บทที่ 1
บทนำ

1.1 ชื่อโครงการวิจัย

ภาษาไทย : โพรเซสเซอร์ร่วมกำลังไฟต่ำสามารถปรับเปลี่ยนโครงสร้างได้แบบไดนามิก

ภาษาอังกฤษ : Low Power Coprocessor using Dynamically Reconfiguration

1.2 สาขาวิชาที่ทำการวิจัย : วิศวกรรมไฟฟ้า เกี่ยวกับการออกแบบวงจรรวมดิจิตอล การออกแบบโพรเซสเซอร์ การออกแบบวงจรพลังงานต่ำ

1.3 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

ปัจจุบันระบบสมองกลฝังตัวได้ถูกนำมาใช้งานกันอย่างแพร่หลายเนื่องมีขนาดเล็ก สะดวกต่อการใช้งานและช่วยลดการใช้พลังงาน ระบบสมองกลฝังตัวได้เริ่มพัฒนาจากระดับเล็กโดยการใช้งานไมโครคอนโทรลเลอร์ขนาด 8 บิตควบคุมอุปกรณ์อิเล็กทรอนิกส์ จากนั้นได้มีการพัฒนาอย่างต่อเนื่องเรื่อยมาจนกระทั่งมีการนำโพรเซสเซอร์ขนาดใหญ่ 16 – 32 บิต เข้ามาใช้งานโดยเฉพาะอย่างยิ่งกับงานที่จะต้องใช้การประมวลผลอย่างหนักเช่น โทรศัพท์มือถือรุ่นที่ 3 (3G) คอมพิวเตอร์พกพาขนาดเล็ก เครื่องเล่นเกมส์ Play Station (PS) เป็นต้น จะพบได้ว่าระบบสมองกลฝังตัวในปัจจุบันต้องรับภาระงานหนักมากขึ้น โดยเฉพาะอย่างยิ่งในการทำงานด้าน Multimedia applications หรือ Recognition applications เป็นต้น ซึ่งงานทางด้านนี้เกี่ยวข้องกับการประมวลผลสัญญาณดิจิตอล (Digital Signal Processing) ที่จะต้องใช้หน่วยประมวลผล (Functional Unit; FU) ทำการคำนวณอยู่ตลอดเวลา ดังนั้นจำเป็นที่ระบบจะต้องการความรวดเร็วในการทำงาน โครงสร้างของโพรเซสเซอร์ทั่วไปจึงไม่เหมาะสำหรับการใช้งานประเภทนี้ นอกจากนี้แล้วระบบสมองกลฝังตัวจะมีขนาดเล็กและสามารถใช้ไฟจากแรงจ่ายแบตเตอร์รี่เท่านั้น ส่งผลให้งานวิจัยที่มุ่งเน้นการออกแบบหน่วยประมวลผลให้มีประสิทธิภาพสูงและใช้พลังงานต่ำเป็นที่น่าสนใจและท้าทาย

เนื่องจากหน่วยประมวลผลที่เป็นแบบ ASIC (application specific integrated circuit) สามารถตอบสนองความต้องการในเรื่องการทำให้พลังงานต่ำและสามารถให้ประสิทธิภาพการสูง แต่ราคาที่สูงของเทคโนโลยีแบบ ASIC ทำให้ไม่เหมาะที่จะนำมาใช้งานสำหรับผลผลิตทางอิเล็กทรอนิกส์ระดับกลาง – เล็ก สำหรับเทคโนโลยีแบบที่สามารถโปรแกรมได้ (programmable device) ที่อาจจะไม่ได้เปรียบในเชิงของการใช้พลังงานและประสิทธิภาพในการทำงาน แต่ถือว่าช่วยเพิ่มความยืดหยุ่นให้กับนักออกแบบและผู้ใช้งาน รวมทั้งช่วยหย่นระยะเวลาของการออกแบบและพัฒนาได้ ฉะนั้นในงานวิจัยนี้จึงใช้โครงสร้างแบบผสมระหว่าง ASIC ในบล็อคถือว่าเป็น element หนึ่ง โดยให้การต่อเชื่อมถึงกัน (Interconnection) นั้นสามารถถูกควบคุมและโปรแกรมได้โดยผู้ใช้งานถือได้ว่าเป็นสถาปัตยกรรมแบบ hybrid โดยจะสามารถสร้างโพรเซสเซอร์อย่างง่ายขนาด

เล็กขึ้นมาได้จากการประกอบกันของ element ต่างๆ ให้เป็น datapath โดยโครงสร้างของ datapath นี้จะสามารถปรับเปลี่ยนได้ตามงานที่ได้รับ เราเรียกว่า dynamically reconfigurable datapath (DRD)

