# การสร้าง FPGA ของการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ สำหรับการบีบอัดภาพ

# FPGA Implementation of Fast 2D-DCT for Image Compression

ธนกร สุขใส[1] เผ่าภัค ศิริสุข[2] และ เอกรัฐ บุญภูงา[2]

[1]ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร

[2]ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีมหานคร

**บทคัดย่อ**

บทความนี้นำเสนอการสร้างตัวประมวลผลที่มีประสิทธิภาพสูงสำหรับการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ แบบเร็ว สำหรับการบีบอัดภาพมาตรฐาน JPEG การแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติแบบทั่วไปจะถูกแปลงให้มีการประมวลผลก่อน เพื่อป้อนให้การแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ หลังจากนั้นเอาต์พุตที่ได้จะถูกทำการประมวลผลหลัง เพื่อให้ได้เอาต์พุตที่ถูกต้องของการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ สถาปัตยกรรมที่ใช้ในการสร้างเป็นแบบขนานโดยใช้เทคนิคการทำงานแบบสายท่อเข้ามาร่วมเพื่อเพิ่มความเร็วของการประมวลผล ตัวดำเนินการที่ใช้ในการสร้างตัวประมวลผลนั้นจะใช้ 2 เทคนิคนั่นคือ เลขคณิตแบบจุดตรึงและเลขโดดมีเครื่องหมายแบบบัญญัติ เพื่อเปรียบเทียบกันทั้งทางด้านการใช้ทรัพยากรและสมรรถนะ บทความนี้จะใช้ FPGA ในการสร้างตัวประมวลผล โดยจะแสดงผลของการใช้ทรัพยากรของ FPGA รวมถึงสมรรถนะที่ได้หลังการสร้าง การทดสอบ FPGA ได้ถูกดำเนินการเพื่อยืนยันให้เห็นว่า การสร้างตัวประมวลผลของการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ สามารถทำงานได้จริงอย่างมีประสิทธิภาพ

**คำสำคัญ :** การแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ, การสร้าง FPGA, ไปล์ไลน์

**Abstract**

This paper presents implementation of an efficient hardware processor of the two-dimensional discrete cosine transform (2D- DCT). The transform comprises pre-processing, DCT and post-processing units. The pre-processing unit is utilized to arrange the input signals which are feed to DCT units. The output of DCT units is slightly modified in the post-processing unit to achieve the 2D-DCT. The architecture employed to implement the 2D-DCT processor is the parallel structure and the pipelining technique is applied to the architecture to reduce its critical path corresponding to the maximum operation frequency. The 2D-DCT processors using fixed-point and canonical signed digit (CSD) arithmetic are implemented on FPGA to investigate their resource utilization and performance. The implementation and testing results shows that the implemented processors can operate well.

**Keyword :** 2D-DCT, FPGA implementation, pipelining

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1. บทนำ

ในปัจจุบันนี้ อุปกรณ์สื่อประสมที่พกพาได้ (Mobile multimedia device) กำลังได้รับความนิยมเป็นอย่างมาก อุปกรณ์พกพาประเภทนี้ สามารถรองรับการใช้งานได้ทั้งทางด้านวีดิทัศน์ดิจิตอล และภาพนิ่ง หัวใจสำคัญของการเข้ารหัสวีดีทัศน์และภาพนิ่ง ที่มีขนาดใหญ่คือการแปลงโคไซน์ไม่ต่อเนื่อง (discrete consign transform หรือ DCT) ประโยชน์ที่สำคัญของการแปลงโคไซน์ไม่ต่อเนื่องคือการทำการบีบอัดภาพ (Image compression) เพื่อให้ภาพมีขนาดเล็กลง ซึ่งจะประหยัด การใช้กำลังงานไฟฟ้า ยืดอายุการใช้งานของแบตเตอรี่ และสามารถส่งข้อมูลผ่านระบบไร้สายได้รวดเร็วขึ้น

การแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ (2-D DCT) โดยเฉพาะอย่างยิ่งการแปลงด้วยขนาด 8x8 เป็นหนึ่งในขั้นตอนที่สำคัญของการเข้ารหัสวีดีทัศน์และภาพนิ่งแบบมาตรฐานที่ใช้ในปัจจุบัน มีบทความจำนวนมากที่ศึกษาการสร้างการแปลงโคไซน์ไม่ต่อเนื่องให้มีความรวดเร็วมากขึ้น มีบทความส่วนหนึ่งที่ได้ใช้วิธีทางอ้อมโดยการใช้วิธีการแยกตัวประกอบของแถวและคอลัมน์ (row-column decomposition) [1]-[4] สถาปัตยกรรมสำหรับการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ เหล่านี้จะคล้ายหรือประกอบด้วยส่วนประมวลผลของการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ โดยมีหน่วยความจำสำหรับเก็บข้อมูลของการสลับระหว่างข้อมูลอินพุตในแถวและคอลัมน์ การสร้างเหล่านี้มักจะใช้สถาปัตยกรรมที่ประกอบด้วยวงจรการสะสมและคูณทั่วไป (Multiplication-Accumulation หรือ MAC) [1]-[2] หรือ เลขคณิตการกระจายที่ไม่มีวงจรคูณโดยใช้ ROM (ROM-based multiplier-less distributed arithmetic) [3]-[4] หรือ การดำเนินการตรีโกณมิติ (trigonometric operations) [5] การสร้างด้วยวิธีทางอ้อมนี้จะมีความซับซ้อนในการคำนวณค่อนข้างสูงเนื่องจากไม่ได้ใช้อัลกอริทึมของการแปลงโคไซน์ไม่ต่อเนื่องแบบเร็วได้อย่างเต็มประสิทธิภาพ

บทความอีกส่วนหนึ่งนำเสนอการแปลงโคไซน์ไม่ต่อเนื่องด้วยวิธีตรงซึ่งมีความซับซ้อนในการคำนวณต่ำกว่าซึ่งมีโครงสร้างต่างจากเดิมเล็กน้อย [6]-[7] บทความเหล่านี้จะใช้อัลกอริทึมการแปลงโคไซน์ไม่ต่อเนื่องแบบเร็วซึ่งสัมพันธ์กับกราฟการไหลของข้อมูล (signal flow-graph) ซึ่งคล้ายกับการแปลงฟูริเยร์ไม่ต่อเนื่อง อย่างไรก็ตามการสร้างการแปลงโคไซน์ไม่ต่อเนื่องด้วยเทคนิคตรงนี้จะทำให้ใช้ทรัพยากรในการสร้างสูง ตัวอย่างหนึ่งของการใช้เทคนิคตรงในการแปลงโคไซน์ไม่ต่อเนื่องคือบทความ [8] ซึ่งนำเสนอการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ โดยบทความนี้เน้นการนำเสนอการแปลงซึ่งจะทำให้การแปลงมีความรวดเร็ว อย่างไรก็ตามยังไม่มีบทความใดศึกษาการสร้างการแปลงโคไซน์ไม่ต่อเนื่องด้วยเทคนิคดังกล่าวโดยใช้อุปกรณ์สารกึ่งตัวนำแถวลำดับเกทที่โปรแกรมได้ (field programmable gate array หรือ FPGA) เพื่อยืนยันความเร็วในการประมวลผลรวมถึงการใช้ทรัพยากรของการสร้าง

บทความนี้นำเสนอการสร้างการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ที่นำเสนอในบทความ [8] ด้วย FPGA โดยใช้การสร้างด้วยสถาปัตยกรรมแบบขนาน มีการนำเทคนิคการทำงานแบบสายท่อ (pipelining technique) มาใช้เพื่อเพิ่มความเร็วในการประมวลผลของการแปลงและยังทำการเปรียบเทียบสมรรถนะ และประสิทธิภาพในการคำนวณของ FPGA ที่สร้าง เมื่อทำการสร้างด้วยการดำเนินการทางคณิตศาสตร์ในรูปแบบเลขคณิตแบบจุดตรึง (fixed-point arithmetic) และใช้เลขโดดมีเครื่องหมายแบบบัญญัติ (canonical signed digit หรือ CSD arithmetic)

