียด และได้นำหลักการดังกล่าวมาออกแบบวงจรกรองปรับตัว ADALINE บน FPGAs ทำให้ได้วงจรที่สามารถทำงานบน FPGAs Xilinx SPARTAN XC3S200 ได้ปริมาณการทำงาน (Throughput) สูงสุดที่ 8.79 ล้านตัวอย่างต่อวินาที (data samples per second)

คำสำคัญ : การใช้ทรัพยากรร่วมกัน ไปป์ไลน์ละเอียด วงจรประมวลผลสัญญาณดิจิตอล วงจรประมวลผลทศนิยม

* รับต้นฉบับเมื่อวันที่ 31 มกราคม 2550 และได้รับบทความฉบับแก้ไขเมื่อวันที่ 8 พฤษภาคม 2550

Design of an ADALINE Adaptive Filter Circuit Based on Fine-grained Pipeline Resource-sharing*

Wut Wiriyasom[1] Nattha Jindapetch[2] and Pornchai Phukpattranont[2]

[1] Master Student, Department of Electrical Engineering, Faculty of Engineering, Prince of Songkla University 90112

[2] Assistant Professor, Department of Electrical Engineering, Faculty of Engineering, Prince of Songkla University 90112

Email: nattha.s@psu.ac.th

ABSTRACT

This article describes a design methodology for digital signal processing circuits in the design limited by the area constraints. A fine-grained pipeline resource-sharing method is presented. In this method, a technique of resource-sharing between operations is applied to minimize the area, and the well designed fine-grained pipelined floating-point arithmetic circuits are used to increase the speed. The appropriate control circuits, FSMs (Finite State Machines), are also concerned. From the experimental result, the circuit based on fine-grained pipeline resource-sharing can operate reasonably fast under the limited resources compared to the fine-grained pipeline circuit without resource-sharing and the circuit based on coarse-grained pipeline resource-sharing. The proposed method is used to realize ADALINE adaptive filter which can work on FPGA Xilinx SPARTAN XC3S200 with highest throughput of 8.79 Million data-samples per second.

Keywords : resource-sharing, fine-grained pipeline, digital signal processing circuits, floating-point arithmetic circuit

* Original manuscript submitted: January 31, 2006 and Final manuscript received: May 8, 2007

บทนำ

การใช้ทรัพยากรร่วมกัน (Resource-sharing) ระหว่างโอเปอเรชันต่าง ๆ ในวงจรประมวลสัญญาณดิจิตอล ช่วยให้สามารถสร้างวงจรได้บนชิพที่มีพื้นที่จำกัดดังเช่น FPGAs (Field Programmable Gate Arrays) แต่หลังจากการใช้ทรัพยากรร่วมกันวงจรจะทำงานช้าลง ยิ่งมีการใช้ทรัพยากรร่วมกันมากขึ้น วงจรก็จะทำงานช้าลงยิ่งขึ้นเช่นกัน

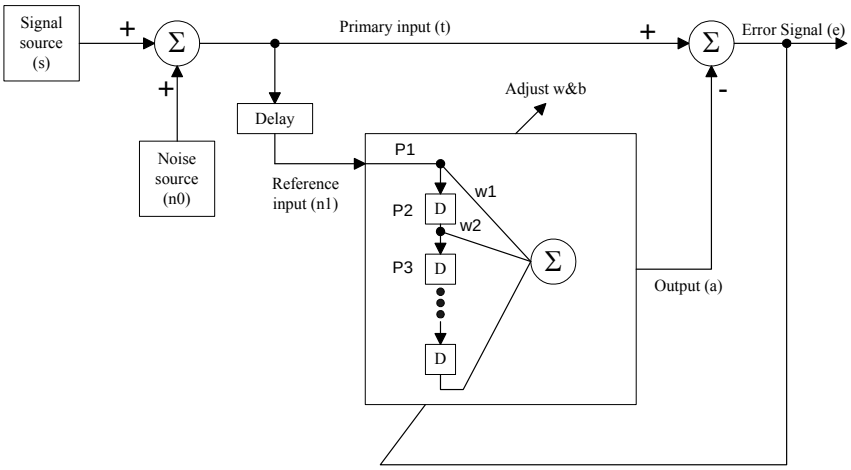
วิธีการทำให้วงจรทำงานได้เร็ววิธีหนึ่งคือ การทำไปป์ไน์ละเอียด (Fine-grained pipeline หรือ micropipeline (Sutherland, 1989) ในวงจรประมวลผลคณิตศาสตร์ เช่น วงจรคูณเลขทศนิยม วงจรบวก/ลบเลขทศนิยม เป็นต้น แต่อย่างไรก็ตามในวงจรไปป์ไลน์ส่วนใหญ่มักจะไม่มีการทำ Resource-sharing เลย ทั้งนี้เนื่องจากวงจรในแต่ละสเตจ (Stage) ของไปป์ไลน์ถูกใช้งานพร้อมกัน มีงานวิจัยของมหาวิทยาลัยแคลิฟอร์เนียได้แสดงให้เห็นว่าการใช้ทรัพยากรร่วมกันในไปป์ไลน์สามารถทำได้และจำเป็นอย่างยิ่งในการออกแบบบนพื้นที่ที่จำกัด (Bakshi *et al*., 1996), (Hsiao-Ping *et al*., 1996), (Chang *et al*., 1996) แต่ก็ยังเป็นเพียงไปป์ไลน์ในระดับหยาบ (Coarse-grained pipeline) (Kim *et al*., 2005) ซึ่งอัลกอริทึมไม่ได้ครอบคลุมถึงการใช้โมดูลที่เป็นไปป์ไลน์ละเอียด

