

โครงสร้างทางฮาร์ดแวร์ของวงจรแปลงแบบตัวประกอบปฐม

Hardware structure of a Prime Factor Finite Field Transform

พงศธร หมายดี[□]สมศักดิ์ ชุมช่วย^{□□}

ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

การแปลงข้อมูลในสนามจำกัดมีประโยชน์มากทั้งในวงจรเข้ารหัสและถอดรหัสรีด-โซโลมอนซึ่งใช้กันแพร่หลาย การแปลงในสนามจำกัด $GF(2^8)$ ขนาด 255 จุดโดยการใช้วิธีตัวประกอบปฐมร่วมกับวิธีการแปลงข้อมูลจำนวนน้อยๆจะทำให้สามารถลดจำนวนการคำนวณลงได้ 57 เท่าเมื่อเทียบกับการคำนวณโดยตรง แต่สำหรับในการออกแบบวงจรแปลงจากวิธีดังกล่าวจะต้องพิจารณาเวลาที่ใช้จริงเพื่อให้วงจรแปลงสามารถทำงานสอดคล้องกับส่วนอื่นๆได้ บทความนี้ได้นำเสนอโครงสร้างของวงจรแปลงต่างๆ ซึ่งใช้เวลาในการแปลงแตกต่างกันไป บางชนิดจะเอื้ออำนวยให้สามารถกำหนดความเร็วในการแปลงได้โดยง่ายทำให้สามารถที่จะออกแบบวงจรแปลงที่สามารถโปรแกรมความเร็วให้เหมาะสมกับการใช้งานได้ ซึ่งวงจรแปลงที่นำเสนอที่มีความเร็วสูงสุดจะมีความเร็วสูงกว่าวงจรแปลงที่ออกแบบจากการคำนวณโดยตรงถึง 63 เท่าโดยมีขนาดใกล้เคียงกัน

Abstract

Prime Factor Algorithm (PFA) and Short Length Algorithm (SLA) can be applied to Finite Field Transform in order to reduce the computation count, especially when it is used in Reed-Solomon encoder/decoder. (Reduced by the factor of 57 when compared to direct computation). Hardware realization is required in real time applications for the speed achievement and the system compaction. This paper presents several transform machines applied to the PFA and SLA. For the maximum speed design, it is shown that the computation speed can be increased by 63 times compared to the direct implementation.

1. บทนำ

ในระบบการติดต่อรับส่งข้อมูลที่ต้องการความถูกต้องของข้อมูลเป็นสิ่งสำคัญ การเข้ารหัสและถอดรหัสเพื่อใช้ในการตรวจหาและแก้ไขข้อมูลที่ผิดพลาดจึงเป็นสิ่งจำเป็น รหัส BCH ที่เป็นที่นิยมใช้กันแพร่หลายในการเข้ารหัสและถอดรหัสชนิดหนึ่งก็คือรหัสรีด-โซโลมอน ขนาด

[□] นักศึกษาปริญญาโท

^{□□} อาจารย์ประจำภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ ฯ

255 จุด ซึ่งข้อมูลถูกกำหนดอยู่ใน $GF(2^8)$ ในการเข้ารหัสและถอดรหัสโดยใช้รหัส BCH นี้จะสามารถทำได้ทั้งในเชิงเวลาและเชิงความถี่ โดยความยุ่งยากของการทำงานจะอยู่ที่การถอดรหัสซึ่งจะนิยมทำในเชิงความถี่ เนื่องจากการถอดรหัสในเชิงความถี่นี้จะต้องใช้การแปลงและการแปลงกลับเป็นส่วนสำคัญ [1] นอกเหนือไปจากการคำนวณอื่นๆ การแปลงและการแปลงกลับจะต้องใช้การคำนวณเป็นจำนวนมากจึงมีการนำวิธีการตัวประกอบปฐม (Prime Factor Algorithm) มาใช้เพื่อลดจำนวนการคำนวณลงโดยใช้ร่วมกับวิธีการแปลงข้อมูลจำนวนน้อยๆ (Short Length Algorithm) [2] ในแง่ของการนำวิธีการดังกล่าวไปใช้งานจริงซึ่งจะต้องพิจารณาเวลาที่ใช้ในการแปลงเป็นสิ่งสำคัญจึงจำเป็นต้องมีการวิเคราะห์โครงสร้างการแปลงดังกล่าวในทางฮาร์ดแวร์ เพื่อพิจารณาดังความเหมาะสมในการนำไปประยุกต์ใช้งานต่างๆ

ในการแปลงและการแปลงกลับจะมีโครงสร้างการทำงานเหมือนกัน โดยจะมีการเปลี่ยนแปลงค่าสัมประสิทธิ์ที่ใช้เท่านั้น ซึ่งค่าสัมประสิทธิ์จะขึ้นอยู่กับ prime polynomial ที่ใช้โดยในบทความนี้ใช้ prime polynomial เป็น $x^8+x^4+x^3+x^2+1$ ซึ่งจะทำให้ขนาดของวงจรแบบขนานชนิดตัวคูณคงที่มีขนาดเล็กที่สุด ในการพิจารณาเลือกใช้งานวงจรแปลงจะพิจารณาจำนวนคาบเวลาที่ใช้ในการแปลงและขนาดของวงจรแปลงเป็นสิ่งสำคัญรวมทั้งความสามารถในการปรับเปลี่ยนความเร็วของวงจรแปลงด้วย เนื่องจากต้องให้มีความเร็วในการแปลงสอดคล้องกับการทำงานในส่วนอื่นๆของการเข้ารหัสและถอดรหัสในขนาดที่เหมาะสมในการพิจารณาเปรียบเทียบจะมีการนำ 2.0 micron standard cell (MDA 20) ของ motorola [3] มาใช้อ้างอิงเพื่อให้เห็นถึงความแตกต่างและข้อจำกัดของฮาร์ดแวร์แต่ละแบบได้อย่างชัดเจนยิ่งขึ้น

