

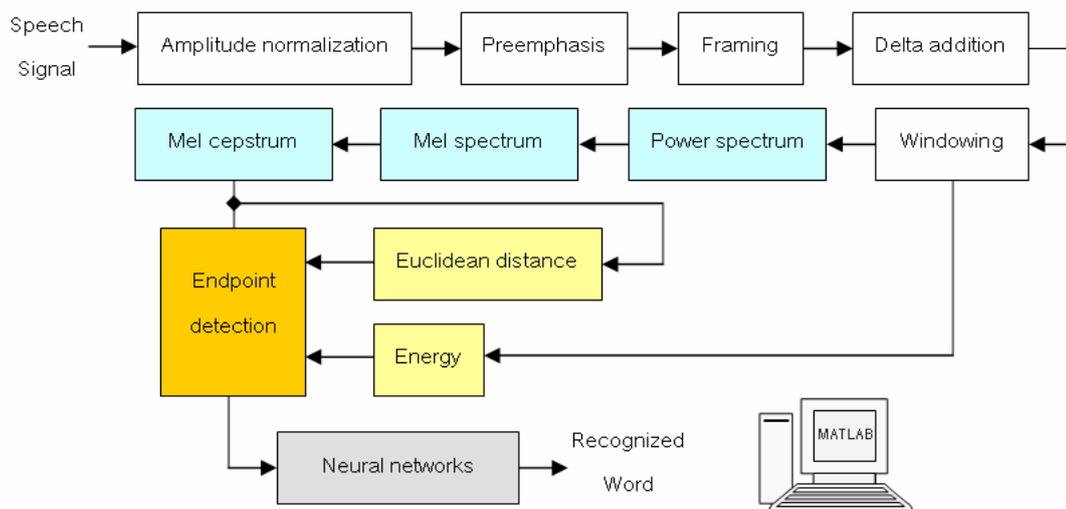
บทที่ 3

ขั้นตอนวิธีในการดำเนินงานวิจัย

หลังจากศึกษาเกี่ยวกับระบบรู้จำเสียงพูดคำโดดในทุกขั้นตอนแล้ว การดำเนินงานวิจัยในขั้นตอนต่อมาคือการออกแบบระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์ส่วนบุคคลและการออกแบบระบบบนอุปกรณ์ FPGA จากนั้นจึงสร้างระบบรู้จำเสียงพูดที่สามารถทำงานได้บนอุปกรณ์ FPGA

3.1 การออกแบบระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์ส่วนบุคคล

ในขั้นตอนแรก ทำการบันทึกเสียงโดยใช้โปรแกรม MATLAB สัญญาณเสียงที่ได้เป็นข้อมูลขนาด 16 บิต และมีอัตราการสุ่มตัวอย่าง (Sampling rate) 11.025 กิโลเฮิร์ต (kHz) ใช้ระยะเวลาในการบันทึก 2 วินาที (22050 ตัวอย่าง) โดยทำการบันทึกเสียงพูดคำโดดภาษาไทย จำนวน 7 คำคือ “เลี้ยวซ้าย” “เลี้ยวขวา” “เดินหน้า” “ถอยหลัง” “เร็วขึ้น” “ช้าลง” และ “หยุด” คำละ 20 ครั้ง แบ่งออกเป็น 2 กลุ่ม กลุ่มที่หนึ่ง (กลุ่มตัวอย่าง) ใช้ในการฝึกโครงข่ายประสาทเทียม และกลุ่มที่สอง (กลุ่มทดสอบ) ใช้ในการทดสอบระบบเบื้องต้น ขั้นตอนต่อมาคือการออกแบบและพัฒนาระบบรู้จำเสียงพูดคำโดดโดยใช้โปรแกรม MATLAB ซึ่งมีกระบวนการทำงานดังรูป



