

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 คำนำ

ในปัจจุบันเทคนิคการประมวลผลสัญญาณเสียงได้ถูกประยุกต์ใช้กันอย่างแพร่หลายเช่น ใน โทรศัพท์เคลื่อนที่ [1] ระบบอินเทอร์เน็ต [2] และระบบ Voice over IP [3] การประยุกต์ใช้สัญญาณเสียงได้พัฒนาขึ้นอย่างรวดเร็วซึ่งเป็นการทำงานควบคู่ไปกับงานวิจัยทางการบีบอัดสัญญาณเสียง สิ่งหนึ่งที่ยังเป็นจุดสนใจและสำคัญสำหรับการวิจัยและประมวลผลทางด้านสัญญาณเสียง นั่นก็คือการสกัดค่าคุณลักษณะของสัญญาณเสียง (Feature extraction) โดยใช้เทคนิคที่แตกต่างกัน เพื่อตรวจสอบและหาคุณลักษณะเด่นทางด้าน amplitude และ frequency เทคนิคต่างๆที่ใช้สกัดค่าคุณลักษณะของสัญญาณเสียงจะถูกออกแบบมาเพื่อสกัดสัญญาณเสียงออกมาจากสัญญาณรบกวน (Background noise) ซึ่งถูกปะปนอยู่ภายในสัญญาณเสียง โดยแบ่งสัญญาณเสียงออกเป็นเฟรมในช่วงเวลาสั้นๆ (Short-time frames) เทคนิคที่นำมาใช้ในการสกัดค่าคุณลักษณะของสัญญาณเสียงเช่น zero-crossing rates [4], linear predictive coding coefficients [5], and cepstral coefficients [6], the Mel-frequency cepstral coefficients[7], the wavelet transforms [8], the correlation coefficients [9], and the likelihood ratios [10, 11]. รวมถึงคณะผู้วิจัยได้ดำเนินการวิจัยและคิดค้นเทคนิคใหม่[12, 13, 14] โดยอาศัยโดยใช้หลักการของ Clustering Technique, A finite vector quantization, k-means algorithm โดยนำค่าคุณลักษณะของสัญญาณเสียงซึ่งสกัดโดยเทคนิค [5, 6] มาประยุกต์ใช้ ซึ่งได้ผลที่สามารถลดจำนวนของอัตราบิตได้มากถึง 30 % แต่เนื่องจาก หลักการของสกัดค่าลักษณะสัญญาณเสียงดังกล่าวเป็นการเทียบกับค่าอ้างอิงซึ่งเป็นค่าคงที่ (Static threshold detection) แต่ในทางปฏิบัติจริงนั้น สัญญาณเสียงและสัญญาณรบกวนเป็นสัญญาณที่ไม่มีรูปแบบคงที่ (nonstationary signal) ดังนั้น หลักการของการสกัดค่าเทียบกับค่าอ้างอิงสามารถให้ประสิทธิภาพที่ดีในช่วงไม่กี่เฟรมเท่านั้น ซึ่งจะส่งผลให้เกิดข้อผิดพลาดในการสกัดค่าคุณลักษณะของเสียงในส่วนที่เหลืออยู่ ทำให้เกิดการสูญเสียคุณภาพทางด้านความถี่ของสัญญาณเสียง ซึ่งส่งผลกระทบต่อตรงต่อคุณภาพของสัญญาณเสียงที่จะนำไปประยุกต์ใช้งานทางการสื่อสารหรือเก็บข้อมูล ดังนั้นเพื่อเป็นการแก้ปัญหาดังกล่าว งานวิจัยนี้จึงมีแนวคิดที่จะศึกษา วิจัยและพัฒนาเทคนิคใหม่สำหรับการสกัดค่าคุณลักษณะของสัญญาณเสียงและสัญญาณรบกวนซึ่งเป็นสัญญาณที่ไม่มีรูปแบบคงที่ด้วยเทคนิคการอ้างอิงแบบเปลี่ยนแปลงตลอดเวลา (dynamic or adaptive thresholds) โดยอาศัยหลักการของ Wavelet transform (Daubechies, Symlet, Coiflet) [15 – 19] ในการสกัดค่าคุณลักษณะของสัญญาณเสียง รวมถึงกำจัดค่าสัญญาณรบกวนออกไป

นอกจากนี้งานวิจัยนี้จะอาศัยหลักการของการเกิดความคล้ายคลึงหรือซ้ำกัน (repetition) ของการเดินทางของคุณลักษณะของสัญญาณเสียง (speech signal trajectory) ซึ่งคณะนักวิจัยได้ดำเนินงาน

วิจัยเพื่อหาความสัมพันธ์ในเบื้องต้น [20, 21] ซึ่งจากงานวิจัยพบว่าสัญญาณเสียงมีรูปแบบไม่คงที่ แต่เมื่อทำการแบ่งเป็นเฟรมด้วยช่วงเวลาสั้นๆแล้ว จะพบว่าการเปลี่ยนแปลงของสัญญาณเสียงจากเฟรมหนึ่งสู่อีกเฟรมหนึ่งมีลักษณะการเปลี่ยนแปลงที่ช้า ดังนั้นคณะนักวิจัยจึงได้ใช้พัฒนาเทคนิค [20, 21] เพื่อใช้ความสัมพันธ์ที่เกิดขึ้นดังกล่าว (Intra-frame and Inter-frame) มาพัฒนาการลดจำนวนอัตราบิทงานวิจัยนี้จะถูกนำไปศึกษา วิจัยและพัฒนาสู่ FPGA Broad เพื่อทดสอบการเข้ารหัสและการบีบอัดสัญญาณเสียงเพื่อที่จะนำไปใช้ประโยชน์ต่อระบบการสื่อสารที่เป็นมาตรฐานต่อไป

2.2 การสกัดค่าคุณลักษณะของสัญญาณเสียง

ในการสกัดค่าคุณลักษณะสำคัญสามารถแบ่งออกได้เป็น 2 ลักษณะด้วยกัน ได้แก่ การสกัดค่าคุณลักษณะสำคัญทางฉันทลักษณ์ (Prosodic Feature) ซึ่งได้แก่ ค่าความถี่มูลฐาน (Fundamental Frequency) ค่าความถี่ฟอร์แมนท์ (Formant Frequency) และค่าพลังงาน (Energy) เป็นต้น อีกลักษณะคือ การสกัดค่าคุณลักษณะสำคัญบนพื้นฐานของสเปกตรัม (Spectral-Based Feature) โดยจะมีอยู่ด้วยกันหลากหลายประเภทแล้วแต่ว่าจะนำไปใช้งานให้ตรงกับทางสายงานวิจัยก็คือ สัมประสิทธิ์การทำนายพหุเชิงเส้น (Linear Prediction Coefficient : LPC) สัมประสิทธิ์เซปสตรัม (Cepstral Coefficient) ซึ่งมีทั้งสัมประสิทธิ์เซปสตรัมแบบคำนวณมาจาก LPC (Linear Predictive Coding Derived Cepstrum : LPCC) และแบบที่คำนวณบนพื้นฐานของชุดตัวกรอง (Filter Bank) และสเกลของความถี่แบบไม่สม่ำเสมอ (Nonuniform Frequency Scale) เช่นสัมประสิทธิ์เซปสตรัมบนสเกลเมล (Mel Frequency Cepstral Coefficient : MFCC) หรือจะเป็นสัมประสิทธิ์คู่เส้นสเปกตรัม (Line Spectral Pairs) เป็นต้น

2.2.1 การวิเคราะห์หาค่าอัตสหสัมพันธ์ (Autocorrelation)

จากการวิเคราะห์ทฤษฎีพื้นฐานพบว่า ตัวอย่างเสียงพูดที่เวลา n สามารถถูกประมาณจากผลรวมเชิงเส้นของตัวอย่างเสียงพูดก่อนหน้าจำนวน p ตัวอย่างได้ใกล้เคียงที่สุด เมื่อมีสัมประสิทธิ์ของผลรวมเชิงเส้นที่เหมาะสม ดังนั้นการวิเคราะห์หาค่าอัตสหสัมพันธ์จะเริ่มจากสมการที่ 2.1

$$\hat{s}(n) \approx \sum_{k=1}^p \alpha_k s(n-k) \quad (2.1)$$

โดยที่ α_k เป็นค่าสัมประสิทธิ์ LPC อันดับที่ k

$\hat{s}(n)$ เป็นค่าประมาณสัญญาณเสียงพูดตัวอย่างที่ n

ค่าประมาณสัญญาณเสียงพูดตัวอย่างที่ n นี้ได้จากผลรวมของผลคูณระหว่างสัมประสิทธิ์ LPC กับสัญญาณเสียงพูดตัวอย่างก่อนหน้าจำนวน p ตัวอย่าง ดังนั้นค่าความผิดพลาดจากการประมาณ $e(n)$ ซึ่งเป็นผลต่างของค่าประมาณสัญญาณเสียงพูดตัวอย่างที่ n กับสัญญาณเสียงพูดจริงหาได้จากสมการที่ 2.2

$$\begin{aligned}
e(n) &= s(n) - \hat{s}(n) \\
&= s(n) - \sum_{k=1}^p \alpha_k s(n-k)
\end{aligned} \tag{2.2}$$

และในการวิเคราะห์การประมาณเชิงเส้นนั้นจำเป็นต้องมีการเลือกค่าสัมประสิทธิ์การประมาณเพื่อทำให้ได้คุณสมบัติทางสเปกตรัมใกล้เคียงกับสเปกตรัมของเสียงพูดในช่วงเวลาที่สนใจอยู่มากที่สุด ทั้งนี้เนื่องจากคุณลักษณะสเปกตรัมของเสียงพูดเปลี่ยนแปลงตลอดช่วงเวลาค่าสัมประสิทธิ์การประมาณ ณ เวลาที่กำหนดให้ n นั้นๆ ดังนั้นจึงมีการคำนวณสัมประสิทธิ์การประมาณที่ทำให้เกิดค่าผิดพลาดกำลังสองเฉลี่ย (Mean Squared Error : E_n) ของการประมาณน้อยที่สุดตลอดช่วงสั้นๆ ของสัญญาณเสียงนั้นเพื่อให้สัมประสิทธิ์ดังกล่าวทำให้เกิดสเปกตรัมของเสียงพูดที่ใกล้เคียงเสียงจริงมากที่สุด ซึ่งโดยปกติการวิเคราะห์สเปกตรัมของช่วงเวลาสั้นๆ แบบนี้ จะกระทำกับเสียงพูดที่ถูกแบ่งเป็นส่วนย่อยหรือเฟรมนั่นเอง

$$\begin{aligned}
E_n &= \sum_m e_n^2(m) \\
&= \sum_m [s_n(m) - \sum_{k=1}^p \alpha_k s_n(m-k)]^2
\end{aligned} \tag{2.3}$$

โดยที่ E_n เป็นค่าความผิดพลาดกำลังสองเฉลี่ย ณ เฟรมที่ n ใดๆ ซึ่งมีตัวอย่างเสียงพูด m ตัวอย่าง

$e(n)$ เป็นค่าความผิดพลาดจากการประมาณ

ในการแก้สมการที่ 2.3 เพื่อหาค่าสัมประสิทธิ์การประมาณ α_k นั้นจำเป็นต้องหาค่าอนุพันธ์ย่อยของ E_n เทียบกับ α_k แล้วกำหนดให้ผลลัพธ์เท่ากับศูนย์ ดังสมการที่ 2.4

$$\frac{\partial E_n}{\partial \alpha_k} = 0 \quad ; k = 1, 2, \dots, p \tag{2.4}$$

จะได้สมการที่ 2.5

$$R(i) = \sum_{k=1}^p \alpha_k R(i-k) \quad ; i = 1, 2, \dots, p \tag{2.5}$$

$$R(i) = \sum_m s(m)s(m+i) \tag{2.6}$$

โดยที่ $R(i)$ เป็นค่าอัตสหสัมพันธ์ (Autocorrelation) ที่ทำได้จากสมการที่ 2.6

2.3 สัมประสิทธิ์คู่เส้นสเปกตรัม (Line Spectrum Pair : LSP)

การพัฒนาในรูปแบบนี้เป็นการพัฒนาที่ปรับและเพิ่มเติมมาจากรูปแบบของการทำนายสัมประสิทธิ์แบบเชิงเส้น เนื่องด้วยพารามิเตอร์ของการทำนายแบบเชิงเส้นนั้นในขั้นตอนของการประมาณค่าพารามิเตอร์ทำให้เกิดความไม่เสถียรของสัญญาณ และทำให้เกิดผลกระทบต่อคุณภาพของสัญญาณเสียงในขณะที่พารามิเตอร์แบบสัมประสิทธิ์คู่เส้นสเปกตรัมจะมีคุณสมบัติที่โดดเด่นคือ ค่าของพารามิเตอร์นั้นจะอยู่ภายในของเขตที่จำกัด มีการเรียงลำดับของค่าพารามิเตอร์และสามารถที่จะ

ตรวจสอบเสถียรภาพของวงจรรองได้ง่าย นอกจากนี้สัมประสิทธิ์คู่เส้นสเปกตรัมยังแสดงในรูปเชิงความถี่ ซึ่งจะนำไปใช้ได้กับการหาคูสมบัติของระบบการรู้จดจำของมนุษย์ได้ ในการคำนวณสัมประสิทธิ์คู่เส้นสเปกตรัม จะเริ่มต้นจากพหุนามอันดับ M ของวงจรรองผกผันในเชิงแชนด์ดังสมการ 2.7

$$A(z) = 1 + \sum_{k=1}^p a_k z^{-k} \quad (2.7)$$

โดยที่ $A(z)$ คือวงจรรองไวเทนนิง (Whitening Filter) มีหน้าที่แยกความสัมพันธ์ช่วงสั้นที่ปรากฏในสัญญาณเสียงพูด ซึ่งก็คือในการวิเคราะห์การเข้ารหัสสัมประสิทธิ์ทำนายพัลเซเชิงเส้นนั้น จะเริ่มต้นจากการพิจารณากรอบสัญญาณเสียงที่มีตัวอย่าง N ตัวอย่างคือ s_1, s_2, \dots, s_N โดยอ้างว่าตัวอย่างสัญญาณปัจจุบันสามารถทำนายได้จากผลรวมของตัวอย่างสัญญาณในอดีต p ตัวอย่าง ดังสมการที่ 2.8

$$\tilde{s}_n = -a_1 s_{n-1} - a_2 s_{n-2} - a_3 s_{n-3} - \dots - a_p s_{n-p} = -\sum_{k=1}^p a_k s_{n-k} \quad (2.8)$$