ในบทความนี้ ระเบียบวิธีสำหรับการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด (Fine-grained pipeline resource-sharing) ถูกนำเสนอเพื่อแก้ปัญหาข้างต้น โดยเน้นถึงเทคนิคการออกแบบหน่วยประมวลผลเลขทศนิยมให้มีความเร็วสูงโดยใช้เนื้อที่บน FPGAs (Field Programmable Gate Arrays) ให้มีประสิทธิภาพมากที่สุด และเพื่อแสดงถึงประสิทธิภาพของเทคนิคที่ได้ ในโครงการวิจัยนี้ได้นำเทคนิคดังกล่าวไปประยุกต์ในการออกแบบวงจร ADALINE ซึ่งเป็นกรองปรับตัว (Adaptive Filter) ชนิดหนึ่งซึ่งถูกใช้งานเป็นวงจรกรองสัญญาณรบกวนที่เกิดขึ้นในเครื่องมือวัดสัญญาณไฟฟ้าของร่างกายที่มีประสิทธิภาพมาก (รักกฤตว์, 2544), (สัญญา, 2548) วงจรสามารถจะทำงานได้ถูกต้องโดยวงจรควบคุมที่เหมาะสม ผลการทดลองแสดงให้เห็นว่า วงจรที่ได้สามารถทำงานได้เร็วสมเหตุสมผลภายใต้ทรัพยากรที่มีจำกัด

หลักการ ADALINE Adaptive Filter

ADALINE Adaptive Filter ชนิดไม่ใช้สัญญาณอ้างอิงจากภายนอก

หลักการของ ADALINE Adaptive Filter ชนิดไม่ใช้สัญญาณอ้างอิงจากภายนอกดังรูปที่ 1 ทำได้โดยนำสัญญาณอินพุท $s + n_0$ มาทำการหน่วงเวลา เพื่อสร้างเป็นสัญญาณอ้างอิงแล้วป้อนให้แก่โครงข่ายประสาท ADALINE ซึ่งโครงข่ายประสาทจะทำการเรียนรู้ และสร้างสัญญาณเอาท์พุท a ให้เหมือนสัญญาณรบกวนที่เกิดขึ้นแล้วนำไปหักล้างกับสัญญาณปฐมภูมิ t จะได้สัญญาณความผิดพลาด e และนำสัญญาณความผิดพลาดนี้ทำการปรับค่าน้ำหนักตามหลักการเรียนรู้ของโครงข่าย ADALINE



รูปที่ 1 *หลักการ ADALINE Adaptive Filter ชนิดไม่ใช้สัญญาณอ้างอิงจากภายนอก*

หลักการเรียนรู้ของโครงข่ายประสาท ADALINE สามารถสรุปได้ดังนี้

ขั้นที่ 1 สมมติค่าน้ำหนัก **w**(k), ค่าไบอัส (b) และค่าอัตราการเรียนรู้ (α) เริ่มต้นให้แก่โครงข่ายประสาท

ขั้นที่ 2 ป้อนอินพุท **p**(k) ให้แก่โครงข่าย

ขั้นที่ 3 คำนวณหาค่าเอาท์พุท a(k) ของโครงข่ายประสาท

$$a(k) = f(\mathbf{w}(k)^T\mathbf{p}(k) + b) \quad (1)$$

ขั้นที่ 4 หาค่าความผิดพลาด e(k) จากผลต่างของเอาท์พุทเป้าหมาย t(k) กับเอาท์พุทของโครงข่ายประสาท a(k)

$$e(k) = t(k) - a(k) \quad (2)$$

ขั้นที่ 5 ปรับค่าน้ำหนักและไบอัสของโครงข่ายประสาทใหม่จากสมการ

$$\mathbf{w}(k+1) = \mathbf{w}(k) + 2\alpha\, e(k)\mathbf{p}(k) \quad (3)$$
$$b(k+1) = b(k) + 2\alpha\, e(k) \quad (4)$$

ขั้นที่ 6 นำค่าน้ำหนักและไบอัสที่ได้จากขั้นตอนที่ 2 ไปป้อนให้กับโครงข่าย ADALINE ในรอบใหม่

พบว่าการทำงานของวงจรกรองปรับตัวโดยใช้โครงข่าย ADALINE ชนิดไม่ใช้สัญญาณอ้างอิงจากภายนอก ซึ่งใช้โครงข่ายประสาทชนิด ADALINE ขนาด 8 แท็ปดีเลย์ (Taps Delay) ระยะเวลาในการหน่วงเวลา (Delay) เท่ากับ 10 และอัตราการเรียนรู้ (α) ของโครงข่ายประสาท ADALINE เท่ากับ 0.005 จะสามารถกำจัดสัญญาณรบกวนที่มีลักษณะเป็นคาบได้ดี โดยในการออกแบบวงจรฮาร์ดแวร์สามารถแบ่งวงจรออกเป็นสองส่วนใหญ่ๆ คือวงจร ADALINE ชนิด 8 แท็ปดีเลย์ และวงจรปรับค่าความผิดพลาด ดังแสดงรายละเอียดในหัวข้อการออกแบบวงจร ADALINE Adaptive Filter

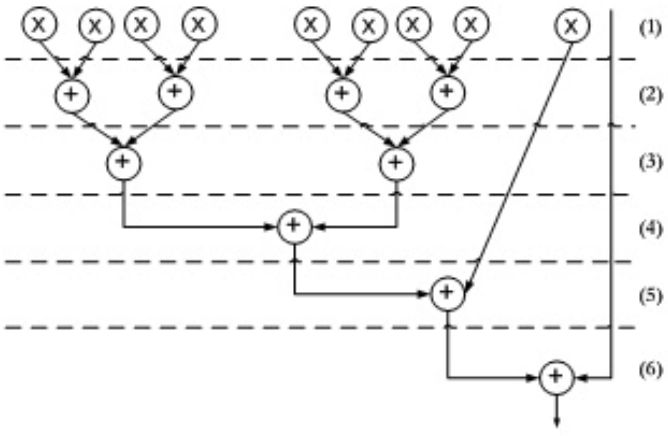
การใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด

การสร้างวงจรประมวลผลสัญญาณทางดิจิตอลบนชิพที่มีพื้นที่จำกัดเช่น FPGAs (Field Programmable Gate Arrays) สามารถแก้ไขปัญหานี้ได้โดยหลักการใช้ทรัพยากรร่วมกันซึ่งอาจทำให้วงจรโดยรวมทำงานได้ช้าลง ยิ่งมีการใช้ทรัพยากรร่วมกันมากวงจรก็ยิ่งจะทำงานได้ช้าลงตามไปด้วย ดังนั้นจึงได้มีการทำไปป์ไลน์ละเอียด (Fine-grained pipeline หรือ micro-pipeline (Sutherland, 1989) ในวงจรประมวลผลทศนิยมเพื่อให้วงจรทำงานได้เร็วขึ้น แต่ในวงจรไปป์ไลน์มักไม่มีการทำ Resource-sharing หรือเท่าที่มีการทำ อัลกอริทึมนั้นก็ไม่ได้ครอบคลุมถึงการใช้โมดูลที่เป็นไปป์ไลน์ละเอียด

ในบทความนี้ระเบียบวิธีสำหรับการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด (Fine-grained pipeline resource-sharing) ถูกนำเสนอเพื่อแก้ปัญหาดังกล่าว ดังตัวอย่างการสร้างวงจร ADALINE ชนิด 8 แท็ปดีเลย์ ในสมการที่ (5)

$$A = (w_1 \times p_1) + (w_2 \times p_2) + \ldots + (w_8 \times p_8) + (w_9 \times p_9) + b \qquad (5)$$

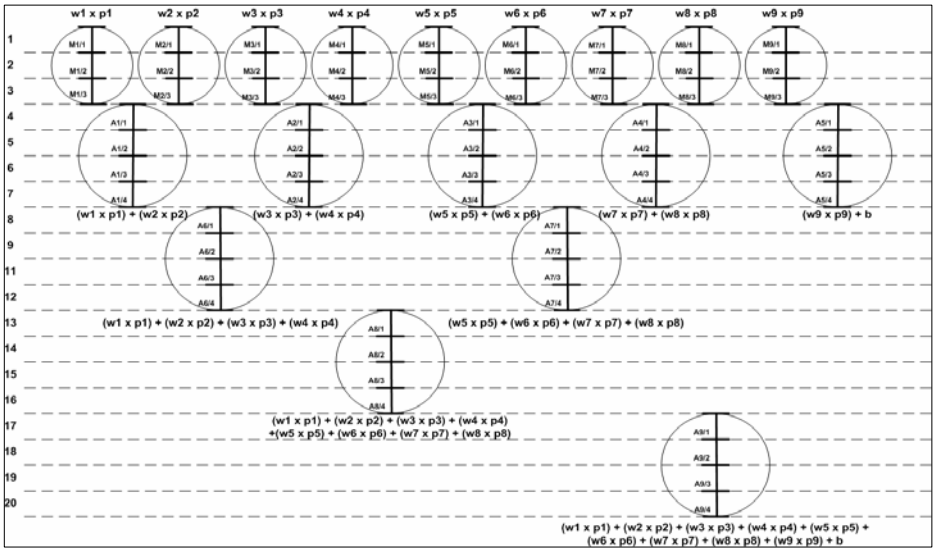
จากสมการที่ (5) กำหนดให้ เอาท์พุท (A) คือ ผลลัพธ์ที่ได้จากสมการที่ (5), w_i คือ ค่าน้ำหนัก (Weight) ตัวที่ i, p_i คือ ค่าอินพุทของโครงข่าย (p) ตัวที่ i และ b คือ ค่าไบอัส (Bias) ตามลำดับ และสามารถแสดงเป็น DFG (Data Flow Graph) (Micheli, 1994) ของการออกแบบวงจรทางดิจิตอลแบบที่ไม่มีการใช้ทรัพยากรร่วมกัน ซึ่งใช้วงจรคูณทศนิยมจำนวน 9 ตัว และวงจรบวกทศนิยมจำนวน 9 ตัว สามารถแบ่งเป็นระดับชั้นของวงจรรวมได้ 6 ระดับ ดังแสดงในรูปที่ 2



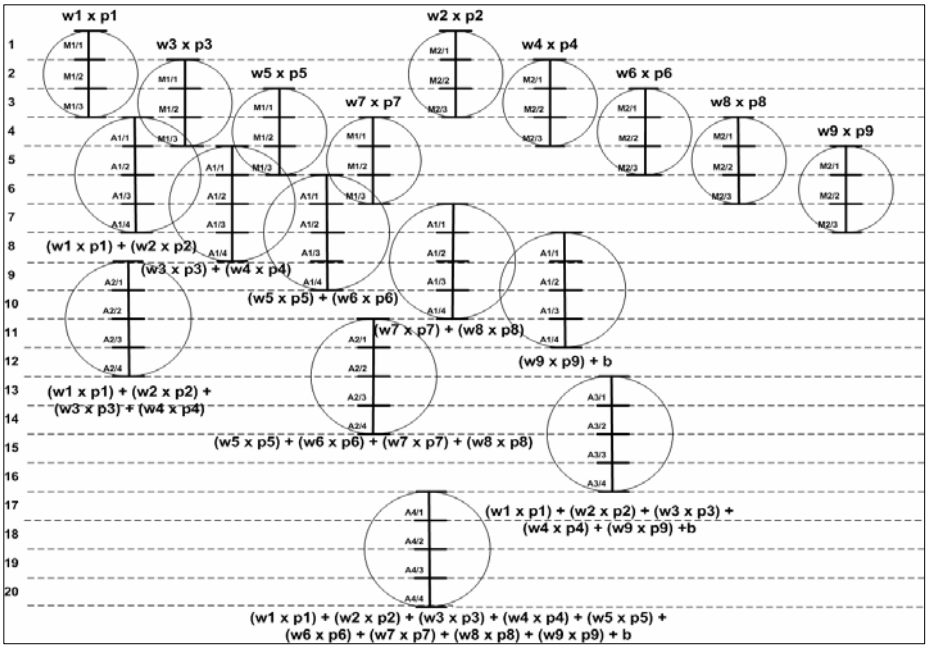
รูปที่ 2 *Data Flow Graph ของวงจรจากสมการที่ (5) แบบไม่ใช้หลักการใช้ทรัพยากรร่วมกัน*