รูปที่ 3.1 การทำงานของระบบรู้จำเสียงพูดคำโดด

สัญญาณเสียง (Speech signal) ที่เข้ามาในระบบจะถูกปรับให้เป็นระดับปกติ (Amplitude normalization) โดยการทำให้ค่าสูงสุดของสัญญาณมีค่าเท่ากับ 1 ดังสมการ

$$S(n) = \frac{\text{Speech}(n)}{\max(\text{Speech})} \quad (3.1)$$

สัญญาณที่ปรับระดับแล้วจะถูกเน้นล่วงหน้า (Preemphasis) และแบ่งออกเป็นเฟรม (Framing) เพื่อหาค่าความแตกต่างของเฟรมรอบข้าง (Delta) เพิ่มให้กับสัญญาณ ก่อนทำการวางกรอบหน้าต่าง (Windowing) ให้กับสัญญาณในแต่ละเฟรม

ขั้นตอนต่อไปคือการหาค่าพลังงานเสียง (Energy) และค่าเคลปสตรัมบนความถี่เมล (Mel-frequency cepstrum) ของสัญญาณในแต่ละเฟรม เมื่อได้ค่าสัมประสิทธิ์เคลปสตรัมครบทุกเฟรมแล้วจึงหาค่าระยะห่างแบบยูคลิเดียน (Euclidean distance) ระหว่างค่าสัมประสิทธิ์ในแต่ละเฟรมกับค่าเฉลี่ยของสัมประสิทธิ์ในช่วงเงียบ จากนั้นใช้ค่าพลังงานเสียงพร้อมกับค่าระยะห่างของสัมประสิทธิ์ตรวจสอบขอบเขตของคำ (Endpoint detection) เมื่อได้คู่ตำแหน่งที่ระบุตำแหน่งเฟรมเริ่มต้นและเฟรมสิ้นสุดของคำแล้ว จะสามารถหาตำแหน่งของสัมประสิทธิ์ช่วงที่เป็นเสียงพูดได้

เนื่องจากข้อมูลในชั้นอินพุท (Input layer) ของโครงข่ายประสาทเทียมจะต้องมีจำนวนข้อมูลเท่ากันทุกรูปแบบ แต่ค่าสัมประสิทธิ์เคลปสตรัมบนความถี่เมล (MFCC) ที่ได้จากการหาขอบเขตของคำมีจำนวนไม่เท่ากันขึ้นอยู่กับการออกเสียงสั้น-ยาวของผู้พูด ดังนั้นจึงต้องมีวิธีการปรับจำนวนของ MFCC ให้เท่ากันก่อน โดยในระบบนี้ใช้วิธีแบ่งค่าลักษณะสำคัญของเสียงออกเป็น 2 ส่วน ส่วนละ N ค่า คือ ค่าลักษณะสำคัญช่วงต้น (M_f) และค่าลักษณะสำคัญช่วงท้าย (M_b) เพื่อป้อนให้โครงข่ายประสาทเทียม 2 โครงข่าย โดยกำหนดให้

$$\text{half} = P/2, \quad \text{overlap} = \text{half}/2, \quad \text{len} = \text{half} + \text{overlap}$$

โดย P คือ จำนวน MFCC ที่ได้จากการหาขอบเขตของคำ

- ถ้า $\text{len} > N$

$$\begin{aligned} M_f(n) &= \text{MFCC}(n) \quad , \quad 1 \leq n \leq N \\ M_b(n) &= \text{MFCC}(m) \quad , \quad 1 \leq n \leq N \quad , \quad (\text{end} - N + 1) \leq m \leq \text{end} \end{aligned} \quad (3.2)$$

- ถ้า $\text{len} < N$

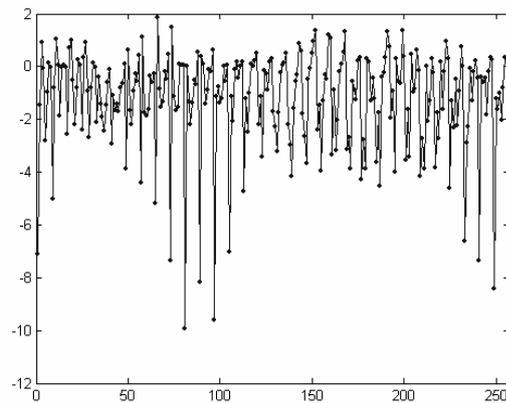
$$\begin{aligned} M_f(n) &= \begin{cases} \text{MFCC}(n) & , \quad 1 \leq n \leq \text{len} \\ 0 & , \quad (\text{len} + 1) \leq n \leq N \end{cases} \\ M_b(n) &= \begin{cases} 0 & , \quad 1 \leq n \leq (N - \text{len}) \\ \text{MFCC}(m) & , \quad (N - \text{len} + 1) \leq n \leq N \quad , \quad (\text{end} - \text{len} + 1) \leq m \leq \text{end} \end{cases} \end{aligned} \quad (3.3)$$

