บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

เนื้อหาในบทนี้กล่าวถึงงานวิจัยและทฤษฎีที่เกี่ยวข้อง ซึ่งงานวิจัยที่เกี่ยวข้องเป็นงานวิจัยที่ใกล้เคียง หรือวิธีการต่างๆ ของงานวิจัยนั้นสอดคล้องและเหมาะสมกับการนำมาประยุกต์ใช้กับระบบรู้จำเสียงพูดคำโดดในวิทยานิพนธ์นี้

2.1 งานวิจัยที่เกี่ยวข้อง

จากการศึกษางานวิจัยที่เกี่ยวกับกรรมวิธีและเทคนิคต่างๆ ของระบบรู้จำเสียงพูดภาษาไทยที่เป็นคำโดด มีดังนี้

สมชาย จิตพันธ์กุล [1] ศึกษาการรู้จำเสียงพูดคำไทยโดดๆ โดยไม่ขึ้นกับผู้พูด ที่เน้นเสียงตัวเลข และอยู่บนกรรมวิธีที่แตกต่างกัน 3 แบบ คือ ไดนามิก ไทม์วาร์ปิง แบบจำลองฮิดเดนมาร์คอฟ และนิวรอลเน็ตเวิร์ก เมื่อพิจารณาโดยรวมแล้ว ในการรู้จำเสียงตัวเลขภาษาไทย กรรมวิธีนิวรอลเน็ตเวิร์ก ควรเป็นกรรมวิธีที่มีสมรรถนะในการรู้จำสูงสุด ทั้งในด้านอัตราการรู้จำและความเร็วในการรู้จำ โดยมีกรรมวิธีแบบจำลองฮิดเดน มาร์คอฟ มีสมรรถนะใกล้เคียงกัน ในเรื่องอัตราการรู้จำ โดยที่ความเร็วในการรู้จำต่ำสุด

ไชยันต์ สุวรรณชีวะสิริ [2] ศึกษาการรู้จำเสียงพูดตัวเลขภาษาไทยแบบขึ้นกับผู้พูด โดยนำเอานิวรัลเน็ตเวอร์คมาประยุกต์ใช้ ทั้งในส่วนการหาพารามิเตอร์และส่วนการตัดสินใจ ซึ่งในส่วนการหาพารามิเตอร์จะเลือกใช้กลุ่มความถี่ฟอร์แมนท์ที่หนึ่งและสองของเสียงคำพูด โดยแบ่งจำนวนกลุ่มความถี่ออกเป็น 16 กลุ่ม ในระบบรู้จำเสียงคำพูดแบบขึ้นกับผู้พูด นิวรัลเน็ตเวอร์คสามารถตัดสินใจเสียงคำพูดได้ถูกต้อง 100% สำหรับกลุ่มตัวอย่าง และ 90% สำหรับกลุ่มทดสอบ ส่วนระบบรู้จำเสียงคำพูดแบบหลายผู้พูด นิวรัลเน็ตเวอร์คสามารถตัดสินใจเสียงคำพูดได้ถูกต้อง 95.4% สำหรับกลุ่มตัวอย่าง และ 87% สำหรับกลุ่มทดสอบ

ปิยสวัสดิ์ นวรัตน์ ณ อยุธยา [3] ศึกษาถึงการดึงคุณลักษณะและลดขนาดข้อมูลโดยอาศัยแบบจำลองของระบบโสตรับเสียงและเวฟเลท ซึ่งเป็นการประมวลข้อมูลเพื่อการรู้จำเสียงพูดที่มีความเหมือนกันสูง มีการหาประสิทธิภาพการย่อข้อมูลและผลการรู้จำเทียบกับการแปลงฟูเรียร์ โดยมีโครงข่ายประสาทเทียมเป็นระบบการตัดสินใจ

กาญจนา ทองบุญนาค [4] ศึกษาการรู้จำเสียงคำโดดด้วยโครงข่ายประสาทเทียม โดยใช้ข้อมูลนำเข้า 330 โหนด จำนวนโหนดในชั้นซ่อนตัวเท่ากับ 100 โหนด และค่าความผิดพลาดเฉลี่ยเท่ากับ 0.02 ซึ่งเป็นค่าที่ใช้เวลาในการฝึกสอนโครงข่ายน้อยที่สุด พบว่ามีอัตราการรู้จำสูงสุด 89% จากผลการวิจัยสรุปว่า ค่าความผิดพลาดเฉลี่ยที่มีความละเอียดสูง จะให้อัตราการรู้จำของชุดข้อมูลที่ใช้ในการฝึกโครงข่ายสูงมาก แต่ให้อัตราการรู้จำต่ำในชุดข้อมูลที่ใช้ทดสอบประสิทธิภาพการรู้จำแบบไม่ขึ้นกับผู้พูด

ชัย วุฒิวิวัฒน์ชัย และคณะ [5] ได้พัฒนาระบบระบุผู้พูดภาษาไทยแบบกำหนดคำพูดตายตัวและใช้ในสภาพแวดล้อมสำนักงาน คำพูดที่ใช้ในการวิจัยเป็นเสียงตัวเลขโดด 0-9 และตัวเลขโดดต่อกัน ใช้กับผู้พูดจำนวน 50 คน ซึ่งได้ผลอัตราการระบุผู้พูดสูงที่สุด 92.30% เมื่อใช้เสียงตัวเลข 0 และเพิ่มขึ้นเป็น 98% เมื่อใช้เสียงตัวเลขโดดต่อกัน 3 ตัว

ยังมีอีกหลายงานวิจัยที่นำเสนอแนวคิดในการนำระบบรู้จำเสียงพูดไปประยุกต์ใช้ให้เกิดประโยชน์ เช่น ธีรพันธ์ ระรื่นรมย์ [6] ได้เสนอการประยุกต์นิวรัลเน็ตเวอร์คในการรู้จำเสียงพูดเพื่อคัดแยกวัสดุ ต่อมา ฐนียา สัตยพานิช [7] เสนอระบบรู้จำเสียงภาษาไทยต่อเนื่องแบบเฉพาะบุคคลสำหรับการใช้งานอีเมลล์ และเอกรินทร์ แซ่เฮ้ง [8] เสนอการประยุกต์ใช้ระบบรู้จำเสียงพูดคำไทยสำหรับงานพิมพ์เอกสารโดยใช้เทคนิควิเคราะห์สเปกตรัมและโครงข่ายประสาทเทียม

นอกจากนั้นมีงานวิจัยของ นวลพิชา เหมทานนท์ [9] ที่นำเสนอการออกแบบวงจรรวม LPC ของระบบรู้จำเสียงแบบขึ้นกับเสียงผู้พูด ซึ่งออกแบบด้วยภาษา VHDL และสร้างเป็นสัญลักษณ์ (Symbol) ของวงจรรวมสำหรับการคำนวณ Floating point 24 บิต แล้วสังเคราะห์ลงบนชิป EP20K200EFC484-2X ด้วยโปรแกรม Quatus II งานวิจัยนี้เป็นเพียงการออกแบบในส่วนของการดึงค่าลักษณะสำคัญของเสียงพูดเท่านั้น ซึ่งต้องพัฒนาวงจรรวมในส่วนประมวลสัญญาณและเปรียบเทียบรูปแบบเพื่อนำมารวมกันให้ได้เป็นระบบรู้จำเสียงพูดที่สมบูรณ์ต่อไป อย่างไรก็ตาม ผลจากงานวิจัยนี้แสดงให้เห็นแนวทางในการพัฒนาระบบรู้จำเสียงพูดที่นอกเหนือไปจากการพัฒนาโปรแกรมเพื่อใช้งานกับเครื่องคอมพิวเตอร์ส่วนบุคคล

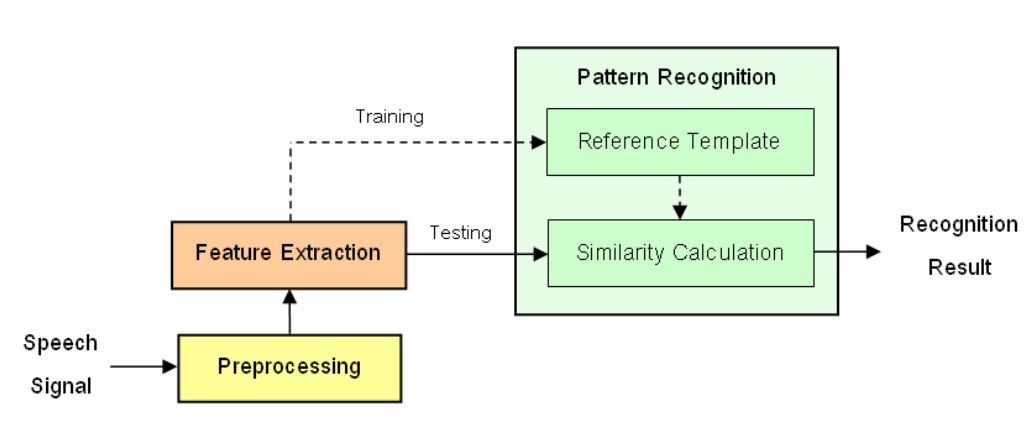
การตรวจหาขอบเขตของเสียงพูด (Endpoint detection) เป็นอีกวิธีการหนึ่งที่มีความสำคัญในระบบรู้จำเสียงพูดคำโดด เพราะนอกจากทำการตัดส่วนที่ไม่ใช่เสียงพูดออกแล้วยังมีโอกาสมากที่ลักษณะสำคัญของเสียงพูดไปอยู่ที่ส่วนต้นและส่วนท้ายของคำ มีงานวิจัยที่น่าสนใจ เช่น Liang-sheng Huang และ Chung-ho Yang [10] เสนอวิธีการตรวจหาขอบเขตของเสียงพูดในสภาพแวดล้อมภายในรถยนต์ โดยการหาลักษณะที่เกิดจากรวมกันของค่า energy และ entropy พบว่าวิธีการใหม่มีความแม่นยำสูงกว่าวิธีการพื้นฐานที่ใช้เฉพาะค่า energy