จากการวิเคราะห์ทางด้านเวลาของวงจรรวมในรูปที่ 2 เมื่อแสดงเป็น Timing Diagram พบว่าวงจรรวมที่ออกแบบจากวงจรคูณทศนิยมแบบไปป์ไลน์ 3 สเตจและวงจรบวกทศนิยมแบบไปป์ไลน์ 4 สเตจ สามารถที่จะทำงานเป็นไปป์ไลน์ละเอียดได้อย่างปกติ ใช้เวลาในการทำงานตั้งแต่เริ่มต้นจนเสร็จสิ้น 20 วงรอบสัญญาณนาฬิกา (Clock Cycles) ดังรูปที่ 3 แต่สังเกตว่าวงจรรวมแบบไปป์ไลน์ละเอียดที่

ไม่มีการใช้ทรัพยากรร่วมกัน เมื่อคำนึงถึงเรื่องพื้นที่ (Area) ของวงจรรวม (วงจรคูณทศนิยม 9 ตัว, วงจรบวกทศนิยม 9 ตัว) จะเห็นได้ว่าใช้พื้นที่จำนวนมาก ซึ่งจะมีปัญหาในกรณีที่สร้างวงจรลงบนชิพที่มีพื้นที่อย่างจำกัด



รูปที่ 3 *Timing Diagram ของวงจรรวมแบบไปป์ไลน์ละเอียดของรูปที่ 2 ที่ไม่มีการใช้ทรัพยากรร่วมกัน*



รูปที่ 4 *Timing Diagram ของวงจรรวมแบบไปป์ไลน์ละเอียดของรูปที่ 2 ที่มีการใช้ทรัพยากรร่วมกัน*

จากรูปที่ 3 และ รูปที่ 4 Mx/y และ Ax/y มีความหมายดังนี้

M คือ วงจรคูณทศนิยม

A คือ วงจรบวกทศนิยม

x คือ วงจรประมวลผลทศนิยมตัวที่ x และ

y คือ สเตจของไปป์ไลน์

เพื่อแก้ปัญหาการใช้ทรัพยากรมากเกินไป หลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด (Fine-grained pipeline resource-sharing) จึงถูกนำเสนอขึ้นมาเพื่อลดทรัพยากรที่ใช้ในวงจรรูปที่ 3 กฎในการใช้ทรัพยากรร่วมกันมีเพียงข้อเดียวคือ *อนุญาตให้มีการใช้ทรัพยากรร่วมกันระหว่างโอเปอเรชันที่อยู่ในระดับเดียวกันเท่านั้น* ตัวอย่างเช่น โอเปอเรชันการคูณทั้ง 9 โอเปอเรชันในระดับที่ 1 ของรูปที่ 2 สามารถใช้วงจรคูณตัวเดียวกันได้ แต่โอเปอเรชันการบวกในระดับที่ 2 และในระดับที่ 3 ไม่สามารถใช้วงจรบวกตัวเดียวกันได้เนื่องจากจะทำให้คุณสมบัติการทำงานพร้อมกันของไปป์ไลน์หมดไป

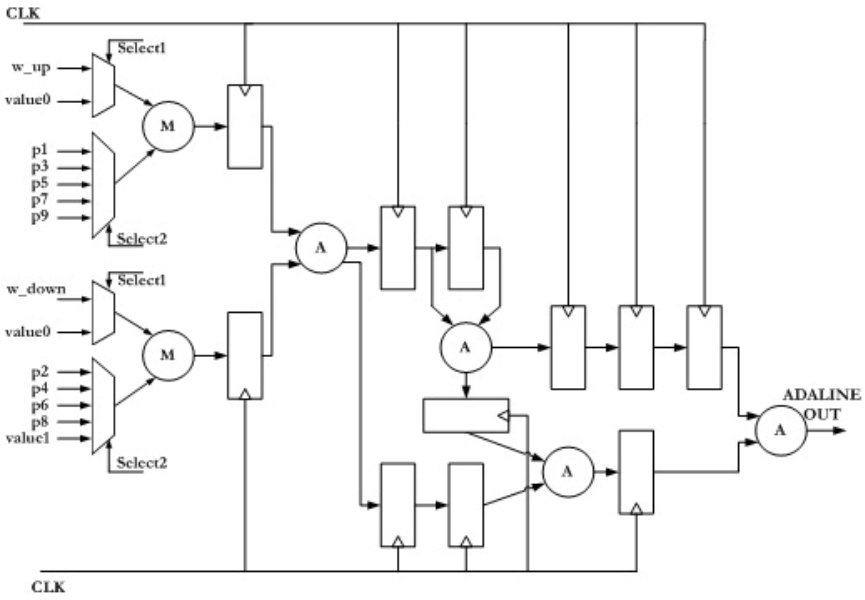
รูปที่ 4 แสดงวงจรรวมแบบไปป์ไลน์ละเอียดของรูปที่ 2 ที่มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด มีการใช้วงจรคูณทศนิยมจำนวน 2 ตัว และวงจรบวกทศนิยมจำนวน 4 ตัว โดยที่วงจรรวมสามารถทำงานได้ผลลัพธ์ถูกต้องเหมือนกับแบบที่ไม่ใช้ทรัพยากรร่วมกันเลย และยังคงทำงานเป็นแบบไปป์ไลน์ละเอียดได้ เมื่อวิเคราะห์ Timing Diagram ดังรูปที่ 4 พบว่าจะใช้เวลาเท่ากับแบบที่ไม่มีการใช้ทรัพยากรร่วมกันในรูปที่ 3