โดยที่ p คืออันดับของการวิเคราะห์การทำนายพัลเซเชิงเส้นและ a_1, a_2, \dots, a_p คือสัมประสิทธิ์การเข้ารหัสการทำนายพัลเซเชิงเส้น ซึ่งจะกำหนดให้ e_n แทนค่าผิดพลาดระหว่างค่าจริงและค่าที่ทำนายได้ โดยจะได้ดังสมการที่ 2.9 และสมการที่ 2.10

$$e_n = s_n - \tilde{s}_n \quad (2.9)$$

$$e_n = s_n + \sum_{k=1}^p a_k s_{n-k} \quad (2.10)$$

โดยที่ e_n เรียกว่าสัญญาณตกค้าง เนื่องจากสัญญาณ e_n ได้มาจากผลของการลบสัญญาณ s_n ด้วย \tilde{s}_n และเนื่องด้วยค่าสหสัมพันธ์ช่วงสั้น (Short-term Correlation) ระหว่างตัวอย่างของสัญญาณตกค้างมีค่าต่ำ ดังนั้นประมาณได้ว่ากรอบสเปกตรัมกำลังของสัญญาณตกค้างจะมีลักษณะราบเรียบ และเมื่อทำการแปลงแชนด์ (Z-Transform) ของสมการที่ 2.10 ก็จะได้ค่า $E(z)$ ดังสมการที่ 2.11

$$E(z) = A(z) \cdot S(z) \quad (2.11)$$

โดยที่ $S(z)$ เป็นผลจากการแปลงแชนด์ของสัญญาณเสียง

$E(z)$ เป็นผลการแปลงแชนด์ของสัญญาณตกค้าง

ในส่วนถัดมาจะทำการแยกส่วนสมการที่ 2.7 ให้เป็นพหุนามอันดับ $M+1$ จำนวน 2 พหุนาม ดังสมการที่ 2.12 และสมการที่ 2.13

$$P(z) = A(z) + z^{-(M+1)} A(z^{-1}) \quad (2.12)$$

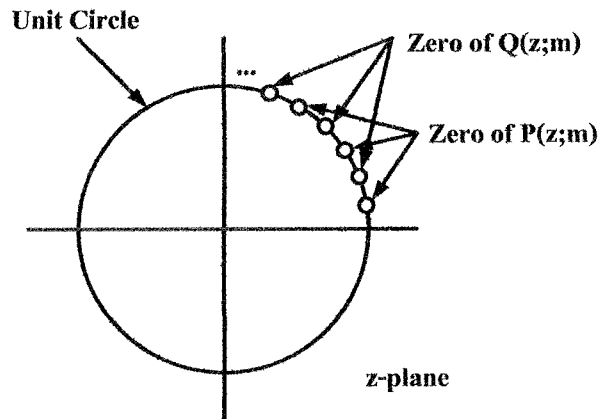
$$Q(z) = A(z) - z^{-(M+1)} A(z^{-1}) \quad (2.13)$$

โดยที่ $P(z)$ และ $Q(z)$ จะเป็นพหุนามที่มีความสัมพันธ์กับ $A(z)$ ตามสมการที่ 2.14

$$A(z) = \frac{P(z) + Q(z)}{2} \quad (2.14)$$

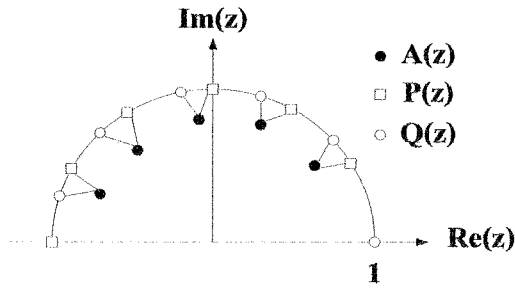
พหุนาม $P(z)$ และ $Q(z)$ สอดคล้องกับรูปแบบช่องทางเดินเสียงที่ได้รับการสูญเสียขณะที่ช่องระหว่างเส้นเสียง (Glottis) ปิดและเปิดตามลำดับ และรากของพหุนาม $P(z)$ และ $Q(z)$ เรียกว่า ความถี่เส้นสเปกตรัม โดยพหุนามทั้งสองมีคุณสมบัติดังนี้

1. ราก (Zeroes) ทั้งหมดของพหุนาม $P(z)$ และ $Q(z)$ นั้นจะตั้งอยู่บนวงกลมหนึ่งหน่วยเสมอ
2. ราก (Zeroes) ของพหุนาม $P(z)$ และ $Q(z)$ จะวางเรียงสลับกันตามลำดับจากน้อยไปหามาก ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 การวางเรียงสลับของรากของพหุนามคู่เส้นสเปกตรัม $P(z)$ และ $Q(z)$

โดยสามารถแสดงให้เห็นได้ว่าวงจรกรองผกผัน $A(z)$ จะมีเฟสต่ำสุด ถ้าคู่เส้นสเปกตรัมของ $A(z)$ สอดคล้องกับคุณสมบัติทั้งสองนี้ ดังนั้นวงจรสังเคราะห์การเข้ารหัสการทำนายพัลส์เชิงเส้นสามารถยืนยันได้ว่ามีเสถียรภาพ โดยการทำการประมวลผลพารามิเตอร์การเข้ารหัสการทำนายพัลส์เชิงเส้นในรูปแบบคู่เส้นสเปกตรัม เมื่อพิจารณารากของพหุนามทั้งสองพบว่าพหุนาม $P(z)$ และ $Q(z)$ มีรากจริงอยู่ที่ -1 และ 1 ตามลำดับ สำหรับรากอื่นๆ อยู่บนวงกลมหนึ่งหน่วยโดยวางเรียงสลับกันตามคุณสมบัติของพหุนามทั้งสอง และรากทั้งสองของพหุนามมีลักษณะเป็นคู่เชิงซ้อนสังยุค ดังนั้นในการเก็บรากของพหุนามเพื่อใช้เป็นพารามิเตอร์จึงเก็บเพียง M ค่า เนื่องจากรากของพหุนาม $A(z)$ แสดงตำแหน่งฟอร์มแมนต์ของเสียงพูด และพหุนาม $P(z)$ และ $Q(z)$ สัมพันธ์กับ $A(z)$ ตามสมการที่ 2.7 – 2.9 ดังนั้นรากของพหุนามทั้งสองจึงสัมพันธ์กับฟอร์มแมนต์ด้วย โดยรากของพหุนาม $A(z)$ แต่ละอันจะจับคู่กับรากของพหุนาม $P(z)$ และ $Q(z)$ อย่างละหนึ่งราก ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 ความสัมพันธ์ระหว่างรากของ $A(z)$ กับรากของคู่เส้นสเปกตรัม $P(z)$ และ $Q(z)$

จากรูปที่ 2.2 รูปวงกลมสีดำจะเป็นรากของพหุนาม $A(z)$, ส่วนรูปสี่เหลี่ยมและรูปวงกลมสีขาวเป็นรากของพหุนาม $P(z)$ และ $Q(z)$ ตามลำดับ การเกาะกลุ่มของรากหรือความถี่เส้นสเปกตรัมจำนวน 2 หรือ 3 รากจะกำหนดลักษณะความถี่พอร์แมนต์และขนาดแบนด์วิดท์ของพอร์แมนต์ โดยขึ้นกับความชิดของกลุ่มรากดังกล่าว ถ้ารากชิดกันมากจะทำให้เกิดพอร์แมนต์และมีแบนด์วิดท์แคบ ในทางตรงข้ามกลุ่มรากจะแสดงถึงสเปกตรัมที่มีแบนด์วิดท์กว้างคือไม่เกิดพอร์แมนต์ นอกจากนี้คุณสมบัติอีกอย่างหนึ่งของคู่เส้นสเปกตรัมคือความไวทางสเปกตรัมของคู่เส้นสเปกตรัมมีลักษณะเฉพาะที่ คือเมื่อมีการเปลี่ยนแปลงคู่เส้นสเปกตรัมใดๆ จะทำให้เกิดการเปลี่ยนแปลงของสเปกตรัมกำลังของการเข้ารหัสการทำนายพันธะเชิงเส้นเฉพาะบริเวณรอบๆ เท่านั้น ทำให้สามารถทำการประเมินค่า (Quantization) ได้อย่างอิสระโดยไม่มีผลกระทบจากการลดทอนเนื่องจากการประเมินค่าจากสเปกตรัมหนึ่งไปสู่สเปกตรัมอื่นๆ

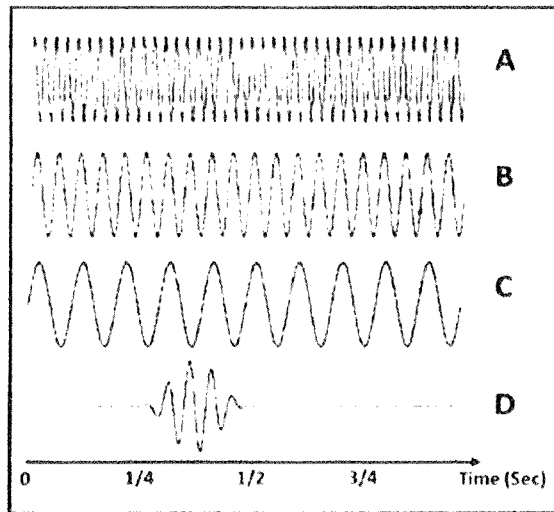
2.4 การแปลงเวฟเล็ต (Wavelet Transform)

การแปลงฟูรีเยร์ [5] เป็นที่นิยมใช้ในงานการประมวลผลสัญญาณ การแปลงฟูรีเยร์ เป็นการสร้างตัวแทนของสัญญาณอินพุตด้วยฟังก์ชันพื้นฐานของฟูรีเยร์นั่นคือ ฟังก์ชันไซน์และโคไซน์ ซึ่งเป็นฟังก์ชันที่มีคุณสมบัติการเปลี่ยนแปลงค่าอย่างต่อเนื่อง (Continuous) มีความถี่คงที่ และมีความถี่คงที่ การวนซ้ำค่า (Periodic Property) โดยส่วนใหญ่สัญญาณที่ถูกแปลงด้วยการแปลงฟูรีเยร์ จะเป็นสัญญาณที่มีการเปลี่ยนแปลงที่มีความถี่คงที่ อย่างไรก็ตาม สัญญาณในการประยุกต์ใช้งานจริงส่วนใหญ่สัญญาณอินพุตสามารถเกิดการเปลี่ยนแปลงความถี่ในแต่ละช่วงเวลาได้เสมอ ดังตัวอย่างเช่น สัญญาณการตรวจสอบสุขภาพของมนุษย์ สัญญาณตำแหน่งและความเร็วของวัตถุบนเรดาร์ พฤติกรรมการลงทุนในตลาดหุ้น หรือแม้แต่ตำแหน่งกักเก็บน้ำมันใต้ดิน ตัวอย่างสัญญาณที่มีการเปลี่ยนแปลงอย่างคงที่และสัญญาณพัลส์ที่มีการเปลี่ยนแปลงความถี่ในเวลาแสดงในรูปที่ 2.3

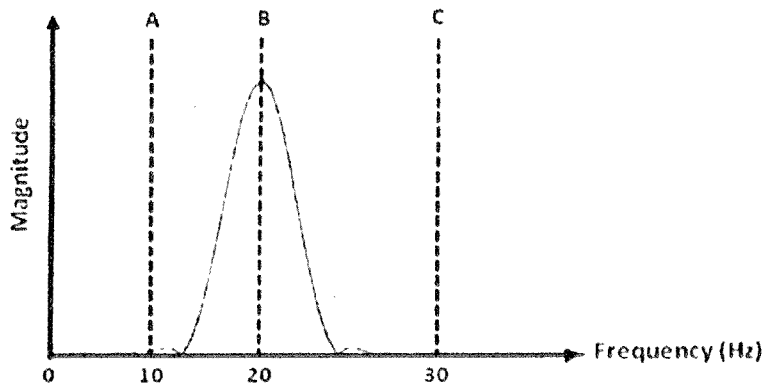
จากรูปที่ 2.3 เมื่อทดลองการหาค่าคอสี่เหลี่ยมระหว่างสัญญาณไซน์ที่มีการเปลี่ยนแปลงอย่างคงที่ (A, B, C) กับสัญญาณที่มีการเปลี่ยนแปลงเฉพาะช่วงเวลา จะพบว่าสัญญาณไซน์ (B) มีความถี่เท่ากับสัญญาณพัลส์ (D) ทำให้ได้ค่าคอสี่เหลี่ยมสูงที่สุด เมื่อเทียบกับสัญญาณไซน์อื่น ๆ (เช่น A และ C) ดังแสดงในรูปที่ 2.3 (ข)

จะเห็นว่าถึงแม้สัญญาณไซน์ (B) จะมีค่าความสัมพันธ์กับสัญญาณพัลส์ (D) มากที่สุด เนื่องจากมีความถี่ที่เท่ากัน แต่ไม่สามารถที่จะเป็นตัวแทนค่าของสัญญาณที่มีการเปลี่ยนแปลงความถี่ไม่ต่อเนื่องเช่น สัญญาณพัลส์ (D) ได้ดี แสดงให้เห็นว่าการประยุกต์ใช้การแปลงฟูรีเยร์ซึ่งในฟังก์ชันตัวแทนเป็นฟังก์ชันไซน์ และโคไซน์เป็นฟังก์ชันที่มีการเปลี่ยนแปลงค่าต่อเนื่องในแต่ละความถี่ไม่เหมาะสมในการใช้เป็นตัวแทนสัญญาณที่มีการเปลี่ยนแปลงความถี่ไม่ต่อเนื่องเช่น สัญญาณพัลส์จะเห็นได้ว่าสิ่งนี้เป็นข้อจำกัดของการแปลงความถี่ฟูรีเยร์สำหรับสัญญาณอินพุตก็คือสัญญาณที่มีคุณสมบัติการเปลี่ยนแปลงแบบค่าเชิงสถิติไม่คงที่ (Non-Stationary Signal)

ดังนั้นเพื่อให้สามารถสร้างตัวแทนค่าของสัญญาณที่มีการเปลี่ยนแปลงความถี่ไม่ต่อเนื่องได้ดี จำเป็นต้องสร้างฟังก์ชันตัวแทนที่มีการเปลี่ยนแปลงที่ไม่ต่อเนื่อง และทำการเลื่อนฟังก์ชันตัวแทนนี้ไปตามช่วงเวลา เพื่อให้สามารถสร้างตัวแทนค่าสัญญาณที่มีการเปลี่ยนแปลงความถี่ไม่ต่อเนื่องและเกิดในช่วงเวลาใด ๆ ได้อย่างมีประสิทธิภาพ และนี่เป็นที่มาของการวิจัยและพัฒนาการแปลงเวฟเล็ต



(ก) แสดงการเปรียบเทียบกับสัญญาณคลื่นพัลส์



(ข) แสดงค่าความสัมพันธ์ระหว่างสัญญาณไซน์ความถี่ต่าง ๆ กับสัญญาณคลื่นพัลส์

รูปที่ 2.3 ความสัมพันธ์ระหว่างคลื่นสัญญาณไซน์ที่ความถี่ต่าง ๆ [5]

เวฟเล็ต เป็นการวิเคราะห์สัญญาณแบบ Multiresolution โดยการแปลงเวฟเล็ตจะได้ผลลัพธ์เป็น ค่าสัมประสิทธิ์เวฟเล็ต (Wavelet Coefficient) การแปลงเวฟเล็ตจะเป็นการแปลงค่าสัมประสิทธิ์ผ่านฟังก์ชันหน้าต่าง แต่ฟังก์ชันหน้าต่างนี้สามารถเลื่อนตำแหน่งและปรับเปลี่ยนช่วงเวลาให้เหมาะสมกับความถี่ที่จะวิเคราะห์ โดยเรียกฟังก์ชันหน้าต่างนี้ว่า “ฟังก์ชันเวฟเล็ต (Wavelet Function)”