เช่นเดียวกับงานวิจัยของ S.E. Bou-Ghazale และ K. Assaleh [11] ที่ใช้ทั้งค่า energy และลักษณะ cepstral ของสัญญาณในการตรวจหาขอบเขตของเสียงพูดเพื่อเพิ่มประสิทธิภาพการรู้จำในสภาพแวดล้อมภายในโรงแรมและสำนักงาน รวมถึงงานวิจัยของ Fan Yingle และคณะ [12] ที่ใช้เทคนิคการเพิ่มคุณภาพของเสียงโดยปรับปรุงสัญญาณทั้งในโดเมนความถี่และโดเมนเวลา ทำให้การตรวจหาขอบเขตของเสียงพูดแม่นยำขึ้นในสภาพแวดล้อมแบบต่างๆ

งานวิจัยอื่นๆ ที่เกี่ยวกับการปรับปรุงค่าลักษณะสำคัญของเสียง เช่น Bo Yin และคณะ [13] เสนอการรวมค่าลักษณะสำคัญของเสียงแบบ cepstral กับค่าทางฉันทลักษณ์เพื่อเพิ่มประสิทธิภาพของระบบระบุภาษา งานวิจัยของ A. Amsalem และ I.D. Shallom [14] เสนอวิธีการหาตัวแทนของเสียงจากแบบจำลองของค่าลักษณะสำคัญที่มีการเปลี่ยนแปลงตามเวลา ทำให้ระบบรู้จำเสียงพูดมีประสิทธิภาพเพิ่มขึ้นในสภาพแวดล้อมที่มีสัญญาณรบกวนสูง เป็นต้น

2.2 ทฤษฎีในการสร้างระบบรู้จำเสียงพูดคำโดด

ระบบรู้จำเสียงพูดคำโดดที่ใช้ในงานวิจัยนี้สามารถจำแนกกระบวนการต่างๆ ออกเป็น 3 ส่วนใหญ่ๆ คือ การประมวลสัญญาณเบื้องต้น (Preprocessing) การดึงค่าลักษณะสำคัญ (Feature Extraction) และการรู้จำรูปแบบ (Pattern Recognition) [5]



รูปที่ 2.1 การทำงานโดยรวมของระบบรู้จำเสียงพูดคำโดด

ในการรู้จำรูปแบบ แบ่งออกเป็น 2 ขั้นตอน ขั้นตอนแรก คือ การฝึกฝน (Training) เป็นการสร้างข้อมูลอ้างอิง (Reference Template) ขึ้นจากคำในชุดฝึกฝน (Training set) และ

ขั้นตอนที่สอง คือ การทดสอบเพื่อหาผลลัพธ์ของระบบ (Testing) เป็นการหาค่าความคล้ายคลึงกันของคำ (Similarity Calculation) โดยอาศัยข้อมูลอ้างอิงที่เก็บไว้ในขั้นตอนแรก

2.2.1 การประมวลสัญญาณเบื้องต้น (Preprocessing)

เนื่องจากสัญญาณเสียงที่รับเข้ามาในระบบจะมีสัญญาณรบกวนรวมอยู่ด้วย จึงต้องมีขั้นตอนการลดสัญญาณรบกวน รวมถึงการแบ่งสัญญาณเสียงซึ่งเป็นวิธีการจัดเตรียมสัญญาณให้เหมาะกับการประมวลผลในขั้นตอนต่อไป

2.2.1.1 การเน้นล่วงหน้า (Preemphasis)

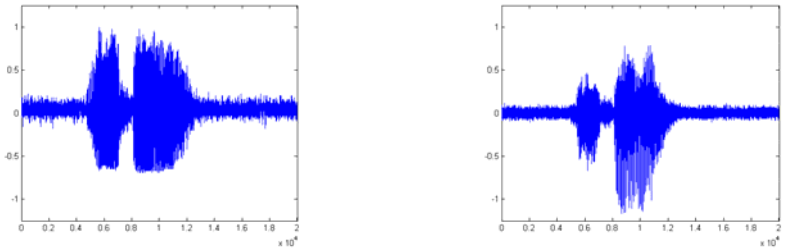
การเน้นล่วงหน้าของสัญญาณ [4]-[6] คือ การใช้วงจรกรองดิจิตอลอันดับหนึ่ง (first-order preemphasis filter [15]) เพื่อทำให้ความลาดเอียงของสเปกตรัมแบนราบลง [16] ซึ่งมีฟังก์ชันถ่ายโอน (Transfer function) ดังสมการ

$$H(z) = 1 - a \cdot z^{-1} \tag{2.1}$$

การเน้นล่วงหน้าของสัญญาณในสมการที่ (2.2) จะช่วยลดผลกระทบจากสัญญาณรบกวนความถี่ต่ำทำให้มีอัตราส่วนสัญญาณเสียงต่อสัญญาณรบกวน (Signal to Noise Ratio: SNR) สูงขึ้น

$$\tilde{S}(n) = S(n) - a \cdot S(n-1), \qquad 0.9 < a < 1 \tag{2.2}$$

โดยที่ a คือ ค่าสัมประสิทธิ์ตัวกรอง



รูปที่ 2.2 (ก) สัญญาณเสียงที่รับเข้ามาในระบบ (ข) สัญญาณเสียงหลังจาก Preemphasis

เนื่องจากเสียงพูดเป็นสัญญาณที่เปลี่ยนแปลงแบบไม่คงที่ จึงจำเป็นต้องแบ่งสัญญาณออกเป็นช่วงสั้นๆ (Frames) โดยทั่วไปในงานวิจัยเกี่ยวกับเสียงพูด [4]-[7] จะแบ่งสัญญาณให้มีขนาดเฟรมละ 10-30 มิลลิวินาที (mSec) เพราะถือว่าสัญญาณภายในช่วงนี้ค่อนข้างคงที่ (Stationary) สามารถที่จะวิเคราะห์คุณสมบัติและหาค่าทางสถิติได้

2.2.1.2 การเพิ่มค่าความแตกต่างของเฟรมรอบข้างให้สัญญาณ

หลังจากที่สัญญาณผ่านการ Preemphasis และถูกแบ่งเป็นเฟรมแล้วจึงทำการหาค่าความแตกต่างของเฟรมรอบข้าง (Difference of pre-frame and post-frame: Delta [17]) ของสัญญาณ ตามสมการ

$$D_j(l) = \mu \sum_{k=-T}^{T} k\tilde{S}_{j+k}(l) \quad , \quad 1 \le l \le L_f \tag{2.3}$$

โดยที่ $\tilde{S}$ คือ สัญญาณหลังผ่านการ Preemphasis ในเฟรมที่ j

k คือ ค่าสัมประสิทธิ์ Delta และ T คือ จำนวนเฟรมรอบข้าง (กำหนดให้มีค่าเท่ากับ 2)

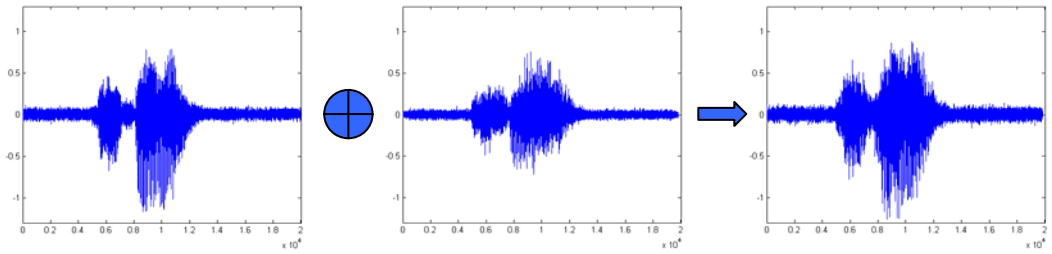
μ คือ อัตราขยายของค่า Delta มีค่าเท่ากับ T^{-2}

$D_j(l)$ คือ ค่าความแตกต่างของเฟรมรอบข้างตัวที่ l ในเฟรมที่ j

L_f คือ จำนวนตัวอย่าง (Samples) ในแต่ละเฟรม

เมื่อรวมสัญญาณหลังผ่านการ Preemphasis กับค่า Delta ของสัญญาณในทุกเฟรมจะได้สัญญาณใหม่ดังสมการ

$$S'(m) = \tilde{S}(m) + D(m) \quad , \quad m < n \tag{2.4}$$



รูปที่ 2.3 การเพิ่มค่า Delta ให้สัญญาณ

สัญญาณใหม่ที่ได้จะมีค่าความแตกต่างระหว่างเฟรมรอบข้างรวมอยู่ทำให้มีลักษณะสำคัญในสัญญาณเสียงมากขึ้นและทำให้พลังงานของเสียงในช่วงเริ่มต้นและสิ้นสุดของคำเพิ่มขึ้นอีกด้วย ซึ่งช่วยให้การตรวจหาขอบเขตของคำแม่นยำยิ่งขึ้น