2. วิธีการตัวประกอบปฐม

การแปลงขนาด 255 จุดในสนามจำกัด $GF(2^8)$ มีสมการเป็น $A_k = \sum_{n=0}^{254} a_n \alpha^{(nk)255}$ โดย $k = 0, 1, \dots, 254$ ซึ่งจะเห็นได้ว่าจะมีการบวก 65025 ครั้งและมีการคูณ 65025 ครั้งเช่นกันจึงทำให้ต้องใช้เวลาในการคำนวณมาก วิธีการตัวประกอบปฐมจะเป็นการแยกข้อมูลจำนวน 255 จุด ออกเป็นส่วนๆ โดยแต่ละส่วนจะมีขนาดเป็นจำนวนปฐมซึ่งเมื่อคูณกันทั้งหมดแล้วจะมีค่าเท่ากับ 255 ซึ่งจะแยกข้อมูลออกได้เป็นขนาด 3, 5 และ 17 จุด ข้อมูลทั้ง 255 จุดจะนำมาผ่านการแปลง 3, 5 และ 17 จุด โดยในการแปลงแต่ละขั้นจะทำงานทั่วข้อมูลทั้ง 255 จุด และใน

ระหว่างการแปลงแต่ละขั้นจะมีการสลับตำแหน่งข้อมูล (mapping) ซึ่งเป็นไปตามวิธีการตัวประกอบปฐม [2] ในการสลับตำแหน่งข้อมูลนี้จะไม่มีการคำนวณเกิดขึ้นเลย (เมื่อออกแบบเป็นวงจรรวม) ดังนั้นจะเห็นได้ว่าเมื่อใช้วิธีการตัวประกอบปฐมจะใช้การบวกและการคูณเพียง 6357 ครั้งซึ่งลดลงจากเดิมประมาณ 10 เท่า และยังจะเห็นได้ว่าการทำงานยังแบ่งออกเป็น 3 ส่วน โดยในส่วนที่มีการคำนวณสูงสุดคือขั้นการแปลง 17 จุดซึ่งมีการบวกและการคูณ 4335 ครั้ง จึงทำให้โครงสร้างการคำนวณที่ได้จากวิธีตัวประกอบปฐมจะมีลักษณะเป็นแบบท่อส่ง (pipeline) ซึ่งจะมีความเร็วกว่าเดิมถึง 15 เท่า และเมื่อนำวิธีการแปลงข้อมูลจำนวนน้อยๆมาใช้ร่วมด้วยจะช่วยลดจำนวนการคูณและการบวกลงเหลือ 1135 ครั้งและ 3857 ครั้งตามลำดับ [2] และเมื่อออกแบบเป็นวงจรรวมด้วยโครงสร้างที่เหมาะสมจะทำให้มีความเร็วเพิ่มขึ้นกว่าเดิมอีกด้วย

3. ส่วนคำนวณและอุปกรณ์ที่ใช้ในการออกแบบวงจรแปลง

ส่วนคำนวณที่ใช้ในการแปลงคือ ส่วนการบวกและส่วนการคูณซึ่งแบ่งออกเป็น ส่วนคำนวณแบบขนานและแบบอนุกรม เนื่องจากข้อมูลที่ใช้ในการแปลงในสนามจำกัด $GF(2^8)$ จะแทนด้วยข้อมูลฐานสองขนาด 8 บิต ดังนั้นวงจรบวกแบบอนุกรมจะใช้เวลามากกว่าวงจรวางแบบขนานอยู่ 8 เท่าในขณะที่มีขนาดเล็กกว่าประมาณ 8 เท่าเช่นกัน [4] ดังแสดงดังตารางที่ 3.1 ส่วนวงจรคูณนั้นนอกจากจะมีทั้งแบบขนานและแบบอนุกรมแล้วแต่ละแบบยังมีทั้งชนิดตัวคูณคงที่และตัวคูณทั่วไป [4] ซึ่งขนาดและเวลาที่ใช้ในการคำนวณของวงจรมูลแต่ละชนิดแสดงดังตารางที่ 3.1 จะเห็นได้ว่าวงจรมูลแบบอนุกรมจะมีขนาดไม่น้อยไปกว่าวงจรมูลแบบขนานเท่าใดนักในขณะที่ต้องใช้เวลาในการคำนวณมากกว่ามากจึงไม่เหมาะที่จะนำมาใช้ในการออกแบบ

การคำนวณ	ชนิดข้อมูล	วิธีการ	ขนาดเกต*	คาบเวลาที่ใช้	เวลาที่ใช้(ns)	เวลาเทียบเท่า*
บวก	อนุกรม	ปกติ	3.3	8	18.32	$8\tau_{xor(2)}$
		บังคับผลลัพธ์**	4.8	8	31.2	$8(\tau_{xor(1)} + \tau_{and(2)})$
	ขนาน	ปกติ	26.4	1	2.29	$\tau_{xor(2)}$
		บังคับผลลัพธ์	38.4	1	3.9	$\tau_{xor(1)} + \tau_{and(2)}$
คูณ	อนุกรม	ตัวคูณคงที่	140.3- 152.3 [□]	8	8.16	$2\tau_{xor(1)} + \tau_{dff(2)}$
		ตัวคูณทั่วไป	152.3	8	9.69	$2\tau_{xor(1)} + \tau_{dff(2)} + \tau_{and(1)}$
	ขนาน	ตัวคูณคงที่	38.8 ^{□□}	1	9.75	$3\tau_{xor(4)}$
		ตัวคูณทั่วไป	165.3	1	12.82	$\tau_{and(1)} + \tau_{xor(4)} + 4\tau_{xor(1)}$