โดยที่ end คือ MFCC ตัวสุดท้าย

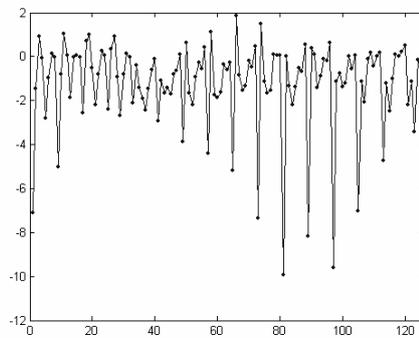
ตัวอย่างที่ 1 จำนวน MFCC ที่ได้จากการหาขอบเขตของค่าเท่ากับ 256 ค่า, $half = 256/2 = 128$ และ $overlap = 128/2 = 64$ จะได้ $len = 128+64 = 192$ กำหนดให้ $N = 125$ เนื่องจาก $len > 125$ ดังนั้น

M_f ตัวที่ 1 ถึงตัวที่ 125 คือ MFCC ตัวที่ 1 ถึงตัวที่ 125

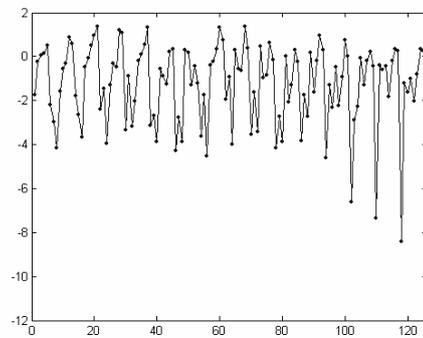
M_b ตัวที่ 1 ถึงตัวที่ 125 คือ MFCC ตัวที่ 132 (ได้จาก $256-125+1$) ถึงตัวที่ 256 (ตัวสุดท้าย)



(ก)



(ข)



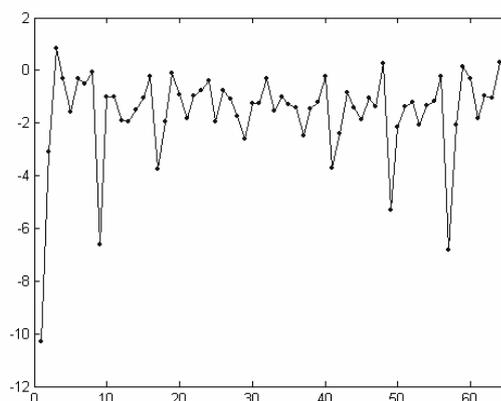
(ค)

รูปที่ 3.2 (ก) ค่าลักษณะสำคัญ (MFCC) ที่ได้จากการหาขอบเขตของค่า จำนวน 256 ค่า

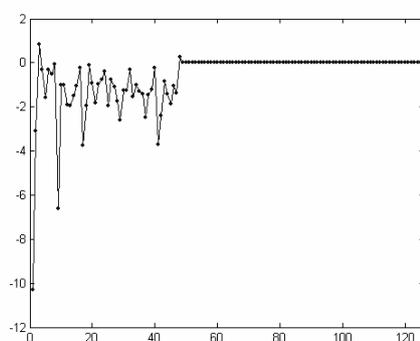
(ข) ค่าลักษณะสำคัญช่วงต้น (M_f) (ค) ค่าลักษณะสำคัญช่วงท้าย (M_b)