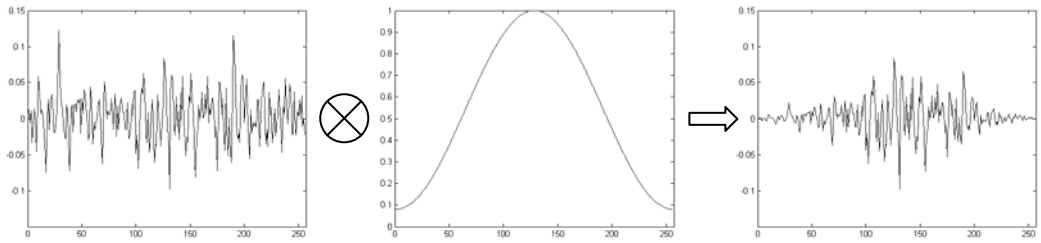
2.2.1.3 การวางกรอบสัญญาณ (Windowing)

เมื่อได้สัญญาณ S' ครบทุกเฟรมแล้ว เพื่อความต่อเนื่องของสัญญาณจึงกำหนดให้แต่ละเฟรมเหลื่อมกัน p ตัวอย่าง จากนั้นนำสัญญาณในแต่ละเฟรมไปคูณกับฟังก์ชันหน้าต่าง (Window function) [1],[4] เพื่อลดการเปลี่ยนแปลงอย่างรวดเร็วที่เกิดขึ้นบริเวณปลายแต่ละข้างของเฟรม ในงานวิจัยนี้ใช้ฟังก์ชันหน้าต่างแบบแฮมมิ่ง (Hamming window) [18] ซึ่งมีสมการ คือ

$$h(l+1) = 0.54 - 0.46\cos\left(2\pi \frac{l}{L_w - 1}\right) \quad , 0 \le l \le L_w - 1 \tag{2.5}$$

โดย L_w คือ ขนาดของกรอบหน้าต่าง มีค่าเท่ากับ $L_f + p$

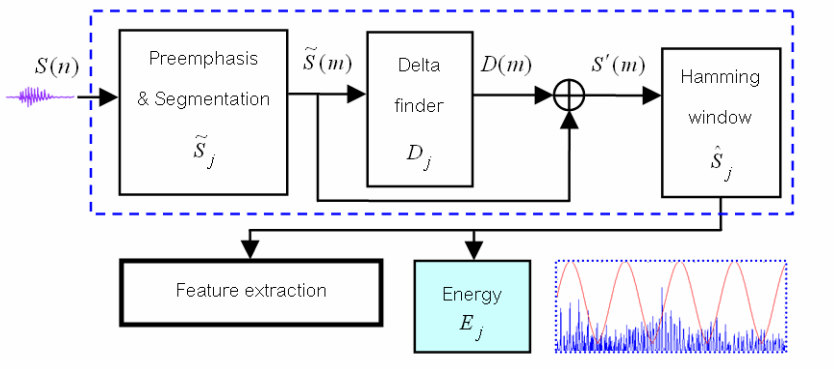
$$\hat{S}_j(l) = S'_j(l) \cdot h(l) \quad , 1 \le l \le L_w \tag{2.6}$$



รูปที่ 2.4 การวางกรอบสัญญาณโดยใช้ Hamming Window

ดังนั้น ค่าพลังงานของเสียง [10],[12] ในเฟรมที่ j ซึ่งนำไปใช้ในการตรวจหาขอบเขตของคำสามารถคำนวณได้จากสมการ

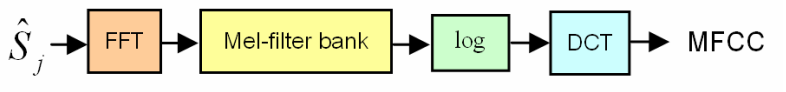
$$E_j = \sum_{l=1}^{L_w} \hat{S}_j^{\,2}(l) \tag{2.7}$$



รูปที่ 2.5 การประมวลสัญญาณเบื้องต้น (Preprocessing)

2.2.2 **การดึงค่าลักษณะสำคัญ** (Feature Extraction)

การดึงค่าลักษณะสำคัญของเสียงพูด คือ การหาค่าลักษณะเด่นที่เป็นตัวแทน (Representation) ของสัญญาณเพื่อใช้ในการลดจำนวนข้อมูล โดยเสียงพูดแบบเดียวกันก็จะมีลักษณะเด่นของเสียงที่เหมือนหรือคล้ายกัน ซึ่งเป็นขั้นตอนสำคัญที่ส่งผลต่อประสิทธิภาพการรู้จำของระบบ วิธีการหาลักษณะสำคัญของเสียงพูดมีอยู่หลายวิธี โดยทั่วไปจะอยู่บนพื้นฐานการวิเคราะห์สเปกตรัม ในงานวิจัยนี้ใช้วิธีการหาค่าสัมประสิทธิ์เคปสตรัมบนความถี่เมล (Mel-Frequency Cepstral Coefficients: MFCCs) [5],[7],[8],[13],[14],[16]



รูปที่ 2.6 ขั้นตอนการดึงค่าลักษณะสำคัญ MFCC

โดยสัญญาณ $\hat{S}$ ในแต่ละเฟรมจะถูกแปลงฟูเรียร์ (FFT) เพื่อหาค่าสเปกตรัมกำลัง (Power spectrum) ส่งเข้าไปในชุดของตัวกรอง (Filter bank) ที่อยู่บนสเกลความถี่แบบเมล (Mel-scale) ได้เป็นค่าสเปกตรัมกำลังแบบเมล (Mel-power spectrum) จากนั้นใส่ค่าลอการิทึม (log) ก่อนที่จะแปลงโคซายน์ (DCT) ให้เป็นค่าสัมประสิทธิ์เคปสตรัมบนความถี่เมล (MFCC)

2.2.2.1 **การแปลงฟูเรียร์อย่างเร็ว** (Fast Fourier Transform: FFT)

DFT (Discrete Fourier Transform) [19] คือ การแปลงสัญญาณแบบไม่ต่อเนื่อง (Discrete signal) ในโดเมนเวลา (Time domain) แทนด้วย $x(n)$ ให้เป็นสัญญาณในโดเมนความถี่ (Frequency domain) แทนด้วย $X(k)$ โดยการแยกสัญญาณออกเป็นส่วนประกอบทางความถี่เพื่อใช้วิเคราะห์ค่าสเปกตรัมของสัญญาณนั้นๆ $X(k)$ จะมีทั้งค่าจริงและค่าจินตภาพ โดยที่ค่าจริงแสดงแอมพลิจูด (Amplitude) ของรูปคลื่น cosine ส่วนค่าจินตภาพแสดงแอมพลิจูดของรูปคลื่น sine ถ้าหากพิจารณาเฉพาะขนาด (Magnitude) ของสัญญาณในโดเมนความถี่จะได้ว่า

$$\left|X(k)\right| = \sqrt{\mathrm{Re}\,X(k)^2 + \mathrm{Im}\,X(k)^2} \tag{2.8}$$

ถ้าให้สัญญาณ $x(n)$ มีเฉพาะค่าจริงจำนวน N ตัวอย่าง หลังจากการแปลงฟูเรียร์แล้วจะใช้ค่า $X(k)$ เพียง $N/2+1$ จุด (0 ถึง $N/2$) แสดงองค์ประกอบทางด้านความถี่ของสัญญาณ ในการวิเคราะห์สเปกตรัมอย่างง่ายจะพิจารณาสเปกตรัมกำลัง (Power spectrum) ดังสมการ

$$P_X(k) = \frac{1}{N}\left|X(k)\right|^2 = \frac{1}{N}\left(\mathrm{Re}\,X(k)^2 + \mathrm{Im}\,X(k)^2\right) \tag{2.9}$$

FFT (Fast Fourier Transform) เป็นวิธีการคำนวณ DFT ที่ใช้เวลาน้อยลง เนื่องจากลดการคำนวณโดยใช้วิธี Radix-2 แบบแตกส่วนย่อยทางเวลา (Decimation in Time) [8] ซึ่งหากพิจารณาสมการที่ใช้คำนวณ DFT (สมการที่ 2.10) ถ้า N เป็นเลขคู่จะสามารถกระจาย $X(k)$ ให้อยู่ในรูปผลบวกของเทอมที่ค่า n เป็นคู่และเทอมที่ค่า n เป็นคี่ได้

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk},\quad 0\le k<N \quad \text{เมื่อ} \quad W_N=e^{(-j2\pi)/N} \tag{2.10}$$

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)W_N^{2nk}+\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)W_N^{(2n+1)k} \tag{2.11}$$

$$X(k)=\underbrace{\sum_{n=0}^{\frac{N}{2}-1}x(2n)W_N^{2nk}}_{even}\quad+\quad\underbrace{\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)W_N^{2nk}W_N^{k}}_{odd} \tag{2.12}$$

พิจารณา

$$W_N^{2nk}=e^{2nk\frac{-j2\pi}{N}}=e^{nk\frac{-j2\pi}{N/2}}=W_{N/2}^{nk} \tag{2.13}$$

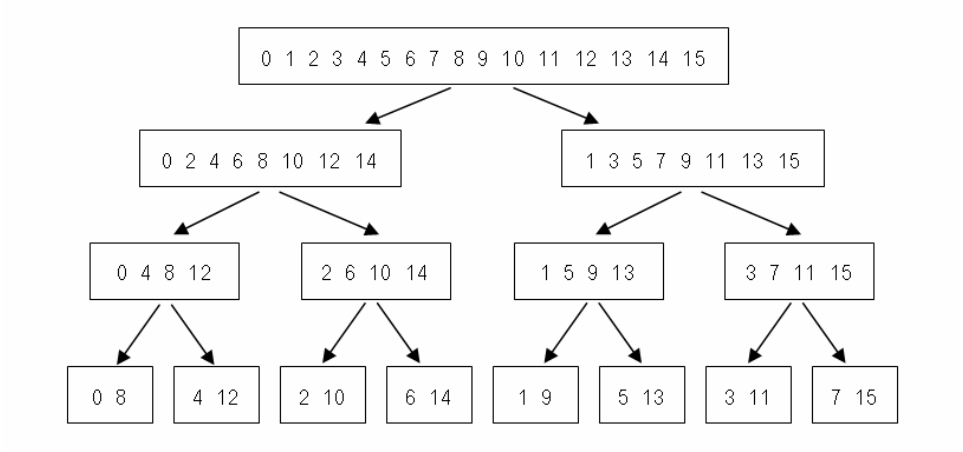
เมื่อแทนค่า W_N^{2nk} ในสมการ (2.12) ด้วย $W_{N/2}^{nk}$ จะได้