จะเห็นว่าพื้นที่ของวงจรรวมใน รูปที่ 3 เปรียบเทียบกับวงจรรวมในรูปที่ 4 ถูกลดลงมา 4.5 เท่าสำหรับวงจรคูณ และ 2.25 เท่าสำหรับวงจรบวก ในขณะที่เวลาในการประมวลผลข้อมูล 1 ข้อมูล หรือ latency ของวงจรรวมในรูปที่ 4 เท่ากันกับค่าที่ได้จากวงจรรวมดัง Timing Diagram ในรูปที่ 3 แสดงให้เห็นว่าการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดมีประโยชน์มาก แต่อย่างไรก็ตามความเร็วหรือ ค่าปริมาณการทำงาน (Throughput) ของวงจรในรูปที่ 4 จะลดลง 4 เท่า เนื่องจากโอเปอเรชันของการคูณต้องรอให้วงจรคูณว่างก่อน ดังนั้นการเลือกระหว่างความเร็วและพื้นที่ (speed-area tradeoff) ขึ้นกับข้อจำกัดในการออกแบบ (design constraints) ที่กำหนดโดยผู้ออกแบบ

การออกแบบวงจรกรองปรับตัว ADALINE

วงจร ADALINE ชนิด 8 แท็ปดีเลย์

เมื่อทำการออกแบบวงจร ADALINE ชนิด 8 แท็ปดีเลย์ตามหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด (Fine-Grained Pipeline Resource-Sharing) ที่ได้อธิบายในหัวข้อที่แล้ว โดยใช้การประกอบกันขององค์ประกอบต่างๆ ทั้งฟังก์ชันประมวลผลทศนิยมและรีจิสเตอร์ให้ยังคงรักษาความเป็นไปป์ไลน์ละเอียด สรุปว่าจะเลือกใช้วงจรประมวลผลทศนิยมจำนวน 6 ตัว (วงจรคูณทศนิยมจำนวน 2 ตัว และวงจรบวกทศนิยมจำนวน 4 ตัว) และรีจิสเตอร์อีก 11 ตัว เพื่อสร้างเป็นวงจร ADALINE ชนิด 8 แท็ปดีเลย์ของ ADALINE Adaptive Filter ชนิดไม่ใช้สัญญาณอ้างอิงจากภายนอกดังแสดงในรูปที่ 5



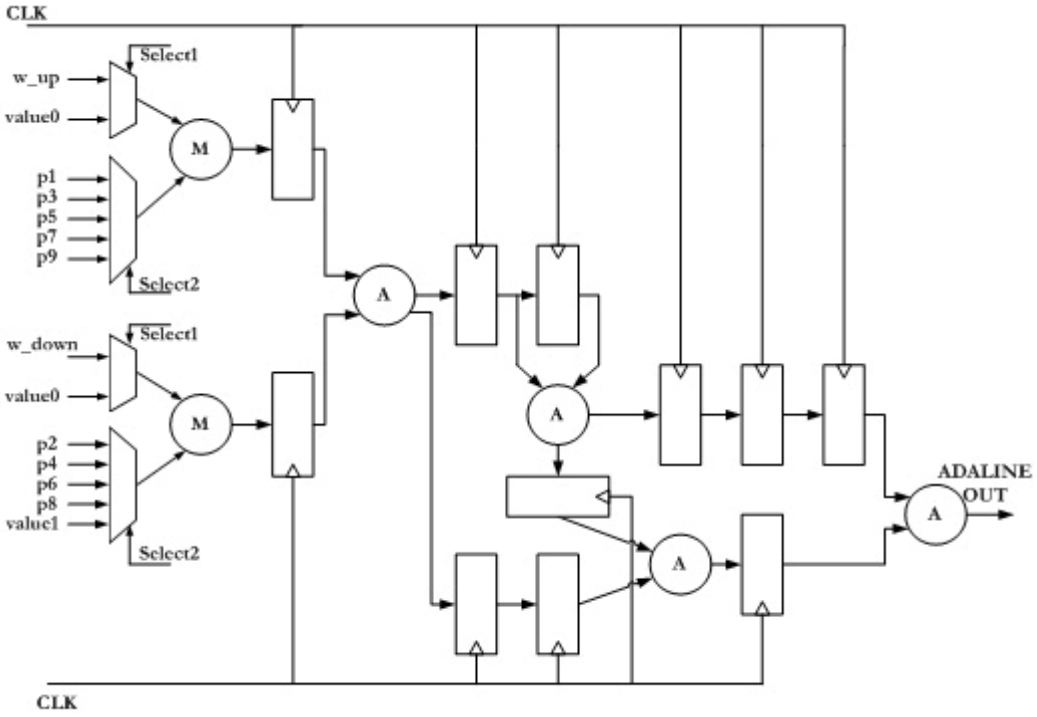
รูปที่ 5 *วงจร ADALINE ชนิด 8 แท็ปดีเลย์ โดยหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด*

จากรูปที่ 5 วงจร ADALINE จะมีอินพุท 2 ประเภทด้วยกันคือ อินพุทที่ไม่ได้มาจากวงจรควบคุม เช่น สัญญาณ p1 - p9, value1 เป็นต้น และอินพุทที่มาจากวงจรควบคุมคือ Select1 และ Select2 โดยที่สัญญาณทั้งสองนี้จะสัมพันธ์กับเอาท์พุทของวงจรควบคุม

วงจรปรับค่าความผิดพลาดของวงจรกรองปรับตัว ADALINE

การออกแบบวงจรปรับค่าความผิดพลาด ตามสมการที่ (3) และ (4) แบบที่ไม่คำนึงถึงปัญหาด้านการใช้ทรัพยากรบน FPGAs นั้นจะใช้วงจรคูณทศนิยมและวงจรบวกทศนิยมอย่างละจำนวน 10 ตัวเท่ากัน แต่เนื่องจากปัญหาข้อจำกัดทางด้านทรัพยากรจึงต้องออกแบบวงจรใหม่ให้สามารถใช้ทรัพยากรได้อย่างเพียงพอ โดยในการวิจัยได้เลือกใช้วงจรคูณทศนิยมจำนวน 2 ตัว และวงจรบวกทศนิยมจำนวน 2 ตัว ดังรูปที่ 6