ทฤษฎีเวฟเล็ตจะใช้ในการอธิบายสิ่งใดสิ่งหนึ่งที่เหมือนกับการแยกสิ่งเหล่านั้นออกเป็น ส่วนประกอบเล็ก ๆ ที่มีความสัมพันธ์กัน ซึ่งอยู่ในรูปของเวฟเล็ตที่ถูกปรับขนาดและเลื่อนตำแหน่ง และมีค่ากว้างน้ำหนัก (สัมประสิทธิ์เวฟเล็ต) ต่าง ๆ กัน คุณอยู่กับเวฟเล็ตแต่ละตัว การทำส่วนการแปลงกลับเวฟเล็ต (Inverse Wavelet Transform : IWT) หรือเรียกอีกอย่างหนึ่งว่า “การแปลงกลับเวฟเล็ต (Wavelet Reconstruction)”

ให้ $f(t)$ เป็นสัญญาณที่ประกอบด้วยหลายความถี่ ถ้าเราพิจารณาในรูปแบบ 2 มิติ โดยแกนหนึ่งเป็นแกนเวลาและแกนหนึ่งเป็นแกนของความถี่ ดังนั้นสามารถเขียนสมการสำหรับการแปลงเวฟเล็ตแบบต่อเนื่อง (CWT) ที่ใช้ในการแตกกระจายเวฟเล็ตของสัญญาณของฟังก์ชันใด ๆ $f(t)$ ด้วยเวฟเล็ตแม่ $M(t)$ เป็นดังนี้

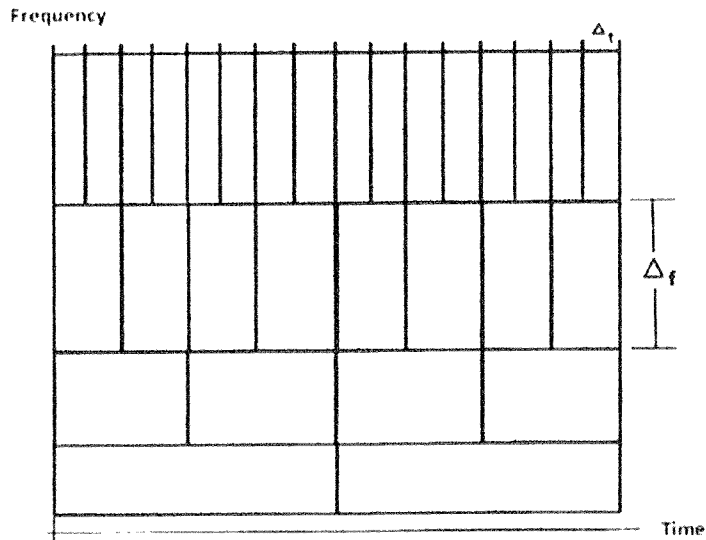
$$W_M(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(t)M^*\left(\frac{t-b}{a}\right)dt \quad (2.15)$$

โดย * คือ Complex Conjugate ของ $M(t)$
 $M(t)$ คือ ค่าสัมประสิทธิ์เวฟเล็ตที่ได้เป็นผลลัพธ์

ค่า $M(t)$ เป็นค่าสัมประสิทธิ์ที่ได้เป็นผลลัพธ์อาจมองได้ว่าเป็นการวิเคราะห์ความสัมพันธ์ (Cross-Correlation) ระหว่างสัญญาณที่พิจารณา $f(t)$ และฟังก์ชันเวฟเล็ตแม่ (Mother Wavelet) โดย $M(t)$ ที่ถูกเลื่อนตำแหน่งด้วยพารามิเตอร์ b และสเกลความถี่ด้วยพารามิเตอร์ a ซึ่งจะมีลักษณะการเปลี่ยนแปลงเป็นช่วงเวลาและความถี่ที่สัมพันธ์กัน คือ ในช่วงเวลาความถี่สูงจะใช้ช่วงเวลาการวิเคราะห์สั้นและในช่วงความถี่ต่ำจะมีช่วงเวลาในการวิเคราะห์ที่ยาวนานขึ้นเป็นไปดังรูปที่ 2.4 ซึ่งเหมาะสมกับลักษณะสัญญาณในทางปฏิบัติ

2.4.1 การแปลงเวฟเล็ตแบบไม่ต่อเนื่อง (Discrete Wavelet Transform : DWT)

การแปลงเวฟเล็ตสามารถทำได้ทั้งแบบต่อเนื่อง (Continuous Wavelet Transform : CWT) และแบบไม่ต่อเนื่อง (Discrete Wavelet Transform : DWT) แต่ในที่นี้จะกล่าวเฉพาะการแปลงแบบไม่ต่อเนื่องเท่านั้น ฟังก์ชันการแปลงเวฟเล็ตแบบไม่ต่อเนื่องนี้เป็นการปรับค่าสเกลความถี่ (Scale) และค่าการเลื่อนตำแหน่ง (Translate) ที่ไม่ต่อเนื่องด้วย โดยการแปลงเวฟเล็ตจะทำการคำนวณหาค่าสัมประสิทธิ์การแปลง $DWT_{j,k}$ จากการหาความสัมพันธ์ของสัญญาณอินพุต $f(x)$ กับฟังก์ชันเวฟเล็ตแม่ $M(x)$ ที่เลือกใช้ในแต่ละสเกล (j) และการเลื่อนตำแหน่ง (k) ดังสมการที่ 2.16 – 2.17



รูปที่ 2.4 ระยะเวลาและความถี่การเปลี่ยนแปลงช่วงเวลาและความถี่ที่สัมพันธ์กันของการ แปลงเวฟ เล็ต [5]

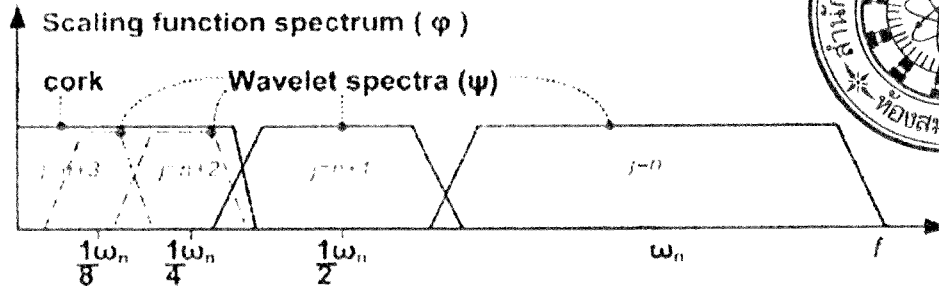
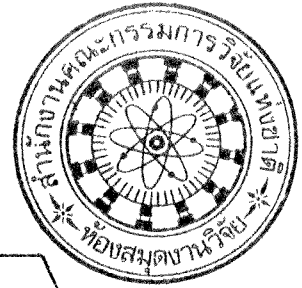
การเลื่อนตำแหน่ง (k) ของฟังก์ชันเวฟเล็ตแม่ ซึ่งแน่นอนย่อมถูกจำกัดด้วยช่วงของสัญญาณที่ทำการวิเคราะห์ $[0, N-1]$ แต่เมื่อพิจารณาขอบเขตการปรับความถี่ของฟังก์ชันเวฟเล็ตแม่ จะพบว่าเมื่อเราทำการปรับขนาดของฟังก์ชันเวฟเล็ตแม่ในแกนเวลาด้วยน้ำหนักกำลังสอง (2^j) เทียบได้กับการลดขอบเขตช่วงของความถี่ลงไปครึ่งหนึ่ง จากคุณสมบัติฟูรีเยร์ในสมการที่ 2.18

$$DWT_{j,k} = \frac{1}{\sqrt{N}} \left[2^j \sum_{x=0}^{N-1} f(x) W_{j,k}(x) \right] \quad (2.16)$$

$$= \frac{1}{\sqrt{N}} \left[2^j \sum_{x=0}^{N-1} f(x) M(2^j x - k) \right] \quad (2.17)$$

$$F\{f(at)\} = \frac{1}{|a|} F\left(\frac{\omega}{a}\right) \quad (2.18)$$

ดังนั้นการยืดขยายฟังก์ชันเวฟเล็ตแม่ไปสองเท่าจะเป็นการลดช่วงกว้างของความถี่วิเคราะห์ลงไปครึ่งหนึ่ง ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การลดช่วงกว้างของความถี่ทีละครึ่งหนึ่งสำหรับค่าแต่ละค่าสเกล [5]

ดังนั้นหากต้องการวิเคราะห์สัมประสิทธิ์ความถี่เวฟเลตย่อยลงไป ในลักษณะที่ความถี่ถูกปรับลดลงทีละครึ่ง และต้องการวิเคราะห์ความถี่ต่ำที่ความถี่ศูนย์ $f = 0$ จำเป็นต้องทำการแบ่งช่วงความถี่ย่อยลงไปถึงจำนวนอนันต์ค่า จึงจะสามารถวิเคราะห์ความถี่ต่ำลงไปจนครอบคลุมถึงความถี่ศูนย์ได้ ซึ่งในทางปฏิบัตินั้นใช้เวลานานเกินไป และถ้าเป็นในกรณีของสัญญาณไม่ต่อเนื่องนั้นเป็นไปได้เลยที่จะทำการวิเคราะห์ความถี่ศูนย์ได้ ดังนั้นสามารถวิเคราะห์ความถี่ต่ำที่ผ่านตำแหน่งความถี่ศูนย์ (Low-pass Spectrum) แยกออกจากช่วงความถี่สูง (High-pass Spectrum) แทนการพยายามวิเคราะห์ความถี่ย่อยลงเป็นจำนวนอนันต์ครึ่ง โดยฟังก์ชันที่วิเคราะห์ช่วงความถี่ต่ำจะเรียกว่า “สเกลลิงฟังก์ชัน ($S(f)$: Scaling Function)” และฟังก์ชันที่วิเคราะห์ช่วงความถี่สูงจะเรียกว่า “เวฟเลตฟังก์ชัน ($W(t)$: Wavelet Function)”

สเกลลิงฟังก์ชัน (Scaling Function)

คุณสมบัติของสเกลลิงฟังก์ชันจะต้องมีค่าโมเมนต์ที่ศูนย์ จะต้องไม่เป็นศูนย์ ดังแสดงในสมการที่ 2.19

$$\int_{-\infty}^{\infty} S(t) dt = 1 \quad (2.19)$$

สามารถทำการแปลงสัมประสิทธิ์ความถี่ต่ำของเวฟเลต (DWT_s : Scaling Coefficients) ด้วยสเกลลิงฟังก์ชัน $S(t)$ ดังแสดงในสมการที่ 2.15

$$DWT_s(j_0, k) = \frac{1}{\sqrt{N}} \left[2^{j_0/2} \sum_{x=0}^{N-1} f(x) S(2^{j_0} x - k) \right] \quad (2.20)$$

โดย j_0 คือ ค่าสเกลความถี่สิ้นสุดของสเกลลิงฟังก์ชัน

k คือ ค่าตำแหน่งการเลื่อน

เวฟเลตฟังก์ชัน (Wavelet Function)

$$DWT_w(j, k) = \frac{1}{\sqrt{N}} \left[2^{j/2} \sum_{x=0}^{N-1} f(x) W(2^j x - k) \right] \quad (2.21)$$

สำนักงานคณะกรรมการวิจัยแห่งชาติ
ห้องสมุดฯ นวท.
วันที่... 0. 2. ๕๖. ๒... 2555
เลขทะเบียน... 248122
เลขเรียกหนังสือ...

โดย j คือ ค่าสเกลความถี่เริ่มต้นที่ j_0
 k คือ ค่าตำแหน่งการเลื่อน

การทำการแปลงสัมประสิทธิ์เวฟเล็ตด้วยสเกลลิ่งฟังก์ชันและเวฟเล็ตฟังก์ชันนี้ บางครั้งถูกเรียกว่า “การแยกองค์ประกอบความถี่ของเวฟเล็ต (Wavelet Decomposition)” ในบางครั้งสเกลลิ่งฟังก์ชันอาจเรียกว่า “ฟังก์ชันการประมาณค่า (Approximation Function)” ในขณะที่เวฟเล็ตฟังก์ชันอาจเรียกว่า “ฟังก์ชันรายละเอียด (Detail Function)” ได้เช่นกัน

ตัวอย่าง การแปลงเวฟเล็ตโดยใช้ฟังก์ชันเวฟเล็ตแม่แบบ Haar

$$F(x) = [1 \ 2 \ 3 \ 4]$$

ฟังก์ชันเวฟเล็ตแม่แบบ Haar

$$M_H(t) = \begin{cases} 1 & 0 < t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.22)$$

ฟังก์ชันเวฟเล็ตแบบ Haar

$$S_H(t) = \begin{cases} 1 & 0 < t < 1/2 \\ 0 & \text{otherwise} \end{cases} \quad (2.23)$$

เวฟเล็ตฟังก์ชันแบบ Haar

$$W_H(t) = \begin{cases} 1 & 0 < t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.24)$$

ตารางที่ 2.1 ตัวอย่างขั้นตอนการคำนวณค่าสัมประสิทธิ์เวฟเล็ตด้วยสเกลลิงฟังก์ชันที่สเกล $j_0 = 0$

สเกล (j)	ตำแหน่งเลื่อน (k)	สัมประสิทธิ์เวฟเล็ต ($DWT_{j,k}$)
$j_0 = 0$	0	$DWT_s(0,0) = \frac{1}{\sqrt{4}} \left[2^{0/2} \sum_{x=0}^3 f(x) S_{H-1,0}(2^0 x) \right]$ $= \frac{1}{\sqrt{4}} (1 \ 2 \ 3 \ 4) \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{4}} [1(1) + 2(1) + 3(1) + 4(1)] = \frac{10}{\sqrt{4}}$
	> 0	ค่าสัมประสิทธิ์เวฟเล็ตของสเกลลิงฟังก์ชันที่ $k > 0$ เป็นต้นไป ตัวฟังก์ชันเวฟเล็ตขยายค่าออกนอกช่วงข้อมูล [0 0 0 0 1 1 1 1] จึงไม่จำเป็นต้องคำนวณ
		$DWT_s = [DWT_s(0,0)] = [10 / \sqrt{4}]$

ตารางที่ 2.2 ตัวอย่างขั้นตอนการคำนวณค่าสัมประสิทธิ์เวฟเล็ตด้วยเวฟเล็ตฟังก์ชัน

สเกล (j)	ตำแหน่งเลื่อน (k)	สัมประสิทธิ์เวฟเล็ต ($DWT_{j,k}$)
$j_0 = 0$	0	$DWT_w(0,0) = \frac{1}{\sqrt{4}} \left[2^{0/2} \sum_{x=0}^3 f(x) W_{H0,0}(2^0 x) \right]$ $= \frac{1}{\sqrt{4}} (1 \ 2 \ 3 \ 4) \begin{pmatrix} 1 \\ 1 \\ -1 \\ -1 \end{pmatrix} = \frac{1}{2} [1(1) + 2(1) + 3(-1) + 4(-1)] = \frac{-4}{\sqrt{4}}$
	> 0	ค่าสัมประสิทธิ์เวฟเล็ตของเวฟเล็ตฟังก์ชันที่สเกลมากกว่า '0' เป็นต้นไป ตัวฟังก์ชันเวฟเล็ตขยายค่าออกนอกช่วงข้อมูล [0 0 0 0 1 1 -1 -1] จึงไม่จำเป็นต้องคำนวณ