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}x(2n)W_{N/2}^{nk}+\sum_{n=0}^{\frac{N}{2}-1}x(2n+1)W_{N/2}^{nk}W_N^{k} \tag{2.14}$$

เมื่อให้ $g(n)=x(2n)$ และ $h(n)=x(2n+1)$ จะได้

$$X(k)=\sum_{n=0}^{\frac{N}{2}-1}g(n)W_{N/2}^{nk}+W_N^{k}\sum_{n=0}^{\frac{N}{2}-1}h(n)W_{N/2}^{nk} \tag{2.15}$$

จะเห็นได้ว่า $X(k)$ เป็นผลบวกของสองเทอม ซึ่งแต่ละเทอมเป็นรูปแบบการคำนวณ DFT $N/2$ จุด โดยเทอมแรกคำนวณกับสัญญาณ $x(0),\ x(2),\ \ldots,\ x(N\text{-}2)$ ส่วนเทอมที่สองคำนวณกับสัญญาณ $x(1), x(3), \ldots, x(N\text{-}1)$ แล้วคูณด้วย W_N^k เช่นเดียวกันหากทำการแตกเทอม DFT $N/2$ จุดต่อไปก็จะสามารถแยกเป็นผลบวกของ DFT $N/4$ จุดสองเทอม โดยสามารถแตกย่อยไปได้อีกเรื่อยๆ จนอยู่ในรูปของการคำนวณ DFT 2 จุด ซึ่งการแตกย่อยในแต่ละครั้งจะส่งผลให้ตำแหน่งของสัญญาณ $x(n)$ ถูกสลับที่ไป (scrambled)



รูปที่ 2.7 การแตกย่อยสัญญาณ สำหรับการคำนวณ FFT 16 จุด

เมื่อ $X(k)$ เป็น DFT 2 จุด จะได้

$$X(k) = \sum_{0}^{1} x(n) W_2^{nk} \tag{2.16}$$

เนื่องจาก $W_2^0 = e^0 = 1$ และ $W_2^1 = e^{(-j2\pi)/2} = e^{-j\pi} = \cos(\pi) - j\sin(\pi) = -1$ จะได้

$$\left.\begin{aligned} X(0) &= x(0) + x(1) \\ X(1) &= x(0) - x(1) \end{aligned}\right\} \tag{2.17}$$

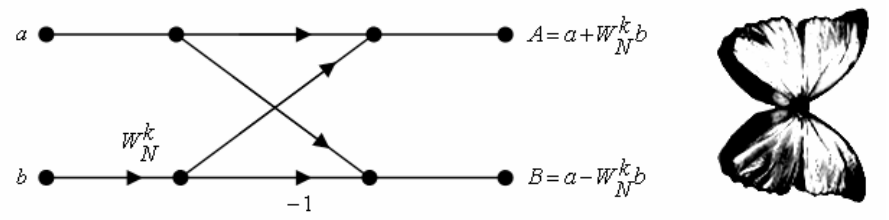
และสามารถจัดรูปแบบ W_N^k ใหม่ โดยใช้คุณสมบัติความสมมาตรของ W_N ได้ดังนี้

$$W_N^{k+N/2} = W_N^k \times W_N^{N/2} = W_N^k(-1) = -W_N^k \tag{2.18}$$

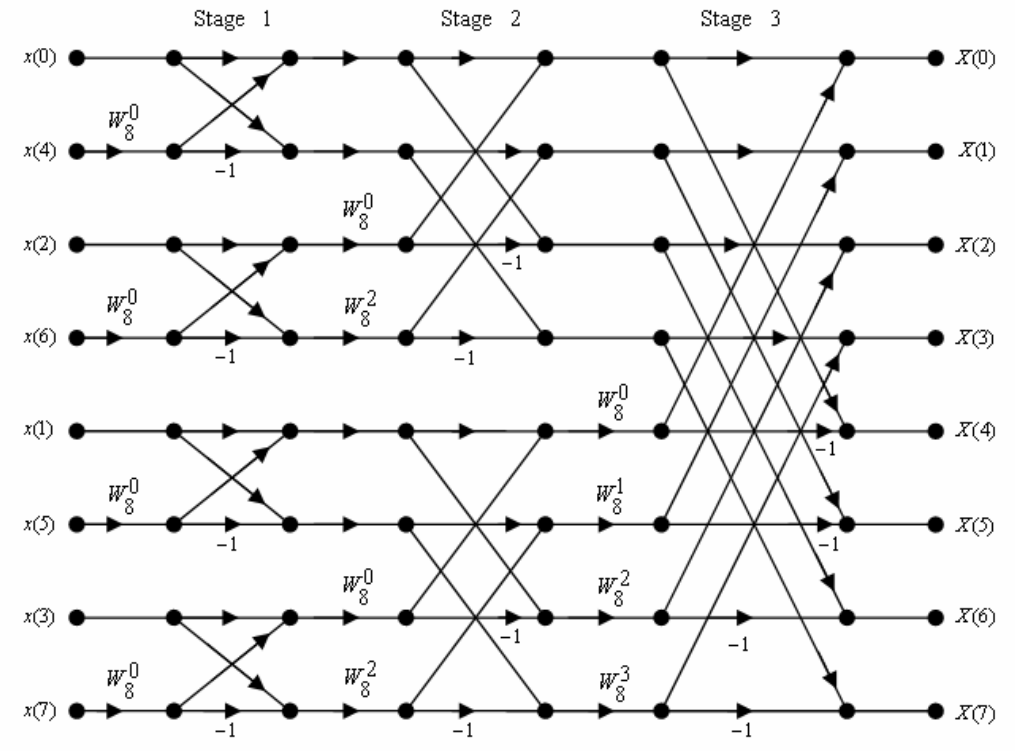
ถ้าให้ N เท่ากับ 8 จะได้ว่า

$$W_8^4 = -W_8^0, \qquad W_8^5 = -W_8^1, \qquad W_8^6 = -W_8^2, \qquad W_8^7 = -W_8^3$$

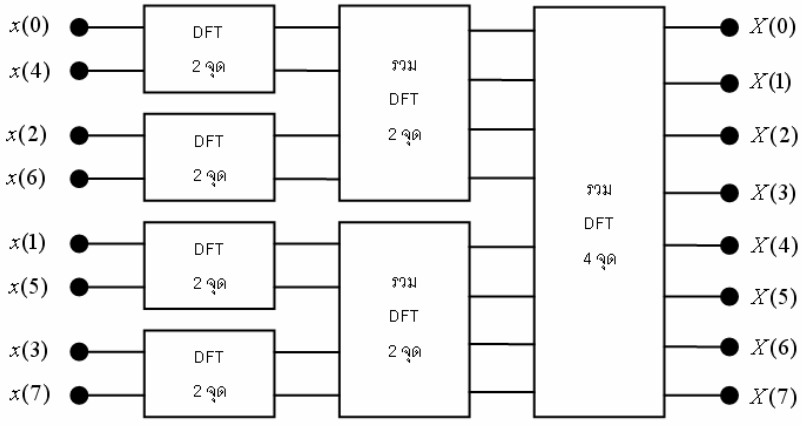
ซึ่งขั้นตอนทั้งหมดของ FFT [20] จะเขียนแทนโดยใช้แผนภาพผีเสื้อ (Butterfly diagram)



รูปที่ 2.8 รูปแบบการคำนวณของแผนภาพผีเสื้อ



รูปที่ 2.9 แผนภาพรวมของการคำนวณ FFT 8 จุด



รูปที่ 2.10 การคำนวณ FFT 8 จุด แบ่งออกเป็น 3 ขั้นตอน

แผนภาพผีเสื้อมีสิ่งที่ควรสังเกต คือ

- วิธี Radix-2 นี้ใช้ได้เฉพาะเมื่อค่า $N = 2^m$ โดย m คือ จำนวนเต็มบวกใดๆ ซึ่งหากจำนวนข้อมูลมีไม่พอสามารถใช้เทคนิคการเติมศูนย์ (Zero padding) ได้
- หากต้องการได้สัญญาณ $X(k)$ เรียงตามลำดับจาก $X(0)$, $X(1)$, …, $X(7)$ ต้องทำการเรียงลำดับสัญญาณ $x(n)$ ใหม่ดังนี้ $x(0), x(4), x(2), x(6), x(1), x(5), x(3)$ และ $x(7)$

- สำหรับการคำนวณ FFT N จุด สามารถคำนวณ W_N^k ที่ค่า k ต่างๆ ไว้ล่วงหน้าได้ เปรียบเสมือนเป็นค่าคงที่
- การคำนวณ FFT N จุด ถูกแบ่งเป็น $\log_2 N$ ขั้นตอน (Stages) โดยอาจประมาณได้ว่าแต่ละขั้นตอนมีการคูณเลขเชิงซ้อนเท่ากับ N ครั้ง (มีเส้นทแยงในแผนภาพ N เส้น ในทุกๆ ขั้นตอน) ดังนั้นจึงมีการคูณทั้งหมดเท่ากับ N $(\log_2 N)$ ซึ่งน้อยกว่าการคำนวณ DFT ที่ต้องใช้การคูณเลขเชิงซ้อนถึง N^2

