บทที่ 5

การคำนวณ 2-D IDCT โดยใช้วงจรคำนวณแบบโครงสร้างผีเสื้อ

ในบทนี้จะกล่าวถึงการออกแบบวงจรคำนวณ 2-D IDCT โดยใช้วงจรคำนวณแบบโครงสร้างผีเสื้อ หรือ Butterfly Structure ของสมการ 1-D IDCT 2 ขั้นตอนด้วยกัน มีการอธิบายแผนภาพบล็อกของวงจรและอธิบายการทำงานของวงจรโดยการยกตัวอย่างข้อมูล

5.1 การออกแบบวงจร 2-D IDCT

การออกแบบ 2-D IDCT โดยใช้วงจรคำนวณแบบโครงสร้างผีเสื้อนั้นใช้โครงสร้างการคำนวณเช่นเดียวกับการใช้ MAC (Multiply Accumulator) โดยการแตกสมการ 2-D IDCT ออกเป็นการคำนวณ 1-D IDCT แต่ละหลักก่อน แล้วจึงนำผลลัพธ์ไปคำนวณในแต่ละแถวต่อไป ซึ่งจากสมการ (7) ในบทที่ 3

$$\left[s(x,y)\right] = \left[C(x,u)\right]\left[S(u,v)\right]\left[C(y,v)\right]^T \qquad (1)$$

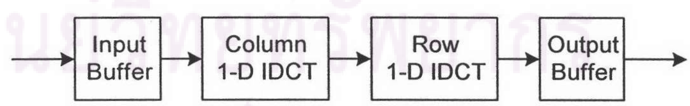
ทำการคำนวณในแต่ละหลักก่อน

$$\left[A(x,v)\right] = \left[C(x,u)\right]\left[S(u,v)\right] \qquad (2)$$

นำผลลัพธ์ที่ได้ คือ $\left[A(x,v)\right]$ไปคำนวณอีกครั้งหนึ่ง ในแต่ละแถว

$$\left[s(x,y)\right] = \left[A(x,v)\right]\left[C(y,v)\right]^T \qquad (3)$$

ซึ่งขั้นตอนการทำงานเป็นไปตามผังงาน



รูปที่ 5.1 ผังงานของ 2-D IDCT

แต่แทนที่จะใช้ MAC ในการคำนวณผลคูณของเมตริกซ์ เราใช้คุณสมบัติสมมาตรของวงกลมในการแก้สมการ 1-D IDCT ดังสมการ (2) ในบทที่ 3

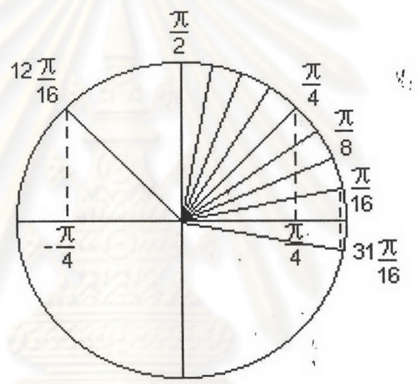
$$s(x) = \sum_{u=0}^{7} \frac{C(u)}{2} S(u)\cos\left(\frac{(2x+1)u\pi}{16}\right) \qquad (4)$$

โดยกำหนดให้

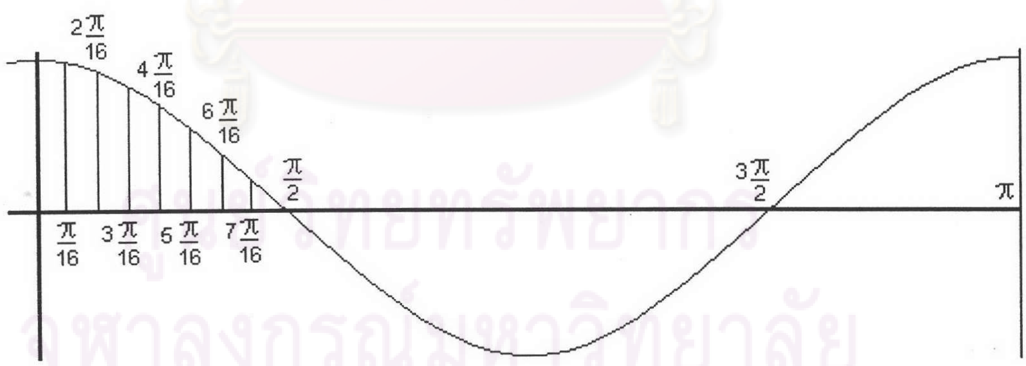
$$C(u) = \frac{1}{\sqrt{2}} \quad \text{สำหรับ } u = 0$$

$$C(u) = 1 \quad \text{สำหรับ } u > 0$$

ค่าโคไซน์ที่นำมาคูณนั้นมีค่าตั้งแต่ cos $\frac{\pi}{16}$ ถึง cos $\frac{105\pi}{16}$ ซึ่งจะมีมุมอยู่หลายมุมที่มีค่าโคไซน์เท่ากัน ดังรูปวงกลมหนึ่งหน่วยที่แสดงในรูปที่ 5.2 (a) และ กราฟของฟังก์ชันโคไซน์ที่แสดงในรูป 5.2 (b) ถ้าเราแบ่งวงกลมออกเป็นสี่ส่วน ค่าโคไซน์จะมีค่าแตกต่างกันเฉพาะหนึ่งในสี่ของวงกลมหรือตั้งแต่มุม 0 ถึง $\frac{\pi}{2}$ เท่านั้น จะเห็นว่า cos $\frac{\pi}{16}$ ค่าเท่ากับ cos $\frac{31\pi}{16}$ (ค่าของโคไซน์วัดที่แกนนอน) และ cos $\frac{12\pi}{16}$ เท่ากับ cos $\frac{\pi}{4}$ แต่เป็นค่าลบ เราจึงสามารถแทน cos $\frac{31\pi}{16}$ ด้วย cos $\frac{\pi}{16}$ และใช้ - cos $\frac{4\pi}{16}$ แทน cos $\frac{12\pi}{16}$ เป็นต้น



(a)



(b)

รูปที่ 5.2 ค่าของโคไซน์ที่มุมต่าง ๆ

เมื่อเราเขียนสมการ (2) ใหม่โดยแตกออกเป็น 8 สมการย่อยดังนี้ (<u>หมายเหตุ</u> $C_k = \mathrm{Cos}\left(\frac{k\pi}{16}\right)$)

$$s(0) = \frac{1}{2}\left(\frac{1}{\sqrt{2}}S(0) + C_1S(1) + C_2S(2) + C_3S(3) + C_4S(4) + C_5S(5) + C_6S(6) + C_7S(7)\right) \quad \text{.......... (5)}$$

$$s(1) = \frac{1}{2}\left(\frac{1}{\sqrt{2}}S(0) + C_3S(1) + C_6S(2) - C_7S(3) - C_4S(4) - C_1S(5) - C_2S(6) - C_5S(7)\right) \quad \text{........... (6)}$$

$$s(2) = \frac{1}{2}\left(\frac{1}{\sqrt{2}} S(0) + C_5S(1) - C_6S(2) - C_1S(3) - C_4S(4) + C_7S(5) + C_2S(6) + C_3S(7) \right) \quad \ldots\ldots\ldots\ldots (7)$$

$$s(3) = \frac{1}{2}\left(\frac{1}{\sqrt{2}} S(0) + C_7S(1) - C_2S(2) - C_5S(3) + C_4S(4) + C_3S(5) - C_6S(6) - C_1S(7) \right) \quad \ldots\ldots\ldots\ldots (8)$$