จากรูปที่ 6 พบว่าวงจรปรับค่าความผิดพลาดก็มีลักษณะการทำงานของวงจรใกล้เคียงกับวงจร ADALINE อีกทั้งยังมีอินพุททั้งสองประเภทเช่นเดียวกัน จะแตกต่างกันตรงที่อินพุทแบบประเภทที่มาจากวงจรควบคุมมีจำนวนสัญญาณมากกว่า เนื่องจากความซับซ้อนของวงจรที่มีมากกว่าด้วย ดังในรูปสัญญาณเหล่านี้คือ Enable2 - Enable4 และ Select3 - Select7

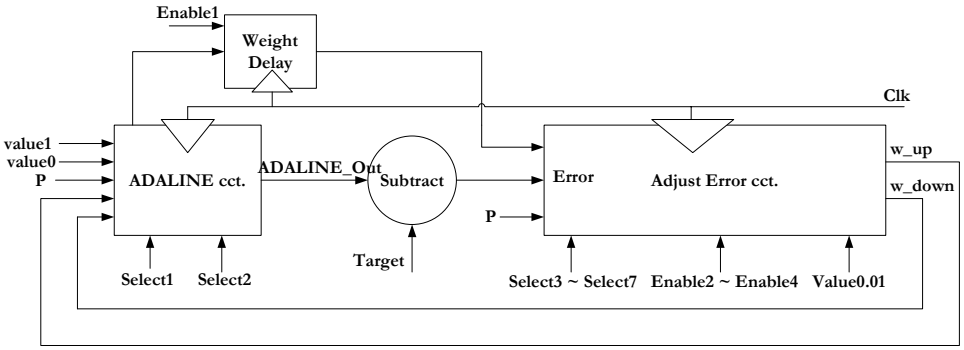


รูปที่ 6 *วงจรปรับค่าความผิดพลาดของวงจร ADALINE รูปที่ โดยหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด*

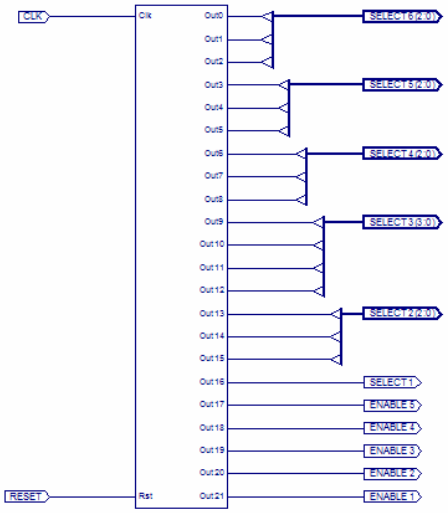
วงจรควบคุมของ ADALINE Adaptive Filter

วงจรควบคุมที่ใช้อยู่ในรูปของ Moore Finite State Machine (Moore FSM) ซึ่งจะเป็นการเขียนสเตตไดอะแกรมรูปแบบที่ค่าของเอาท์พุตในแต่ละสเตตจะขึ้นอยู่กับค่าของสเตตปัจจุบันเท่านั้น จากหลักการข้างต้นเมื่อนำมาควบคุมวงจร ADALINE Adaptive Filter ซึ่งออกแบบโดยหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด (Fine-grained Pipeline Resource-sharing) ในรูปที่ 7 จะต้องมีสัญญาณเอาท์พุทจำนวน 11 สัญญาณด้วยกัน แบ่งออกเป็นสัญญาณ Select ที่ควบคุมมัลติเพล็กเซอร์จำนวน 7 สัญญาณ และสัญญาณ Enable เพื่อควบคุมรีจิสเตอร์อีก 4 สัญญาณ ตามลำดับ โดยวงจรควบคุมนี้มีอินพุทเดียวคือสัญญาณ Clock ของวงจรเท่านั้น

วงจรควบคุมชนิด Moore FSM ของวงจร ADALINE Adaptive Filter นี้มีทั้งหมด 43 สเตต เนื่องจากวงจรส่วนข้อมูลได้ออกแบบเป็นไปป์ไลน์ละเอียดถึง 39 สเตจไปป์ไลน์ รวมถึงต้องพิจารณาในกรณีการวนกลับของค่าน้ำหนักอีกด้วย โดยสัญญาณเอาท์พุททั้งหมดในแต่ละสเตตจะแตกต่างกันไป ดังแสดงเป็นบล็อกไดอะแกรม รูปที่ 8



รูปที่ 7 *วงจร ADALINE Adaptive Filter ที่มีหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด*



รูปที่ 8 *บล็อกไดอะแกรมวงจรควบคุม ADALINE Adaptive Filter*

ผลการทดลอง

การวิเคราะห์ประสิทธิภาพของหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด

ในการทดลองใช้ภาษา VHDL เพื่อบรรยายลักษณะของวงจร การสังเคราะห์และการสร้างวงจรใช้ FPGAs ของ Xilinx รุ่น SPARTAN XC3S200 ซึ่งมีข้อจำกัดด้านพื้นที่ 3,840 LUTs เมื่อนำวงจรเป้าหมายที่ออกแบบไว้ทั้งวงจรคูณทศนิยมและวงจรบวกทศนิยมสร้างวงจร ADALINE ชนิด 8 แท็ปดีเลย์ ตามสมการที่ (5) โดยใช้หลักการใช้ทรัพยากรร่วมกันของไปป์ไลน์ละเอียด เปรียบเทียบกับการสร้างโดยไม่ใช้หลักการใช้ทรัพยากรร่วมกัน อีกทั้งเปรียบเทียบกับวงจรที่ใช้ทรัพยากรร่วมกันแต่ไม่เป็นไปป์ไลน์ละเอียด (ไปป์ไลน์หยาบ) ด้วย ซึ่งได้ผลการทดลองดังตารางที่ 1 (วงจรที่เป็นไปป์ไลน์ละเอียดทั้งสองจะมีความเร็วสัญญาณนาฬิกาในไปป์ไลน์ละเอียดเท่ากัน)