2.2.2.2 **การแปลงโคซายน์ไม่ต่อเนื่อง** (Discrete Cosine Transform: DCT)

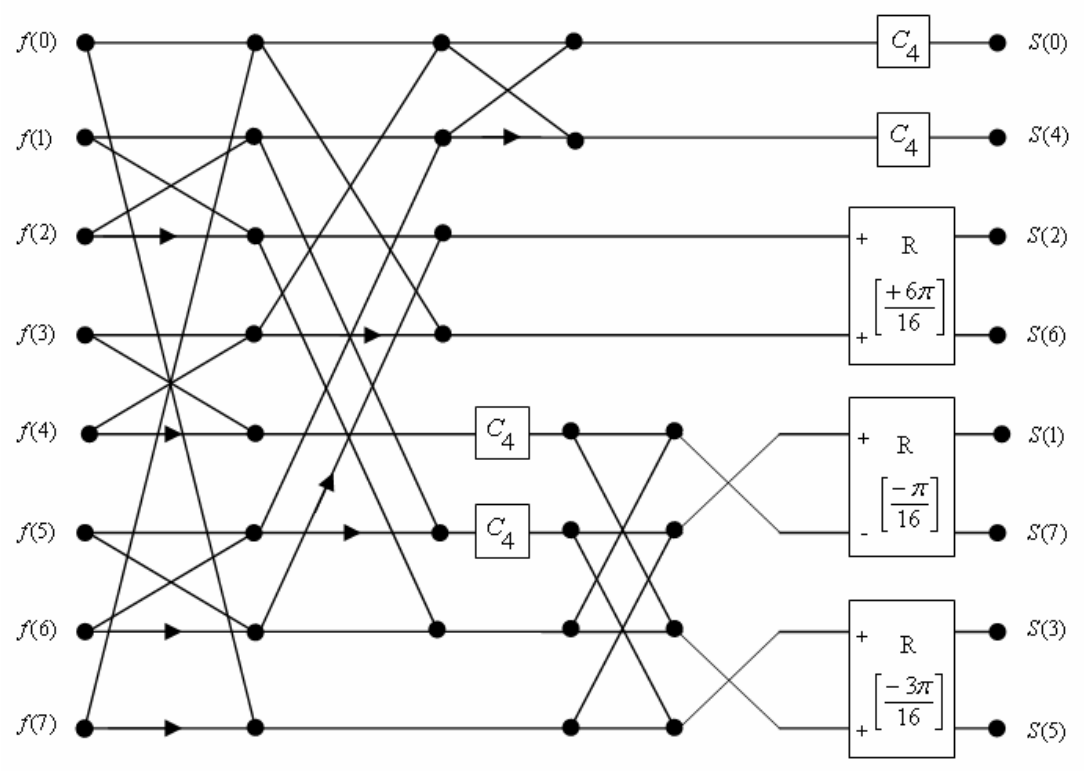
DCT [21] เป็นการแปลงสัญญาณแบบไม่ต่อเนื่องที่มีขอบเขตจำกัด เช่นเดียวกับ DFT เพียงแต่ใช้เฉพาะรูปคลื่น cosine เป็นฟังก์ชันฐาน (cosine basis function) ผลของการแปลงจึงมีเฉพาะสัมประสิทธิ์ที่เป็นค่าจริง ซึ่ง DCT มีคุณสมบัติ Energy compaction ที่ดี คือ สามารถรวมพลังงานส่วนใหญ่ของสัญญาณไปไว้ในสัมประสิทธิ์ย่านความถี่ต่ำในโดเมนของการแปลง โดยสัมประสิทธิ์ที่มีค่าใกล้ศูนย์สามารถประมาณให้เป็นศูนย์ได้ เมื่อทำการแปลงกลับ (Inverse DCT) จะได้สัญญาณที่มีความใกล้เคียงกับสัญญาณเดิม จึงเป็นคุณสมบัติที่มีประโยชน์มากในการลดจำนวนข้อมูล สมการพื้นฐานของ DCT 1 มิติ (1-Dimensional) คือ

$$C(u)=\alpha(u)\sum_{x=0}^{N-1} f(x)\cos\left[\frac{\pi(2x+1)u}{2N}\right],\ 0\le u<N,\ \alpha(u)=\begin{cases}\dfrac{1}{\sqrt{N}}, & u=0\\ \sqrt{\dfrac{2}{N}}, & u\neq 0\end{cases} \tag{2.19}$$

การคำนวณ DCT อย่างเร็ว คือ การปรับเปลี่ยนขั้นตอนการคำนวณ (Algorithms) เพื่อทำให้ได้ผลการแปลงในเวลาที่เร็วขึ้น [22] ซึ่งมีอยู่หลายวิธี ในงานวิจัยนี้ใช้แผนภาพการไหลของสัญญาณ (Flow-graph) ของ Vetterli และ Ligtenberg [23] ในการคำนวณ Scaled DCT 8 จุด โดยมีสมการดังนี้

$$S(u)=\alpha(u)\sum_{x=0}^{7} f(x)\cos\left[\frac{\pi(2x+1)u}{16}\right],\ \alpha(u)=\begin{cases}\dfrac{1}{\sqrt{2}} & ,u=0\\ 1 & ,u>0\end{cases} \tag{2.20}$$

โดยที่ $S(u)$ จะมีค่าเป็น 2 เท่าของ $C(u)$ เนื่องจากเป็นผลการคำนวณของ Scaled DCT 8 จุด



รูปที่ 2.11 แผนภาพการไหลของสัญญาณของ Vetterli และ Ligtenberg

จุดในแผนภาพหมายถึงการรวมกันของสัญญาณ ถ้ามีลูกศรที่เส้นหมายถึงสัญญาณถูกกลับเครื่องหมาย (คูณด้วย -1) และการคูณกับค่าคงที่จะแทนด้วยรูปสี่เหลี่ยม โดย R (Rotation) คือ ผลบวกและผลต่างของ 2 จำนวน (x และ y) ที่คูณกับค่าคงที่ C_k และ S_k ดังสมการ

$$\begin{aligned} X &= C_k x + S_k y \\ Y &= -S_k x + C_k y \end{aligned} \quad , \quad C_k = \cos\left(\frac{k\pi}{16}\right), \; S_k = \sin\left(\frac{k\pi}{16}\right) \tag{2.21}$$

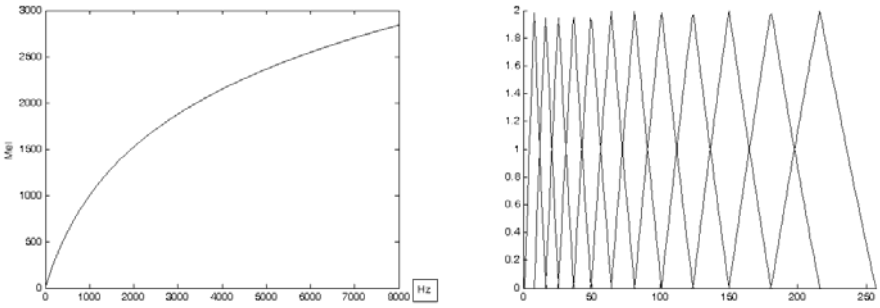
จากแผนภาพการไหลของสัญญาณ เมื่อให้ $s_{xy} = f(x) + f(y)$ และ $d_{xy} = f(x) - f(y)$ จะสามารถเขียนเป็นสมการได้ดังนี้

$$\left.\begin{aligned} S(0) &= C_4\{(s_{07} + s_{34}) + (s_{56} + s_{12})\} \\ S(4) &= C_4\{(s_{07} + s_{34}) - (s_{56} + s_{12})\} \\ \\ S(2) &= C_6(d_{12} - d_{56}) + S_6(s_{07} - s_{34}) \\ S(6) &= -S_6(d_{12} - d_{56}) + C_6(s_{07} - s_{34}) \\ \\ S(3) &= C_3\{d_{07} - C_4(s_{12} - s_{56})\} - S_3\{d_{34} - C_4(d_{12} + d_{56})\} \\ S(5) &= S_3\{d_{07} - C_4(s_{12} - s_{56})\} + C_3\{d_{34} - C_4(d_{12} + d_{56})\} \\ \\ S(1) &= C_1\{d_{07} + C_4(s_{12} - s_{56})\} + S_1\{d_{34} + C_4(d_{12} + d_{56})\} \\ S(7) &= S_1\{d_{07} + C_4(s_{12} - s_{56})\} - C_1\{d_{34} + C_4(d_{12} + d_{56})\} \end{aligned}\right\} \tag{2.22}$$

2.2.2.3 **ชุดตัวกรองความถี่แบบเมล** (Mel-filter bank)

เนื่องจากหูของมนุษย์จะแยกรายละเอียดของสัญญาณเสียงที่ความถี่ต่ำได้ดีกว่าความถี่สูง ดังนั้นจึงมีการออกแบบชุดตัวกรองที่ให้ความสำคัญกับสเปกตรัมย่านความถี่ต่ำ เรียกว่า Mel-filter bank ที่จะเน้นความถี่ในช่วงกลางของตัวกรอง โดยความถี่กลางของตัวกรองแต่ละช่องนั้นอยู่บนสเกลความถี่แบบเมล (Mel-scale) [7],[8] ซึ่งเป็นมาตราส่วนความถี่แบบไม่สม่ำเสมอ ดังสมการ

$$Mel(f) = 2595 \times \log_{10}(1 + \frac{f}{700}) \tag{2.23}$$



รูปที่ 2.12 (ก) สเกลความถี่แบบเมล (ข) ชุดตัวกรองความถี่แบบสามเหลี่ยม [24]