การปรับเปลี่ยนโครงสร้างภายในเช่นนี้ช่วยให้งานแต่ละชนิดสามารถใช้ทรัพยากรที่มีอยู่ได้อย่างคุ้มค่า ส่งผลให้เกิดการใช้พลังงานได้อย่างเหมาะสมที่สุด แต่เนื่องจากการปรับเปลี่ยนและก่อตัวเป็นโครงสร้าง datapath นั้นจะต้องใช้เวลา overhead ที่เกิดขึ้นจนเป็นปัญหาที่น่าสนใจคือ *จะต้องปรับเปลี่ยนโครงสร้างแบบ dynamically ในลักษณะใดจึงจะช่วยให้ประสิทธิภาพและการใช้พลังงานโดยรวมของระบบดีขึ้น* ดังนั้นหัวข้อสำคัญในการศึกษาค้นคว้าและทำวิจัยในโครงการนี้คือ

- ลด overhead ที่เกิดจากระยะเวลาในการ form โครงสร้างการทำงาน เพื่อให้เป็นระบบที่สามารถ reconfigurable แบบ dynamically อย่างแท้จริง
- เปรียบเทียบประสิทธิภาพและพลังงานระหว่างโครงสร้างที่ปรับเปลี่ยนได้อย่างเร็ววรวดกับโครงสร้างที่ปรับเปลี่ยนได้ภายในระยะเวลาที่กำหนดแต่จะให้มีจำนวนครั้งในการปรับเปลี่ยนน้อยที่สุด
- เพื่อให้ได้ประสิทธิภาพอย่างแท้จริง ดังนั้นการปรับเปลี่ยนโครงสร้างควรเป็น datapath ในระดับของ Functional Unit (FU) ขึ้นไป หรือรวมถึงโครงสร้างภาย Functional Unit ด้วย
- โครงสร้างในการเชื่อมต่อที่มีประสิทธิภาพและพลังงานต่ำ ที่รวมถึง routing และ switching ซึ่งที่มีอยู่จะเป็นแบบใช้สัญญาณนาฬิกา synchronize ถึงกัน แต่ในงานวิจัยนี้มุ่งเน้นแบบไม่ใช้สัญญาณนาฬิกา (Asynchronous)
- ลักษณะการวาง hardwired unit ในที่นี้ควรจะเป็นระดับ FU หรือ element สำหรับการประมวลผลทั่วไป เพื่อให้ประหยัดพื้นที่และการเชื่อมต่อสายสัญญาณระหว่างกัน
- โมเดลที่ช่วยวิเคราะห์การออกแบบและกำหนดรูปแบบการปรับเปลี่ยนโครงสร้างที่ให้ได้ประสิทธิภาพตามที่ต้องการ
- นำเสนอ Design flow ที่ช่วยให้การออกแบบและพัฒนาสะดวกและมีประสิทธิภาพมายิ่งขึ้น

1.4 วัตถุประสงค์ของโครงการวิจัย

1.4.1 เพื่อศึกษา วิจัย พัฒนาการปรับเปลี่ยนโครงสร้างของ datapath ให้ได้ประสิทธิภาพและประหยัดพลังงานมากที่สุด

1.4.2 เพื่อลด overhead (เวลา, พื้นที่, ขนาดของ configuration bit, พลังงาน) ที่เกิดจากการปรับเปลี่ยนโครงสร้างแบบ dynamically

1.4.3 เพื่อสร้างโพรเซสเซอร์ร่วมที่สามารถปรับเปลี่ยนโครงสร้างแบบ dynamically ที่มีประสิทธิภาพและประหยัดพลังงานมากที่สุดได้

1.4.4 เพื่อพัฒนา design flow ในการ map โค้ดโปรแกรมที่ออกแบบโดยโปรแกรมเมอร์ลงบนโพรเซสเซอร์ที่ได้ออกแบบ

1.5 ผลงานวิจัยที่เกี่ยวข้องและเอกสารอ้างอิง

โครงสร้างที่สามารถโปรแกรมได้ (programmable architecture) จะถูกนำมาใช้งานเพื่อสร้างสถาปัตยกรรมที่สามารถปรับเปลี่ยนโครงสร้าง (reconfigurable) การทำงานได้ตามการใช้งาน มีงานวิจัยจำนวนมากเกี่ยวกับโพรเซสเซอร์ที่สามารถโปรแกรมได้สร้างอยู่บน reconfigurable logic โดยมีจุดประสงค์เพื่อเพิ่มความเร็วในการทำงานสำหรับประยุกต์ใช้งานเฉพาะด้าน โดยที่โพรเซสเซอร์ที่เข้ามาช่วยเหล่านี้สามารถแบ่งออกได้เป็น 2 ประเภทคือ