สเกล (j)	ตำแหน่ง เลื่อน (k)	สัมประสิทธิ์เวฟเล็ด ($DWT_{j,k}$)
	0	$DWT_w(1,0) = \frac{1}{\sqrt{4}} \left[2^{1/2} \sum_{x=0}^3 f(x)_{H1,0} (2^1 x) \right]$ $= \frac{1}{\sqrt{4}} (1 \ 2 \ 3 \ 4) \begin{pmatrix} \sqrt{2} \\ -\sqrt{2} \\ 0 \\ 0 \end{pmatrix} = \frac{1}{\sqrt{4}} [1(\sqrt{2}) + 2(-\sqrt{2}) + 3(0) + 4(0)] = \frac{-\sqrt{2}}{\sqrt{4}}$
$j_0 = 1$	1	$DWT_w(0,1) = \frac{1}{\sqrt{4}} \left[2^{1/2} \sum_{x=0}^3 f(x)_{H1,1} (2^1 x - 1) \right]$ $= \frac{1}{\sqrt{4}} (1 \ 2 \ 3 \ 4) \begin{pmatrix} 0 \\ 0 \\ \sqrt{2} \\ -\sqrt{2} \end{pmatrix} = \frac{1}{\sqrt{4}} [1(0) + 2(0) + 3(\sqrt{2}) + 4(-\sqrt{2})] = \frac{-\sqrt{2}}{\sqrt{4}}$
	≥ 2	เวฟเล็ดฟังก์ชัน $W_{H1,k} (2^1 x - k)$ ถูกเลื่อนตำแหน่งออกไปนอกขอบเขตของค่าช่วงข้อมูล $[0 \ 0 \ 0 \ 1 \ -1]$ จึงไม่จำเป็นต้องคำนวณ
		$DWT_w = [DWT_w(0,0), DWT_w(1,0), DWT_w(1,1)]$ $= \frac{1}{\sqrt{4}} [-4, -\sqrt{2}, -\sqrt{2}]$

จากผลลัพธ์จะได้จำนวนสัมประสิทธิ์การแปลงเวฟเล็ดจะมีจำนวน 4 ค่า เท่ากับจำนวนค่าของสัญญาณอินพุต คือ

$$DWT = [DWT_s(0,0), DWT_w(0,0), DWT_w(1,0), DWT_w(1,1)] = \frac{1}{\sqrt{4}} [10, -4, -\sqrt{2}, -\sqrt{2}]$$

การแปลงกลับเวฟเลต (Inverse Discrete Wavelet Transform)

การแปลงกลับเวฟเลตเพื่อทำการสร้างคืน (Reconstruction) สัญญาณอินพุตฉบับ $f(x)$ โดยการคำนวณผลรวมของความสัมพันธ์ของค่าสัมประสิทธิ์เวฟเลตและฟังก์ชันการแปลงสัมประสิทธิ์ทั้งจากทางสเกลถึงฟังก์ชันและเวฟเลตฟังก์ชันของแต่ละสเกล (j) และการเลื่อนตำแหน่ง (k) ดังแสดงในสมการที่ 2.25

$$f(x) = \frac{1}{\sqrt{N}} \sum_k DWT_s(j_0, k) 2^{j_0} / 2S_{j_0, k} (2^{j_0} x - k) + \frac{1}{\sqrt{N}} \sum_{j=j_0}^{\infty} \sum_k DWT_w(j, k) 2^{j/2} W_{j, k} (2^j x - k) \quad (2.25)$$

ตัวอย่าง การแปลงกลับสัมประสิทธิ์เวฟเลต โดยใช้ฟังก์ชันเวฟเลตแม่แบบ Haar ผลลัพธ์การแปลงกลับสำหรับสเกลถึงฟังก์ชัน

$$F_{s(0,0)} = \frac{1}{\sqrt{4}} DWT_s(0,0) \cdot S_{0,0}(x) = \frac{1}{\sqrt{4}} \left\{ \frac{10}{\sqrt{4}} [1,1,1,1] \right\} = \frac{1}{\sqrt{4}} [10,10,10,10]$$

ผลลัพธ์การแปลงกลับสำหรับเวฟเลตฟังก์ชัน

$$F_{w(0,0)} = \frac{1}{\sqrt{4}} DWT_w(0,0) \cdot W_{0,0}(x) = \frac{1}{\sqrt{4}} \left\{ \frac{-4}{\sqrt{4}} [1,1,-1,-1] \right\} = \frac{1}{4} [-4, -4, 4, 4]$$

$$F_{w(1,0)} = \frac{1}{\sqrt{4}} DWT_w(1,0) \cdot W_{1,0}(x) = \frac{1}{\sqrt{4}} \left\{ \frac{-\sqrt{2}}{\sqrt{4}} [\sqrt{2}, -\sqrt{2}, 0, 0] \right\} = \frac{1}{4} [-2, 2, 0, 0]$$

$$F_{w(1,1)} = \frac{1}{\sqrt{4}} DWT_w(1,1) \cdot W_{1,1}(x) = \frac{1}{\sqrt{4}} \left\{ \frac{-\sqrt{2}}{\sqrt{4}} [0, 0, \sqrt{2}, -\sqrt{2}] \right\} = \frac{1}{4} [0, 0, -2, 2]$$

2.4.2 ฟังก์ชันเวฟเลตแม่ (Mother Wavelet)

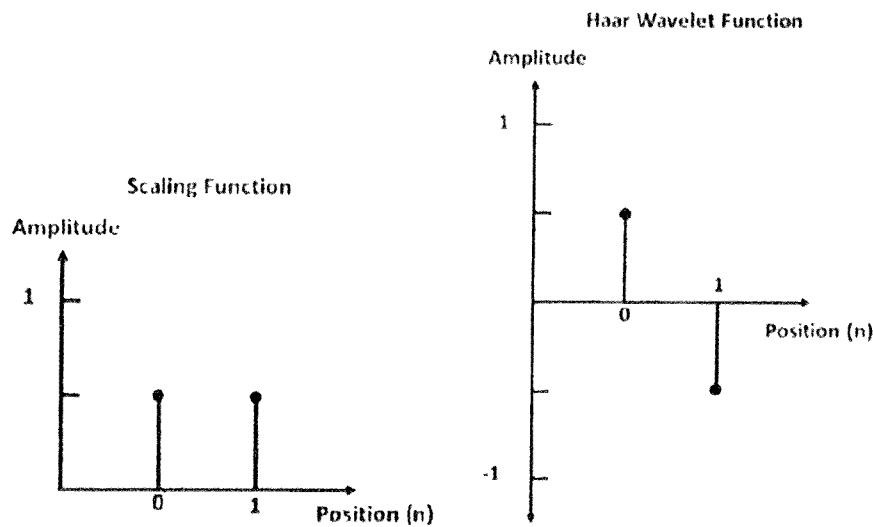
ฟังก์ชันเวฟเลตแม่เป็นฟังก์ชันต้นแบบที่จะถูกนำไปใช้เพื่อแปลงสัมประสิทธิ์เวฟเลตสำหรับแต่ละสเกลและตำแหน่ง โดยมีผู้นำเสนอฟังก์ชันเวฟเลตแม่หลากหลายฟังก์ชัน ตัวอย่างเช่น Haar, Daubachies Gaussian or Spline, Biorthogonal, Mexican Hat, Coiflet Wavelet เป็นต้น

2.4.2.1 ฟังก์ชันเวฟเลตแม่แบบ Haar (Haar Mother Wavelet)

ฟังก์ชันเวฟเลตแม่แบบ Haar เป็นฟังก์ชันเวฟเลตที่นำมาประยุกต์ใช้ได้ง่ายที่สุด ใช้เวลาในการประมวลผลสั้นและใช้หน่วยความจำในการคำนวณอย่างมีประสิทธิภาพ เนื่องจากไม่จำเป็นต้องใช้อะเรย์ชั่วคราว (Temporary Array) ในการเก็บข้อมูลระหว่างการประมวลผล ดังเช่น

ฟังก์ชันเวฟเลตแม่ตัวอื่น ๆ นอกจากนี้ฟังก์ชันเวฟเลตแม่แบบ Haar ยังมีคุณสมบัติตั้งฉาก (Orthogonality) นั่นคือ ฟังก์ชันเวฟเลตทุกฟังก์ชันจะตั้งฉากกันทั้งหมด โดยค่าของฟังก์ชันเวฟเลตแม่แบบ Haar เป็นดังสมการที่ 2.26 และรูปที่ 2.6

$$g_H(t) = \begin{cases} 1 & 0 < t < 1/2 \\ -1 & 1/2 \leq t < 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.26)$$



(ก) สเกลลิงฟังก์ชัน

(ข) เวฟเลตฟังก์ชันแบบ Haar

รูปที่ 2.6 คุณสมบัติของเวฟเลตแบบ Haar [5]

ค่าของฟังก์ชันเวฟเลตแบบ Haar จะทำการหาค่าเฉลี่ยและค่าผลต่างของข้อมูลในตำแหน่งข้างเคียง โดยค่าเฉลี่ยที่คำนวณได้จะเป็นค่าสัมประสิทธิ์การแปลงเวฟเลตในช่วงความถี่ต่ำ และค่าผลต่างที่คำนวณได้จะเป็นค่าสัมประสิทธิ์การแปลงเวฟเลตในช่วงความถี่สูง

อย่างไรก็ดีข้อจำกัดของฟังก์ชันเวฟเลตแบบ Haar คือ ไม่สามารถรวบรวมค่าการเปลี่ยนแปลงทันทีทันใดขนาดใหญ่ (Sharp Spike) ไว้ในสัมประสิทธิ์ความถี่สูงได้ เมื่อการเปลี่ยนแปลงนั้นเกิดขึ้นในบางลักษณะทำให้การแปลงสัมประสิทธิ์เวฟเลตด้วยฟังก์ชัน Haar ไม่สามารถแยกข้อมูลระหว่างความถี่ต่ำและความถี่สูงออกจากกันได้อย่างมีประสิทธิภาพ ทำให้ไม่เหมาะสมสำหรับการนำไปประยุกต์ใช้ในการบีบอัดสัญญาณและการกำจัดสัญญาณรบกวน และเนื่องจากความไม่ต่อเนื่องของฟังก์ชัน Haar

ทำให้ไม่เหมาะสมที่จะนำไปใช้ในการวิเคราะห์สัญญาณต่อเนื่อง (Continuous Signal) ที่มีค่าของอนุพันธ์ที่มีลำดับต่อเนื่อง (Continuous Derivatives)

2.4.2.2 ฟังก์ชันเวฟเลตแม่แบบ Daubechies (Daubechies Mother Wavelet)

ฟังก์ชันแม่ของการแปลงเวฟเลตแบบนี้ เรียกชื่อตามนักคณิตศาสตร์ผู้คิดค้น คือ “Ingrid Daubechies” เป็นหนึ่งในกลุ่มของฟังก์ชันเวฟเลตแม่ที่มีคุณสมบัติตั้งฉาก (Orthogonal Wavelet) โดยเป็นฟังก์ชันไม่ต่อเนื่อง (Discrete Function) ที่คุณสมบัติของฟังก์ชันถูกกำหนดด้วยจำนวนค่าของโมเมนต์ที่หายไป (Vanishing Moment) สูงสุด ตามเงื่อนไขต่อไปนี้

1. ฟังก์ชันพื้นฐาน D_M มีคุณสมบัติมีค่ากระชับ (Compact Support) ในช่วง $[0, 2r+1]$ และจะลู่เข้าสู่ศูนย์เมื่อออกนอกช่วง
2. ค่าโมเมนต์ลำดับที่ $0 - r$ มีค่าเป็นศูนย์

$$\int_{-\infty}^{\infty} D_M(x) dx = \dots = \int_{-\infty}^{\infty} x^r D_M(x) dx = 0$$

3. ฟังก์ชันพื้นฐาน D_M มีค่าอนุพันธ์ต่อเนื่องได้จนถึงอนุพันธ์อันดับที่ $r/5$

ฟังก์ชันเวฟเลตแม่แบบ Daubechies มีตั้งแต่ $D_2 - D_{20}$ โดยค่าตัวเลขจะกำหนดจำนวนของสัมประสิทธิ์ทั้งหมดของฟังก์ชันเวฟเลต Daubechies และแต่ละฟังก์ชันจะมีค่าโมเมนต์ที่หายไปเป็นจำนวนครึ่งหนึ่งของจำนวนค่าสัมประสิทธิ์ทั้งหมด ตัวอย่างเช่น D_2 หรือเทียบได้กับฟังก์ชันเวฟเลตแม่แบบ Haar ซึ่งมีจำนวนโมเมนต์ที่หายไป 1 ค่า และเช่นเดียวกันกับ D_4 จะมีจำนวนโมเมนต์ที่หายไป 2 ค่า จำนวนค่าโมเมนต์ที่หายไปเป็นตัวกำหนดความสามารถในการแทนค่าสมการที่มีความสัมพันธ์แบบโพลิโนเมียล (Polynomial) ของพฤติกรรมหรือข้อมูลของสัญญาณที่ต้องการพิจารณา ตัวอย่างเช่น D_2 ที่มีเพียงโมเมนต์ค่าเดียวสามารถสร้างเป็นตัวแทนฟังก์ชันโพลิโนเมียลสำหรับหนึ่งสัมประสิทธิ์หรือแทนได้เพียงองค์ประกอบที่คงที่ของสัญญาณเท่านั้น ในขณะที่ D_4 สามารถเข้ารหัสฟังก์ชันโพลิโนเมียลได้ 2 สัมประสิทธิ์ คือ ค่าคงที่และความสัมพันธ์เชิงเส้นขององค์ประกอบของสัญญาณและ D_6 ทำให้เราสามารถประมาณสัญญาณที่อยู่ในรูปของฟังก์ชันโพลิโนเมียล 3 สัมประสิทธิ์ คือ ค่าคงที่ ค่าความสัมพันธ์เชิงเส้น และความสัมพันธ์เชิงซ้อนกำลังสอง (Second Order Polynomial) ขององค์ประกอบของสัญญาณที่ทำการวิเคราะห์

ฟังก์ชันเวฟเลตแม่แบบ Daubechies นิยมใช้ในการแก้ไขปัญหอย่างกว้างขวาง ตัวอย่างเช่น ฟังก์ชันเวฟเลตแม่แบบ Daubechies สำหรับ D_2, D_{12} และ D_{20} ตัวอย่างฟังก์ชันเวฟเลตแม่แบบ Daubechies4 (D_4) จะมีสัมประสิทธิ์การกรองทั้งหมด 4 ค่า (4 Tap Filters) ซึ่งประกอบไปด้วยค่าสัมประสิทธิ์สำหรับการกรองสัญญาณความถี่ต่ำและค่าสัมประสิทธิ์สำหรับการกรองสัญญาณความถี่สูงจำนวน 4 ค่า ดังแสดงในตารางที่ 2.3