*ดูรายละเอียดได้จากภาคผนวก **สามารถบังคับผลลัพธ์ให้เป็นศูนย์
[□] ขึ้นอยู่กับจำนวนบิตที่เป็นหนึ่งของตัวคูณ ^{□□} ค่าเฉลี่ยจากวงจรมูลที่ใช้ทั้งหมด

ตารางที่ 3.1 แสดงเวลาที่ใช้และขนาดของวงจรคำนวณแบบต่างๆ

ในการออกแบบส่วนคำนวณที่มีชนิดของข้อมูลเป็นทั้งแบบขนานและอนุกรมนั้นจะต้องมีการเพิ่มบัพเฟอร์เพื่อเปลี่ยนชนิดของข้อ

มูลซึ่งทำให้เกิดความยุ่งยากในการออกแบบและยังต้องเพิ่มขนาดของวงจรมูลในส่วนบัพเฟอร์อีกด้วยดังนั้นในการออกแบบที่พิจารณาต่อไปจึงจะใช้เฉพาะวงจรมูลและวงจรวางแบบขนานเท่านั้น

บทความนี้จะออกแบบเฉพาะวงจรแปลงที่สามารถเปลี่ยนแปลงค่าสัมประสิทธิ์ได้ ซึ่งสามารถทำได้โดยใช้วงจรมูลชนิดตัวคูณทั่วไป ทำให้จะต้องเพิ่มอุปกรณ์ที่ใช้ในการป้อนตัวคูณให้กับวงจรมูลด้วย อุปกรณ์ดังกล่าวอาจจะเป็น rom/ram cell หรือเป็นบัพเฟอร์เลื่อน (shift register) ก็ได้ โดยขนาดของอุปกรณ์ที่ใช้ป้อนตัวคูณนี้จะคงที่ไม่ว่าจะออกแบบวงจรแปลงแบบใดและมีความเร็วเท่าใด ทั้งนี้เนื่องมาจากจำนวนสัมประสิทธิ์และจำนวนการคูณจะมีจำนวนเท่าเดิมนั่นเอง ดังนั้นจึงไม่นำขนาดของอุปกรณ์ในส่วนนี้มาใช้ในการพิจารณาขนาดของวงจรแปลงที่ออกแบบ

การใช้อุปกรณ์ป้อนตัวคูณแบบ rom/ram cell จะทำให้วงจรมูลขนาดเล็กเนื่องจากเราสามารถเก็บค่าตัวคูณไว้เท่าที่จำเป็นโดยการเลือกค่าตัวคูณไปใช้นั้นจะถูกควบคุมโดยสัญญาณเลือก (ข้อมูลแอดเดรสของ rom/ram cell) ดังนั้นจึงต้องมีการออกแบบวงจรในส่วนกำเนิดสัญญาณเลือกนี้เพิ่มเติม

การออกแบบอุปกรณ์ป้อนตัวคูณแบบบัพเฟอร์เลื่อนนั้นทำได้ง่ายกว่าแบบ rom/ram cell เนื่องจากเราจะใช้เพียงสัญญาณนาฬิกาเพื่อให้อุปกรณ์ส่วนนี้ทำงานซึ่งก็เป็นสัญญาณนาฬิกาเดียวกับที่ใช้ในระบบ แต่เนื่องจากต้องมีการเก็บค่าตัวคูณไว้เท่ากับจำนวนการคูณทั้งหมดซึ่งก็คือ 1135 ตัว ซึ่งข้อมูลใน $GF(2^8)$ จะสามารถแทนด้วยเลขฐานสองจำนวน 8 บิต ดังนั้นบัพเฟอร์เลื่อนขนาด 1 จุดใน $GF(2^8)$ จึงประกอบไปด้วย D/F/F จำนวน 8 ตัว โดย 1 อินพุต D/F/F จะมีขนาด 6.5 เกท [3] และ 2 อินพุต D/F/F จะมีขนาดเป็น 8.5 เกท [3] ทำให้อุปกรณ์ป้อนตัวคูณ (สร้างจาก 1 อินพุต D/F/F) แบบบัพเฟอร์เลื่อนมีขนาดที่ใหญ่กว่าประมาณ 77180 เกท

การเปลี่ยนแปลงค่าสัมประสิทธิ์ของอุปกรณ์ป้อนตัวคูณแบบ rom cell นั้นทำได้โดยการเปลี่ยนชุดเก็บข้อมูล (rom cell) ในขณะที่แบบ ram cell จะทำได้โดยการรับข้อมูลสัมประสิทธิ์ชุดใหม่จากภายนอก จะเห็นได้ว่าในกรณีที่ต้องการเปลี่ยนชุดของสัมประสิทธิ์เพียง 2-3 ชุดนั้นอาจจะใช้อุปกรณ์ป้อนตัวคูณแบบ rom cell แต่ถ้าต้องการเปลี่ยนชุดของสัมประสิทธิ์เป็นจำนวนมากๆ หรือไม่รู้ค่าของสัมประสิทธิ์ที่แน่นอนนั้นควรจะใช้อุปกรณ์ป้อนตัวคูณแบบ ram cell

การเปลี่ยนแปลงค่าสัมประสิทธิ์ในอุปกรณ์ป้อนตัวคูณแบบ บัพเฟอร์เลื่อน นั้นเราสามารถทำได้โดยการส่งสัญญาณกำหนดค่าให้กับ flip-flop ทั้ง 1135 ตัว จึงต้องมีการออกแบบวงจรมูลสัญญาณนี้ทุกค่าสัมประสิทธิ์ที่ใช้ด้วยเหตุผลที่อุปกรณ์ป้อนตัวคูณแบบนี้มีขนาดใหญ่ อีกทั้งยังทำการเปลี่ยนชุดสัมประสิทธิ์ได้ยากจึงทำให้ไม่เหมาะที่จะใช้ในการออกแบบวงจรแปลง