## 2. การบีบอัดภาพมาตรฐาน JPEG

การบีบอัดภาพตามมาตรฐาน JPEG [9] ได้รับการพัฒนาจากกลุ่ม Joint Photographic Expert Group ในการบีบอัดภาพ โดยใช้การแปลงโคไซน์ไม่ต่อเนื่องนั้นเป็นการแปลงข้อมูลภาพให้อยู่ในย่านความถี่เชิงพื้นที่ ซึ่งโดยปรกติแล้วพิกเซลที่ใกล้เคียงกันในภาพ จะมีความคล้ายคลึงกันโดยมีสีที่แตกต่างกันเล็กน้อย การแปลงโคไซน์ไม่ต่อเนื่องนั้น จะเป็นการจับกลุ่มของแอมปลิจูดที่สูงกว่า ในย่านความถี่ที่ต่ำกว่า ดังนั้นจึงสามารถตัดย่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่สูงออกไปได้ซึ่งทำให้มีอัตราการบีบอัดภาพได้สูง ซึ่งจะทำให้สูญเสียคุณภาพของภาพที่บีบอัดเพียงเล็กน้อย การบีบอัดภาพตามมาตรฐาน JPEG ได้รับความนิยมมาก สำหรับภาพถ่ายเนื่องจากภาพวาดโดยทั่วไป จะมีข้อมูลย่านความถี่สูงเป็นจำนวนมาก และเมื่อมีการคลายการบีบอัดภาพ ก็จะทำให้ได้ภาพที่มีความเพี้ยนจากต้นฉบับเดิมค่อนข้างมาก

เราสามารถแบ่งการบีบอัดข้อมูลเป็น 5 ขั้นตอนด้วยกัน คือ 1. การแปลงเชิงพื้นที่ของสี (Color space conversion) 2. การชักตัวอย่างให้ต่ำลง (Down sampling) 3. การแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ 4. การแบ่งนับ (quantization) 5. การเข้ารหัสเอนโทรปี (Entropy coder) ขั้นตอนที่ 1 และ 2 จะใช้เฉพาะในกรณีที่บีบอัดภาพสีเท่านั้น การแปลงสีเป็นการเปลี่ยนภาพสีจาก โหมดการผสมสี แดง เขียว น้ำเงิน (RGB mode) เป็นความสว่าง (Luminance) ความพร่าสี (Chrominance) หรือเป็นการแทนแบบ YCbCr การทำให้อัตราการชักตัวอย่างต่ำลงเป็นการลดอัตราการสุ่มตัวอย่างของข้อมูลสี เนื่องจากส่วนประกอบของ chrominance มีผลต่อการมองเห็นของตามนุษย์ การดำเนินการแบ่งนับนั้น จะตัดความถี่สูงและสัมประสิทธิ์ที่มีค่าน้อยๆออกไปและการเข้ารหัสเอนโทรปีใช้เข้ารหัสแบบ run-length encoding (RLE) Huffman หรือ variable length coding (VLC) และการเข้ารหัสแบบอนุพันธ์ (differential coding) เพื่อลดจำนวนการใช้บิตที่ใช้แทนภาพ

การนำภาพกลับมาเหมือนเดิมนั้น ก็จะใช้ขั้นตอนการย้อนกลับ ในทำนองเดียวกันกับการบีบอัดโดยมีการถอดรหัสเอนโทรปี (Entropy decoding) การคลายการแบ่งนับ (Dequatization) การแปลงโคไซน์ไม่ต่อเนื่องผกผัน 2 มิติ (Inverse discrete consign transform) การทำให้อัตราการชักตัวอย่างสูงขึ้น (Upsampling) และ ทำกลับการแปลงสี

## 3. การแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติแบบเร็ว

ในหัวข้อนี้ จะอธิบายถึงการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ แบบเร็ว ซึ่งได้ถูกนำเสนอในบทความ [8] การแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติแบบทั่วไปจะถูกปรับให้เป็นการแปลงซึ่งใช้การแปลงโคไซน์ 1 มิติ โดยมีการประมวลผลก่อน (preprocessing) สำหรับจัดกลุ่มของสัญญาณอินพุต และเอาต์พุตที่ได้จากการแปลง ก็จะถูกทำการประมวลผลหลัง (post processing) พิจารณาข้อมูลอินพุต 2 มิติ $\{x_{ij}: i,j = 0, 1, \ldots, N\text{-}1\}$ ที่ได้จากภาพ สามารถแสดงเอาต์พุต ของการแปลงโคไซน์ 2 มิติ $\{y_{mn}: m, n= 0, 1, \ldots, N\text{-}1\}$ ด้วยสมการ

$$y_{mn} = \frac{4}{N^2}u(m)u(n) \cdot \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x_{ij}\cos\frac{(2i+1)m}{2N}\pi\cos\frac{(2j+1)n}{2N}\pi \qquad (1)$$