Daubechies สามารถนำมาประยุกต์ใช้เป็นฟังก์ชันการกรองความยาวจำกัด (Finite Length Filter) ได้ง่าย ด้วยฟังก์ชันการกรองความยาวจำกัดนี้เป็นคุณสมบัติที่สำคัญในการวิเคราะห์

ข้อมูลในเชิงตำแหน่งเฉพาะพื้นที่ได้ (Spatial Domain Localization) และยังมีคุณสมบัติที่สามารถวิเคราะห์ความถี่เนื่องของการเปลี่ยนแปลงฟังก์ชันต่อการทำงานอนุพันธ์ต่อเนื่อง (Continuous Derivatives Analyze) และสามารถหลีกเลี่ยงปัญหาการเกิดสัญญาณรบกวนบริเวณขอบของการเปลี่ยนแปลงของข้อมูลหรือเส้นขอบของวัตถุ (Edge Artifacts) ได้อย่างมีประสิทธิภาพ

ตารางที่ 2.3 ค่าสัมประสิทธิ์การกรองของฟังก์ชันเวฟเล็ตแบบ Daubechies จำนวน 4 ค่า (D_4)

สัมประสิทธิ์การกรอง ความถี่ต่ำ (Scaling Function)	$h_0 = \frac{1 + \sqrt{3}}{4\sqrt{2}}$	$h_1 = \frac{3 + \sqrt{3}}{4\sqrt{2}}$	$h_2 = \frac{3 - \sqrt{3}}{4\sqrt{2}}$	$h_3 = \frac{1 - \sqrt{3}}{4\sqrt{2}}$
สัมประสิทธิ์การกรอง ความถี่สูง (Wavelet Function)	$g_0 = h_3$	$g_1 = -h_2$	$g_2 = h_1$	$g_3 = -h_0$

การแปลงสัมประสิทธิ์เวฟเล็ต มีลักษณะคล้ายกับการหาเฉลี่ยค่าของข้อมูลด้วยตัวถ่วงน้ำหนักซึ่งค่าตัวถ่วงน้ำหนัก คือ ค่าสัมประสิทธิ์ของฟังก์ชันเวฟเล็ตที่เลือกใช้นั่นเอง และด้วยการออกแบบที่เหมาะสมของค่าสัมประสิทธิ์ของฟังก์ชันเวฟเล็ตแม่แบบ Daubechies ทำให้สามารถรักษาแนวโน้มของข้อมูลได้เป็นอย่างดี โดยเฉพาะเมื่อพิจารณาสัมประสิทธิ์การแปลงเวฟเล็ตที่ความถี่ต่ำ ถึงแม้ว่าการแปลงสัมประสิทธิ์เวฟเล็ตด้วยเวฟเล็ตแบบ Daubechies อาจไม่สามารถให้ผลดีกว่าเวฟเล็ตแบบ Haar ได้ทุกรูปแบบของการประยุกต์ใช้งาน แต่มีงานวิจัยที่ศึกษาพบว่าการแปลงสัมประสิทธิ์ด้วยเวฟเล็ตแบบ Daubechies ให้ผลการวิเคราะห์สัญญาณได้ดีในหลากหลายรูปแบบการทดลอง จากการพิจารณารูปร่างของฟังก์ชันเวฟเล็ตแบบ Haar เทียบกับฟังก์ชันเวฟเล็ตแบบ Daubechies พบว่าสัญญาณที่มีลักษณะการเปลี่ยนแปลงอย่างทันทีทันใด (Sharp Spike) จะสามารถวิเคราะห์ได้ดีกว่าด้วยเวฟเล็ตแบบ Daubechies นอกจากนี้ยังสามารถถ่ายโอนพลังงาน (Energy) ส่วนใหญ่ของสัญญาณไว้ในช่วงความถี่ต่ำและเหลือพลังงานปริมาณน้อยอยู่ในช่วงความถี่สูง ซึ่งแตกต่างจากฟังก์ชันเวฟเล็ตแม่แบบ ซึ่งจะมีการถ่ายโอนพลังงานไปยังช่วงความถี่สูงมากกว่าเมื่อข้อมูลถูกบีบอัดในช่วงความถี่สูงนี้ มักจะถูกตัดทิ้งเพื่อลดขนาดของข้อมูล

2.4.2.3 ฟังก์ชันเวฟเล็ตแม่แบบ Biorthogonal (Biorthogonal Mother Wavelet)

ในการแยกองค์ประกอบหรือการหาสัมประสิทธิ์การแปลงฟังก์ชันพื้นฐานส่วนใหญ่ มักจะมีคุณสมบัติตั้งฉาก แต่ในความเป็นจริงฟังก์ชันพื้นฐานการแปลงไม่จำเป็นต้องมีคุณสมบัติตั้งฉากเสมอไป ดังนั้นเพื่อความยืดหยุ่นในการสร้างฟังก์ชันเวฟเล็ตแม่แบบที่จะใช้เป็นพื้นฐานในการแปลงสัมประสิทธิ์เวฟเล็ตสามารถลดเงื่อนไขการตั้งฉาก เนื่องจากการออกแบบให้ฟังก์ชันเวฟเล็ตมีคุณสมบัติทั้งตั้งฉาก (Biorthogonal) กระจาย (Compactly Support) และสมมาตร (Symmetric) พร้อมๆ กันนั้นเป็นไปได้ยาก ส่วนใหญ่แล้วฟังก์ชันที่มีคุณสมบัติกระจายและสมมาตรมักจะต้องสูญเสียคุณสมบัติตั้ง

ฉากไป แต่การสูญเสียนี้อาจไม่ส่งผลเสียร้ายแรงนัก ถ้าสามารถออกแบบฟังก์ชันการแปลงหลายระดับความละเอียด (Multi-resolution) เป็นฟังก์ชันชุดคู่ที่ประกอบด้วยฟังก์ชันหลัก (Primary Basis Function) และฟังก์ชันคู่เทียบ (Dual Basis Approximation Wavelet) 2 ชุด คือ $A_w(j), \tilde{A}_w(j)$ และฟังก์ชันเวฟเล็ตที่ใช้กรองสัญญาณความถี่สูง (Wavelet Function : Detail Wavelet ($D_w(j)$)) 2 ชุด คือ $D_w(j), \tilde{D}_w(j)$ โดยเป็นฟังก์ชันชุดคู่

1. มีคุณสมบัติการตั้งฉากระหว่างฟังก์ชันคู่เทียบของตัวเอง

$$\langle A_w(t), \tilde{A}_w(t-k) \rangle = \delta(k)$$

$$\langle D_w(t), \tilde{D}_w(t-k) \rangle = \delta(k)$$

2. มีคุณสมบัติตั้งฉากระหว่างฟังก์ชันหลักของฟังก์ชันเวฟเล็ตกรองความถี่ต่ำ $A_w(j)$ และฟังก์ชันคู่เทียบของฟังก์ชันเวฟเล็ตกรองความถี่สูง $\tilde{D}_w(j)$

$$\langle A_w(t), \tilde{D}_w(t-k) \rangle = 0$$

3. มีคุณสมบัติตั้งฉากระหว่างฟังก์ชันหลักของฟังก์ชันเวฟเล็ตกรองความถี่สูง $D_w(j)$ และ ฟังก์ชันคู่เทียบของฟังก์ชันเวฟเล็ตกรองความถี่ต่ำ $\tilde{A}_w(j)$

$$\langle D_w(t), \tilde{A}_w(t-k) \rangle = 0$$

ฟังก์ชันเวฟเล็ตแม่ที่มีคุณสมบัติ Biorthogonal มีความสามารถในการรวบรวม (Capturing) ค่ากำลังของสัญญาณที่ต้องการได้อย่างรวดเร็วด้วยการใช้จำนวนของค่าสัมประสิทธิ์ที่น้อย ดังนั้นเทคนิคของการแปลงเวฟเล็ตแบบนี้ถูกนำมาประยุกต์ใช้กับงานทางด้านการประมวลผลสัญญาณ (Signal Approximation) สำหรับการประมาณสัญญาณนี้เป็นปัญหาที่สำคัญของการแทนสัญญาณด้วยองค์ประกอบจำนวนน้อยที่สุดเท่าที่มีความเป็นไปได้ ซึ่งคุณสมบัติการลดขนาดของข้อมูลภาพด้วยการแทนด้วยสัมประสิทธิ์การแปลงเวฟเล็ตจำนวนน้อย เป็นพื้นฐานการบีบอัดภาพด้วยเวฟเล็ต ตัวอย่างฟังก์ชันเวฟเล็ตแม่ เช่น Daubechies 9/7 ซึ่งนิยมใช้ในการแปลงสัมประสิทธิ์เวฟเล็ตเพื่อบีบอัดไฟล์รูปภาพแบบสูญเสียข้อมูล (Lossy Compression) และ LeGall 5/3 ซึ่งนิยมเพื่อบีบอัดไฟล์รูปภาพแบบไม่มีการสูญเสียข้อมูล (Lossless Compression) ในมาตรฐาน JPEG2000 และได้รับการยอมรับว่าเป็นฟังก์ชันเวฟเล็ตแม่ที่ให้ผลการบีบอัดข้อมูลที่ดี จะสังเกตได้ว่าสัมประสิทธิ์ในการแปลงเวฟเล็ตของฟังก์ชันแม่แบบ Biorthogonal นี้จะมีจำนวนค่าสัมประสิทธิ์ในการกรองสำหรับฟังก์ชันหลักและฟังก์ชันคู่เทียบไม่เท่ากัน เช่น ฟังก์ชันเวฟเล็ตแม่แบบ Daubechies 9/7 โดยจำนวนค่าสัมประสิทธิ์การกรองมีจำนวน 9 และ 7 ค่า เพื่อใช้ในการแปลงเวฟเล็ต (Decomposition Wavelet : Analysis Filter) และค่าสัมประสิทธิ์ในการแปลงกลับเวฟเล็ต (Reconstruction Wavelet : Synthesis Filter) ตัวอย่างสัมประสิทธิ์การกรองของฟังก์ชันเวฟเล็ตแม่แบบ Daubechies 9/7 และ Le Gall 5/3 แสดงในตารางที่ 2.4 และ 2.5 นอกจากนี้ฟังก์ชันเวฟเล็ตแม่แบบ Biorthogonal ยังมีคุณสมบัติสมมาตรและกระชับคุณสมบัติสมมาตรนี้เป็นคุณสมบัติที่ต้องการในการสร้างฟังก์ชันการกรองเนื่องจากให้ผลลัพธ์ของฟังก์ชันการแปลงที่ให้ความสัมพันธ์ของเฟสเชิงเส้น (Linear Phase)

2.4.3 การสร้างเวฟเล็ตแม่แบบจากฟังก์ชันพัลส์สี่เหลี่ยม (Two-Scale Relation from Rectangular Pulse)

การสร้างฟังก์ชันเวฟเล็ตแม่แบบสามารถสร้างได้จากการถ่วงน้ำหนักของลำดับของพัลส์สี่เหลี่ยมที่มีการสเกลและเลื่อนตำแหน่ง ซึ่งการออกแบบค่าถ่วงน้ำหนักสามารถออกแบบได้เช่นเดียวกับการออกแบบสัมประสิทธิ์การกรองสัญญาณ สมการฟังก์ชันเวฟเล็ตแม่แบบเป็นดังสมการที่ 2.27

$$M(x) = \sum_{k=-\infty}^{\infty} h_k P\left(\frac{2x}{s} - k\right) \quad (2.27)$$

$$P(x) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases} \quad (2.28)$$

ตารางที่ 2.4 สัมประสิทธิ์การแปลงเวฟเล็ตและการแปลงเวฟเล็ตสำหรับการคำนวณสัมประสิทธิ์ความถี่ต่ำและความถี่สูงของฟังก์ชันเวฟเล็ตแม่แบบ Daubechies 9/7

K	Analysis Filter		Synthesis Filter	
	Approximation Wavelet : Low Pass Filter (Dual Function)	Detail Wavelet : High Pass Filter (Dual Function)	Approximation Wavelet : Low Pass Filter (Primary Function)	Detail Wavelet : High Pass Filter (Primary Function)
-4,4	0.026748757411	0	0	0.026748757411
-3,3	-0.016864118443	-0.091271763114	-0.091271763114	0.016864118443
-2,2	-0.078223266529	0.057543526229	-0.057543526229	-0.078223266529
-1,1	0.266864118443	-0.591271763114	0.591271763114	-0.266864118443
0	0.602949018236	1.11508705	1.11508705	0.602949018236

ตารางที่ 2.5 สัมประสิทธิ์การแปลงเวฟเล็ตและการแปลงเวฟเล็ตสำหรับการคำนวณสัมประสิทธิ์ความถี่ต่ำและความถี่สูงของฟังก์ชันเวฟเล็ตแม่แบบ Le Gall 5/3

k	Lowpass Filter (h _k)	Highpass Filter (g _k)
0	6/8	1
±1	2/8	1/2
±2	-1/8	

1. ทำการแทนค่าฟังก์ชันเวฟเล็ดแม่ซึ่งอยู่ในรูปแบบของผลรวมของการถ่วงน้ำหนักฟังก์ชันพัลส์สี่เหลี่ยมในสมการแปลง

$$\begin{aligned}
 W_M(t) &= \sum_{n=-\infty}^{\infty} h_n \sqrt{2P(2t-n)} \\
 W_M(2^j x - k) &= \sum_{n=-\infty}^{\infty} h_n \sqrt{2P(2(2^j x - k) - n)} \\
 &= \sum_{n=-\infty}^{\infty} h_n \sqrt{2P(2^{j+1} x - 2k - n)}; \quad m = 2k + n \\
 &= \sum_{n=-\infty}^{\infty} h_{m-2k} \sqrt{2P(2^{j+1} x - m)}; \quad n = m - 2k
 \end{aligned}$$

2. นำฟังก์ชันเวฟเล็ดแม่มาทำการแปลงเวฟเล็ด

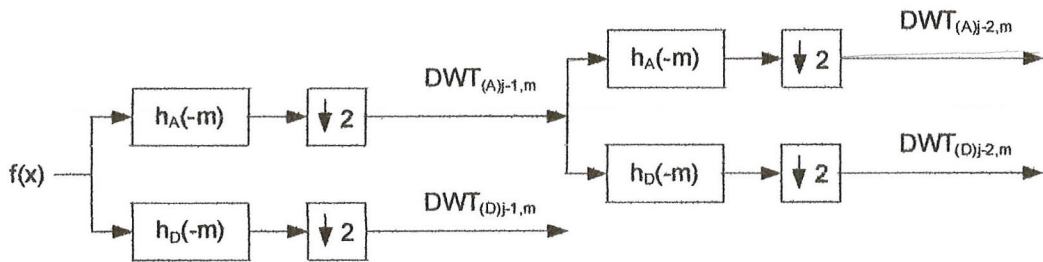
$$DWT_{j,k} = \frac{2^{j/2}}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) \sum_{n=-\infty}^{\infty} h_{m-2k} \sqrt{2P(2^{j+1} x - m)}$$