วงจรรวม	ระยะเวลาประมวลผล (นาโนวินาที)	พื้นที่ (LUTs)	ปริมาณการทำงาน (ล้านตัวอย่าง/วินาที)
ไม่มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด*	216.86	5,453	97.253
มีการใช้ทรัพยากรร่วมกันแต่ไม่เป็นไปป์ไลน์ละเอียด	230.168	2,454	17.379
มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด*	216.86	2,218	24.313

ตารางที่ 1 *ผลการวัดประสิทธิภาพของวงจรที่สังเคราะห์จากสมการที่ (5)*

(*หมายเหตุ วงจรประมวลผลทศนิยมที่ใช้ในไปป์ไลน์ละเอียดมีรายละเอียดทางเทคนิคดังนี้คือ

- วงจรคูณทศนิยมมีความเร็วสัญญาณนาฬิกา 10.284 นาโนวินาที (ns), พื้นที่ 125 LUTs และปริมาณการทำงาน 97.23 ล้านตัวอย่างต่อวินาที (data-samples/s)
- วงจรบวกทศนิยมมีความเร็วสัญญาณนาฬิกา 8.622 นาโนวินาที, พื้นที่ 492 LUTs และปริมาณการทำงาน 115.98 ล้านตัวอย่างต่อวินาที)

จากตารางที่ 1 สังเกตได้จากผลการทดลองค่าปริมาณการทำงาน (Throughput) ของวงจรแบบที่ใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด ที่ได้จากการสังเคราะห์เมื่อเปรียบเทียบกับวงจรแบบที่มีการใช้ทรัพยากรร่วมกันแต่ไม่ได้เป็นไปป์ไลน์ละเอียด (ไปป์ไลน์หยาบ) จะเพิ่มขึ้น 1.40 เท่า ในขณะที่พื้นที่มีขนาดใกล้เคียงกัน จึงถือได้ว่าหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดช่วยเพิ่มความเร็วให้กับวงจรได้อย่างมีประสิทธิภาพมากขึ้น

นอกจากนี้ยังพบว่าปริมาณการทำงาน (Throughput) ของวงจรรวมแบบที่มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดจะมีค่าน้อยกว่าแบบที่ไม่มีการใช้ทรัพยากรร่วมกันเท่ากับ 72.940 ล้านตัวอย่างต่อวินาที ซึ่งจะลดลงคิดเป็นประมาณ 4 เท่า แต่ในขณะเดียวกันเมื่อเปรียบเทียบพื้นที่การใช้งานบน FPGAs แบบที่ไม่มีการใช้ทรัพยากรร่วมกันจะใช้พื้นที่สูงกว่าเท่ากับ 3,235 LUTs คิดเป็นประมาณ 2.46 เท่า ของวงจรแบบที่ใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด และเมื่อคำนึงถึงพื้นที่การใช้ทรัพยากรบน FPGAs ที่มีอยู่จำกัดจะเห็นได้ว่าถึงแม้วงจรที่ไม่มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดจะให้ Throughput ที่สูงกว่าประมาณ 4 เท่าของวงจรที่ออกแบบ แต่ก็จะไม่สามารถสร้างได้บน FPGAs ที่มีขนาดเล็กเช่น SPATAN XC3S200 ดังการทดลอง เป็นต้น

ผลการสังเคราะห์วงจร ADALINE Adaptive Filter

การสังเคราะห์วงจร ADALINE Adaptive Filter ให้เป็นวงจรดิจิตอลบน FPGAs นั้นจำเป็นต้องออกแบบในรูปของฮาร์ดแวร์ แยกเป็นสองส่วนประกอบใหญ่ ๆ คือ วงจรโครงข่ายประสาท ADALINE ชนิด 8 แท็ปดีเลย์และวงจรปรับค่าความผิดพลาด ดังที่กล่าวไปแล้ว นอกจากนี้จากรูปที่ 7 เมื่อนำวงจร ADALINE ชนิด 8 แท็ปดีเลย์ มาทำการรวมกับวงจรปรับค่าความผิดพลาด แล้วเพื่อให้เป็นวงจรกรองปรับตัวแบบโครงข่าย ADALINE โดยยังคงให้วงจรรวมสามารถที่จะยังคงทำงานแบบไปป์ไลน์ละเอียด แต่ในการวิจัยนี้ต้องการให้เป็นวงจรกรองปรับตัวที่ไม่มีสัญญาณอ้างอิงจากภายนอกดังนั้น

จึงต้องเพิ่มวงจรการดีเลย์ 10 ตัวอย่างข้อมูล (Data-samples) ของสัญญาณ Electromyography (EMG) เข้าไป และวงจรแท็ปดีเลย์เพื่อสร้างอินพุทเพื่อที่จะป้อนให้กับวงจร ADALINE เข้าไปซึ่งผลการวัดประสิทธิภาพของวงจรที่สังเคราะห์แสดงในตารางที่ 2

วงจร	ระยะเวลาประมวลผล (นาโนวินาที)	พื้นที่ (LUTs)	ปริมาณการทำงาน (ล้านตัวอย่าง/วินาที)
ADALINE 8 แท็ปดีเลย์	216.86	2,218	24.313
ปรับค่าความผิดพลาด	195.42	1,246	8.79
วงจรกรองปรับตัว ADALINE	555.95	4,754	8.79

ตารางที่ 2 *ผลการวัดประสิทธิภาพจากการสังเคราะห์วงจร ADALINE Adaptive Filter ที่ไม่ใช้สัญญาณอ้างอิงจากภายนอก*