$$s(4) = \frac{1}{2}\left(\frac{1}{\sqrt{2}} S(0) - C_7S(1) - C_2S(2) + C_5S(3) + C_4S(4) - C_3S(5) - C_6S(6) + C_1S(7) \right) \quad \ldots\ldots\ldots\ldots (9)$$

$$s(5) = \frac{1}{2}\left(\frac{1}{\sqrt{2}} S(0) - C_5S(1) - C_6S(2) + C_1S(3) - C_4S(4) - C_7S(5) + C_2S(6) - C_3S(7) \right) \quad \ldots\ldots\ldots\ldots (10)$$

$$s(6) = \frac{1}{2}\left(\frac{1}{\sqrt{2}} S(0) - C_3S(1) + C_6S(2) + C_7S(3) - C_4S(4) + C_1S(5) - C_2S(6) + C_5S(7) \right) \quad \ldots\ldots\ldots\ldots (11)$$

$$s(7) = \frac{1}{2}\left(\frac{1}{\sqrt{2}} S(0) - C_1S(1) + C_2S(2) - C_3S(3) + C_4S(4) - C_5S(5) + C_6S(6) - C_7S(7) \right) \quad \ldots\ldots\ldots\ldots (12)$$

เมื่อพิจารณาสมการ (5) ถึงสมการ (12) จะสังเกตได้ว่ามีการคูณกันที่ซ้ำกันหลายคู่ในสมการ ยกตัวอย่างเช่น $C_4S(4)$ จะปรากฏอยู่ในทุกสมการ หรือ $C_6S(6)$ อยู่ในสมการ (5) สมการ (8) สมการ (9) และสมการ (12) นอกจากนี้แล้วยังมีอีกหลายคู่ที่ไม่ได้กล่าวถึง ถ้าเราสามารถดึงผลคูณร่วมเหล่านี้ออกมาทำการคำนวณเพียงครั้งเดียว แล้วเอาเฉพาะผลลัพธ์ไปใช้เราก็สามารถประหยัดเวลา และประหยัดวงจรคำนวณผลคูณได้ จากสมการ (5) ถึง สมการ (12) สามารถจัดรูปสมการใหม่และแทนค่า $\frac{1}{\sqrt{2}}$ ในสมการด้วย $\cos \frac{4\pi}{16}$ ได้ดังนี้

$$2s(0) = (C_4S(0) + C_4S(4)) + (C_2S(2) + C_6S(6)) + [(C_1S(1) + C_7S(7)) + (C_3S(3) + C_5S(5))] \quad \ldots\ldots\ldots (13)$$

$$2s(1) = (C_4S(0) - C_4S(4)) + (C_6S(2) - C_2S(6)) + [(C_3S(1) - C_5S(7)) - (C_1S(3) + C_7S(5))] \quad \ldots\ldots\ldots (14)$$

$$2s(2) = (C_4S(0) - C_4S(4)) - (C_6S(2) - C_2S(6)) + [(C_5S(1) + C_3S(7)) - (C_1S(3) - C_7S(5))] \quad \ldots\ldots\ldots (15)$$

$$2s(3) = (C_4S(0) + C_4S(4)) - (C_2S(2) + C_6S(6)) + [(C_7S(1) - C_1S(7)) + (C_5S(3) - C_3S(5))] \quad \ldots\ldots\ldots (16)$$

$$2s(4) = (C_4S(0) + C_4S(4)) - (C_2S(2) + C_6S(6)) - [(C_7S(1) - C_1S(7)) + (C_5S(3) - C_3S(5))] \quad \ldots\ldots\ldots (17)$$

$$2s(5) = (C_4S(0) - C_4S(4)) - (C_6S(2) - C_2S(6)) - [(C_5S(1) + C_3S(7)) - (C_1S(3) - C_7S(5))] \quad \ldots\ldots\ldots (18)$$

$$2s(6) = (C_4S(0) - C_4S(4)) + (C_6S(2) - C_2S(6)) - [(C_3S(1) - C_5S(7)) - (C_1S(3) + C_7S(5))] \quad \ldots\ldots\ldots (19)$$

$$2s(7) = (C_4S(0) + C_4S(4)) + (C_2S(2) + C_6S(6)) - [(C_1S(1) + C_7S(7)) + (C_3S(3) + C_5S(5))] \quad \ldots\ldots\ldots (20)$$

เมื่อพิจารณาดูจากสมการ (13) ถึง สมการ (20) จะเห็นได้ว่าสมการทั้ง 8 มีการจัดกลุ่มอย่างเป็นระบบ ถ้าแบ่งสมการออกเป็น 2 ซีกเท่าๆ กันคือแต่ละซีกจะมีการคูณกัน 4 คู่ ในซีกซ้ายนั้นจะมีการคูณกันน้อยกว่าเช่น S(0) และ S(4) จะคูณกับ C_4 เท่านั้น S(2) และ S(6) จะคูณกับ C_2 หรือ C_6 เท่านั้น แต่ในซีกขวาทุกๆค่าจะต้องคูณกับสัมประสิทธิ์โคไซน์ถึง 4 ค่าด้วยกัน เราจึงพยายามหาความสัมพันธ์ของ C_1 C_3 C_5 และ C_7

เป็นที่ทราบกันดีว่า $cos\ \frac{4\pi}{16} = \frac{\sqrt{2}}{2}$ และจากสูตร

$$cos\ \frac{A}{2} = \sqrt{\frac{1+\cos A}{2}} \quad \text{......................} \quad (21)$$

$$sin\ \frac{A}{2} = \sqrt{\frac{1-\cos A}{2}} \quad \text{......................} \quad (22)$$

เราสามารถหาค่าของ $cos\ \frac{2\pi}{16}$ และค่าของ $cos\ \frac{\pi}{16}$ จากสมการ (21) และ สมการ (22) ได้ดังนี้

$$cos\ \frac{2\pi}{16} = \sqrt{\frac{1+\frac{\sqrt{2}}{2}}{2}} = \frac{\sqrt{2+\sqrt{2}}}{2}$$

$$cos\ \frac{\pi}{16} = \sqrt{\frac{1+\frac{\sqrt{2+\sqrt{2}}}{2}}{2}} = \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2}$$

$$sin\ \frac{2\pi}{16} = \sqrt{\frac{1-\frac{\sqrt{2}}{2}}{2}} = \frac{\sqrt{2-\sqrt{2}}}{2}$$

$$sin\ \frac{\pi}{16} = \sqrt{\frac{1-\frac{\sqrt{2+\sqrt{2}}}{2}}{2}} = \frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}$$

ซึ่ง $sin\ \frac{2\pi}{16} = cos\ \frac{6\pi}{16}$ และ $sin\ \frac{\pi}{16} = cos\ \frac{7\pi}{16}$ นั่นเอง

และจากสูตรฟังก์ชันของมุมบวกกันและลบกัน

cos (A+B) = cos A cos B - sin A sin B .... (23)

cos (A-B) = cos A cos B + sin A sin B .... (24)