3. ทำการจัดเรียงสมการใหม่จะพบว่า การแปลงสัมประสิทธิ์เวฟเล็ดเป็นการแปลงแบบเป็นลำดับที่เกิดจากการทำคอนโวลูชัน (Convolution) ระหว่างสัมประสิทธิ์ของฟังก์ชันเวฟเล็ดแม่กับสัมประสิทธิ์เวฟเล็ดที่สเกลก่อนหน้า และทำการลดการสุ่มตัวอย่าง (Down Sampling) ด้วยองค์ประกอบ (Factor) ของสอง

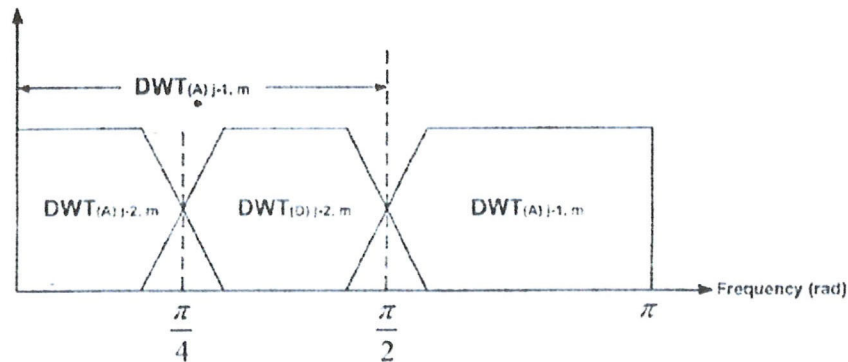
$$\begin{aligned}
 DWT_{j,k} &= \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} \sum_{n=-\infty}^{\infty} 2^{j/2} f(x) h_{m-2k} P(2^{j+1} x - m) \\
 &= \sum_{m=-\infty}^{\infty} h_{m-2k} \frac{2^{j/2}}{\sqrt{N}} \sum_{x=0}^{N-1} f(x) P(2^{j+1} x - m) \\
 &= \sum_{m=-\infty}^{\infty} h_{m-2k} DWT_{j+1,m} \\
 &= \sum_{m=-\infty}^{\infty} h_{(-2k - (-m))} DWT_{j+1,m} \\
 &= h_{-m} * DWT_{j+1,m} \quad \Big| \quad m = 2k, k \geq 0
 \end{aligned}$$

ดังนั้นค่าสัมประสิทธิ์ของตัวถ่วงน้ำหนักในฟังก์ชันเวฟเล็ดแม่นี้จึงเป็นค่าสัมประสิทธิ์ตัวกรองสัญญาณนั่นเอง และการแปลงสัมประสิทธิ์เวฟเล็ดจะทำการหาความสัมพันธ์ของสัญญาณกับตัวกรองสัญญาณความถี่ต่ำ (Approximation Function : $h_A(-m)$) และตัวกรองความถี่สูง (Detail Function :

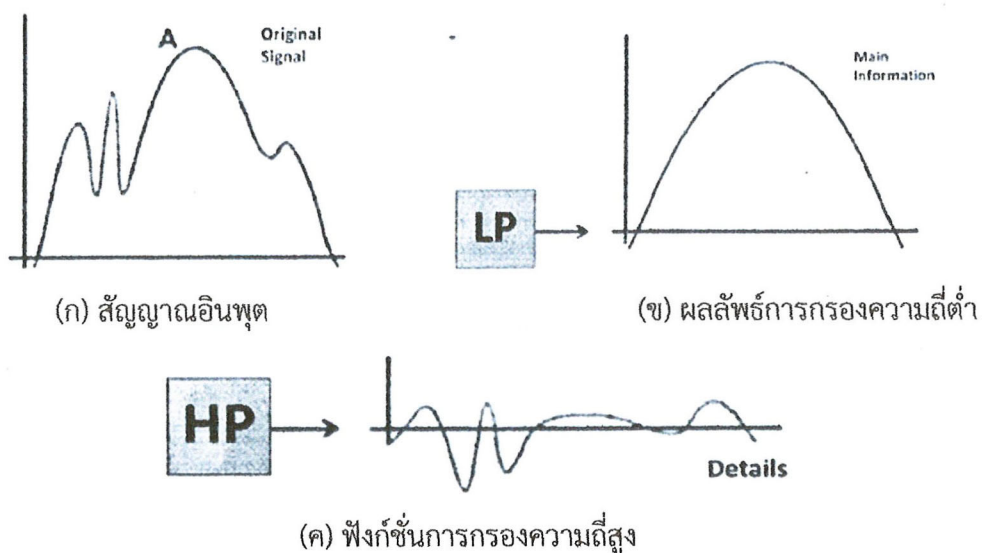
$h_D(-m)$) ซึ่งผลลัพธ์ที่ได้จากการกรองนี้จะถูกลดความละเอียดลงครึ่งหนึ่ง จึงทำให้ความละเอียด (Resolution) ของผลลัพธ์เหลือเพียงครึ่งหนึ่งของสัญญาณที่นำมาวิเคราะห์ และสามารถทำการแปลงต่อไปอย่างเป็นลำดับของความละเอียด ดังแสดงในรูปที่ 2.7 – 2.9 ทำให้สามารถวิเคราะห์ข้อมูลได้ละเอียดขึ้นในช่วงความถี่เฉพาะที่สนใจ



รูปที่ 2.7 ขั้นตอนการแปลงสัมประสิทธิ์เวฟเล็ตแบบ Multiresolution [5]



รูปที่ 2.8 การแบ่งสัดส่วนความถี่ในการกรองแต่ละลำดับขั้น [5]



รูปที่ 2.9 ขั้นตอนการแปลงเวฟเล็ตแบบหลายระดับ [5]

2.5 FPGA

ในปัจจุบันมี FPGA อยู่ 4 ชนิดได้แก่ Symmetrical array, Row based, Hierarchical PLD และ Sea of gates ซึ่งแต่ละชนิดก็มีลักษณะการเชื่อมต่อภายในและการโปรแกรมที่แตกต่างกันไป นอกจากนี้ในการแบ่งประเภทของ FPGA อาจแบ่งได้ตามเทคโนโลยีที่ใช้ในการโปรแกรมซึ่งก็จะมีอยู่ด้วยกัน 2 แบบคือ การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพของตัวชิพ และการโปรแกรมโดยการใช้หน่วยความจำ

1) การโปรแกรมโดยการทำให้เกิดการเปลี่ยนแปลงทางกายภาพ

- Fuse เป็นวิธีการโปรแกรมที่สามารถทำได้เพียงครั้งเดียว ซึ่งหลังจากที่โปรแกรมแล้วจุดเชื่อมต่อจะขาดจากกัน

- Anti fuse เป็นวิธีการโปรแกรมที่คล้ายกับแบบ Fuse แต่ต่างกันที่หลังจากทำการโปรแกรมแล้วจุดเชื่อมต่อจะเชื่อมถึงกัน

2) การโปรแกรมโดยใช้หน่วยความจำ

- Ethically Erasable Programmable Read Only Memory (EEPROM) based FPGA

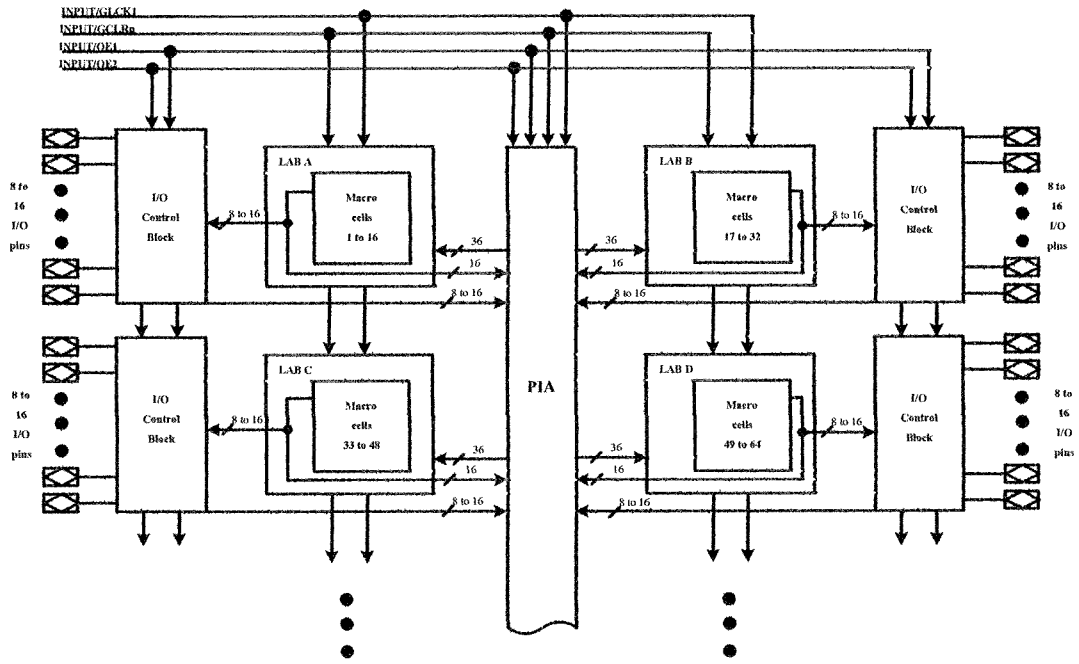
FPGA ที่ใช้การโปรแกรมแบบนี้มักเรียกว่า CPLD ซึ่งเทคโนโลยีที่ใช้จะเหมือนกับ EEPROM ทำให้มีความจำของเกตต่ำ โดยทั่วไปจะน้อยกว่า 20,000 เกต แต่ข้อดีของ EEPROM based FPGA คือสามารถเก็บข้อมูลที่โปรแกรมลงไปได้โดยไม่จำเป็นต้องมีไฟเลี้ยง และในการโปรแกรมจะใช้ทรานซิสเตอร์ 1 ตัวต่อ 1 บิต ซึ่งการโปรแกรมสามารถทำได้ประมาณ 10,000 ครั้ง

- Static Read Only Memory (SRAM) based FPGA

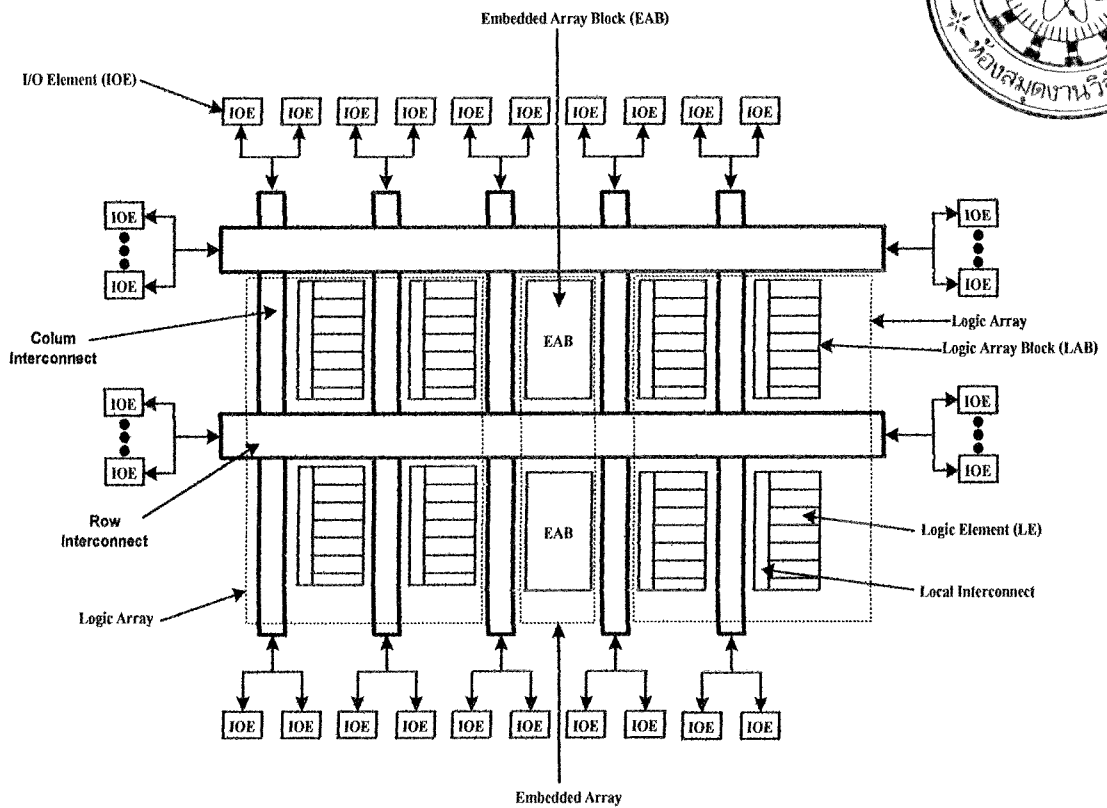
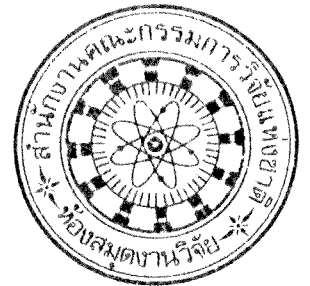
FPGA แบบนี้จะใช้เทคโนโลยีในการโปรแกรมเหมือนกับ SRAM ทำให้สามารถโปรแกรมซ้ำได้โดยไม่จำกัดจำนวนครั้ง นอกจากนี้ยังมีความจุของเกตในระดับปานกลาง ถึงสูงมาก (ประมาณ 10,000 - 1,000,000 เกต) ซึ่งข้อดีของ SRAM based FPGA คือ ใช้เวลาในการโปรแกรมน้อย การโปรแกรมทำได้ง่ายเทียบได้กับการเขียน SRAM ทั่วไป และเหมาะสำหรับการออกแบบวงจรที่มีแบบวงจรที่มีความสลับซับซ้อน ส่วนข้อเสียคือ ไม่สามารถเก็บโปรแกรมในภาวะที่ไม่มีไฟเลี้ยงได้ ดังนั้น FPGA ชนิดนี้จึงมักใช้ควบคู่กับ ROM เพื่อเก็บโปรแกรมและทำการโหลดโปรแกรมลงในตัวชิพในขณะเริ่มต้นใช้งาน

2.5.1 โครงสร้างภายในของ FPGA

ลักษณะโครงสร้างภายในของ FPGA จะเป็นอะเรย์ของบล็อกลอจิกที่สามารถทำการโปรแกรมได้ดังรูปที่ 2.10 และ 2.11