โดยที่

$$u(m) = \begin{cases} 1/\sqrt{2}\,, & m = 0 \\ 1\,, & Otherwise. \end{cases}$$

เมื่อทำการแปลงโคไซน์ 2 มิติตามที่นำเสนอในบทความ [8] เอาต์พุตที่ได้หลังการแปลง สามารถแสดงได้ดังสมการ

$$y_{mn} = \begin{cases} 1/2\sum\limits_{\substack{p=1 \\ (p:odd)}}^{N-1}\left[\sum\limits_{i=0}^{N-1}\left(x_{ij(p;a)} + x_{ij(p;b)}\right)\cdot \cos\frac{(2i+1)(m+np)}{2N}\pi\right. \\ \left. + \sum\limits_{i=0}^{N-1}\left(x_{ij(p;a)} + x_{ij(p;b)}\right)\cdot \cos\frac{(2i+1)(m-np)}{2N}\pi\right], \\ \qquad \text{เมื่อ } n \text{ เป็นเลขคู่} \\ 1/2\sum\limits_{\substack{p=1 \\ (p:odd)}}^{N-1}\left[\sum\limits_{i=0}^{N-1}(-1)^{qpi}\left(x_{ij(p;a)} - x_{ij(p;b)}\right)\cdot \cos\frac{(2i+1)(m+np)}{2N}\pi\right. \\ \left. + \sum\limits_{i=0}^{N-1}(-1)^{qpi}\left(x_{ij(p;a)} - x_{ij(p;b)}\right)\cdot \cos\frac{(2i+1)(m-np)}{2N}\pi\right] \\ \qquad \text{เมื่อ } n \text{ เป็นจำนวนคี่} \end{cases} \qquad (2)$$

เมื่อ

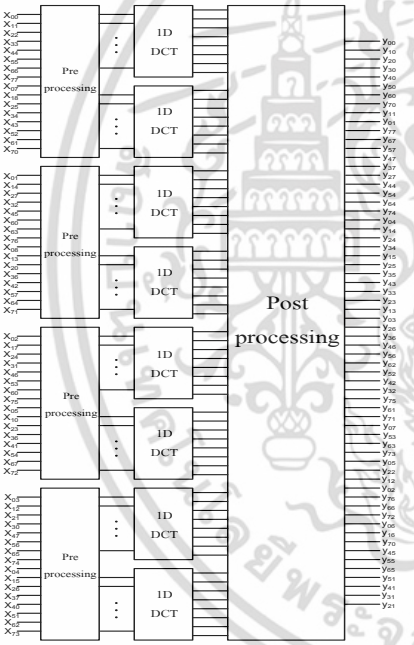
$$f_{pl} = \sum_{i=0}^{N-1}\left(x_{ij(p;a)} + x_{ij(p;b)}\right)\cos\frac{(2i+1)l}{2N}\pi \qquad (3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$g_{pl} = \sum_{i=0}^{N-1} \frac{(-1)^{q_{pi}} \left( x_{ij(p;a)} - x_{ij(p;b)} \right) \bullet}{\cos \frac{(2i+1)l}{2N} \pi} \qquad (4)$$

หลังจากที่ได้ $f_{pl}$ และ $g_{pl}$ แล้ว พิจารณาการบวกและตัวดำเนินการอื่นๆ ที่ต้องการสำหรับการคำนวณ DCT ขนาด $N \times N$ จากสมการ (2) และข้อกำหนดในสมการ (3) และ (4) จะเห็นได้ว่าเอาต์พุต $y_{mn}$ จะแสดงในรูปแบบของการรวมกันของ $f_{pl}$ และ $g_{pl}$ บทความนี้นำเสนอการสร้างการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติแบบเร็ว เมื่อ $N$=8