ค่าสเปกตรัมกำลังของสัญญาณในแต่ละเฟรมจะถูกคูณกับตัวกรองแต่ละช่อง ตามสมการ

$$\mathrm{M}_j(k) = \sum_{i=0}^{N/2} P_j(i)\Theta_k(i) \tag{2.24}$$

โดย $P_j(i)$ คือ ค่า Power spectrum ตัวที่ i ในเฟรมที่ j

$\Theta_k(i)$ คือ ค่าสัมประสิทธิ์ของตัวกรอง (ความสูงของตัวกรองช่องที่ k ตำแหน่งที่ i)

$\mathrm{M}_j(k)$ คือ ค่า Mel-power spectrum ตัวที่ k ในเฟรมที่ j

2.2.2.4 **รูปร่างของสเปกตรัม** (Spectral envelope)

จากแนวคิดพื้นฐานที่ว่าสัญญาณเสียงเกิดจากการคูณประสาน (Convolution) ในโดเมนเวลาของสัญญาณกระตุ้น (Excitation signal) กับผลตอบสนองของช่องทางเดินเสียง (Vocal tract impulse response) [18] หลังจากการแปลงฟูเรียร์จะได้ผลคูณ (Product) ของ 2 สัญญาณ (Harmonics และ Resonances)

$$x[n] = h[n] \otimes g[n] \tag{2.25}$$

หรือ

$$X[k] = H[k] \cdot G[k] \tag{2.26}$$

ดังนั้นสเปกตรัมของสัญญาณเสียงจึงประกอบด้วย 2 ส่วนคือ ส่วนรูปร่าง (Spectral envelope) และส่วนโครงสร้างรายละเอียด (Spectral fine structure) ทั้งสองส่วนนี้สามารถแยกกันได้โดยการใส่ลอการิทึม (log) ดังสมการ

$$\log(X[k]) = \log(H[k]) + \log(G[k]) \tag{2.27}$$

เคปสตรัม (Cepstrum) [25] คือ การแปลงกลับฟูเรียร์ (IFFT) ของค่า log ของ Magnitude ของ spectrum ได้เป็นสัมประสิทธิ์ในโดเมนควิเฟรนซี (Quefrency domain) โดยที่สัมประสิทธิ์ในช่วง 5 mSec แรก จะแทนเฉพาะรูปร่างของสเปกตรัม [26] ที่แสดงความถี่ฟอร์แมนท์ (Formant frequency) และคาบเวลาพิทช์ (Pitch periods) ซึ่งเป็นส่วนประกอบสำคัญของเสียง [2]

เคปสตรัมบนความถี่เมล (Mel-frequency cepstrum) คือ เคปสตรัมที่ถูกปรับให้เหมาะสมกับการได้ยินของมนุษย์ โดยการแปลงโคซายน์ (DCT) ของค่า log ของ Mel-power spectrum เพื่อแทนค่าพลังงานของสเปกตรัมกำลัง โดยที่ค่าสัมประสิทธิ์ในช่วงแรก (low quefrency) จะใช้แทนลักษณะสำคัญของเสียง ซึ่งเป็นค่าสัมประสิทธิ์ที่นิยมใช้ในการรู้จำเสียงพูด

2.2.2.5 ค่าความแตกต่างระหว่างสัมประสิทธิ์ในแต่ละเฟรมกับค่าเฉลี่ยของสัมประสิทธิ์ในช่วงเงียบ

หลังจากผ่านการหาค่าลักษณะสำคัญในแต่ละเฟรมแล้ว สมมติว่าสัญญาณในช่วง 3 เฟรมแรกและ 3 เฟรมสุดท้าย เป็นสัญญาณในช่วงเงียบ จึงกำหนดให้

$$C_s(i) = [C_1(i) \quad C_2(i) \quad C_3(i) \quad C_{end\text{-}2}(i) \quad C_{end\text{-}1}(i) \quad C_{end}(i)]$$

โดย s คือ เฟรมที่เป็นช่วงเงียบ และ end คือ เฟรมสุดท้าย

$C_s(i)$ คือ ค่า MFCC ตัวที่ i ในเฟรมที่เป็นช่วงเงียบ

ดังนั้นค่าเฉลี่ยของสัมประสิทธิ์ในช่วงเงียบ (Background characteristics) หาได้จากสมการ

$$C_{mean}(i) = \frac{1}{F}\sum C_s(i) \quad , \quad 1 \le i \le I \tag{2.28}$$

โดยที่ I คือ จำนวนสัมประสิทธิ์ MFCC ทั้งหมดใน 1 เฟรม

F คือ จำนวนเฟรมในช่วงเงียบ (กำหนดให้ $F = 6$)

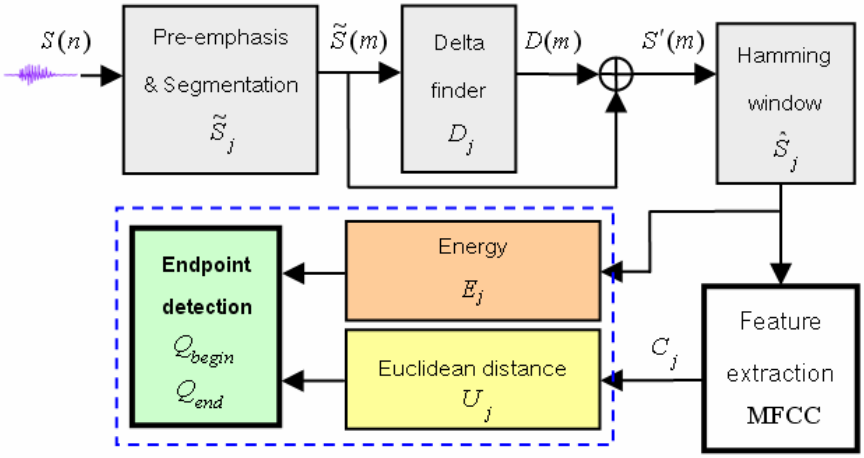
ดังนั้น ค่าระยะห่างแบบยูคลีเดียน (Euclidean distance) ระหว่างสัมประสิทธิ์ในแต่ละเฟรมกับค่าเฉลี่ยของสัมประสิทธิ์ในช่วงเงียบ [11] หาได้จากสมการ

$$U_j = \sum_{i=1}^{I}\left(C_j(i) - C_{mean}(i)\right)^2 \tag{2.29}$$

ค่าระยะห่างแบบยูคลีเดียนแสดงถึงความแตกต่างระหว่างสัญญาณในแต่ละเฟรมกับสัญญาณอ้างอิง (ค่าเฉลี่ยของสัมประสิทธิ์ในช่วงเงียบ) ในโดเมนควิเฟรนซี ซึ่งจะแยกระหว่างสัญญาณเสียงช่วงที่มีพลังงานต่ำ เช่น เสียงพ่นลมในพยัญชนะ ฟ ส ซ (Fricative) กับสัญญาณในช่วงเงียบ (Background noise) ได้เด่นชัดยิ่งขึ้น

2.2.2.6 การตรวจหาขอบเขตของคำ (Endpoint Detection)

สัญญาณในช่วงที่ยังไม่มีการเปล่งเสียงหรือสภาวะเงียบ (Silence) จะมีค่าพลังงานต่ำและค่อนข้างเรียบถ้าไม่มีสัญญาณรบกวนจากภายนอก ส่วนในช่วงเสียงพูดแบบโฆษะ (Voiced) สัญญาณจะมีค่าพลังงานสูง ในงานวิจัยนี้จึงนำค่าพลังงานของเสียง (**E**) พร้อมด้วยค่าความแตกต่างระหว่างสัมประสิทธิ์ในแต่ละเฟรมกับค่าเฉลี่ยของสัมประสิทธิ์ในช่วงเงียบ (**U**) มาใช้ในการตรวจหาขอบเขตของคำ



รูปที่ 2.13 การหาค่า **E** และ **U** ที่ใช้ในการตรวจหาขอบเขตของคำ

เริ่มจากการหาค่าเฉลี่ยของพลังงานเสียงในช่วงเงียบเพื่อใช้เป็นระดับอ้างอิง โดยกำหนดให้

$$E_s = [E_1 \quad E_2 \quad E_3 \quad E_{end\text{-}2} \quad E_{end\text{-}1} \quad E_{end}]$$

ดังนั้นค่าเฉลี่ยของ **E** ในช่วงเงียบ คือ

$$E_{sil} = \left(\frac{1}{F}\sum E_s\right) + t \qquad (2.30)$$

โดย t คือ ค่าที่ใช้ปรับระดับอ้างอิง (threshold adjustment)

และกำหนดให้

$$U_s = [U_1 \quad U_2 \quad U_3 \quad U_{end\text{-}2} \quad U_{end\text{-}1} \quad U_{end}]$$

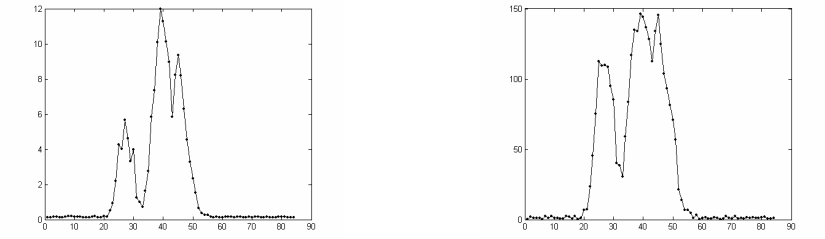
ดังนั้นค่าเฉลี่ยของ **U** ในช่วงเงียบ คือ

$$U_{sil} = \frac{1}{F}\sum U_s \tag{2.31}$$

จากนั้นเริ่มการตรวจหาขอบเขตของคำโดยให้สัญญาณในเฟรมที่ j เป็นเสียงพูด ถ้าเฟรมนั้นมีค่า E_j และ U_j ตรงตามเงื่อนไขข้อใดข้อหนึ่ง ต่อไปนี้