ดังนั้นเราได้ว่า

$$cos\ \frac{3\pi}{16} = cos\left(\frac{4\pi}{16} - \frac{\pi}{16}\right)$$

$$= \frac{\sqrt{2}}{2}\ \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} + \frac{\sqrt{2}}{2}\ \frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}$$

$$= \frac{\sqrt{4+2\sqrt{2+\sqrt{2}}} + \sqrt{4-2\sqrt{2+\sqrt{2}}}}{4}$$

$$= \frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2}$$

$$\cos\frac{5\pi}{16} = \cos\left(\frac{4\pi}{16}+\frac{\pi}{16}\right)$$

$$= \frac{\sqrt{2}}{2}\,\frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} - \frac{\sqrt{2}}{2}\,\frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2}$$

$$= \frac{\sqrt{4+2\sqrt{2+\sqrt{2}}}-\sqrt{4-2\sqrt{2+\sqrt{2}}}}{4}$$

$$= \frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2}$$

สรุปค่าของโคไซน์ที่มุมต่างๆ และความสัมพันธ์ที่ได้

$$\cos\frac{\pi}{16} = \frac{\sqrt{2+\sqrt{2+\sqrt{2}}}}{2} = \frac{C_3+C_5}{\sqrt{2}}$$

$$\cos\frac{3\pi}{16} = \frac{\sqrt{2+\sqrt{2-\sqrt{2}}}}{2} = \frac{C_1+C_7}{\sqrt{2}}$$

$$\cos\frac{5\pi}{16} = \frac{\sqrt{2-\sqrt{2-\sqrt{2}}}}{2} = \frac{C_1-C_7}{\sqrt{2}}$$

$$\cos\frac{7\pi}{16} = \frac{\sqrt{2-\sqrt{2+\sqrt{2}}}}{2} = \frac{C_3-C_5}{\sqrt{2}}$$

จัดรูปสมการ (13) ถึง (20) ด้วยความสัมพันธ์ข้างต้นได้ดังนี้

$$2s(0) = (C_4S(0) + C_4S(4)) + (C_2S(2) + C_6S(6)) + [(C_1S(1) + C_7S(7)) + (C_3S(3) + C_5S(5))] \quad \text{........... (25)}$$

$$2s(1) = (C_4S(0) - C_4S(4)) + (C_6S(2) - C_2S(6)) + \left[\frac{(C_1S(1)+C_7S(7))-(C_3S(3)+C_5S(5))}{\sqrt{2}} + \frac{(C_7S(7)-C_1S(1))-(C_5S(5)+C_3S(3))}{\sqrt{2}}\right] \quad \text{........... (26)}$$

$$2s(2) = (C_4S(0) - C_4S(4)) - (C_6S(2) - C_2S(6)) + \left[\frac{(C_1S(1)+C_7S(7))-(C_3S(3)+C_5S(5))}{\sqrt{2}} - \frac{(C_7S(7)-C_1S(1))-(C_5S(5)+C_3S(3))}{\sqrt{2}}\right] \quad \text{............ (27)}$$

$$2s(3) = (C_4S(0) + C_4S(4)) - (C_2S(2) + C_6S(6)) + [(C_7S(1) - C_1S(7)) + (C_5S(3) - C_3S(5))] \quad \text{............ (28)}$$

$$2s(4) = (C_4S(0) + C_4S(4)) - (C_2S(2) + C_6S(6)) - [(C_7S(1) - C_1S(7)) + (C_5S(3) - C_3S(5))] \quad \text{............ (29)}$$

$$2s(5) = (C_4S(0) - C_4S(4)) - (C_6S(2) - C_2S(6)) - \left[\frac{(C_1S(1)+C_7S(7))-(C_3S(3)+C_5S(5))}{\sqrt{2}} - \frac{(C_7S(7)-C_1S(1))-(C_5S(5)+C_3S(3))}{\sqrt{2}}\right] \quad \text{............ (30)}$$

$$2s(6) = (C_4S(0) - C_4S(4)) + (C_6S(2) - C_2S(6)) - \left[\frac{(C_1S(1)+C_7S(7))-(C_3S(3)+C_5S(5))}{\sqrt{2}} + \frac{(C_7S(7)-C_1S(1))-(C_5S(5)+C_3S(3))}{\sqrt{2}}\right] \quad \text{........... (31)}$$

$$2s(7) = (C_4S(0) + C_4S(4)) + (C_2S(2) + C_6S(6)) - [(C_1S(1) + C_7S(7)) + (C_3S(3) + C_5S(5))] \quad \text{............. (32)}$$

เฉพาะสมการ (26) สมการ (27) สมการ (30) และ สมการ (31) ที่มีการเปลี่ยนแปลง จากความสัมพันธ์ระหว่าง C_1 C_3 C_5 และ C_7 ทำให้สามารถลดคู่ของการคูณลงไปได้มากคือ S(1) และ S(7) จะคูณกับ C_1 หรือ C_7 เท่านั้นและ S(3) กับ S(5) จะคูณกับ C_3 หรือ C_5 เท่านั้น แต่จะต้องหารด้วย $\sqrt{2}$ เราจึงจำเป็นต้องคูณด้วย $\sqrt{2}$ ตลอดทุกสมการ

$$\sqrt{2}\,2s(0) = (S(0) + S(4)) + \sqrt{2}\,(C_2S(2) + C_6S(6)) + \sqrt{2}\,[(C_1S(1) + C_7S(7)) + (C_3S(3) + C_5S(5))] \quad (33)$$

$$\sqrt{2}\,2s(1) = (S(0) - S(4)) + \sqrt{2}\,(C_6S(2) - C_2S(6)) + [(C_1S(1) + C_7S(7)) - (C_3S(3) + C_5S(5)) + [(C_7S(1) - C_1S(7)) - (C_5S(3) - C_3S(5))]] \quad (34)$$

$$\sqrt{2}\,2s(2) = (S(0) - S(4)) - \sqrt{2}\,(C_6S(2) - C_2S(6)) + [(C_1S(1) + C_7S(7)) - (C_3S(3) + C_5S(5)) - [(C_7S(1) - C_1S(7)) - (C_5S(3) - C_3S(5))]] \quad (35)$$

$$\sqrt{2}\,2s(3) = (S(0) + S(4)) - \sqrt{2}\,(C_2S(2) + C_6S(6)) + \sqrt{2}\,[(C_7S(1) - C_1S(7)) - (C_5S(3) - C_3S(5))] \quad (36)$$

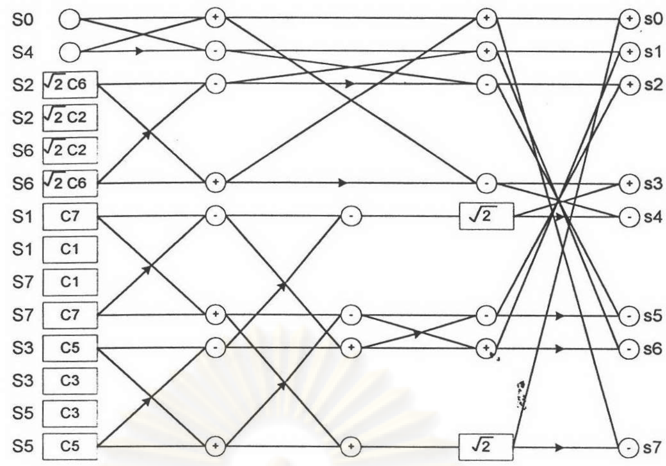
$$\sqrt{2}\,2s(4) = (S(0) + S(4)) - \sqrt{2}\,(C_2S(2) + C_6S(6)) - \sqrt{2}\,[(C_7S(1) - C_1S(7)) - (C_5S(3) - C_3S(5))] \quad (37)$$

$$\sqrt{2}\,2s(5) = (S(0) - S(4)) - \sqrt{2}\,(C_6S(2) - C_2S(6)) - [(C_1S(1) + C_7S(7)) - (C_3S(3) + C_5S(5)) - [(C_7S(1) - C_1S(7)) - (C_5S(3) - C_3S(5))]] \quad (38)$$

$$\sqrt{2}\,2s(6) = (S(0) - S(4)) + \sqrt{2}\,(C_6S(2) - C_2S(6)) - [(C_1S(1) + C_7S(7)) - (C_3S(3) + C_5S(5)) - [(C_7S(1) - C_1S(7)) - (C_5S(3) - C_3S(5))]] \quad (39)$$

$$\sqrt{2}\,2s(7) = (S(0) + S(4)) + \sqrt{2}\,(C_2S(2) + C_6S(6)) - \sqrt{2}\,[(C_1S(1) + C_7S(7)) + (C_3S(3) + C_5S(5))] \quad (40)$$