## 4. การออกแบบและสร้างโคไซน์ไม่ต่อเนื่อง 2 มิติด้วย FPGA




**รูปที่ 1** โครงสร้างพื้นฐานการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ แบบเร็ว

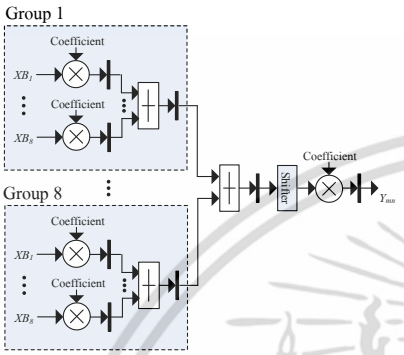
ตามที่ได้กล่าวไปข้างต้นแล้วนั้น บทความนี้จะนำเสนอการสร้างตัวประมวลผล สำหรับใช้ในการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ โดยการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ที่จะนำมาสร้างนั้นได้อธิบายไว้ในหัวข้อที่ 3 รูปที่ 1 แสดงถึงโครงสร้างพื้นฐานของการสร้าง โดยมีส่วนประกอบที่สำคัญคือ 1. ส่วนประมวลผลก่อนคือ การจัดเรียงรูปแบบสำหรับอินพุต ของการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ การจัดเรียงนี้เป็นการบวกหรือลบกันของสัญญาณอินพุต $x_{ij}$ ความสัมพันธ์ของการบวกหรือลบกันดังกล่าวแสดงในสมการ (3) และ (4) 2. การแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ เอาต์พุตที่ได้จากการจัดเรียงรูปแบบของอินพุตใหม่ ก็จะถูกส่งให้ตัวประมวลผลย่อย (processing element) ของแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ ขนาด $N$=8 เอาต์พุตที่ได้จากการแปลงถูกแสดงด้วย $f_{pl}$ และ $g_{pl}$ ดังแสดงในสมการที่ (3) และ (4) 3. การประมวลผลหลังคือการประมวลผลหลังจากที่ได้ทำการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติแล้ว โดยการประมวลผลนี้เป็นความสัมพันธ์ของการบวกหรือลบของฟังก์ชั่น $f_{pl}$ และ $g_{pl}$ ซึ่งจะทำให้ได้เอาต์พุตของการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ดังที่ได้แสดงไว้ในสมการที่ (2)

ในการสร้างตัวประมวลผล สำหรับการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ด้วยโครงสร้างที่แสดงในรูปที่ 1 นั้น บทความนี้จะทำการสร้างด้วยสถาปัตยกรรมแบบขนาน ซึ่งก็จะทำให้ได้การประมวลผลที่รวดเร็วตามหลักการพื้นฐานของการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติแบบเร็ว ซึ่งถูกเสนอไว้ในบทความ [8] แต่ยังไม่ได้ถูกนำไปพิจารณาร่วมกับการสร้างด้วย FPGA อย่างจริงจัง ในการสร้างสถาปัตยกรรมแบบขนานนั้น ได้มีการแทรกรีจิสเตอร์ (register) หลังตัวดำเนินการไว้เป็นช่วงๆ เพื่อลดเส้นทางวิกฤติ (critical path) ซึ่งจะส่งผลทำให้ความถี่ที่สามารถใช้งานได้สูงสุด (maximum operating frequency) เพิ่มขึ้น การดำเนินการในแบบดังกล่าว คือเทคนิคการทำงานแบบสายท่อ (pipelining technique) นอกจากเราจะพิจารณาการสร้างด้วยสถาปัตยกรรมแบบขนานด้วยเทคนิค pipelining แล้ว เรายังศึกษาการสร้างFPGA ด้วยการดำเนินการทางคณิตศาสตร์ในรูปแบบเลขคณิตแบบจุดตรึง (fixed-point arithmetic) และเลขโดดมีเครื่องหมายแบบบัญญัติ (canonical signed digit หรือ CSD arithmetic

รูปที่ 2 แสดงสถาปัตยกรรมของการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ ที่ใช้ในการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ จากรูปประกอบด้วย วงจรบวก ลบ คูณ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

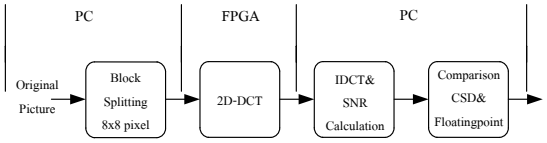
รีจิสเตอร์สำหรับลดเส้นทางวิกฤติ ตัวประมวลผลการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติที่สร้างด้วย fixed-point และ CSD arithmetic (การสร้างนี้ไม่รวมวงจรสำหรับสร้างสัมประสิทธิ์) บน FPGA ยี่ห้อ Xilinx รุ่น XC4VLX200FF1513-10ใช้เกทสมมูลทั้งหมด 1,060,037 และ 1,042,808 ตามลำดับ