จะสังเกตได้ว่าถ้าจำนวน MFCC ที่ได้จากการหาขอบเขตของค่ามากกว่า $2N$ ค่า จะทำให้ MFCC ในช่วงกลางถูกตัดทิ้งไป

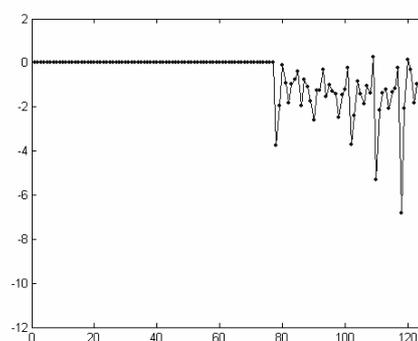
ตัวอย่างที่ 2 จำนวน MFCC ที่ได้จากการหาขอบเขตของค่าเท่ากับ 64 ค่า, $half = 64/2 = 32$ และ $overlap = 32/2 = 16$ จะได้ $len = 32+16 = 48$ กำหนดให้ $N = 125$ เนื่องจาก $len < 125$ ดังนั้น M_f ตัวที่ 1 ถึงตัวที่ 48 คือ MFCC ตัวที่ 1 ถึงตัวที่ 48 M_b ตัวที่ 78 ถึงตัวที่ 125 คือ MFCC ตัวที่ 17 (ได้จาก $64-48+1$) ถึงตัวที่ 64 (ตัวสุดท้าย) ในกรณีนี้ต้องเติม 0 (Zero padding) เพื่อให้มีจำนวน M_f และ M_b ครบ 125 ค่า



(ก)



(ข)

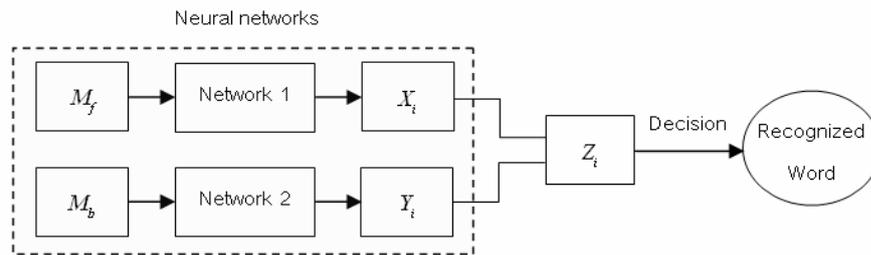


(ค)

รูปที่ 3.3 (ก) ค่าลักษณะสำคัญ (MFCC) ที่ได้จากการหาขอบเขตของค่า จำนวน 64 ค่า

(ข) ค่าลักษณะสำคัญช่วงต้น (M_f) (ค) ค่าลักษณะสำคัญช่วงท้าย (M_b)

จากนั้นจะทำการบันทึกค่า M_f และ M_b (ของสัญญาณเสียงในกลุ่มตัวอย่าง) ไว้เป็นฐานข้อมูลชุดฝึก (Database) สำหรับใช้ในขั้นตอนการฝึกโครงข่ายประสาทเทียมทั้ง 2 โครงข่าย ส่วนในขั้นตอนการทดสอบระบบเบื้องต้นจะใช้ค่า M_f และ M_b บ่อนให้ชั้นอินพุทของโครงข่าย โดยนำค่าถ่วงน้ำหนักที่ได้จากขั้นตอนการฝึกมาใช้คำนวณหาค่าผลลัพธ์ของทั้ง 2 โครงข่าย



รูปที่ 3.4 การหาค่าผลลัพธ์ของโครงข่ายประสาทเทียม

ถ้ากำหนดให้ผลลัพธ์ของโครงข่ายที่ 1 คือ X_i และผลลัพธ์ของโครงข่ายที่ 2 คือ Y_i แล้ว ผลลัพธ์ที่ได้จากโครงข่ายประสาทเทียม (Z_i) คือ

$$Z_i = X_i \times Y_i \quad , \quad 1 \leq i \leq 7 \quad (3.4)$$

โดยที่ i คือ จำนวนโหนดในชั้นเอาต์พุตของโครงข่ายเท่ากับจำนวนคำที่ต้องการรู้จำ จากนั้น ระบบจะทำการตัดสินใจ (Decision) ว่าสัญญาณเสียงที่รับเข้ามาตรงกับคำพูดคำใด โดยใช้วิธีการตรวจเลือกค่าสูงสุดของผลลัพธ์ที่ได้จากโครงข่ายประสาทเทียม โดยค่าสูงสุดนั้นจะต้องมากกว่าค่าที่กำหนดไว้จึงจัดเป็นคำพูดในกลุ่มคำที่ต้องการรู้จำ