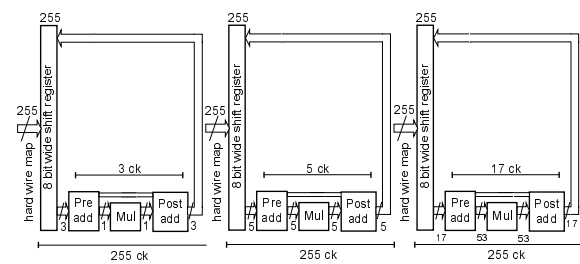
4. การออกแบบโครงสร้างของวงจรที่ใช้ในการแปลง

การออกแบบวงจรแปลงสามารถพิจารณาได้หลายวิธีดังนี้

4.1 การออกแบบโดยตรงจากขั้นตอนวิธี

วิธีนี้เป็น การนำโครงสร้างการคำนวณที่กล่าวถึงในหัวข้อที่ 2 มา ออกแบบซึ่งวงจรแปลงจะแบ่งออกเป็น 3 ขั้นตอนดังรูปที่ 4.1 จะเห็นว่าการแปลงแต่ละขั้นจะประกอบไปด้วยส่วนป้อนข้อมูล, ส่วนคำนวณ และส่วนรวบรวมข้อมูล โดยส่วนป้อนข้อมูลและรวบรวมข้อมูลจะใช้ โครงสร้างเป็นบัพเฟอร์เลื่อน (ออกแบบด้วย 2 อินพุต D/F/F เนื่องจาก ต้องมีการถ่ายเทข้อมูลเข้าแบบขนานอีกด้วย) ซึ่งเป็นโครงสร้างที่ทำให้ การป้อนข้อมูลให้กับส่วนคำนวณทำได้ง่ายอีกทั้งยังทำให้สามารถใช้ ตัวป้อนและรวบรวมข้อมูลร่วมกันได้เพื่อเป็นการลดขนาดของวงจร รวมเนื่องจากบัพเฟอร์เลื่อนที่ใช้เป็นส่วนป้อนหรือรวบรวมข้อมูลนี้มี ขนาดถึงประมาณ 17340 เกท สำหรับข้อมูลจำนวน 255 จุด

ส่วนป้อนข้อมูลจะป้อนข้อมูลให้กับส่วนคำนวณในแบบขนาน และจะมีการเลื่อนข้อมูลชุดใหม่ให้มาอยู่ทางด้านล่างของส่วนป้อนข้อมูลเพื่อป้อนให้กับส่วนคำนวณในรอบการคำนวณถัดไป ซึ่งต้องใช้ เวลาในการเลื่อนเท่ากับจำนวนข้อมูลที่ต้องใช้ในส่วนคำนวณ การ เลื่อนจะทำแบบไม่ป้อนกลับเนื่องจากข้อมูลที่ถูกลำเลียงไปคำนวณแล้วจะ ไม่ถูกนำมาใช้อีก ทำให้เราสามารถจะเก็บผลลัพธ์ที่ได้จากการคำนวณ ซึ่งมีขนาดเท่ากับข้อมูลที่นำมาใช้คำนวณลงที่ด้านบนของส่วนป้อนข้อมูลได้ จะเห็นว่า การเลื่อนทำไปอย่างต่อเนื่องทำให้เวลาที่สามารถใช้ได้ ในส่วนคำนวณนับเป็นคาบเวลาจะเท่ากับจำนวนของข้อมูลที่ใช้ใน ส่วนคำนวณนั้นๆนั่นเอง ซึ่งเท่ากับ 3, 5 และ 17 คาบเวลาสำหรับการ แปลง 3, 5 และ 17 จุดตามลำดับ จะเห็นได้ว่าเมื่อเราทำการเลื่อนข้อมูล จนครบ 255 คาบเวลา ก็จะเป็นการป้อนข้อมูลให้ส่วนคำนวณทำการ คำนวณครบทั้ง 255 จุด ดังนั้นวงจรแปลงที่มีโครงสร้างแบบนี้จะมีความเร็วในการแปลง 255 คาบเวลาและมี latency time เป็น 765 คาบ เวลา



รูปที่ 4.1 แสดงโครงสร้างวงจรแปลงที่ออกแบบโดยตรงจากขั้นตอนวิธี

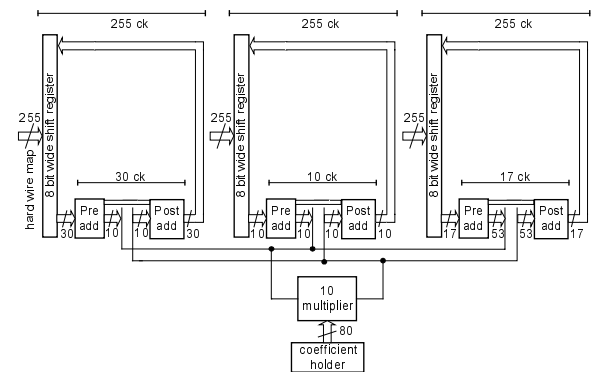
เมื่อพิจารณาในส่วนคำนวณการแปลง 3 จุด จะเห็นว่าเวลาที่ใช้ ในการ pre-add และ post-add เป็น $2\tau_{xor(2)}$ เนื่องจากประกอบด้วย การบวก 2 ชั้น [2] และในการคูณจะใช้เวลาเท่ากับ $3\tau_{xor(4)}$ เมื่อใช้วงจรคูณ แบบขนานชนิดตัวคูณคงที่ เนื่องจาก $4\tau_{xor(2)} < 3\tau_{xor(4)} < 2*4\tau_{xor(2)}$ ดังนั้น เราสามารถใช้คาบเวลาพิกษาเป็น $4\tau_{xor(2)}$ ซึ่งจะทำให้ใช้เวลาในการ