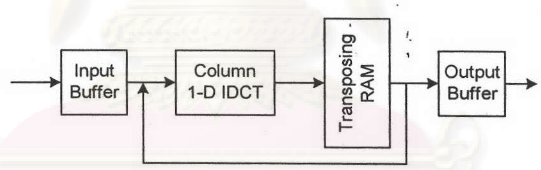
เนื่องจาก $C_4 = \frac{\sqrt{2}}{2}$ เมื่อคูณด้วย $\sqrt{2}$ เท่ากับ 1 ทำให้ประหยัดตัวคูณไปได้ ส่วนผลลัพธ์ของทุกสมการนั้นจะเป็น $2\sqrt{2}$ เท่า ของค่า s(x) แต่เราจะต้องคำนวณ 1-D IDCT อีกครั้งหนึ่งดังนั้น ผลลัพธ์ที่ได้จึงเป็น 8 เท่าของ s(x,y) การหารด้วย 8 นั้นเท่ากับเราตัดบิตข้อมูลทิ้งไป 3 บิต ซึ่งไม่จำเป็นต้องใช้วงจรหาร จากสมการ (33) ถึง (40) สามารถนำมาเขียนเป็นแผนผังได้ดังรูปที่ 5.3 ที่มีโครงสร้างคล้ายรูปผีเสื้อ บล็อกสี่เหลี่ยมหมายถึงปฏิบัติการคูณกับค่าในบล็อกนั้น ส่วนรูปวงกลมหมายถึงปฏิบัติการบวก ถ้าเส้นที่มีลูกศรนั้นต้องแปลงเป็นค่าลบก่อนที่จะนำมาบวก



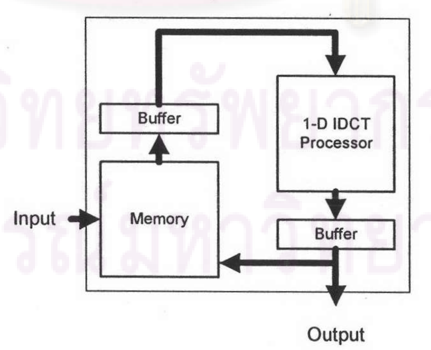
รูปที่ 5.3 ผังงานของ 1-D IDCT

5.2 แผนภาพบล็อกของวงจร

จากผังงานในรูปที่ 5.1 เราสามารถใช้วงจร 1-D IDCT เพียงวงจรเดียว สำหรับการคำนวณ 2-D IDCT โดยจะคำนวณ 1-D IDCT ทุกหลักแล้วสลับตำแหน่ง (transpose) เพื่อคำนวณตามแนวแถวโดยใช้วงจร 1-D IDCT วงจรเดิม เพื่อเป็นการประหยัดพื้นที่ในการออกแบบ ตามผังงานได้ใหม่ ดังแสดงในรูปที่ 5.4



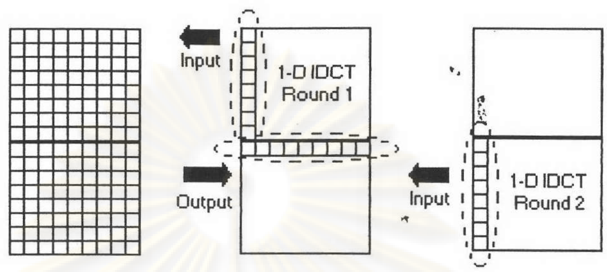
รูปที่ 5.4 ผังงานของ 2-D IDCT



รูปที่ 5.5 แผนภาพบล็อกของ 2-D IDCT โดยใช้ 1-D IDCT 1 ตัว

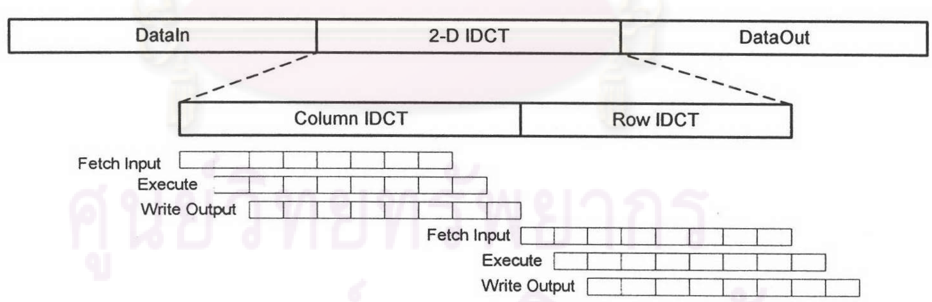
รูปที่ 5.5 แผนภาพบล็อกของ 2-D IDCT โดยใช้วงจรคำนวณ 1-D IDCT วงจรเดียว ข้อมูลเข้าจะถูกเก็บไว้ในหน่วยความจำโดยหน่วยความจำจะถูกแบ่งออกเป็นสองส่วนด้วยกันคือ ส่วนที่เป็นข้อมูลเข้าและส่วนที่เก็บผลลัพธ์จากการคำนวณ 1-D IDCT ครั้งแรก และผลลัพธ์นี้จะเป็นข้อมูลซึ่งจะใช้คำนวณ 1-D IDCT ในรอบที่สองโดยในการออกแบบใช้วงจรเดิม ลักษณะการใช้งานหน่วยความจำแสดงในรูปที่ 5.6 สังเกตว่าข้อมูลชุดแรก

ที่เรียกใช้จะเอาข้อมูลหลักแรกไปคำนวณทั้งหลัก ผลลัพธ์ของการคำนวณหลักแรกจะมี 8 ค่าเท่ากัน แต่แทนที่เราจะวางเรียงผลลัพธ์ในแนวหลักเหมือนเดิมกลับวางเรียงในแนวของแถว ที่เป็นเช่นนี้เพราะการคำนวณ 2-D IDCT คือการคำนวณ 1-D IDCT แต่ละหลัก 1 ครั้งแล้วจึงคำนวณแต่ละแถวอีก 1 ครั้ง เมื่อเราวางผลลัพธ์ในแนวแถวและดึงข้อมูลออกในแนวหลักเหมือนเดิมก็เป็นการคำนวณแต่ละแถวนั่นเอง ทำให้เราไม่จำเป็นต้องเปลี่ยนวิธีการดึงข้อมูลออกมาคำนวณใหม่ซึ่งยังคงเป็นแนวเดิม