- fine-grain ตัวอย่างเช่น GARP[2], NAPA[3], Chimaera[4] และ PRISC[5]
- coarse-grain ตัวอย่างเช่น Pleiades[6], PipeRench[7], Chameleon[8], RAPID[9] และ MorphoSys[10]

จากการศึกษาในงานวิจัยข้างต้นพบว่ารูปแบบ coarse-grain มีข้อดีเหนือกว่าแบบ fine-grain ในแง่ของความเร็วในการปรับเปลี่ยนโครงสร้าง (reconfigurable) ใช้ขนาดของ configuration bit ที่จะไปโปรแกรม FPGA น้อย และสามารถทำงานได้ด้วยสัญญาณนาฬิกาความถี่สูง ดังนั้น coarse-grain จึงเหมาะกับงานที่มีการใช้ข้อมูลจำนวนมากอย่างเช่น multimedia application และ communication ส่วน fine-grain เหมาะสำหรับงานที่ต้องการคำนวณในระดับบิต ซึ่งในงานวิจัยนี้ใช้แบบ coarse-grain แต่จะทำการศึกษาและวิเคราะห์ว่าการทำงานแบบ coarse-grain ในระดับ FU หรือ block element (multiplier, adder) ที่จะให้การใช้พลังงานอย่างมีประสิทธิภาพ

เมื่อมีการอนุญาตให้ปรับเปลี่ยนโครงสร้างบนอุปกรณ์ที่สามารถโปรแกรมได้ทำให้ส่วนของการ routing เชื่อมต่อแต่ละหน่วยถึงกันให้มีประสิทธิภาพมีงานวิจัย [11, 12] ได้เสนออัลกอริทึมในการ routing แต่งานวิจัยจำนวนหนึ่งที่มุ่งเน้นในถึงการโครงสร้างของการเชื่อมต่ออาทิเช่นในงานวิจัยใช้การเชื่อมต่อแบบบัสหลายทาง[13] และใช้การเชื่อมต่อเป็นระบบเครือข่ายบนชิป [14,15,16] ซึ่งในงานวิจัยนี้จะใช้แนวทางของการเชื่อมต่อเป็นเครือข่ายเนื่อง overhead ที่เกิดจากบัสจะเพิ่มขึ้นตามจำนวนของ element ที่ใช้ ในงานวิจัย [17] ชี้ให้เห็นว่าสามารถนำการเชื่อมต่อแบบเครือข่ายและใช้การส่งผ่าน packet ลงใน Network on Chip (NOC) [15]

เมื่อได้โพรเซสเซอร์ที่สามารถปรับเปลี่ยนโครงสร้างการเชื่อมต่อได้แล้ว การที่จะให้ผู้ใช้สามารถทำให้โพรเซสเซอร์ได้อย่างมีประสิทธิภาพ ฉะนั้นการพัฒนาเครื่องมือช่วยในการ map การแปลโปรแกรมภาษาระดับสูงอย่างเช่นภาษาซี จึงต้องถูกนำมาพิจารณา ดังนั้นจึงต้องศึกษางานวิจัยที่เกี่ยวข้องกับการพัฒนา design flow ในอุปกรณ์ที่สามารถโปรแกรมได้[18,19,20] รวมถึงเทคนิคการจัดแบ่งงานระหว่างซอฟต์แวร์และฮาร์ดแวร์ [21,22]

1.6 ระเบียบวิธีวิจัย

ขั้นตอนการดำเนินงาน

1.6.1 ศึกษาโครงสร้างทั่วไปของสถาปัตยกรรมแบบ reconfigurable

1.6.2 ศึกษาการปรับเปลี่ยนโครงสร้างแบบ dynamically

1.6.3 ศึกษาและวิเคราะห์ความละเอียดของโครงสร้างที่จะทำการปรับเปลี่ยนใหม่ (ความละเอียดของการ Coarse-grain)

1.6.4 ออกแบบและพัฒนาวิธีการปรับเปลี่ยนโครงสร้างแบบ dynamically ที่ใช้เวลาน้อย

1.6.5 ออกแบบและพัฒนาวิธีการ Place และ Route ที่เหมาะกับโครงสร้างที่สามารถปรับเปลี่ยนได้แบบ dynamically

1.6.6 สร้างโมเดลขึ้นเพื่อทดสอบสถาปัตยกรรมที่ออกแบบขึ้นใหม่

1.6.7 พัฒนาโพรเซสเซอร์ที่สามารถปรับเปลี่ยนโครงสร้างได้แบบ dynamically บนเทคโนโลยี FPGA