**รูปที่ 2** สถาปัตยกรรมการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ

ตัวประมวลผล DCT 1 มิตินี้จะถูกนำไปใช้สร้างตัวประมวลผล DCT 2 มิติ โดยการสร้างส่วนประมวลผลก่อนและหลังนั้นก็ใช้ fixed-point และ CSD arithmetic เช่นเดียวกัน โดยในการสร้างนั้นก็จะแทรกรีจิสเตอร์หลังการคูณและการบวกการสร้างตัวประมวลผลโคไซน์ไม่ต่อเนื่อง 2 มิติ จะสร้างโดยใช้ FPGA รุ่นเดียวกันกับที่กล่าวข้างต้นและขนาดความยาวคำ (word length) สำหรับอินพุตและเอาต์พุตของตัวประมวลผลมีขนาด 8 บิต ตารางที่ 1 แสดงถึงการใช้ทรัพยากรและสมรรถนะของ FPGA ที่ถูกสร้าง จากตารางจะสังเกตได้ว่า ถึงแม้ทรัพยากรสำหรับใช้ในการสร้าง FPGA ซึ่งได้แก่ จำนวนสไลซ์ จำนวนสไลซ์ของฟลิปฟลอบ (Slice Flip Flops) และจำนวนตารางค้นหา (look-up tables หรือ LUTs) ด้วยการใช้ CSD arithmetic จะน้อยกว่าทรัพยากรที่ใช้สร้างด้วย fixed-point arithmetic เพียงเล็กน้อย แต่ความถี่ที่สามารถใช้งานได้สูงสุด เพิ่มขึ้นจาก 140 MHz เป็น 164 MHz หรือเพิ่ม 17 %

## 5. ผลการทดลอง




**รูปที่ 3** ขั้นตอนทดสอบ FPGA สำหรับการบีบอัดภาพ

เพื่อให้เห็นถึงความสามารถของตัวประมวลผลการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ที่ได้สร้างด้วย FPGA ดังที่ได้อธิบายไว้ข้างต้นนั้น เราจะทดสอบความสามารถด้วยการทดสอบในกระบวนการบีบอัดภาพดังรูปที่ 3 ที่แสดงถึงขั้นตอนของการบีบอัดข้อมูลภาพ ที่ใช้ในการทดสอบภาพขนาด 128×128 พิกเซล ถูกแปลงให้อยู่ในระดับสีเทาหรือเกรย์สเกล (Gray Scale) หลังจากนั้นถูกนำมาแบ่งเป็นส่วนย่อย ๆ ที่มีขนาดเมตริกซ์เท่ากับ 8 × 8 พิกเซล ขั้นตอนเหล่านี้ได้ถูกทำโดยใช้ภาษาระดับสูง (high level language) ด้วยคอมพิวเตอร์และมัลติเพล็กซ์ข้อมูลจัดเรียงข้อมูลในรูปแบบอนุกรม RS232 แล้วส่งต่อให้ FPGA ที่ได้ออกแบบไว้ ซึ่งทำหน้าที่แปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ก่อนที่จะแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ข้อมูลที่รับเข้ามาจะถูกจัดเรียงให้เป็นแบบขนานและแก้มัลติเพล็กซ์ ข้อมูลภาพขนาด 8×8 พิกเซลนี้จะถูกแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ จนกระทั่งครบทั้ง 8 ชุด ซึ่งเท่ากับภาพที่ใช้ในการทดสอบ การทำซ้ำลักษณะแบบนี้ใช้การมัลติเพล็กซ์ข้อมูลทางขาเข้า และแก้มัลติเพล็กซ์ข้อมูลทางขาออก ซึ่งข้อมูลที่ได้จะถูกจัดเรียงเป็นแบบอนุกรม RS232 อีกครั้งและป้อนกลับคืนให้คอมพิวเตอร์ เพื่อคำนวณการแปลงโคไซน์ไม่ต่อเนื่องผกผัน เพื่อที่จะแสดงภาพที่ได้หลังจากผ่านกระบวนการบีบอัดภาพ