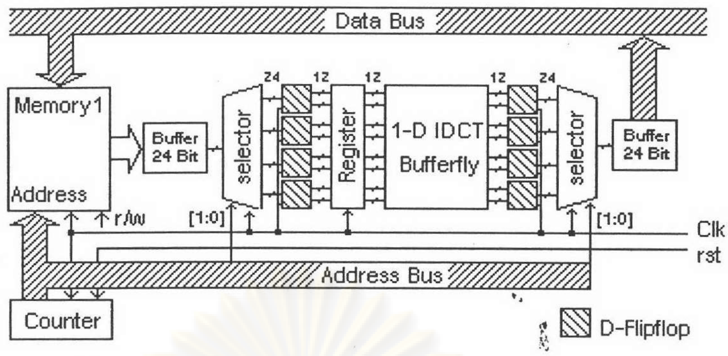
รูปที่ 5.6 การจัดสรรหน่วยความจำ

เมื่อคำนวณผลลัพธ์ได้ในแต่ละหลักจะได้ผลลัพธ์ออกมา 8 ค่า จะต้องนำเอาผลลัพธ์นั้นเขียนลงในหน่วยความจำชุดเดียวกันกับที่เก็บข้อมูลอินพุต เพื่อที่จะให้การทำงานเป็นไปอย่างรวดเร็วเราจึงออกแบบให้เป็นสถาปัตยกรรมแบบไพป์ไลน์ คือในขณะที่ทำการคำนวณให้การอ่านข้อมูลชุดต่อไปและบันทึกผลลัพธ์ลงบนหน่วยความจำได้เลยไม่จำเป็นต้องรอให้เขียนผลลัพธ์เสร็จเสียก่อน แต่เมื่อจะเริ่มการคำนวณในรอบที่ 2 คือการคำนวณแต่ละแถวนั้นจำเป็นต้องรอให้มีการเขียนผลลัพธ์จากการคำนวณหลักสุดท้ายให้เสร็จเสียก่อน เพราะต้องการข้อมูลจากหลักแรกของทุกๆแถว ดังแสดงแผนผังเวลาในรูปที่ 5.7



รูปที่ 5.7 ผังเวลาในการคำนวณ 2-D IDCT

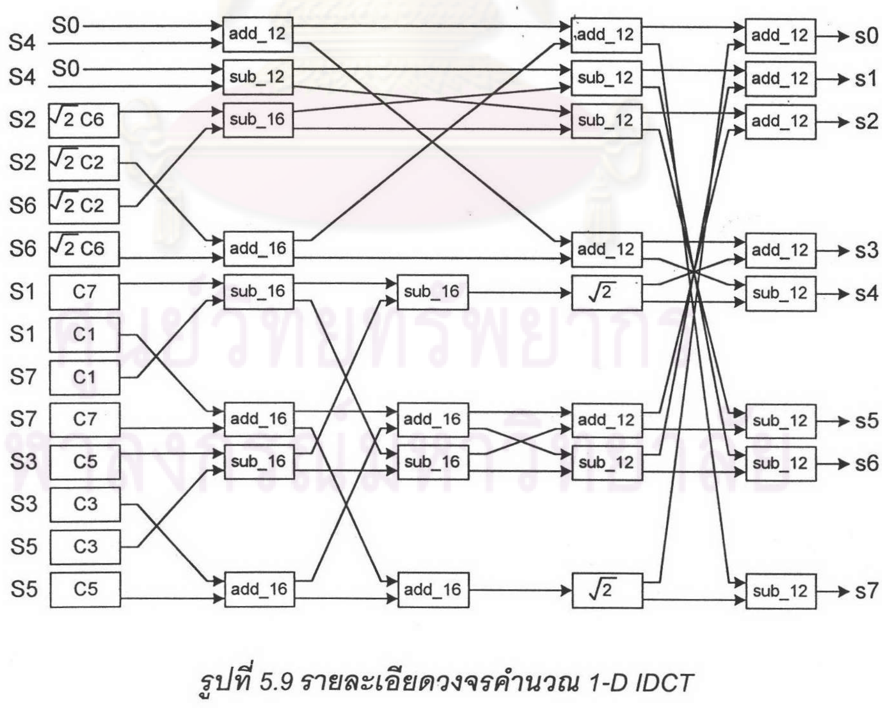
ในรูปที่ 5.8 การเชื่อมต่อทางตรรกของวงจรคำนวณ 2-D IDCT เนื่องจากการอ่าน/เขียนข้อมูลกับหน่วยความจำมีขนาด 24 บิต แต่ข้อมูลที่เข้า/ออกของวงจรคำนวณแบบโครงสร้างผีเสื้อที่ได้ออกแบบขึ้นนั้นจะมีขนาด 96 บิต (ข้อมูล 8 ค่า ค่าละ 12 บิต) จึงจะต้องมีบัฟเฟอร์ขนาด 96 บิตที่ส่วนอินพุตและเอาท์พุต โดยใช้ตัวเลือก (selector) เก็บข้อมูลทีละ 24 บิตทุกๆสัญญาณนาฬิกา ที่ส่วนของอินพุตของวงจรคำนวณแบบโครงสร้างผีเสื้อจะมีเรจิสเตอร์ขนาด 96 บิตทำให้สามารถอ่านข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์ในขณะที่ทำการคำนวณได้



รูปที่ 5.8 การเชื่อมต่อทางตรรกของวงจรคำนวณ 2-D IDCT

5.3 วงจรคำนวณ 1-D IDCT

รูปที่ 5.7 ผังเวลาของการคำนวณวงจร 1-D IDCT เมื่อทำการออกแบบโดยใช้วงจรบวกและวงจรคูณจะได้ดังรูปที่ 5.9 วงจรคำนวณนี้ประกอบไปด้วยวงจรบวกขนาด 16 บิตและวงจรลบขนาด 16 บิต อย่างละ 5 วงจร วงจรบวกขนาด 12 บิตและวงจรลบขนาด 12 บิตอย่างละ 8 วงจร และวงจรคูณอีก 14 วงจรด้วยกัน โดยแบ่งเป็นวงจรคูณด้วย $\sqrt{2}\,C_2$ $\sqrt{2}\,C_6$ C_1 C_7 C_3 C_5 และ $\sqrt{2}$ อย่างละ 2 วงจร



รูปที่ 5.9 รายละเอียดวงจรคำนวณ 1-D IDCT

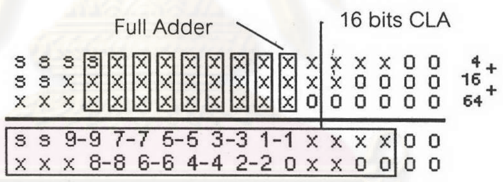
วงจรบวกออกแบบโดยใช้วงจรบวกการทดมองล่วงหน้า (Carry Look Ahead หรือ CLA) วงจรลบก็เช่นเดียวกับแต่ทำการผกผันบิตตัวลบจาก 1 เป็น 0 และ 0 เป็น 1 ก่อนแล้วในตัวทดเข้าวงจรเป็น 1 ซึ่งเป็นหลักการทำเลข ส่วนเติมเต็ม 2 (two's complement)

ในส่วนของวงจรคูณนั้นเนื่องจากเราทราบค่าของตัวคูณล่วงหน้าอยู่แล้ว เราจึงสร้างวงจรเฉพาะสำหรับการคูณแต่ละค่าดังต่อไปนี้

$64 * \sqrt{2}\ C_2$	$= 0.923879532 * \sqrt{2} * 64 \sim = 84$	
$64 * \sqrt{2}\ C_6$	$= 0.382683432 * \sqrt{2} * 64 \sim = 35$	
$64 * C_1$	$= 0.980785280 * 64$	$\sim = 63$
$64 * C_7$	$= 0.195090322 * 64$	$\sim = 12$
$64 * C_3$	$= 0.831469612 * 64$	$\sim = 53$
$64 * C_5$	$= 0.555570233 * 64$	$\sim = 35$
$64 * \sqrt{2}$	$= 1.414213562 * 64$	$\sim = 91$