1.6.8 ทดสอบและเก็บผล

1.6.9 พัฒนาเครื่องมือช่วยในการออกแบบ การ mapping และการแปลโปรแกรมจากภาษาซี

1.6.10 วิเคราะห์และสรุปทำงานรายงาน

1.7 ขอบเขตของโครงการวิจัย

1.7.1 สร้างโพรเซสเซอร์ที่สามารถปรับเปลี่ยนโครงสร้างแบบ dynamically ให้มีประสิทธิภาพสูงและใช้พลังงานต่ำ

1.7.2 โครงสร้างโพรเซสเซอร์ใหม่ที่สามารถปรับเปลี่ยนได้แบบ dynamically จะถูกทดสอบบน FPGA

1.7.3 ใช้วิธีการจำลองการทำงานเพื่อตรวจสอบและวัดผล

1.8 แผนการดำเนินงานวิจัยตลอดโครงการ

กิจกรรมตลอดโครงการแสดงดังตารางที่ 1 โดยมีภาพรวมดังนี้

ปีที่ 1 เดือนที่ 1-6: กิจกรรมตามขั้นตอนที่ 1 โดยมีผลลัพธ์ที่คาดว่าจะได้รับคือ โมเดลวงจรใหม่ที่สามารถสร้างระบบ DSP บน FPGAs ได้

ปีที่ 1 เดือนที่ 7-12: กิจกรรมตามขั้นตอนที่ 2 พัฒนาอัลกอริทึมสำหรับโมเดลวงจรที่พัฒนาได้

ปีที่ 2 เดือนที่ 1-6: กิจกรรมตามขั้นตอนที่ 3 พัฒนาซอฟแวร์ตามอัลกอริทึมที่พัฒนาได้

ปีที่ 2 เดือนที่ 7-12: กิจกรรมตามขั้นตอนที่ 4 ทดสอบและวิเคราะห์ประสิทธิภาพของอัลกอริทึม

ตารางที่ 1 แผนการดำเนินการตลอดโครงการวิจัย

ปีที่ 1			
ลำดับ	กิจกรรม	เวลา	ผล
1	ศึกษาโครงสร้างสถาปัตยกรรมแบบปรับเปลี่ยนโครงสร้างได้, ขั้นตอนกระบวนการ ของการปรับเปลี่ยนโครงสร้างแบบ dynamic และโครงสร้างแบบ coarse-grain และทำการหาความหยาบของการทำ coarse-grain	3 เดือน	White paper
2	โมเดลเพื่อช่วยคาดคะแน power, delay โครงสร้างที่ปรับเปลี่ยนได้	3 เดือน	Technical report
3	สร้าง functional unit ด้วยภาษา HDL เพื่อใช้ในการทดสอบ	2 เดือน	Soft core
4	พัฒนาวงจรของการ routing ให้ใช้พลังงานต่ำ	2 เดือน	International paper
5	โมเดลของโพรเซสเซอร์ที่สามารถปรับเปลี่ยนโครงสร้างได้	2 เดือน	-

1.9 ผลงานวิจัย

ผลงานการตีพิมพ์ในที่ประชุมวิชาการ

1. Sasatorn somwatee, Wannarat Suntiamorntut, survey of finding empty space algorithm for partial reconfigurable FPGAs, Proceedings *of International Joint Conference on Computer Science and Software Engineering* (JCSSE2009), pp. 489-491, 2009.
2. Sasatorn somwatee, Wannarat Suntiamorntut, performance evaluation of the efficient algorithms for free resources management on the FPGA, Proceedings *of International Joint Conference on Computer Science and Software Engineering* (JCSSE2009), pp. 303-305, 2009.
3. W. Suntiamorntut, C. Vongchumyen, Design Techniques for Energy Efficient Multiplier, Ladkrabang Journal, June, 2007
4. W. Suntiamorntut, Hamming Distance and Normalization Circuits, ICCCAS07, July, pp.1053-1056, 2007.
5. W. Suntiamorntut, L.E.M. Brackenbury, Jim Garside Design and Implementation of an Energy Efficient, Parallel, Asynchronous DSP, ITC-CSCC07, July, pp. xxx, 2007.

1.10 เอกสารอ้างอิง

1. W.Suntiamorntut, L.E.M. Brackenbury, Functional Unit for Low-Power DSP Architecture, *IEE System-On-Chip*, Cardiff, UK, 2-3 Sept. 2003.

2. HAUSER, J. R. ANDWAWRZYNEK, J., Garp: A MIPS processor with a reconfigurable coprocessor. In *Proceedings of the IEEE Symposium on Field-Programmable Custom Computing Machines* (FCCM '97). 24–33, 1997.