pre-add และ post-add รวมกันเป็น 1 คาบเวลา และการคูณใช้ 2 คาบ เวลา รวมเป็น 3 คาบเวลาเท่ากับที่กำหนดไว้สำหรับการแปลง 3 จุด ใน การแปลง 5 จุดจะใช้เวลาในการบวกเป็น $3\tau_{xor(2)}$ สำหรับการ pre-add และ post-add และใช้เวลาในการคูณเป็น $3\tau_{xor(4)}$ ดังนั้นจากคาบเวลาที่ กำหนดเป็น $4\tau_{xor(2)}$ ทำให้เราใช้เวลาในส่วนการแปลง 5 จุดนี้เป็น 4 คาบเวลา และในการแปลง 17 จุด จะใช้เวลาในการ pre-add และ post-add เป็น $4\tau_{xor(2)}$ และ $5\tau_{xor(2)}$ ดังนั้นเราจะใช้เวลาในการแปลง 17 จุดเท่า กับ 5 คาบเวลา วงจรแปลงในลักษณะนี้จะมีขนาด เป็น 59603 เกท

ถ้าเราจะออกแบบให้วงจรแปลงแบบนี้สามารถที่จะเปลี่ยนชุด สัมประสิทธิ์ได้โดยใช้ตัวคูณแบบขนานชนิดตัวคูณทั่วไปจำนวน 59 ตัว จะเห็นว่าเราจะต้องใช้เวลาในการคูณเพิ่มขึ้นเป็น $\tau_{and(1)} + \tau_{xor(4)} + 4\tau_{xor(1)}$ ซึ่งจะมีค่ามากกว่า $3\tau_{xor(4)}$ อยู่เล็กน้อย แต่ $\tau_{and(1)} + \tau_{xor(4)} + 4\tau_{xor(1)} < 2*4\tau_{xor(2)}$ ($2*4\tau_{xor(2)}$ คือเวลาที่ใช้ในการคูณเมื่อใช้วงจรคูณแบบขนาน ชนิดตัวคงที่) ดังนั้นจึงทำให้มีเวลาในการคำนวณเท่าเดิมแต่จะมีขนาด เพิ่มขึ้นเป็น 67069 เกท

4.2 การออกแบบโดยใช้วงจรคูณซ้ำ

ในการออกแบบโดยตรงจากขั้นตอนวิธีจะเห็นได้ว่าเมื่อเรา ต้องการให้วงจรแปลงสามารถเปลี่ยนแปลงค่าสัมประสิทธิ์ได้นั้นเราจะต้องเพิ่มเกทถึงเกือบ 10000 ซึ่งเป็นขนาดที่มากเกินไปจนจำเป็น และ จะสังเกตได้ว่าการแปลงในแต่ละขั้นจะเสร็จไม่พร้อมกันทำให้ต้องมีการรอเกิดขึ้น ซึ่งแสดงถึงการใช่วงจรคูณไม่เต็มประสิทธิภาพ ดังนั้น เราสามารถที่จะนำวงจรคูณมาใช้ร่วมกันได้ดังรูปที่ 4.2 โดยต้องเพิ่ม ระบบบัสอีกเล็กน้อย



รูปที่ 4.2 แสดงโครงสร้างวงจรแปลงโดยใช้วงจรคูณซ้ำ

การคูณในการแปลง 3, 5 และ 17 จุดมีการคูณ 85, 255 และ 795 ครั้งตามลำดับรวมเป็น 1135 ครั้ง ถ้าเราต้องการทำการคูณให้เสร็จภายใน 255 คาบเวลา เราจะต้องใช่วงจรคูณแบบขนานชนิดตัวคูณทั่วไปซึ่ง ใช้เวลาในการคูณเป็น 2 คาบเวลา (เมื่อคาบเวลาพิกษาที่ใช้คือ $4\tau_{xor(2)}$) เป็นจำนวนเท่ากับ $1135*2/255 = 8.9$ หรือเท่ากับ 9 ตัว แต่เนื่องจากชุด วงจรคูณจะคำนวณการแปลงในแต่ละขั้นทั้งชุดวงจรคูณ ดังนั้นเพื่อให้สอดคล้องกับจำนวนการคูณในการแปลง 5 จุด ที่มีการคูณ 5 ครั้ง ซึ่งเมื่อมีการเพิ่มวงจร pre-add และ post-add อีก 1 ชุดจะทำให้มีการ คูณเป็น 10 ครั้ง จึงต้องเพิ่มวงจรคูณในชุดวงจรคูณให้เป็น 10 ตัว และ