ค่า 64 ที่นำเข้าไปคูณตัวคูณแต่ละตัวเพื่อทำให้ตัวคูณมีค่ามากขึ้น เนื่องจากว่าตัวคูณนั้นมีค่าน้อยมาก ถ้าปัดเป็นเลขจำนวนเต็มแล้วค่าจะกลายเป็น 1 หมด ที่เลือกใช้เลข 64 เนื่องจาก $64 = 2^6$ และเป็นเลขที่มากที่สุดเมื่อนำไปคูณแล้วผลลัพธ์มีค่าระหว่าง -128 และ 127 (8 บิต) เมื่อได้ผลลัพธ์แล้วเราสามารถที่จะหารเอาค่า 64 ออกเพียงตัด 6 บิตทางขวาสุดทิ้งไปเท่านั้น

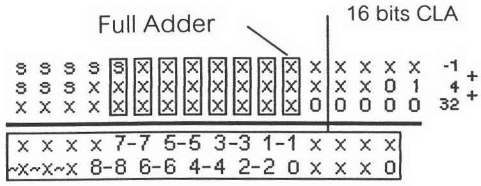
วงจรคูณ $64 * \sqrt{2}\ C_2$ เท่ากับการคูณด้วย 84 = 64+16+4 ซึ่งผลคูณย่อยทั้ง 3 แสดงในรูปที่ 5.10



รูปที่ 5.10 วงจรคูณ $64 * \sqrt{2}\ C_2$

ผลคูณย่อยตัวแรกได้จากการคูณตัวตั้งด้วย 4 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 2 บิต ผลคูณย่อยตัวที่สองได้จากการคูณตัวตั้งด้วย 16 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 4 บิต ผลคูณย่อยตัวที่สามได้จากการคูณตัวตั้งด้วย 64 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 6 บิต ตัวตั้งเป็นข้อมูลขนาด 12 บิต การออกแบบใช้ Full Adder ทั้งสิ้น 9 ตัว (บิตที่ 6 -14) ผลลัพธ์จากวงจรบวกจะมีขนาด 2 บิต (1-1 คือ ผลลัพธ์จาก Full Adder ตัวที่ 1 2-2 คือ ผลลัพธ์จาก Full Adder ตัวที่ 2 .. 9-9 คือ ผลลัพธ์จาก Full Adder ตัวที่ 9) ซึ่งจะนำไปคำนวณโดย CLA ต่อไป บิตที่ 15-17 นั้นแต่ละหลักเป็นการรวมของ บิตเครื่องหมาย 2 ตัวและข้อมูล 1 ตัว บิตเครื่องหมายนั้นจะเหมือนกันเสมอไม่ว่าจะเป็น 0 หรือ 1 เพราะผลคูณย่อยเป็นบวกทั้งสิ้น เมื่อรวมกับข้อมูล 1 ตัวจะได้ข้อมูลนั้นเสมอ เช่น 0+0+x = x และมีเลขทดเป็น 0 หรือ 1+1+x = x และมีเลขทดเป็น 1 ซึ่งเลขทดนี้จะมีค่าเดียวกับบิตเครื่องหมายนั่นเอง เราจึงนำเอาบิตเครื่องเป็นเลขทดไปบวกกับบิตต่อไป สุดท้ายนำผลลัพธ์จากวงจรบวกเข้าวงจร CLA ขนาด 16 บิตเพื่อให้ได้ผลลัพธ์ความสูงเพียงชั้นเดียว

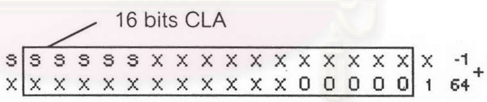
วงจรคูณ $64 * \sqrt{2}\ C_6$ เท่ากับการคูณด้วย 35 = 32+4-1 ซึ่งผลคูณย่อยทั้ง 3 แสดงในรูปที่ 5.11



*รูปที่ 5.11 วงจรคูณ 64 * $\sqrt{2}$ C_6*

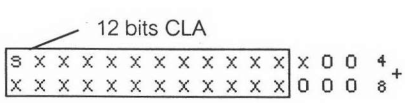
ผลคูณย่อยตัวแรกได้จากการคูณตัวตั้งด้วย -1 ซึ่งเท่ากับการผกผันตัวตั้งและบวกด้วย 1 ที่ข้อมูลบิต 0 ผลคูณย่อยตัวที่สองได้จากการคูณตัวตั้งด้วย 4 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 2 บิต ผลคูณย่อยตัวที่สามได้จากการคูณตัวตั้งด้วย 32 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 5 บิต ตัวตั้งเป็นข้อมูลขนาด 12 บิต การออกแบบใช้ Full Adder ทั้งสิ้น 8 ตัว (บิตที่ 5 -12) ผลลัพธ์จากวงจรบวกจะมีขนาด 2 บิต (1-1 คือ ผลลัพธ์จาก Full Adder ตัวที่ 1 2-2 คือ ผลลัพธ์จาก Full Adder ตัวที่ 2 .. 8-8 คือ ผลลัพธ์จาก Full Adder ตัวที่ 8) ซึ่งจะนำไปคำนวณต่อไป บิตที่ 13-16 นั้น แต่ละหลักเป็นการรวมของ บิตเครื่องหมาย 2 ตัวและข้อมูล 1 ตัว บิตเครื่องหมายนั้นจะต่างกันเสมอ เพราะผลคูณย่อยตัวแรกเป็นลบแต่ตัวที่สองเป็นบวก เมื่อรวมกับข้อมูล 1 ตัวจะได้ค่าผกผันข้อมูลนั้นเสมอ คือ 1+0+x = ~x และมีเลขทดเป็น x ซึ่งตัวทดนี้จะใช้รวมกับข้อมูลถัดไป สุดท้ายนำผลลัพธ์จากวงจรบวกเข้าวงจร CLA ขนาด 16 บิตเพื่อให้ได้ผลลัพธ์ความสูงเพียงชั้นเดียว

วงจรคูณ 64 * C_1 เท่ากับการคูณด้วย 63 = 64-1 ซึ่งผลคูณย่อยทั้ง 2 แสดงในรูปที่ 5.12 ผลคูณย่อยตัวแรกได้จากการคูณตัวตั้งด้วย -1 ซึ่งเท่ากับการผกผันตัวตั้งและบวกด้วย 1 ที่ข้อมูลบิต 0 ผลคูณย่อยตัวที่สองได้จากการคูณตัวตั้งด้วย 64 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 6 บิต เนื่องจากความสูงมีเพียง 2 ชั้นจึงบวกด้วยวงจร CLA ขนาด 16 บิตเพื่อให้ได้ผลลัพธ์ความสูงเพียงชั้นเดียวได้เลย



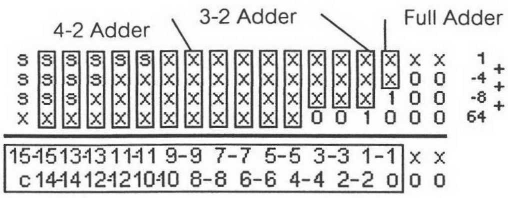
*รูปที่ 5.12 วงจรคูณ 64 * C_1*

วงจรคูณ 64 * C_7 เท่ากับการคูณด้วย 12 = 8+4 ซึ่งผลคูณย่อยทั้ง 2 แสดงในรูปที่ 5.13 ผลคูณย่อยตัวแรกได้จากการคูณตัวตั้งด้วย 4 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 2 บิต ผลคูณย่อยตัวที่สองได้จากการคูณตัวตั้งด้วย 8 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 3 บิต ตัวตั้งเป็นข้อมูลขนาด 12 บิต เนื่องจากความสูงมีเพียง 2 ชั้นจึงบวกด้วยวงจร CLA ขนาด 12 บิตเพื่อให้ได้ผลลัพธ์ความสูงเพียงชั้นเดียวได้เลย