3. RUPP, C. R., LANDGUTH, M., GARVERICK, T., GOMERSALL, E., HOLT, H., ARNOLD, J. M., AND GOKHALE, M., The NAPA adaptive processing architecture. In *IEEE Symposium on Field-Programmable Custom Computing Machines*, 1998.

4. HAUCK, S., FRY, T. W.,HOSLER, M. M., AND KAO, J. P., The Chimaera reconfigurable functional unit. In *IEEE Symposium on Field-Programmable Custom Computing Machines*, 1997.

5. RAZDAN, R. AND SMITH, M., High-performance micro-architectures with hardware programmable functional units. In *Proceedings of the 27th Annual IEEE/ACM International Symposium on Microarchitecture*. 172–180, 1994.

6. WAN, M., ZHANG, H., GEORGE, V., BENES, M., ABNOUS, A., PRABHU, V., AND RABAEY, J., Design methodology of a low-energy reconfigurable single-chip DSP system. *J. VLSI Signal Process,* 2000.

7. GOLDSTEIN, S. C., SCHMIT, H., MOE, M., BUDIU, M., CADAMBI, S., TAYLOR, R., AND LAUFER, R., PipeRench: A coprocessor for streaming multimedia acceleration. In *Proceedings of the 26th Annual International Symposium on Computer Architecture*. 28–39, 1999.

8. SALEFSKI, B. AND CAGLAR, L., Re-configurable computing in wireless. In *Proceedings of the 38th Design Automation Conference*, Las Vegas, Nevada, 178–183, 2001.

9. EBELING, C., CRONQUIST, D., AND FRANKLIN, P., RaPiD—Reconfigurable Pipelined Datapath. In *The 6th International Workshop on Field-Programmable Logic and Applications*, 1996.

10. H. Singh et al., MorphoSys: An Integrated Reconfigurable System for Data-Parallel and Communication-Intensive Applications, in *IEEE Trans. on Computers*, vol. 49, no. 5, pp. 465-481, May 2000.

11. A. Ahmadinia et al., A New Approach for On-Line Placement on Reconfigurable Devices, Proc. 18th Int'l Parallel and Distributed Processing Symp. (IPDPS 04), IEEE CS Press, p. 134, 2004.

12. C. Steiger et al., Online Scheduling and Placement of Real-Time Tasks to Partially Reconfigurable Devices, Proc. 24th Int'l Real-Time Systems Symp. (RTSS 03), IEEE CS Press, pp. 224-235, 2003.

13. R. Vaidyanathan and J.L. Trahan, Dynamic Reconfiguration: Architectures and Algorithms, Plenum US, 2004.

14. C. Bobda et al., A Dynamic NoC Approach for Communication in Reconfigurable Devices, Proc. 14th Int'l Field-Programmable Logic Conf. (FPL 04), LNCS vol. 3203, Springer, pp. 1032-1036, 2004.

15. L. Benini and G. Micheli, Networks on Chips: A New SoC Paradigm, J. Computer, vol. 35, no. 1, Jan., pp. 70-78, 2001.

16. T. Marescaux et al., Networks on Chip as Hardware Components of an OS for Reconfigurable Systems, Proc. 13th Int'l Field-Programmable Logic Conf. (FPL 03), LNCS vol. 2778, Springer, pp. 595-605, 2003.

17. V. Baumgarte et al., Pact XPPA Self-Reconfigurable Data Processing Architecture, J. Supercomputing, vol. 26, no. 2, Sept., pp. 167-184, 2003.

18. J. Becker et al., Datapath and Compiler Integration of Coarse-grain Reconfigurable XPP-Arrays into Pipelined RISC Processor, in *Proc. of IFIP VLSI SoC*, pp. 288-293, 2003.

19. J. Villareal et al., Improving Software Performance with Configurable Logic, in *Design Automation for Embedded Systems*, Springer, vol. 7, pp. 325-339, 2002.

20. G. Stitt et al., Energy Savings and Speedups from Partitioning Critical Software Loops to Hardware in Embedded Systems, in *ACM TECS*, vol.3, no.1, pp. 218-232, Feb., 2004.

21. D. D. Gajski et al., SpecSyn: An environment supporting the specify-explore-refine paradigm for hardware/software system design, in *IEEE Trans. on VLSI Syst.*, vol. 6, no. 1, pp. 84–100, 1998.

22. J. Henkel, A low power hardware/software partitioning approach for core-based embedded systems, in *Proc. of the 36th ACM/IEEE DAC*, pp. 122–127, 1999.