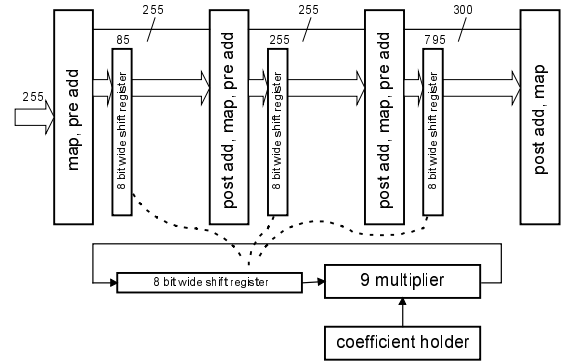
ในการแปลง 3 จุดซึ่งมีการคูณเพียง 1 ครั้งจึงจำเป็นต้องเพิ่มวงจรในส่วน pre-add และ post-add เพื่อให้มีการคูณเป็นจำนวน 10 ครั้งดังแสดงในรูปที่ 4.2 การใช้งานชุดวงจรคูณเพื่อคำนวณในการแปลงต่างๆ จะควบคุมด้วยวงจรควบคุมที่ต้องออกแบบเพิ่มเติม โดยจะต้องควบคุมให้วงจรคูณทำการคูณให้กับการแปลง 3 จุด 1 ครั้งภายในทุกๆ 30 คาบเวลา, 1 ครั้งภายในทุกๆ 10 คาบเวลาสำหรับการแปลง 5 จุด และ 6 ครั้งภายในทุกๆ 17 คาบเวลาสำหรับการแปลง 17 จุด ซึ่งจะเห็นได้ว่าเรามีเวลาพอที่จะใช้ในการทำ pre-add และ post-add ของการแปลงในแต่ละขั้น ในการออกแบบวิธีนี้ทำให้ขนาดของวงจรเพิ่มขึ้นอีกเล็กน้อยในส่วนของ pre-add และ post-add ที่เพิ่มเข้ามา 1637 เกท แต่สามารถลดขนาดของส่วนวงจรคูณลงไป 8100 เกท เมื่อเทียบกับการออกแบบโดยตรงจากขั้นตอนวิธีที่สามารถเปลี่ยนค่าสัมประสิทธิ์ได้ จะเห็นได้ว่าคาบเวลาสัญญาณนาฬิกาที่ใช้คือ $4\tau_{xor(2)}$ และเวลาที่ใช้ในการคูณในแต่ละรอบการคูณคือ 2 คาบเวลา ดังนั้นการออกแบบลักษณะนี้จึงมีความเร็วและ latency time เท่ากับการออกแบบโดยตรงจากขั้นตอนวิธี โดยมีขนาดเป็น 60606 เกท

4.3 การออกแบบโดยย้ายตำแหน่งบัพเฟอร์

ในการออกแบบโดยใช้วงจรถูกซ้ำนั้นจะเห็นได้ว่าการที่จะเพิ่มความเร็วในการแปลงนั้นทำได้โดยการเพิ่มจำนวนวงจรถูกซ้ำในชุดวงจรคูณซึ่งจะต้องมีการเพิ่มขนาดของ pre-add และ post-add ของการแปลงในขั้นต่างๆ ให้มีจำนวนที่สอดคล้องกับจำนวนวงจรถูกซ้ำด้วย อีกทั้งยังต้องมีการเปลี่ยนแปลงขนาดของระบบบัสและที่สำคัญก็คือต้องมีการออกแบบวงจรที่จะใช้ควบคุมชุดวงจรถูกซ้ำใหม่ทั้งหมด จึงทำให้เกิดความไม่สะดวกในการที่จะเปลี่ยนแปลงความเร็วของวงจรแปลง เราสามารถจะแก้ไขปัญหานี้ทั้งหมดที่กล่าวมาแล้วได้โดยใช้โครงสร้างของวงจรถูกซ้ำดังรูปที่ 4.3 โดยจะมีเพิ่มจำนวนของ pre-add และ post-add ให้มีขนาดเต็มจำนวนข้อมูลทั้ง 255 จุด จึงเป็นการเพิ่มขนาดของส่วน pre-add และ post-add จาก 6733 เกท ที่ใช้ในวงจรถูกซ้ำเป็น 101834 เกท การทำเช่นนี้ทำให้เราสามารถที่จะย้ายบัพเฟอร์เข้ามาอยู่ก่อนหน้าการคูณได้ จะเห็นได้ว่าจำนวนของบัพเฟอร์ที่ใช้จะมีขนาดเปลี่ยนไปจากวิธีอื่นๆ ที่กล่าวไปแล้ว ทั้งนี้เนื่องมาจากจำนวนของการคูณในการแปลงแต่ละขั้นไม่เท่ากับจำนวนข้อมูลที่ใช้ในการแปลงนั่นเอง โดยจะมีจำนวนบัพเฟอร์รวมเป็น 1135 จุดซึ่งมีขนาดเป็น 77180 เกท จะเห็นว่าเมื่อนำบัพเฟอร์ทั้งหมดมาวางเรียงกันเพื่อป้อนตัวตั้งให้กับชุดวงจรถูกซ้ำเราจะควบคุมการคูณของวงจรถูกซ้ำนี้ได้โดยใช้สัญญาณนาฬิกาเพียงอย่างเดียว

จะเห็นว่า $4\tau_{xor(2)} < 7\tau_{xor(2)} + \tau_{and(1)} + \tau_{xor(4)} + 4\tau_{xor(1)} < 2 * 4\tau_{xor(2)}$ ($4\tau_{xor(2)}$) คือคาบสัญญาณนาฬิกาที่ใช้ ดังนั้นเวลาที่ใช้ในการ pre-add, map และ post-add ของขั้นที่ใช้เวลาในการคำนวณมากที่สุดคือในขั้นที่ 3 ซึ่งใช้เวลาในการคำนวณเป็น 2 คาบเวลา ถ้าเราต้องการใช้เวลาในการแปลงทั้งหมดเป็น 255 คาบเวลา เราต้องทำการคูณให้เสร็จภายใน 253