*รูปที่ 5.13 วงจรคูณ 64 * C_7*

วงจรคูณ 64 * C_3 เท่ากับการคูณด้วย 53 = 64-8-4+1 ซึ่งผลคูณย่อยทั้ง 4 แสดงในรูปที่ 5.14 ตัวตั้งเป็นข้อมูลขนาด 12 บิต

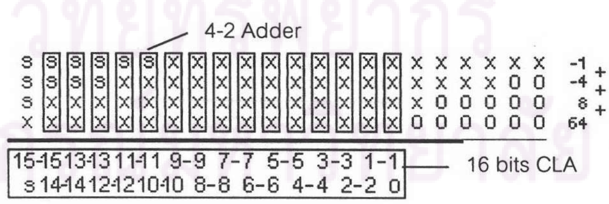


รูปที่ 5.14 วงจรคูณ 64 * C_3

ผลคูณย่อยตัวแรกได้จากการคูณตัวตั้งด้วย 1 ซึ่งเท่ากับค่าตัวตั้ง ผลคูณย่อยตัวที่สองได้จากการคูณตัวตั้งด้วย -4 ซึ่งเท่ากับการผกผันตัวตั้งและบวกด้วย 1 ที่ข้อมูลบิต 0 เลื่อนข้อมูลไปทางซ้าย 2 บิต ผลคูณย่อยตัวที่สามได้จากการคูณตัวตั้งด้วย -8 ซึ่งเท่ากับการผกผันตัวตั้งและบวกด้วย 1 ที่ข้อมูลบิต 0 และเลื่อนข้อมูลไปทางซ้าย 3 บิต ผลคูณย่อยตัวที่สี่ได้จากการคูณตัวตั้งด้วย 64 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 6 บิต ตัวตั้งเป็นข้อมูลขนาด 12 บิต การออกแบบใช้ Full Adder บิตที่ 2 และ 3-2 Adder ในบิตที่ 3-5 ส่วนบิตที่ 6-12 ใช้ 4-2 Adder และในบิตที่ 13-16 ใช้ 3-2 Adder เพราะว่าใน 4 บิตนี้แต่ละหลักมีบิตเครื่องหมายของผลคูณย่อยตัวที่ 1 และ 2 ซึ่งจะมีค่าแตกต่างกันเสมอเนื่องจากตัวหนึ่งเป็นบวกอีกตัวจะเป็นลบ (เป็น 0 และ 1) ดังนั้นจึงเท่ากับบวกด้วย 1 ใช้ 3-2 Adder ก็เพียงพอ ส่วนหลักสุดท้าย (บิตที่ 17) คือค่าบิตเครื่องหมายของผลคูณย่อยทุกตัว ซึ่งเป็นบวก 2 ค่าและลบ 2 ค่าดังนั้น ผลบวกจึงเท่ากับ 0 เสมอเมื่อรวมกับตัวทดเข้าจึงมีค่าเท่ากับกับตัวทดเข้านั่นเอง 1-1 คือ ผลลัพธ์จาก Full Adder ตัวที่ 1 2-2 คือ ผลลัพธ์จาก 3-2 Adder ตัวที่ 1 .. 15-15 คือ ผลลัพธ์จาก 3-2 Adder บิตที่ 16 สุดท้ายนำผลลัพธ์จากวงจรบวกเข้าวงจร CLA ขนาด 16 บิตเพื่อให้ได้ผลลัพธ์ความสูงเพียงชั้นเดียว

วงจรคูณ 64 * C_5 เท่ากับการคูณด้วย 35 และเท่ากับการคูณด้วย 64 * $\sqrt{2}$ C_6พอดี ใช้วงจรเดียวกันกับที่แสดงในรูปที่ 5.10

วงจรคูณ 64 * $\sqrt{2}$ เท่ากับการคูณด้วย 91 = 64+32-4-1 ซึ่งผลคูณย่อยทั้ง 4 แสดงในรูปที่ 5.15



รูปที่ 5.15 วงจรคูณ 64 * $\sqrt{2}$

ผลคูณย่อยตัวแรกได้จากการคูณตัวตั้งด้วย -1 ซึ่งเท่ากับการผกผันตัวตั้งและบวกด้วย 1 ที่ข้อมูลบิต 0 ตัวตั้ง ผลคูณย่อยตัวที่สองได้จากการคูณตัวตั้งด้วย -4 ซึ่งเท่ากับการผกผันตัวตั้งและบวกด้วย 1 ที่ข้อมูลบิต 0 เลื่อนข้อมูลไปทางซ้าย 2 บิต ผลคูณย่อยตัวที่สามได้จากการคูณตัวตั้งด้วย 32 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 5 บิต ผลคูณย่อยตัวที่สี่ได้จากการคูณตัวตั้งด้วย 64 ซึ่งเท่ากับการเลื่อนข้อมูลไปทางซ้าย 6 บิต ตัวตั้งเป็นข้อมูลขนาด 16 บิต การออกแบบใช้ 3-2 Adder ในบิตที่ 6 ส่วนบิตที่ 7-16 ใช้ 4-2 Adder และในบิตที่ 17- 20

ใช้ Full Adder เพราะว่าใน 4 บิตนี้แต่ละหลักมีบิตเครื่องหมายของผลคูณย่อยตัวที่ 1 และ 2 ซึ่งจะมีค่าเหมือนกันเสมอเนื่องจากเป็นลบทั้งคู่รวมกันเป็น 0 เสมอและมีตัวทดออกเป็นค่าบิตเครื่องหมายเอง ส่วนหลักสุดท้าย (บิตที่ 21) คือค่าบิตเครื่องหมายของผลคูณย่อยทุกตัว ซึ่งเป็นบวก 2 ค่าและลบ 2 ค่าดังนั้น ผลบวกจึงเท่ากับ 0 เสมอเมื่อรวมกับตัวทดเข้าจึงมีค่าเท่ากับกับตัวทดเข้านั่นเอง 1-1 คือ ผลลัพธ์จาก 4-2 Adder ตัวที่ 1 2-2 คือ ผลลัพธ์จาก 4-2 Adder ตัวที่ 2 .. 15-15 คือ ผลลัพธ์จาก Full Adder บิตที่ 20 สุดท้ายนำผลลัพธ์จากวงจรบวกเข้าวงจร CLA ขนาด 16 บิตเพื่อให้ได้ผลลัพธ์ความสูงเพียงชั้นเดียว ข้อมูล 6 บิตล่างไม่นำมาคำนวณเนื่องจากเป็นการหารเอาค่า 64 ออก

5.4 กระบวนการทำงานของวงจร

ในหัวข้อนี้จะอธิบายกระบวนการทำงานของวงจรอย่างละเอียดโดยการยกตัวอย่าง ข้อมูลเข้าเป็นสัมประสิทธิ์ DCT 1 บล็อกขนาด 8 x 8 ตัวอย่างแสดงในรูปที่ 5.16

$$\begin{bmatrix} 736 & -15 & -19 & -21 & 0 & 0 & 0 & 0 \\ 31 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 21 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 25 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 001011100000 & 111111110001 & 111111101101 & 111111101011 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000000011111 & 000000001111 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000000010101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000000010101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 111111100111 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000000001101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \end{bmatrix}$$

(b)