จากตารางที่ 2 วงจร ADALINE ชนิด 8 แท็ปดีเลย์ ซึ่งออกแบบตามหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด เลือกใช้วงจรคูณทศนิยมจำนวน 2 ตัว และวงจรบวกทศนิยมจำนวน 4 ตัว ได้ผลเป็นดังนี้ คือ ระยะเวลาประมวลผล (Latency) 216.86 นาโนวินาที (ns) พื้นที่ (Area) 2,218 LUTs และ ปริมาณการทำงาน (Throughput) 24.313 ล้านตัวอย่างต่อวินาที (data-samples per second) ในขณะที่วงจรปรับค่าความผิดพลาด ก็จำเป็นต้องออกแบบโดยใช้หลักการเช่นเดียวกับการออกแบบวงจร ADALINE เนื่องจากสาเหตุของข้อจำกัดด้านทรัพยากรของ FPGAs โดยวงจรปรับค่าความผิดพลาดที่ออกแบบในการวิจัยนี้ เลือกใช้วงจรคูณทศนิยมจำนวน 2 ตัว และวงจรบวกทศนิยมจำนวน 2 ตัว เช่นเดียวกัน ได้ผลจากการสังเคราะห์วงจรเป็นดังนี้คือ ระยะเวลาประมวลผล 195.42 นาโนวินาที พื้นที่ 2,318 LUTs และ ปริมาณการทำงาน 8.79 ล้านตัวอย่างต่อวินาที ตามลำดับ

จากการจำลองการทำงานวงจร ADALINE Adaptive Filter รวมทั้งหมดสามารถทำงานได้เร็วสูงสุดที่ค่าปริมาณการทำงานเท่ากับ 8.79 ล้านตัวอย่างต่อวินาที

สรุป

บทความนี้ได้เริ่มต้นนำเสนอการออกแบบวงจร ADALINE ชนิด 8 แท็ปดีเลย์ ซึ่งเป็นส่วนประกอบของวงจรกรองปรับตัว ADALINE ชนิดไม่ใช้สัญญาณอ้างอิงภายนอก ออกแบบโดยหลักการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียด (Fine-grained Pipeline Reource-sharing) เพื่อลดปัญหาการข้อจำกัดของพื้นที่บน FPGAs ผลการทดลองแสดงให้เห็นว่าวงจรที่ได้ออกแบบเพื่อการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดจะลดการใช้พื้นที่บน FPGAs ลง เมื่อเปรียบเทียบกับแบบที่ไม่มีการใช้ทรัพยากรร่วมกัน แต่ขณะเดียวกันยังคงรักษาการทำงานของวงจรเป็นแบบไปป์ไลน์ละเอียดได้เช่นเดียวกับการออกแบบที่ไม่มีการใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดด้วย เพียงแต่อาจจะสูญเสียปริมาณการทำงาน (Throughput) ลดลงไปบ้าง แต่ก็ยังถือว่าทำงานได้เร็วสมเหตุสมผลภายใต้ทรัพยากร

ที่จำกัด และเมื่อนำหลักการนี้ออกแบบเป็นวงจรกรองแบบปรับตัว ADALINE โดยวงจรปรับค่าความผิดพลาดที่ใช้ก็เป็นการออกแบบที่ใช้ทรัพยากรร่วมกันในไปป์ไลน์ละเอียดเช่นเดียวกัน ได้ผลการสังเคราะห์ว่าสามารถทำงานได้ปริมาณการทำงานสูงสุด 8.79 ล้านตัวอย่างต่อวินาที

กิตติกรรมประกาศ

งานวิจัยนี้ได้รับการสนับสนุนจากบัณฑิตวิทยาลัย มหาวิทยาลัยสงขลานครินทร์

เอกสารอ้างอิง

รักกฤตว์ ดวงทองสร้อย. 2544. “การประยุกต์ใช้โครงข่ายประสาทเพื่อลดสัญญาณรบกวนที่เกิดจากการวัดสัญญาณ Somatosensory Evoked Potentials และสัญญาณไฟฟ้าของกล้ามเนื้อลาย.” **วิทยานิพนธ์** วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์.

วุฒิ วิริยะสม ณัฏฐา จินดาเพ็ชร์ และพรชัย พฤกษ์ภัทรานนท์. 2549. “การออกแบบวงจรประมวลผลสัญญาณดิจิตอลโดยหลักการใช้ทรัพยากรร่วมกันของไปป์ไลน์ละเอียด.” **การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่29**, หน้า 1037-1040.

สัญญา ผาสุก. 2548. “การประยุกต์ใช้ไมโครคอนโทรลเลอร์เป็นโครงข่ายประสาทเพื่อลดสัญญาณรบกวน.” **วิทยานิพนธ์** วิศวกรรมศาสตรมหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยสงขลานครินทร์.

Sutherland I.E. 1989. “Micropipelines.” **Communications of the ACM**. 32 (6): 720 – 738.

Bakshi S., Gajski D.D., and Hsiao-Ping J. 1996. “Component Selection in Resource Shared and Pipelined DSP Applications.” **Proceedings EURO-DAC '96**, European Design Automation Conference, with EURO-VHDL '96 and Exhibition. Page(s):370 – 375.

Hsiao-Ping J., Gajski D.D., and Bakshi S. 1996. “Clock Optimization for High-performance Pipelined Design.” **Proceedings EURO-DAC '96**, European Design Automation Conference, with EURO-VHDL '96 and Exhibition. Page(s):330 – 335.

Chang E., Gajski D.D., and Narayan S. 1996. “An Optimal Clock Period Selection Method Based on Slack Minimization Criteria.” **ACM Transactions on Design Automation of Electronic Systems**. 1(3): 352 – 370.

Kim Y., Kiemb M., Park C., Jung j., and Choi K. 2005. “Resource Sharing and Pipelining in Coarse-Grained Reconfigurable Architecture for Domain-specific Optimization.” **Proceedings Design, Automation and Test in Europe 2005**. 1:12 – 17.

Micheli G. 1994. **Synthesis and Optimization of Digital Circuits**. McGraw-Hill.

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