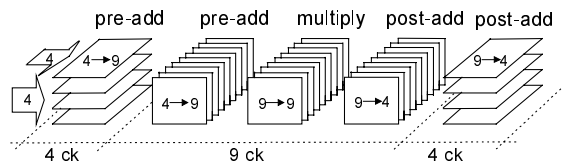
คาบเวลา ซึ่งจะต้องใช้วงจรถูกซ้ำทั้งหมด $1135 * 2 / 253 = 8.97$ หรือเท่ากับ 9 ตัว จะเห็นว่าเราสามารถเพิ่มจำนวนวงจรถูกซ้ำเพื่อให้ใช้จำนวนคาบเวลาในการแปลงน้อยลงได้โดยง่าย โดยจะต้องจัดรูปแบบการวางของบัพเฟอร์ที่ใช้ป้อนตัวตั้งเสียใหม่เพิ่มเติมเพียงอย่างเดียวเท่านั้น แต่วงจรถูกซ้ำแบบนี้จะมีขนาด 180502 เกท ซึ่งจะมีขนาดใหญ่กว่าถึงประมาณ 3 เท่าเนื่องจากการเพิ่มขนาดในส่วน pre-add และ post-add นั่นเอง



รูปที่ 4.3 แสดง โครงสร้างวงจรถูกซ้ำแบบใช้วงจรถูกซ้ำ

4.4 การออกแบบโดยใช้ส่วนคำนวณร่วมกัน

เมื่อพิจารณาผังงานในส่วนของการ pre-add และ post-add ของการแปลง 17 จุด [2] ในลักษณะ 3 มิติซึ่งแสดงได้ดังรูปที่ 4.4.1 ซึ่งไม่ได้แสดงการบวกค่า a_0 เข้ากับผลลัพธ์ของการแปลงไว้ด้วย ข้อมูลขนาด $4 * 4$ ($a_1 - a_{16}$ ยกเว้น a_0) จะถูกทำการ pre-add 4 ครั้ง ในแนวนอนได้เป็นข้อมูลขนาด $4 * 9$ เป็นขั้นที่หนึ่ง หลังจากนั้นจะมีการเปลี่ยนแกนของการคำนวณไปเป็นแนวตั้งผ่านการคำนวณขั้นที่สองคือ pre-add, multiply และ post-add เป็นจำนวน 9 ครั้ง ได้เป็นข้อมูลขนาด $4 * 9$ เช่นเดิม (โดยขนาดของข้อมูลในแต่ละการคำนวณแสดงดังรูปที่ 4.4.1) หลังจากนั้นจะมีการเปลี่ยนแกนการคำนวณไปเป็นแนวนอนอีกครั้งหนึ่งเพื่อทำการคำนวณในขั้นที่สามคือ post-add 4 ครั้ง ได้เป็นข้อมูลขนาด $4 * 4$ ที่ผ่านการแปลงแล้ว จะเห็นได้ว่าถ้าการทำงานในแต่ละครั้งของทั้งสามขั้นสามารถทำได้ภายใน 1 คาบเวลาจะทำให้เราสามารถทำการแปลง 17 จุดได้ภายใน 17 คาบเวลา



รูปที่ 4.4.1 แสดงการแปลง 17 จุด [2] ในลักษณะ 3 มิติ

เมื่อพิจารณาผังงานในส่วนของการ pre-add และ post-add ของส่วนคำนวณในขั้นตอนการแปลงต่างๆ [2] จะเห็นว่าเราสามารถที่จะทำการ pre-add และ post-add ของการแปลง 3 และ 17 จุดได้โดยใช้วงจรถูกซ้ำ pre-add และ post-add ของการแปลง 5 จุด โดยจะมีการบังคับกับผลของการบวกวงจรถูกซ้ำให้เป็นศูนย์ และในขั้นของการคูณของการแปลงขนาดต่างๆ [2] เราสามารถที่จะแทนการคูณและการส่งผ่านข้อมูลโดยไม่มีการคูณได้โดยใช้วงจรถูกซ้ำ 9 ตัวซึ่งจะมีความคุ้มค่าให้ด้วย

โครงสร้างที่มีประสิทธิภาพมากที่สุด แต่จะเห็นได้ว่าการออกแบบนั้นยุ่งยากกว่าวงจรแปลงที่ได้จากการออกแบบโดยตรงจากขั้นตอนวิธีซึ่งมีค่าอัตราขนาดเร็วใกล้เคียงกัน แต่สำหรับการออกแบบวงจรแปลงให้สามารถเปลี่ยนแปลงความเร็วต่าง ๆ กัน โดยไม่ต้องการเปลี่ยนแปลงโครงสร้างของวงจรมานักก็ควรจะใช้โครงสร้างแบบย้ายตำแหน่งบัพเฟออร์ ในขณะที่ถ้าต้องการวงจรแปลงที่มีขนาดเล็กโดยไม่คำนึงถึงความเร็วในการแปลงนักโครงสร้างแบบใช้ส่วนคำนวณร่วมกันจะเป็นโครงสร้างที่เหมาะสมที่สุด

วิธีการ	ขนาดของวงจร (เกต)	minimum clock periode		ความเร็วในการแปลง		latency time		อัตราเร็ว
		(ns)	(คาบเวลา)	(ns)	(คาบเวลา)	(ns)	(คาบเวลา)	
ออกแบบโดยตรงจากขั้นตอนวิธี	67069	$4\tau_{xor(2)}$	9.16	255	2335.8	765	7007.4	1.57
ใช้วงจรคูณซ้ำ	60606	$4\tau_{xor(2)}$	9.16	255	2335.8	765	7007.4	1.42
ย้ายตำแหน่งบัพเฟออร์	180502	$4\tau_{xor(2)}$	9.16	255	2335.8	765	7007.4	4.22
ใช้ส่วนคำนวณร่วมกัน	25165	$2\tau_{inv(1)} + \tau_{and(1)}$ $6\tau_{xor(1)} + \tau_{and(2)}$ $8\tau_{xor(2)} + \tau_{xor(4)}$	39.89	395	15551.15	395	15551.15	3.91