รูปที่ 2.10 โครงสร้างภายในของ FPGA ตระกูล MAX7000S



รูปที่ 2.11 โครงสร้างภายในของ FPGA ตระกูล FLEX10K

2.5.2 ปัจจัยที่ทำให้การออกแบบ FPGA ทำได้ง่ายและสะดวกรวดเร็ว

1) ผู้ออกแบบไม่จำเป็นต้องทราบถึงโครงสร้างภายในของตัวชิพ เพียงแต่มีความรู้เกี่ยวกับขั้นตอนการออกแบบลอจิกก็เพียงพอแล้ว ต่างกับการใช้ไมโครโปรเซสเซอร์ซึ่งจำเป็นต้องศึกษาโครงสร้างภายในรวมถึงภาษา Assembly ของไมโครโปรเซสเซอร์ตัวนั้นด้วย

2) มีการออกแบบโดยใช้ภาษาในการอธิบายการทำงานของวงจรถือ Hardware Description Language (HDL) เป็นเครื่องมือในการออกแบบ ซึ่งเป็นวิธีการที่มีความยืดหยุ่นสูงสามารถทำได้รวดเร็วและไม่จำเป็นต้องทราบถึงลักษณะของวงจรที่ต้องการว่าจะเชื่อมต่อกันอย่างไร เพียงแต่กำหนดลักษณะการทำงานให้มัน จากนั้นตัวซอฟต์แวร์จะทำ Synthesis and optimize ให้ทั้งหมด นอกจากนี้ภาษาที่ใช้ยังเป็นมาตรฐานเดียวกันสามารถใช้ได้กับชิพทุกตัวและทุกบริษัท

3) การโปรแกรมสามารถทำได้เองและใช้เวลาไม่นาน เพียงแค่ส่งข้อมูลผ่านสาย ดาวินโหลดทางพอร์ตของคอมพิวเตอร์ก็สามารถโปรแกรมตัวชิพ ขณะที่อยู่ในระบบได้โดยไม่ต้องถอดมาโปรแกรมข้างนอก และที่สำคัญสามารถโปรแกรมได้หลายครั้ง จึงทำให้ง่ายในการแก้ไขและพัฒนาโดยไม่ต้องเสียค่าใช้จ่ายเพิ่มแต่อย่างใด

2.5.3 การออกแบบภาษาฮาร์ดแวร์

ในการออกแบบวงจรดิจิทัลนั้นสามารถทำได้โดยการวาดวงจรหรือใช้ภาษาอธิบายพฤติกรรมของฮาร์ดแวร์ ในกรณีของการออกแบบวงจรด้วย ASICs ชนิด Full custom ผู้ออกแบบจะต้องเขียนวงจรด้วยการวาดวงจร จากนั้นจะนำวงจรที่ออกแบบไว้ไปทำการจำลองการทำงาน ซึ่งหากผลที่ออกมาเป็นที่ยอมรับก็จะต้อง Layout เป็นชั้นสาร และในการออกแบบ ASICs ในชนิดนี้ผู้ออกแบบจำเป็นต้องทราบถึงเทคโนโลยีที่ใช้ในการสร้างด้วย หลังจากได้ Layout ที่สมบูรณ์แล้วจึงจะทำการส่งไปเข้ากระบวนการสร้างวงจรรวมเพื่อสร้างเป็นชิพวงจรรวมออกมา แต่ในการออกแบบวงจรด้วย FPGA โดยการใช้การวาดวงจร หรือใช้ภาษาอธิบายการทำงานของวงจรจะทำได้สะดวกกว่าเนื่องจากวิธีการนี้ผู้ออกแบบไม่จำเป็นต้องคำนึงถึงเทคโนโลยีที่จะใช้สร้างวงจรรวมและที่สำคัญการออกแบบโดยวิธีนี้สามารถแก้ไขโมเดลหรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่า เพราะไม่ต้องวาดวงจรใหม่นั้นคือ การออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยีสำหรับภาษาที่ใช้สำหรับอธิบายพฤติกรรมของฮาร์ดแวร์ที่ใช้กันก็มี VHDL และ Verilog เป็นต้น ส่วนรายละเอียดของขั้นตอนในการออกแบบสามารถอธิบายได้ดังนี้

1) การสังเคราะห์วงจร (Logic synthesis) ในส่วนของขั้นตอนนี้จะใช้ซอฟต์แวร์ในการสังเคราะห์วงจร (Synthesis tools) ทำการสังเคราะห์พฤติกรรมของวงจรที่ได้จากการออกแบบด้วยการวาดวงจรหรือ VHDL ซึ่งต้องทำการตรวจสอบด้วยว่าซอฟต์แวร์นั้นสนับสนุนเทคโนโลยี FPGA ที่ต้องการหรือไม่ดังตัวอย่างเช่น FPGA ของบริษัท XILINX และบริษัท ALTERA นั้นจะมีซอฟต์แวร์หลายตัวที่สามารถใช้ได้เช่น Max Plus II ในขั้นตอนนี้ซอฟต์แวร์สังเคราะห์วงจรจะทำการแปลงโค้ด VHDL

และจะทำการ Optimize เพื่อให้ได้วงจรตามเทคโนโลยีที่จะเลือกใช้ในการสังเคราะห์วงจรนั้น วงจรระดับเกตจะไม่เหมาะสมกับโครงสร้างที่มีอยู่ในอุปกรณ์ FPGA ดังนั้นในการ Optimize ซอฟต์แวร์สังเคราะห์วงจร จะต้องทำการ Optimize ให้ได้เป็นวงจรที่ประกอบด้วยกลุ่มของลอจิกที่เหมาะสมกับอุปกรณ์ FPGA นั้นๆ จึงทำให้ผลที่ได้มีประสิทธิภาพและในขั้นตอนการสังเคราะห์วงจรนี้ ผู้ออกแบบสามารถกำหนดข้อบังคับสำหรับโมเดลแต่ละตัวได้เช่น ข้อบังคับในเรื่องเวลา หรือข้อบังคับในเรื่องของพื้นที่ หรือกำหนดชนิดและตำแหน่งของอินพุตและเอาต์พุต ซึ่งข้อบังคับเหล่านี้จะถูกนำไปใช้ในขั้นตอน Optimize เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนดส่วนสำคัญในการ Optimize คือ การได้เทียบโมเดลให้เข้ากับเทคโนโลยีที่ใช้ เพื่อให้ได้วงจรที่เหมาะสมกับโครงสร้างและสถาปัตยกรรมภายในของตัวอุปกรณ์ FPGA เมื่อทำการสังเคราะห์วงจรเสร็จแล้ว ซอฟต์แวร์การสังเคราะห์วงจรก็จะมีกรรายงานผลว่าโมเดลที่ออกแบบไปนั้นเป็นอย่างไรเช่น อาจมีค่าความหน่วงเวลาเท่าใด ใช้ทรัพยากรต่างๆ ใน FPGA อะไรบ้าง เมื่อมาถึงขั้นตอนนี้ผู้ออกแบบก็จะทราบว่าโมเดลเป็นไปตามข้อบังคับหรือไม่ ถ้าไม่ก็สังเคราะห์ใหม่จนกว่าจะเป็นไปตามที่กำหนด

2) การแบ่งวงจร (Partitioning) ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ เป็นส่วนย่อยๆ สำหรับลงในองค์ประกอบภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือ ให้แต่ละส่วนที่จะแยกออกจากกัน มีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำโดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจรเช่น เกต, ฟลิปฟลอป ลงในทรัพยากรต่างๆ ที่มีอยู่ในอุปกรณ์ FPGA หลังจากทำขั้นตอนนี้เสร็จแล้วผู้ออกแบบสามารถที่จะทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนข้อมูลทางเวลานั้นผู้ออกแบบจะทราบเฉพาะความหน่วงภายในแต่ละส่วนเท่านั้น หรือที่เรียกว่าความหน่วงลอจิก ส่วนซอฟต์แวร์จะรวมเอาซอฟต์แวร์ย่อยต่างๆ อีกเพื่อจะให้การทำ Partitioning Placement & Routing (PPR) เป็นไปอย่างต่อเนื่อง

3) การวางอุปกรณ์ (Placement) ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจรมาแล้วว่าจะอยู่ ณ ตำแหน่งไหนในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดเช่น วงจรส่วนไหนควรอยู่ใกล้กัน เพื่อจะได้ค้นหาเส้นทางได้ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้ว จะทำให้ความหน่วงเพิ่มขึ้นหรือ Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด การวางอุปกรณ์ที่ดีควรวางส่วนต่างๆ ให้อยู่ใกล้กันโดยเฉพาะส่วน ที่มีการเชื่อมต่อสัญญาณด้วยกัน นอกจากนั้นการกำหนดตำแหน่งขาอินพุตและตำแหน่งเอาต์พุตตามตำแหน่งขาของ FPGA บนแผ่น Printed Circuit Board (PCB) ก็จะมีผลโดยตรงเลยคือ ซอฟต์แวร์จะวางอินพุตและเอาต์พุตลงในตำแหน่งที่ผู้ออกแบบกำหนด ซึ่งบางครั้งตำแหน่งที่กำหนดไปไม่เหมาะสม ดังนั้นการกำหนดขา อินพุตและเอาต์พุตควรกำหนดตำแหน่งให้เหมาะสม หรือไม่ก็ให้ซอฟต์แวร์จัดการเอง

4) การเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะเป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่างๆ ภายในอุปกรณ์ FPGA ขั้นตอนนี้จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมด หรือเกิดความห่วงใยค่าที่กำหนดในข้อบังคับ ผู้ออกแบบสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟต์แวร์หรือผู้ออกแบบจะทำการเชื่อมต่อสัญญาณด้วยตนเองก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่า นอกจากนั้นการกำหนดข้อบังคับทางเวลาจะช่วยให้ผลที่ได้จากการเชื่อมต่อสัญญาณดีขึ้นได้

5) ความหน่วงด้านเวลา (Delay) คือ ในการทำ FPGA นั้นความหน่วงที่เกิดขึ้นจะเป็นความหน่วงที่เกิดจากการวางตำแหน่งของอุปกรณ์ ซึ่งผู้ออกแบบจะไม่สามารถเข้าไปแก้ไขได้ แต่สามารถทำให้มีความหน่วงน้อยที่สุดได้ สำหรับความหน่วงที่เกิดขึ้นนั้นแยกได้เป็น 2 ประเภทคือ ความหน่วงลอจิก เป็นความหน่วงภายในองค์ประกอบของอุปกรณ์ FPGA ความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณ เป็นความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณระหว่างองค์ประกอบภายในอุปกรณ์ FPGA โดยปกติแล้วค่าความหน่วงลอจิกไม่ควรเกิน 50% ของค่าความหน่วงที่ยอมรับได้ เพราะความหน่วงที่เกิดจากการเชื่อมต่อสัญญาณมักจะมีค่ามากกว่าค่าความหน่วงลอจิก ดังนั้นในการวางอุปกรณ์และเชื่อมต่อสัญญาณ ผู้ออกแบบควรกำหนดข้อบังคับทางเวลาเพื่อให้ซอฟต์แวร์ได้ทำงานอย่างมีประสิทธิภาพเพิ่มขึ้น และเพื่อให้ได้ผลลัพธ์ที่ดีขึ้นค่าความหน่วงที่ได้หลังจากการวางอุปกรณ์และเชื่อมต่อสัญญาณแล้วจะมีค่าความหน่วงที่ค่อนข้างแน่นอน ซึ่งผู้ออกแบบสามารถทราบได้ว่าโมเดลที่ออกแบบนั้นเป็นไปตามข้อกำหนดหรือไม่

6) การจำลองการทำงานของวงจร (Simulation) ในขั้นตอนนี้เป็นขั้นตอนที่สำคัญอีกขั้นตอนหนึ่ง เพราะเป็นขั้นตอนที่ผู้ออกแบบตรวจสอบฟังก์ชันการทำงานของโมเดลว่าถูกต้องหรือไม่ มีข้อผิดพลาดตรงไหนเพื่อจะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้มีซอฟต์แวร์ที่ใช้สำหรับทำการจำลองการทำงานของวงจรที่ใช้อยู่เช่น Model sim ของบริษัท Model technology หรือ Max plus II ของบริษัท Altera ในการจำลองการทำงานของวงจรควรทำทุกครั้งหลังจากที่มีการทำแต่ละขั้นตอนหลักเสร็จแล้ว เพื่อจะได้ทราบว่าข้อผิดพลาดของโมเดลเกิดขึ้นตอนไหน จะได้แก้ไขข้อผิดพลาดตรงขั้นตอนนั้นๆ ได้เลยไม่ต้องมาคอยตรวจหาขั้นตอนที่ทำให้เกิดข้อผิดพลาด นั่นคือ การทำการจำลองการทำงานของวงจรต้องทำทั้งหลังการเขียนคำสั่ง การสังเคราะห์วงจร และการจำลองการทำงานของวงจร หลังจากเขียนคำสั่งเสร็จแล้วนั้น ผู้ออกแบบสามารถทราบได้แค่โมเดลทำงานถูกต้องหรือไม่เท่านั้น ยังไม่สามารถตรวจสอบการทำงานในเชิงเวลาได้ถูกต้อง ในการจำลองการทำงานของวงจรหลังจากที่สังเคราะห์เป็นวงจรแล้ว เพื่อตรวจสอบว่าฟังก์ชันการทำงานยังคงถูกต้องหรือไม่ และค่าความหน่วงที่เกิดขึ้นเป็นไปตามข้อบังคับหรือไม่ มีข้อผิดพลาดเกิดขึ้นหรือไม่ถ้ามีจะแก้ไขให้ถูกต้องในการจำลองการทำงานของวงจรหลังจากทำการวางอุปกรณ์การเชื่อมต่อสัญญาณแล้วก็มีความสำคัญเช่นกัน เพราะผลที่ได้จากการจำลองการทำงานของวงจรในตอนนี้ จะเป็นผลลัพธ์ของโมเดลเลยซึ่งผู้ออกแบบนอกจากจะตรวจสอบฟังก์ชันการทำงานแล้วยังต้องตรวจสอบคุณสมบัติอื่นๆ เช่น ความหน่วงที่ได้จากการจำลอง

การทำงานของวงจรหลังจากที่เขียนโค้ดในรูปแบบค่าความหน่วงมาตรฐาน (Standard Delay Format : SDF) ว่าตรงตามที่กำหนดหรือไม่ หรือตรวจสอบว่าวงจรรวมสามารถใช้งานที่ความถี่สูงสุดเท่าไร ในการจำลองการทำงานของวงจรควรใช้ ซอฟต์แวร์ตัวเดียวกันตลอดเพื่อจะได้เปรียบเทียบผลที่ได้จาก ขั้นตอนต่างๆ