- $U_j > \alpha * U_{sil}$ & $E_j > \beta * E_{sil}$
- $U_j > U_{sil}$ & $E_j > \eta * E_{sil}$
- $E_j > \gamma * E_{sil}$

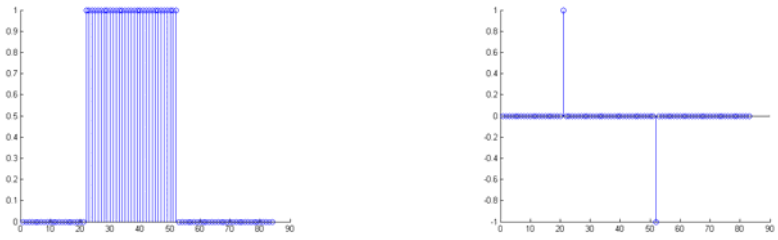
โดยที่ $\alpha = 3;\ \beta = 0.75;\ \eta = 1.1;\ \gamma = 1.3$ เป็นค่าจากการทดลอง [11]



รูปที่ 2.14 (ก) ค่าพลังงานของเสียง (**E**) (ข) ค่าระยะห่างของสัมประสิทธิ์ (**U**)

กำหนดให้ค่า $P = 1$ เมื่อเฟรมที่ j เป็นเสียงพูด และให้ค่า $P = 0$ เมื่อเฟรมที่ j ไม่ใช่เสียงพูด จากนั้นระบุคู่ตำแหน่งเริ่มต้นและสิ้นสุดของคำได้จากสมการ

$$Q_j = P_{j+1} - P_j \tag{2.32}$$



รูปที่ 2.15 (ก) เฟรมที่เป็นเสียงพูด (**P**) (ข) คู่ตำแหน่งของเสียงพูด (**Q**)

โดยตำแหน่งที่ $Q = 1$ คือ ตำแหน่งเฟรมเริ่มต้นของคำ (Q_{begin}) และตำแหน่งที่ $Q = -1$ คือ ตำแหน่งเฟรมสิ้นสุดของคำ (Q_{end}) ในคำที่มากกว่า 1 พยางค์อาจจะมีคู่ตำแหน่งซึ่งระบุตำแหน่งเฟรมเริ่มต้นและเฟรมสิ้นสุดมากกว่า 1 คู่ก็ได้ เนื่องจากเป็นระบบรู้จำเสียงคำโดด ถ้ามีคู่ตำแหน่งสองคู่

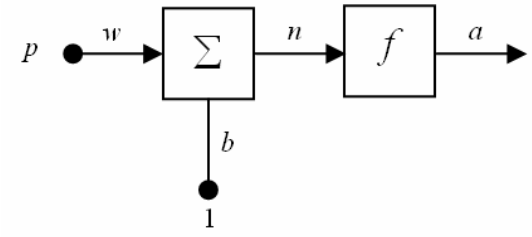
ที่ห่างกันเพียง 1 เฟรม (มีช่วงเงียบระหว่างพยางค์ 1 เฟรม) จะรวมให้เป็นคู่ตำแหน่งเดียวกันโดยถือว่าเฟรมนั้นเป็นเสียงพูด เมื่อได้คู่ตำแหน่งของเสียงพูดแล้ว จะสามารถระบุขอบเขตของ MFCC ในช่วงที่เป็นเสียงพูดได้

2.2.3 **การรู้จำรูปแบบ** (Pattern Recognition)

เมื่อได้ลักษณะสำคัญของเสียงแล้วจะนำค่าที่ได้มาคำนวณเทียบเคียงกับข้อมูลอ้างอิงเพื่อหาคำตอบว่าค่าลักษณะสำคัญนั้นตรงหรือคล้ายคลึงกับเสียงพูดคำใด ซึ่งขั้นตอนในการฝึกฝนเพื่อให้ได้ข้อมูลอ้างอิงนั้นขึ้นอยู่กับวิธีในการรู้จำของระบบ เช่น วิธี Dynamic Time Warping: DTW [1],[5] เพียงแค่เก็บข้อมูลตัวอย่างไว้เปรียบเทียบกับข้อมูลชุดทดสอบเท่านั้น ในขณะที่วิธี Artificial Neural Networks: ANN จะนำข้อมูลชุดฝึกฝนไปแปลงเป็นค่าอ้างอิงที่ต้องการ โดยนำข้อมูลไปผ่านโครงข่ายที่สร้างขึ้นเพื่อจดจำรูปแบบและเก็บเป็นค่าถ่วงน้ำหนัก หรือวิธี Hidden Markov Model: HMM [1],[7] จะนำข้อมูลชุดฝึกฝนไปผ่านแบบจำลองที่สร้างขึ้นเพื่อจดจำรูปแบบโดยเก็บค่าทางสถิติและค่าความน่าจะเป็นของแต่ละสถานะไว้ ในงานวิจัยนี้เลือกใช้วิธีการของโครงข่ายประสาทเทียม (Artificial Neural Networks) ในการจดจำรูปแบบของเสียงพูด

2.2.3.1 **โครงข่ายประสาทเทียม** (Artificial Neural Networks)

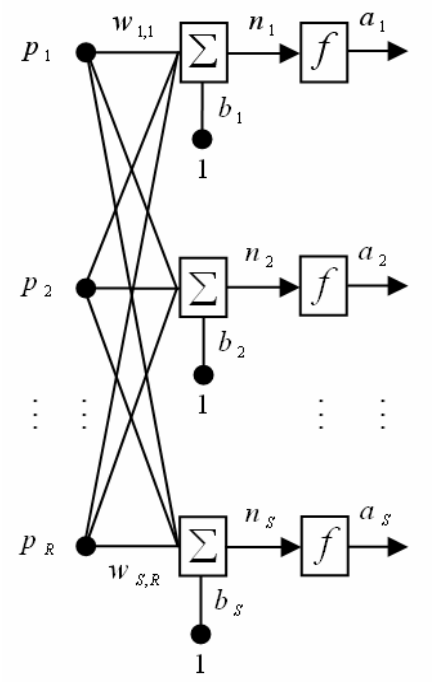
โครงข่ายประสาทเทียม [1]-[6],[8] คือ ระบบคอมพิวเตอร์ที่มีโครงสร้างและการทำงานเลียนแบบระบบประสาทของมนุษย์ ซึ่งแบบจำลองเซลล์ประสาทในรูปที่ 2.16 ประกอบด้วยค่าอินพุท (input) p คูณกับค่าน้ำหนัก (weight) w แล้วบวกกับค่าถ่วง (bias) b จากนั้นส่งผ่านฟังก์ชันกระตุ้น (activation function) f เพื่อจำกัดขอบเขตค่าเอาท์พุต (output) a ให้อยู่ในช่วงที่ต้องการ



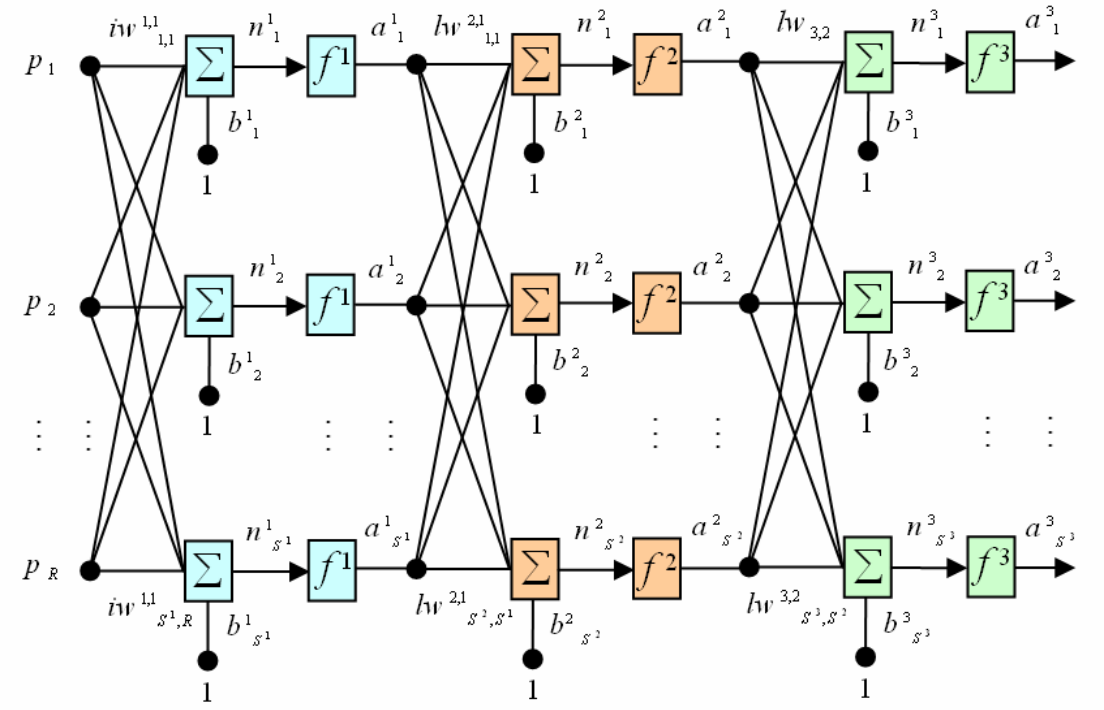
$$a = f(wp + b)$$

รูปที่ 2.16 แบบจำลองเซลล์ประสาท

เมื่อมีเซลล์ประสาทมากกว่า 1 โหนด (node) เชื่อมต่อกันจะเรียกว่าชั้น (layer) ของเซลล์ประสาท ดังแสดงในรูปที่ 2.17 ซึ่งภายในโครงข่ายสามารถมีจำนวนชั้นได้มากกว่า 1 ชั้น (Multiple layers of neurons) โดยเซลล์ประสาทที่เรียกว่า เพอเซปตรอน (Perceptron) มีความสามารถในการแยกประเภทของข้อมูลที่ป้อนเข้ามาโดยใช้ค่าถ่วงน้ำหนักที่ได้จากการเรียนรู้แบบมีการสอน (Supervised learning) คือ มีการกำหนดชุดตัวอย่างที่ใช้ในการเรียนรู้ให้