ตารางที่ 5.1 แสดงการเปรียบเทียบขนาดและความเร็วของวงจรแปลงที่ออกแบบ

เมื่อเปรียบเทียบขนาดของวงจรแปลงที่ออกแบบได้ (แสดงดังตารางที่ 5.1) เทียบกับวงจรแปลงที่ไม่ใช้วิธีตัวประกอบปฐม (แสดงในภาคผนวก) จะเห็นว่ามีความใกล้เคียงกัน (สำหรับวงจรแปลงแบบใช้วงจรคูณซ้ำและจากการออกแบบโดยตรงจากขั้นตอนวิธี) แต่จะมีความเร็วมากกว่าประมาณ 63.75 เท่า ในขณะที่วงจรแปลงที่ออกแบบโดยใช้ส่วนคำนวณร่วมกันจะมีขนาดเล็กกว่าวงจรแปลงที่ไม่ใช้วิธีตัวประกอบปฐมประมาณ 2.8 เท่าแต่จะมีความเร็วกว่าเพียง 9.6 เท่า

จะเห็นได้ว่าวงจรแปลงที่ออกแบบโดยใช้วิธีตัวประกอบปฐมจะสามารถเพิ่มความเร็วในการแปลงได้ 9.6-63.75 เท่าโดยมีขนาดเล็กกว่า 2.8 เท่าหรือมีขนาดเท่ากัน ในขณะที่มีจำนวนการคำนวณการคูณลดลงจากเดิม 57 เท่า จากโครงสร้างวงจรแปลงที่ได้จากวิธีตัวประกอบปฐม (ยกเว้นวิธีการย้ายตำแหน่งบัพเฟออร์) ทำให้เราสามารถที่จะออกแบบวงจรแปลงด้วย FPGA, EPF10K100 ของ Altera ซึ่งสามารถออกแบบได้ 100,000 เกต ซึ่งทำให้มีวงจรเหลือพอสำหรับการออกแบบวงจรใช้งานร่วมกับอื่น ๆ ได้อีกด้วย

เอกสารอ้างอิง

[1] Richard E. Blahut, "Theory and practice of error control codes.", Addison-Wesley publishing Company, Inc., 1984

[2] พงศธร หมายดี, สมศักดิ์ ชุ่มช่วย, "วิธีการตัวประกอบปฐมเพื่อเพิ่มความเร็วของการแปลงในสนามจำกัด", วิศวกรรมลาดกระบัง ฉบับที่ 1 ปีที่ 13 หน้า 62-71, กรกฎาคม 2539

[3] Motorola, "Motorola MDA20 standard cell data", Motorola INC., Chandler, AZ, 1990

[4] S.choomchuay, "On the Implementation of Finite Field Operations", วิศวกรรมลาดกระบัง ฉบับที่ 1 ปีที่ 11 หน้า 7-17, มิถุนายน 2537

ภาคผนวก

-อักษรและสัญลักษณ์ย่อที่ใช้ในบทความ

เกต = equivalent gates

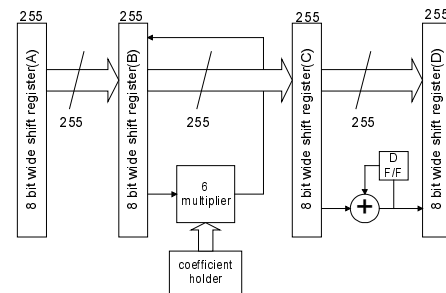
$\tau_{inv(n)}$ = propagation delay of inverter gate (fan out = n)

$\tau_{and(n)}$ = propagation delay of and gate (fan out = n)

$\tau_{xor(n)}$ = propagation delay of xor gate (fan out = n)

$\tau_{d/f(n)}$ = propagation delay of 1 input D F/F (fan out = n)

-วงจรแปลงขนาด 255 จุดในสนามจำกัด $GF(2^8)$ ซึ่งไม่ใช้วิธีตัวประกอบปฐมแสดงได้ดังรูปที่ 5.1



รูปที่ 5.1 แสดงวงจรแปลงขนาด 255 จุดใน $GF(2^8)$ ที่ไม่ใช้วิธีตัวประกอบปฐม

วงจรแปลงดังรูปที่ 5.1 จะทำการแปลงโดยจะได้ข้อมูลที่ผ่านการแปลง 1 ตัวต่อรอบการคำนวณ รอบการคำนวณเริ่มจากข้อมูลจากบัพเฟออร์ A ถูกถ่ายเทเข้าบัพเฟออร์ B ซึ่งจะทำหน้าที่ป้อนข้อมูลให้กับชุดวงจรคูณทำการคูณข้อมูลทั้ง 255 ตัวด้วยตัวคูณที่ป้อนให้โดยอุปกรณ์เก็บตัวคูณ เมื่อทำการคูณเสร็จแล้วข้อมูลทั้ง 255 ตัวจะถูกถ่ายเทไปยังบัพเฟออร์ C ซึ่งจะทำการคำนวณในขั้นตอนการบวกเพื่อให้ได้ผลลัพธ์ 1 ตัวจากการแปลงซึ่งจะนำไปเก็บไว้ที่ บัพเฟออร์ D และจะมีการเลื่อนขึ้นเพื่อรอรับผลจากการคำนวณในรอบต่อไป จะเห็นว่ากระบวนการแปลงจะทำงานเช่นนี้ไปจนครบ 255 รอบจึงจะได้ผลลัพธ์จากการแปลงทั้ง 255 ตัว การแปลงจากโครงสร้างนี้จะมีคาบเวลาสัญญาณนาฬิกาเป็น $\tau_{xor(2)}$ และจะใช้เวลาในการแปลงเท่ากับ $255 * 255 \tau_{xor(2)} = 148907.25$ ns และมีขนาดของวงจรแปลงเป็น 70385 เกต