7) การโปรแกรมอุปกรณ์ FPGA (Configuration) หลังจากที่ไม่ได้ผ่านขั้นตอนต่างๆ จนกระทั่งผ่านการจำลองการทำงานของวงจรหลังจากที่เขียนคำสั่ง แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลดลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้เป็น ข้อมูลวงจร ซึ่งอยู่ในรูปของบิตสตรีม (Bit stream) ก่อนแล้วจึงดาวน์โหลดไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามโมเดลที่ผู้ออกแบบต้องการ ซึ่งในขั้นตอนนี้จะใช้วิธีที่แตกต่างกันออกไปสำหรับ อุปกรณ์ FPGA ของแต่ละบริษัทผู้ผลิตคือ ในกรณีที่เป็นอุปกรณ์ FPGA ชนิดที่ต้องโปรแกรมโดยวิธี SRAM นั้นในการใช้งานผู้ออกแบบจะต้องเก็บข้อมูลวงจรไว้ในหน่วยความจำประเภท EPROM ด้วยเพื่อ จะใช้งานสะดวกขึ้นคือ ในการใช้งานโมเดลครั้งต่อไป ไม่ต้องดาวน์โหลดข้อมูลวงจรจากเครื่อง คอมพิวเตอร์อีก เพราะมีข้อมูลวงจรเก็บอยู่ในหน่วยความจำอยู่แล้ว แต่กรณีที่อุปกรณ์ FPGA เป็นชนิด ที่โปรแกรมโดยวิธี EPROM หรือ Anti fuse ก็ไม่จำเป็นต้องมีหน่วยความจำสำหรับเก็บข้อมูลวงจร เพราะว่าอุปกรณ์ FPGA ชนิดนี้เมื่อดาวน์โหลดข้อมูลวงจรลงไปแล้ว ข้อมูลที่ดาวน์โหลดไปก็ยังคงอยู่ใน อุปกรณ์ FPGA และครั้งต่อไปก็ใช้งานโมเดลที่ออกแบบไว้ได้เลย

คุณลักษณะพื้นฐานของระบบดิจิทัลตั้งอยู่บนแนวคิดของหน่วยอิสระ การเชื่อมต่อ การกระทำพร้อมกัน และจังหวะเวลา

- หน่วยอิสระ (Entity) เป็นแนวคิดในการจำลองส่วนต่างๆ ของระบบ ซึ่งแต่ละส่วนจะทำงานได้ด้วยตัวเอง และมีความเป็นอิสระ ไม่มีสารสนเทศของส่วนอื่นแอบแฝงอยู่
- การเชื่อมต่อ (Connectivity) เป็นแนวคิดในการจำลองสายสัญญาณ ที่เชื่อมต่อส่วนต่างๆ ของระบบซึ่งเป็นวิธีที่หน่วยอิสระต่างๆ ของระบบได้ต่อบริเวณกัน
- การทำงานพร้อมกัน (Concurrency) คือ เมื่อหากสัญญาณในสายสัญญาณใดก็ตาม เปลี่ยนแปลง จะทำให้หน่วยอิสระที่รับสัญญาณนั้นเป็นอินพุตทำงานทันที เนื่องจากสายสัญญาณในระบบมีจำนวนมาก หน่วยอิสระจึงทำงานพร้อมกันเป็นจำนวนมาก
- จังหวะเวลา (Timing) การที่หน่วยอิสระของระบบดิจิทัลทำงานพร้อมกันทำให้จังหวะเวลา มีความสำคัญเพราะถ้าจังหวะการทำงานของหน่วยต่างๆ ไม่สอดคล้องกัน ระบบก็จะไม่สามารถทำงานที่ต้องการให้สำเร็จได้

จุดประสงค์สำคัญของภาษาอธิบายฮาร์ดแวร์คือ เพื่อให้ข้ออธิบายฮาร์ดแวร์ให้ตรงตามจริง และอย่างเที่ยงตรง การที่จะบรรลุวัตถุประสงค์นี้ได้ นั้น หลักสำคัญของภาษาอธิบายฮาร์ดแวร์คือ ต้องตั้งอยู่บนแบบจำลองการทำงานของฮาร์ดแวร์นั้นคือ มันจะต้องสามารถนำเอาคุณลักษณะของฮาร์ดแวร์ทั้ง 4 ประการข้างบนมาเขียนอธิบายได้ว่า เราแทนฮาร์ดแวร์เป็น 3 มุมและในแต่ละมุมสามารถแยกมองเป็น

4 ระดับของความคิดรวบยอด ในทุกมุมมองและทุกระดับนั้น ฮาร์ดแวร์จะมีคุณลักษณะพื้นฐานทั้ง 4 ประการข้างบนนี้เหมือนกัน แต่ระดับความละเอียดของการอธิบายอาจแตกต่างกันมาก เราอยากจะได้ภาษาอธิบายฮาร์ดแวร์ตัวเดียวสำหรับอธิบายฮาร์ดแวร์ในทุกมุมมองและทุกระดับ แต่ความต้องการนี้เป็นไปได้อย่างยาก เพราะจะทำให้ภาษานั้นซับซ้อนเกินไป เนื่องจากความแตกต่างของมุมมอง และระดับความคิดรวบยอด ภาษาอธิบายฮาร์ดแวร์สมัยใหม่นี้มักจะครอบคลุมมุมมอง 2 มุมคือ มุมมองด้านโครงสร้างและมุมมองด้านพฤติกรรมเท่านั้น ไม่ครอบคลุมมุมมองด้านกายภาพ ส่วนในมิติของระดับความคิดรวบยอดนั้น ภาษาอธิบายฮาร์ดแวร์จะมีรูปประโยคที่สามารถใช้อธิบายฮาร์ดแวร์ได้อย่างดีในระดับตัวประมวลผลและระดับเกท และสามารถอธิบายฮาร์ดแวร์ระดับตัวประมวลผลและระดับทรานซิสเตอร์ได้จำกัด รายการต่อไปนี้เป็นคุณสมบัติเด่นของภาษาอธิบายฮาร์ดแวร์

ก. ภาษาต้องมีความหมายของประโยคที่รวมเอาแนวคิดเรื่องหน่วยอิสระ การเชื่อมต่อการทำงานพร้อมกันและจังหวะเวลาไว้ได้หมด

ข. ภาษาจะต้องสามารถนำเอา Propagation delay และนำข้อมูลทางจังหวะเวลาอื่นๆของฮาร์ดแวร์เข้าในการอธิบายได้

ค. ภาษาต้องมีรูปประโยคที่อธิบายโครงสร้างของฮาร์ดแวร์ได้อย่างชัดเจน

ง. ภาษาต้องมีรูปประโยคที่สามารถอธิบายพฤติกรรมของระบบได้ รวมถึงทั้งรูปประโยคที่ใช้อธิบายการทำงานของกระบวนการแบบเป็นลำดับคล้ายกับในภาษาสำหรับเขียนโปรแกรมทั่วไป ทั้งนี้เพื่อให้สามารถอธิบายระบบในมุมมองของพฤติกรรมได้

จ. ภาษาจะต้องสามารถอธิบายลักษณะการทำงานและโครงสร้าง ที่ในระดับรีจิสเตอร์ ทรานส์เฟอร์และในระดับเกทได้อย่างดี

ฉ. ภาษาต้องมีรูปประโยคสำหรับกระบวนการออกแบบอย่างเป็นขั้นๆ ได้

2.5.4 ภาษา VHDL

VHDL และ Verilog เป็นภาษาอธิบายฮาร์ดแวร์ที่ได้รับความนิยมสูงสุด ภาษาทั้งสองนี้มีความสามารถและการครอบคลุมได้คล้ายๆ กันแต่กฎทางไวยากรณ์และสิ่งที่เห็นค่อนข้างแตกต่างกัน ทั้งสองเป็นภาษาที่มีมาตรฐานทางอุตสาหกรรมรองรับและซอฟต์แวร์ช่วยการออกแบบ ส่วนใหญ่สามารถรองรับได้ทั้งสองภาษา ซึ่ง VHDL ใช้ในการออกแบบแบบปรับพารามิเตอร์ได้ดีกว่า

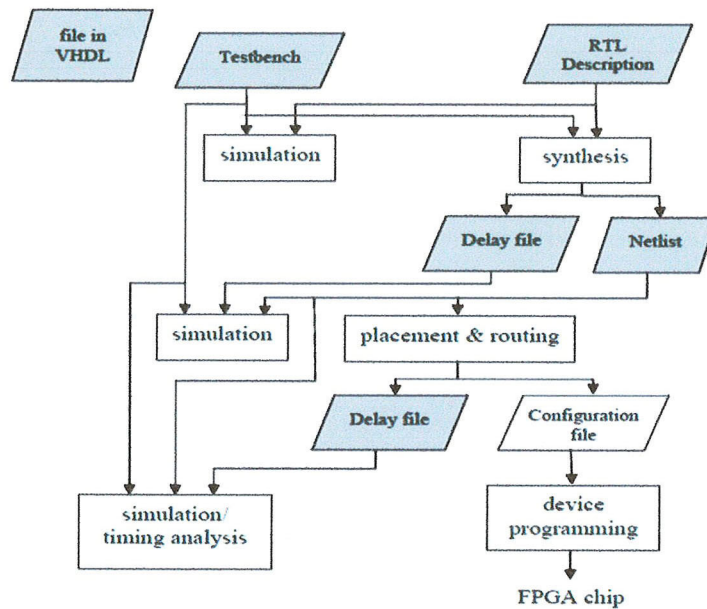
2.5.4.1 แนวคิดพื้นฐานของ VHDL ภายในกระบวนการพัฒนา

ตามที่ขี้อบอกไว้ภาษาอธิบายฮาร์ดแวร์คือ ภาษาสำหรับการอธิบายฮาร์ดแวร์ ดังนั้นจึงมีความสำคัญมากที่ผู้ที่จะใช้ภาษานี้ต้องอ่านและเขียนภาษานี้จากมุมมองของฮาร์ดแวร์ ในหัวข้อย่อยต่อไปนี้จะเราจะใช้ตัวอย่างของระบบดิจิทัลอย่างง่าย ๆ คือ วงจรตรวจจับพาริตีคู่ ในการเรียนรู้แนวคิดสำหรับจำลองแบบระบบที่ใช้ใน VHDL และในขณะเดียวกันเรียนรู้ความแตกต่างของความหมาย

ของประโยค ของภาษาอธิบายฮาร์ดแวร์ และของภาษาสำหรับเขียนโปรแกรมทั่วไป ตัวอย่างนี้จะทำให้เราเห็นภาพใหญ่ของภาษาอธิบายฮาร์ดแวร์และ VHDL ในขั้นตอนต่างๆ ของกระบวนการพัฒนา

1) ขอบเขตของ VHDL (Scope of VHDL)

เราจะใช้รูปที่ 2.12 ในการอธิบายถึงขอบเขตของการใช้ภาษา VHDL นี้ในขั้นแรกของกระบวนการพัฒนาระบบดิจิทัล



รูปที่ 2.12 การใช้ VHDL ในขั้นตอนต่างๆ ของการพัฒนา

หลังจากนั้นต้องการอธิบายระบบในรูปการกระทำต่างๆ เพื่อตรวจสอบระบบในรูปพฤติกรรมก่อน ดังนั้นในขั้นนี้ต้องสร้าง VHDL เชนพฤติกรรม ดังตัวอย่างของ Beh2_arch ของ Even_detector พร้อมกับพัฒนา VHDL Testbench ที่ใช้สำหรับตรวจสอบการทำงานของระบบได้อย่างครบถ้วน ดังตัวอย่างของ Testbench ของ Even_detector เมื่อระบบได้รับการตรวจสอบแล้วว่าสามารถทำงานได้ถูกต้อง ครบถ้วนตามข้อกำหนด เราจะทำการสังเคราะห์ระบบโดยใช้ไฟล์ VHDL ในเชิงพฤติกรรมเป็นอินพุต อย่างไรก็ตามถ้าระบบมีความซับซ้อนสูงการสังเคราะห์จาก VHDL บายพฤติกรรมเพื่อสร้าง Netlist โดยตรงอาจนำไปสู่ฮาร์ดแวร์ที่ซับซ้อนโดยไม่จำเป็น หรือสังเคราะห์ไม่ได้เลย ดังนั้นวิธีปฏิบัติคือ เราจะอธิบายระบบในระดับที่สังเคราะห์ได้ xor_arch และ beh1_arch เป็นตัวอย่างของ VHDL ที่เขียนอธิบายระบบโดยมุ่งที่จะนำไปสังเคราะห์ แต่สำหรับระบบที่ใหญ่และซับซ้อนนั้นเราอาจจำเป็นต้องแยกระบบเป็นระบบย่อยก่อนแล้วนำระบบย่อยนั้นไปเขียน VHDL ที่สังเคราะห์ได้ หลังจากนั้นเราจะใช้ Synthesis tool สังเคราะห์ Netlist และ Delay file ออกมา เราสามารถกำหนดให้ซอฟต์แวร์ตัวสังเคราะห์ให้ผลลัพธ์ทั้งสองไฟล์นี้ในรูปแบบของ VHDL ซึ่งตัวอย่างของ VHDL ของ Netlist ที่ได้จากการสังเคราะห์

คือ Str_arch ของ Even_detector สำหรับระบบที่ใหญ่ Netlist นั้นซับซ้อนเกินไปที่จะดูได้ด้วยตาเปล่า แต่เราใช้ Netlist นี้สำหรับการจำลองการทำงาน โดยใช้ร่วมกับ Testbench ที่ได้พัฒนาไว้แล้วเช่น ในตัวอย่างที่สามารถนำ Netlist ตาม Str_arch ไปจำลองการทำงานโดยการเชื่อมต่อ Even_detector กับ Str_arch การจำลองการทำงานของ Netlist จากการสังเคราะห์ก็เพื่อตรวจสอบก่อนว่า Netlist นั้นยังทำงานถูกต้อง ครบถ้วนตามข้อกำหนดอยู่ ยิ่งไปกว่านั้นการจำลองการทำงานในขั้นนี้ จะมีการนำเอาข้อมูลทางด้านเวลาจาก Delay file เข้าไปจำลองด้วยเพื่อตรวจสอบด้านจังหวะเวลาของระบบด้วย เมื่อตรวจสอบครบถ้วนแล้ว เราจึงจะนำ Netlist ไปทำการ Placement และ Routing ซึ่งผลจากขั้นตอนนี้คือ Layout หรือ Configuration file ที่พร้อมจะนำไปโปรแกรมลงบนชิพ FPGA พร้อมกันนั้นเครื่องมือในการทำ Placement และ Routing ก็มักจะให้ข้อมูล Delay ที่ละเอียดถูกต้องยิ่งขึ้นออกมาด้วย ข้อมูลเหล่านี้สามารถนำไปจำลองการทำงานร่วมกับ Netlist และ Testbench เดิม หรือนำไปวิเคราะห์ด้านจังหวะเวลาด้วย Timing analyzer ก็ได้ เมื่อทดสอบแล้วว่าระบบทำงานได้ตามข้อกำหนดทั้งทางด้านฟังก์ชันและเวลาเราจึงนำ Configuration file ไปโปรแกรมลงบนชิพ เพื่อทดสอบการทำงานจริงต่อไปจะเห็นว่า VHDL ทำให้เรามีกระบวนการพัฒนาที่เป็นหนึ่งเดียว นอกจาก VHDL จะเป็นเครื่องมือในการอธิบายระบบในทุกมุมมอง (ยกเว้นด้านกายภาพ) และทุกระดับตั้งแต่ระดับที่ความคิดรวบยอดสูงถึงระดับ Netlist ของเกตและเซลล์ แล้วมันยังเป็นกรอบในการจำลองการทำงานและการตรวจสอบระบบให้กับกระบวนการพัฒนา