รูปที่ 2.17 เซลล์ประสาทที่เชื่อมต่อกันเป็นชั้น



$$\mathbf{a}^1 = \mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) \qquad \mathbf{a}^2 = \mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{a}^1 + \mathbf{b}^2) \quad \mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{a}^2 + \mathbf{b}^3)$$

$$\mathbf{a}^3 = \mathbf{f}^3(\mathbf{LW}^{3,2}\mathbf{f}^2(\mathbf{LW}^{2,1}\mathbf{f}^1(\mathbf{IW}^{1,1}\mathbf{p} + \mathbf{b}^1) + \mathbf{b}^2) + \mathbf{b}^3)$$

รูปที่ 2.18 เซลล์ประสาทที่เชื่อมต่อกันแบบหลายชั้น

2.2.3.2 การเรียนรู้ของโครงข่ายประสาทเทียม

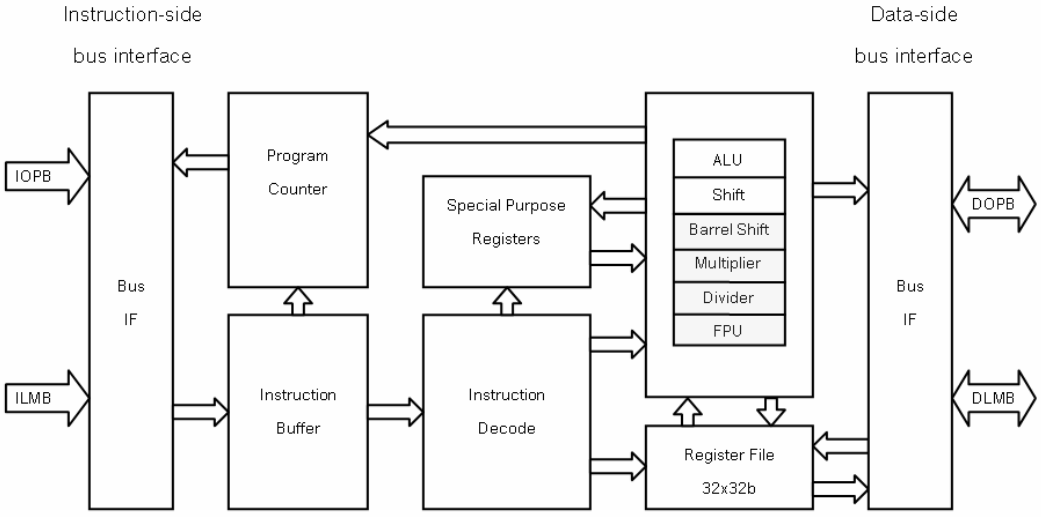
ในขั้นตอนการฝึก (Training) โครงข่ายประสาทเทียมแบบ Feed-forward ข้อมูลในชุดตัวอย่างจะถูกป้อนเข้าทางชั้นอินพุท (Input layer) และถูกส่งผ่านชั้นซ่อนตัว (Hidden layer) จากชั้นหนึ่งไปอีกชั้นหนึ่งโดยไม่มีการย้อนกลับ จนกระทั่งถึงชั้นเอาท์พุต (Output layer) จากนั้นหาค่าผิดพลาดระหว่างข้อมูลในชั้นเอาท์พุตกับค่าที่ต้องการ (Target) เพื่อใช้ในการปรับค่าถ่วงน้ำหนัก ค่าผิดพลาดจะถูกคำนวณจากชั้นเอาท์พุตย้อนกลับไปถึงชั้นอินพุท จึงเรียกขั้นตอนนี้ว่า การแพร่ย้อนกลับ (Backpropagation) เมื่อเสร็จสิ้นการปรับค่าถ่วงน้ำหนักก็จะกลับสู่โครงข่ายที่ป้อนข้อมูลไปข้างหน้าอีกครั้ง เพื่อหาค่าเอาท์พุตมาเปรียบเทียบกับค่าที่ต้องการ ทำเช่นนี้สลับกันไปจนกว่าค่าผิดพลาดที่ได้จะลดลงต่ำกว่าค่าที่กำหนดไว้

ในขั้นตอนการทดสอบ (Testing) เป็นการหาค่าเอาท์พุตโดยใช้ค่าถ่วงน้ำหนักที่ได้มาจากขั้นตอนการฝึก ซึ่งมีลักษณะการคำนวณเช่นเดียวกับการฝึกโครงข่ายในส่วนที่ป้อนข้อมูลไปข้างหน้า แต่จะใช้เวลาน้อยกว่ามากเพราะเป็นการส่งข้อมูลไปข้างหน้าเพียงรอบเดียว

โดยข้อมูลที่ใช้จะเป็นข้อมูลในชุดทดสอบและเอาท์พุตที่ได้คือผลลัพธ์ของการทดสอบ ซึ่งผลลัพธ์ที่ได้สามารถนำไปตัดสินใจได้ว่าข้อมูลที่เข้ามานั้นตรงกับเสียงพูดคำใด

2.3 MicroBlaze

MicroBlaze เป็นไมโครโปรเซสเซอร์ที่สร้างขึ้นจาก Configurable Logic Blocks (CLBs) ภายใน FPGA [27] สิ่งที่น่าสนใจ คือ การเป็นตัวประมวลผลที่มีความยืดหยุ่นในการใช้งาน (Soft-core microprocessor) เนื่องจากสามารถสร้าง MicroBlaze ให้มีความสามารถในการทำงานเท่าที่จำเป็น (เช่น Barrel shifter, Multiplier, Divider, Floating-Point Unit และ อื่นๆ) ความสามารถที่ไม่ได้เลือกใช้งานจะไม่ถูกนำไปสร้าง จึงไม่ใช้พื้นที่ logic ของ FPGA นอกจากนั้นยังสามารถปรับเพิ่ม-ลดอุปกรณ์เชื่อมต่อ (Peripherals) ต่างๆ ได้ (เช่น UART, GPIO และอื่นๆ) ทำให้ผู้ใช้สามารถออกแบบระบบที่มีประสิทธิภาพการทำงานคุ้มค่ากับทรัพยากรที่เสียไป



รูปที่ 2.19 โครงสร้างพื้นฐานของ MicroBlaze

MicroBlaze [28] เป็นสถาปัตยกรรมแบบ 32-bit Reduced Instruction Set Computer (RISC) ที่ถูกพัฒนาเพื่อนำมาใช้ใน FPGA ของบริษัท Xilinx ซึ่งมีการแยก Local Memory Bus (LMB) ที่ใช้ติดต่อกับหน่วยความจำภายใน FPGA (Block RAM แบบ Dual port) ออกเป็นส่วนของชุดคำสั่ง (Instruction-side) และส่วนของข้อมูล (Data-side) ทำให้มีความเร็วเต็มที่ในการกระทำคำสั่ง (execute) และการเข้าถึงข้อมูล (access) นอกจากนั้นยังมี On-chip Peripheral Bus (OPB) ที่ใช้ติดต่อกับอุปกรณ์รอบข้างรวมถึงหน่วยความจำภายนอกด้วย สำหรับ

ประเภทข้อมูลที่สนับสนุน คือ Word (32 บิต), Half word (16 บิต) และ Byte (8 บิต) โดยใช้รูปแบบ Big-Endian ในการแทนบิตของข้อมูล (ลำดับของบิต คือ Bit 0 Bit 1 ..... Bit 30 Bit 31 โดยที่ Bit 0 คือ MSB และ Bit 31 คือ LSB)

EDK (Embedded Development Kit) เป็นเครื่องมือที่ช่วยสร้าง MicroBlaze โดยจำเป็นต้องมี ISE (Integrated Software Environment) ซึ่งเป็นซอฟท์แวร์ที่ใช้พัฒนาการออกแบบอุปกรณ์ของ Xilinx ติดตั้งอยู่ด้วย สำหรับขั้นตอนการออกแบบระบบจะใช้ซอฟท์แวร์ XPS (Xilinx Platform Studio) ในการสร้างและปรับแต่งส่วนของ Hardware (Processor core, Memory-controller, I/O peripherals) นอกจากนั้นยังสามารถสร้างอุปกรณ์ที่นอกเหนือไปจากรายการที่มีอยู่ (Custom peripherals [29]) เพื่อใช้ทำงานที่แตกต่างออกไปตามความต้องการ ในส่วนการทำงานของ MicroBlaze จะเขียน Software ควบคุมโดยใช้ภาษา C ซึ่งใน EDK มีตัวแปลภาษา (C compilers) เพื่อสร้างรหัสคำสั่ง (Machine code) ให้กับ MicroBlaze