3.2 การออกแบบระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA

หลังจากศึกษาการทำงานของระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์แล้ว ได้แบ่งการออกแบบระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA ออกเป็น 3 ระยะ คือ

ระยะที่ 1 เริ่มต้นจากการใช้โปรแกรม XPS (Xilinx Platform Studio) สร้างไมโครโปรเซสเซอร์แบบ soft-core ที่ชื่อว่า MicroBlaze ขึ้นภายในอุปกรณ์ FPGA จากนั้นทำการเขียนชุดคำสั่งควบคุมการทำงานของ MicroBlaze เริ่มตั้งแต่ขั้นตอนการปรับระดับสัญญาณให้เป็นปกติ (Amplitude normalization) จนถึงขั้นตอนการหาค่าผลลัพธ์ของโครงข่ายประสาทเทียม โดยสัญญาณเสียง (Speech signal) และค่าถ่วงน้ำหนัก (Weight & bias) ของโครงข่ายที่ใช้ในระบบได้มาจากโปรแกรม MATLAB บนเครื่องคอมพิวเตอร์

ระยะที่ 2 เป็นการออกแบบอุปกรณ์เชื่อมต่อกับระบบรู้จำเสียงพูดภายใน MicroBlaze ในส่วนอุปกรณ์แปลงสัญญาณอนาล็อกเป็นดิจิตอล (Analog to Digital Converter: ADC) โดยเสียงพูดจะถูก Microphone เปลี่ยนเป็นสัญญาณทางไฟฟ้าส่งไปที่วงจร Pre-mic ซึ่งทำหน้าที่ขยายสัญญาณก่อนที่จะส่งต่อไปเข้าตัวแปลงสัญญาณแบบ ADC หน้าทีของวงจร ADC คือ

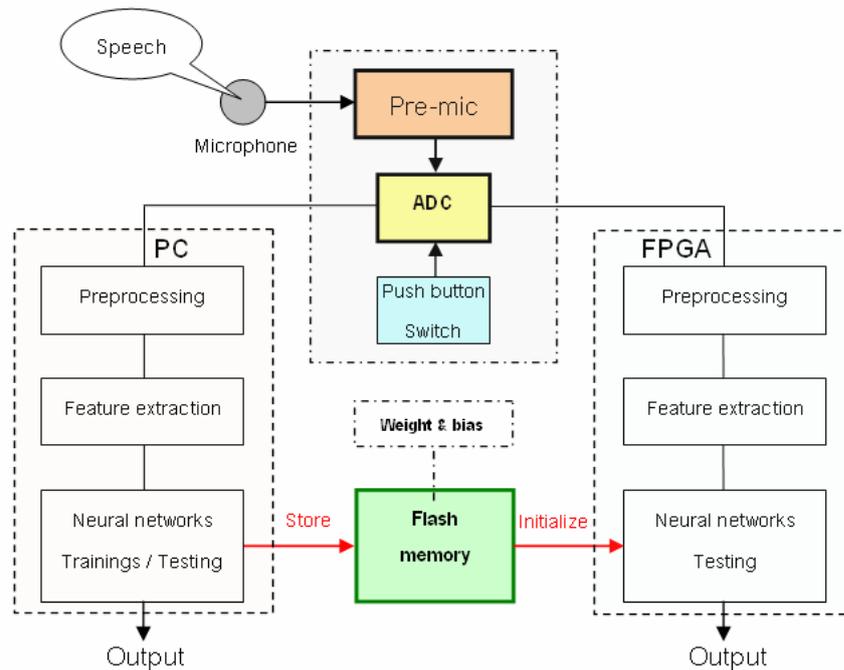
แปลงสัญญาณที่มีค่าต่อเนื่อง (Analog signal) ให้เป็นสัญญาณที่มีค่าไม่ต่อเนื่อง (Digital signal) โดยมีสวิตช์แบบกด (Push button Switch) เป็นตัวควบคุมการส่งสัญญาณเข้าไปในระบบ