รูปที่ 5.16 ข้อมูลตัวอย่างในระบบ (a) เลขฐานสิบ (b) เลขฐานสอง

วงจรคำนวณนี้รับข้อมูลเข้าครั้งละ 8 ค่า จึงมีบัฟเฟอร์อ่านข้อมูลจากหน่วยความจำทีละ 24 บิต ซึ่งเท่ากับข้อมูลขนาด 12 บิต 2 ค่า เมื่ออ่านข้อมูลหลักแรกของเมตริกซ์ จนครบแล้ววงจรจึงเริ่มทำงาน ค่าของ S0 S1 S2 S3 S4 S5 S6 และ S7 มีค่าดังแสดงในรูป 5.7 คือค่า 736 31 0 21 21 -25 25 และ 0 ตามลำดับ ข้อมูลที่เข้าวงจรคูณ $64 * \sqrt{2}\, C_2$ $64 * \sqrt{2}\, C_6$ $64 * C_1$ $64 * C_7$ $64 * C_3$ และ $64 * C_5$ จะต้องตัด 6 บิตล่างทิ้งเพื่อหารเอาค่า 64 ออกไป แต่ไม่ได้ตัดทิ้งทั้ง 6 บิตในทีเดียว โดยข้อมูลจากวงจรคูณเข้าวงจรบวก/ลบจะตัดทิ้ง 3 บิตก่อนและจะตัดทิ้งอีก 4 บิตสำหรับข้อมูลเข้าวงจรบวก/ลบในชุดที่ 4 ที่ตัดทิ้งไป 4 บิตแทนที่จะเป็น 3 เนื่อง

จาก ผลลัพธ์ในขั้นตอนนี้เป็น $2\sqrt{2}$ เท่าของ s(x) (ดูสมการ (33) ถึง สมการ (40)) ส่วนข้อมูลที่ไม่ได้ผ่านวงจรคูณจะตัดออกเพียงบิตเดียว ดังนั้นผลลัพธ์ s0 s1 s2 s3 s4 s5 s6 และ s7 จึงยังติดเพียง $\sqrt{2}$ ซึ่งผลลัพธ์ในหลักนี้คือ 408 2 -10 11 0 0 0 และ 0 ตามลำดับ ผลลัพธ์ทั้ง 8 หลักแสดงไว้ในรูปที่ 5.17

$$\begin{bmatrix} 408 & 371 & 369 & 353 & 391 & 379 & 309 & 360 \\ 2 & 0 & -2 & -6 & -10 & -14 & -16 & -18 \\ -10 & -10 & -10 & -10 & -10 & -10 & -10 & -10 \\ -11 & -11 & -11 & -11 & -11 & -11 & -11 & -11 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

รูปที่ 5.17 ผลลัพธ์จาก 1-D IDCT

เมื่อได้ผลลัพธ์ทั้ง 8 ค่าแล้วจะต้องเขียนลงในหน่วยความจำ เช่นเดียวกับการอ่าน คือ เขียนทีละ 24 บิตข้อมูลเท่ากับ 2 ค่า โดยจะเขียนสลับในตำแหน่งแถว เพื่อที่จะนำไปคำนวณในขั้นต่อไป ดังแสดงในรูป 5.18

$$\begin{bmatrix} 408 & 2 & -10 & -11 & 0 & 0 & 0 & 0 \\ 371 & 0 & -10 & -11 & 0 & 0 & 0 & 0 \\ 369 & -2 & -10 & -11 & 0 & 0 & 0 & 0 \\ 353 & -6 & -10 & -11 & 0 & 0 & 0 & 0 \\ 391 & -10 & -10 & -11 & 0 & 0 & 0 & 0 \\ 379 & -14 & -10 & -11 & 0 & 0 & 0 & 0 \\ 309 & -16 & -10 & -11 & 0 & 0 & 0 & 0 \\ 360 & -18 & -10 & -11 & 0 & 0 & 0 & 0 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 000110011000 & 000000000010 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000101110011 & 000000000000 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000101110001 & 111111111110 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000101100001 & 111111111010 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000110000111 & 111111110110 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000101111011 & 111111110010 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000100110101 & 111111110000 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \\ 000101101000 & 111111101110 & 111111110110 & 111111110101 & 000000000000 & 000000000000 & 000000000000 & 000000000000 \end{bmatrix}$$

(b)

รูปที่ 5.18 ผลลัพธที่บันทึกบนหน่วยความจำในระบบ (a) เลขฐานสิบ (b) เลขฐานสอง

ซึ่งเมตริกซ์นี้จะเป็นข้อมูลเข้า โดยกระบวนการจะเหมือนเดิม แต่ผลลัพธ์สุดท้ายจะต้องหารด้วย 2 เป็นการเก็บค่า $\sqrt{2}$ ของทั้งสองขั้นตอนซึ่งเท่ากับ 2 พอดี ผลลัพธ์จากขั้นตอนนี้ภายหลังจากตัดบิตขวาไป 1 บิต ดังแสดงในรูปที่ 5.19

$$\begin{bmatrix} 95 & 101 & 107 & 107 & 103 & 99 & 99 & 101 \\ 85 & 91 & 98 & 98 & 94 & 90 & 91 & 92 \\ 84 & 90 & 97 & 97 & 93 & 90 & 91 & 92 \\ 79 & 85 & 92 & 93 & 90 & 87 & 88 & 90 \\ 87 & 93 & 100 & 102 & 99 & 97 & 98 & 101 \\ 82 & 89 & 97 & 99 & 97 & 95 & 97 & 99 \\ 64 & 71 & 79 & 81 & 79 & 78 & 80 & 82 \\ 76 & 83 & 91 & 94 & 92 & 91 & 93 & 96 \end{bmatrix}$$

(a)

$$\begin{bmatrix} 000001011111 & 000001100101 & 000001101011 & 000001101011 & 000001100111 & 000001100011 & 000001100011 & 000001100101 \\ 000001010101 & 000001011011 & 000001100010 & 000001100010 & 000001011110 & 000001011010 & 000001011011 & 000001011100 \\ 000001010100 & 000001011010 & 000001100001 & 000001100001 & 000001011101 & 000001011010 & 000001011011 & 000001011100 \\ 000001001111 & 000001010101 & 000001011100 & 000001011101 & 000001011010 & 000001010111 & 000001011000 & 000001011010 \\ 000001010111 & 000001011101 & 000001100100 & 000001100110 & 000001100011 & 000001100001 & 000001100010 & 000001100101 \\ 000001010010 & 000001011001 & 000001100001 & 000001100011 & 000001100001 & 000001011111 & 000001100001 & 000001100011 \\ 000001000000 & 000001000111 & 000001001111 & 000001010001 & 000001001111 & 000001001110 & 000001010000 & 000001010010 \\ 000001001100 & 000001010011 & 000001011011 & 000001011110 & 000001011100 & 000001011011 & 000001011101 & 000001100000 \end{bmatrix}$$

(b)

รูปที่ 5.19 ผลลัพธ์ขั้นสุดท้าย ในระบบ (a) เลขฐานสิบ (b) เลขฐานสอง

5.5 สรุป

ในบทนี้กล่าวถึงการออกแบบวงจรคำนวณ 2-D IDCT โดยการแตกเป็นการคำนวณ 1-D IDCT 2 ขั้นตอน และการออกแบบวงจรคำนวณ 1-D IDCT แบบโครงสร้างผีเสื้อ เป็นการจัดกลุ่มการคำนวณใหม่พยายามลดความซ้ำซ้อน เปรียบได้กับการดึงตัวประกอบออกมา ทั้งนี้เพื่อลดปริมาณการปฏิบัติการ ไม่ว่าจะเป็นการบวก การลบหรือการคูณ เพื่อให้เวลาในการคำนวณผลลัพธ์น้อยที่สุด ในขณะเดียวกันก็ทำให้ขนาดของวงจรเล็กที่สุด เนื่องจากเราทราบตัวคูณล่วงหน้า จึงออกแบบวงจรคูณที่ใช้เป็นวงจรคูณเฉพาะเพื่อให้มีขนาดเล็กกว่าวงจรคูณโดยทั่วไปอีกด้วย