รูปที่ 4 (ก) และ (ข)แสดงภาพต้นฉบับที่ใช้ในการบีบอัดและภาพที่ได้หลังจากการบีบอัดด้วยการใช้การคำนวณแบบ floating-point ในโปรแกรมคอมพิวเตอร์ รูปที่ 6 (ก) และ (ข) แสดงภาพที่ได้หลังการบีบอัดด้วย FPGA โดยใช้ fixed point และ CSD arithmetic ตามลำดับ จากรูปแสดงให้เห็นว่าผลที่ได้จากตัวประมวลผลการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ ที่สร้างโดยใช้ fixed point และ CSD arithmetic นั้น มีผลที่ได้ใกล้เคียงกันกับผลที่ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจำลอง ในแบบจุดลอยตัว (floating point arithmetic) 32 บิต และมีความแตกต่างในระดับที่ยอมรับได้ เมื่อเทียบกับรูปต้นฉบับที่ใช้ในการทดสอบการบีบอัด รูปที่4 (ก) นอกจากนี้เราสามารถสรุปได้ว่าตัวประมวลผลการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ สามารถใช้งานจริงได้เป็นอย่างดี ซึ่งเหมาะสำหรับการประมวลผลที่ต้องการความเร็วสูง และมีขนาดกะทัดรัด เช่น ในการส่งภาพที่ได้หลังจากการถ่ายภาพจากกล้องและต้องการส่งผ่านระบบไร้สาย

(ก) (ข)

**รูปที่ 4** (ก) ภาพต้นฉบับที่ใช้ในการทดสอบ (ข) Floating-point arithmetic

(ก) (ข)

**รูปที่ 5** ภาพที่ได้หลังการบีบอัดภาพด้วยการแปลโคไซน์ไม่ต่อเนื่อง 2 มิติ แบบเร็วด้วย (ก) CSD (ข) Fixed-point

## 6. สรุป

บทความนี้ ได้นำเสนอการสร้างตัวประมวลผลของการแปลงโคไซน์ไม่ต่อเนื่อง 2 มิติ แบบเร็วด้วย FPGA โดยการสร้างนั้นมีส่วนประกอบหลักได้แก่ ตัวประมวลผลก่อน ตัวประมวลผลย่อยของการแปลงโคไซน์ไม่ต่อเนื่อง 1 มิติ และตัวประมวลผลหลัง ในการสร้างตัวประมวลผล ได้มีการเปรียบเทียบการคำนวณทางคณิตศาสตร์ของตัวดำเนินการระหว่าง fixed-point และ CSD arithmetic พร้อมทั้งได้มีการแสดงให้เห็นผลการสร้างตัวประมวลผลระหว่าง 2 เทคนิคนี้ โดยมีการใช้ทรัพยากรที่ใกล้เคียงกัน แต่ความถี่ที่สามารถใช้งานได้สูงสุดของการสร้างด้วย CSD arithmetic จะสูงกว่า fixed-point arithmetic 17% ซึ่งผลการทดสอบแสดงให้เห็นว่า FPGA ที่สร้างสามารถใช้ได้ในการบีบอัดภาพมาตรฐาน JPEG

## 7. เอกสารอ้างอิง


[1] A. Madisetti and A. N. Willson, “A 100 MHz 2-D 8x8 DCT/IDCT processor for HDTV applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 5, pp. 158–165, Apr. 1995.

[2] D. Stawecki andW. Li, “DCT/IDCT processor design for high data rate image coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 2, pp. 135–146, June 1992.

[3] M. T. Sun, T. C. Chen, and A. M. Gottlieb, “VLSI implementation of a 16×16 discrete cosine transform,” IEEE Trans. Circuits Syst., vol. 36, pp. 610–617, Apr. 1989.

[4] S. Uramoto et al., “A 100-MHz 2-D discrete cosine transform core processor,” IEEE J. Solid-State Circuits, vol. 27, pp. 492–498, Apr. 1992.

[5] J. H. Hsiao, L. G. Chen, T. D. Chiueh, and C. T. Chen, “High throughput CORDIC-based systolic array design for the discrete cosine transform,” IEEE Trans. Circuits Syst. Video Technol., vol. 5, pp. 218–225, June 1995.

[6] Y. Sungwook and E.E.S. Jr., “DCT implementation with distributed arithmetic,” IEEE Trans. Comput., vol.50, no.9, pp.985-991, 2001.

[7] N. I. Cho, I. D. Yun, and S. U. Lee, “On regular structure for the fast 2-D DCT algorithm,” IEEE Trans. Circuits Syst. II, vol. 40, pp. 259–266, Apr. 1993.

[8] Nam IK Cho and Sang Uk Lee, “Fast Algorithm and Implementation of 2-D Discrete Cosine Transform,” IEEE Transaction on circuit and system, Vol.38, no.3, March 1991.

[9] E. Magli, “The JPEG Family of Coding Standard,” Part of “Document and Image Compression,” New York: Taylor and Francis, 2004.


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้