ระยะที่ 3 เป็นการออกแบบอุปกรณ์เชื่อมต่อกับระบบในส่วนอุปกรณ์เก็บข้อมูล (Flash memory) ซึ่งเป็นหน่วยความจำขนาดเล็กที่สามารถบันทึกข้อมูลลงไปได้ โดยข้อมูลที่บันทึกไม่มีการสูญหายแม้จะไม่มีไฟเลี้ยง (non-volatile) และสามารถลบหรือแก้ไขข้อมูลภายในได้ ใช้สำหรับเก็บค่าถ่วงน้ำหนัก (Weight & bias) ของโครงข่ายประสาทเทียมหลังจากขั้นตอนการฝึก โดยสามารถเปลี่ยนค่าถ่วงน้ำหนักใหม่ได้ในกรณีที่มีการเปลี่ยนแปลงผู้พูดหรือกลุ่มคำในระบบ ซึ่งเมื่อระบบเริ่มทำงาน MicroBlaze จะนำข้อมูลภายในอุปกรณ์ตัวนี้ไปไว้ในหน่วยความจำ (RAM) ของระบบเป็นขั้นตอนแรก

3.2.1 โครงสร้างของระบบรู้จำเสียงพูดที่ทำงานบนอุปกรณ์ FPGA

ระบบรู้จำเสียงพูดที่สามารถทำงานบนอุปกรณ์ FPGA ได้นั้น จะต้องมี 2 ส่วนที่สัมพันธ์กัน คือ ค่าสัญญาณเสียงแบบดิจิตอลและค่าถ่วงน้ำหนักของโครงข่ายประสาทเทียม ซึ่งหาก 2 ส่วนนี้ไม่สัมพันธ์กันจะทำให้ผลการรู้จำผิดพลาด เนื่องจากในระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA จำเป็นต้องอาศัยค่าถ่วงน้ำหนักจากการฝึกโครงข่ายที่สร้างขึ้นด้วยโปรแกรม MATLAB ดังนั้นจึงต้องสร้างระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์ที่มีกระบวนการทำงานในทุกขั้นตอนตรงกับระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA และสัญญาณเสียงแบบดิจิตอลที่ส่งให้ระบบบนเครื่องคอมพิวเตอร์นั้น ต้องมาจากวงจรแปลงสัญญาณ ADC ชุดเดียวกับระบบบนอุปกรณ์ FPGA เพื่อให้ค่าถ่วงน้ำหนักที่ได้จากการฝึกโครงข่ายบนเครื่องคอมพิวเตอร์ มีความสัมพันธ์กับสัญญาณเสียงที่ส่งให้ระบบบนอุปกรณ์ FPGA

หลังจากสร้างระบบรู้จำเสียงพูดแล้ว ทำการบันทึกเสียงโดยใช้โปรแกรม MATLAB สัญญาณเสียงที่ได้เป็นข้อมูลขนาด 8 บิต และมีอัตราการสุ่มตัวอย่าง (sampling rate) 10 kHz ใช้ระยะเวลาในการบันทึก 2 วินาที (20000 ตัวอย่าง) โดยทำการบันทึกเสียงพูดคำโดดภาษาไทย จำนวน 7 คำคือ “เลี้ยวซ้าย” “เลี้ยวขวา” “เดินหน้า” “ถอยหลัง” “เร็วขึ้น” “ช้าลง” และ “หยุด” คำละ 20 ครั้ง สำหรับใช้ในการฝึกและตรวจสอบการทำงานของระบบ



รูปที่ 3.5 โครงสร้างของระบบรู้จำเสียงพูดคำโดดที่ทำงานบนอุปกรณ์ FPGA

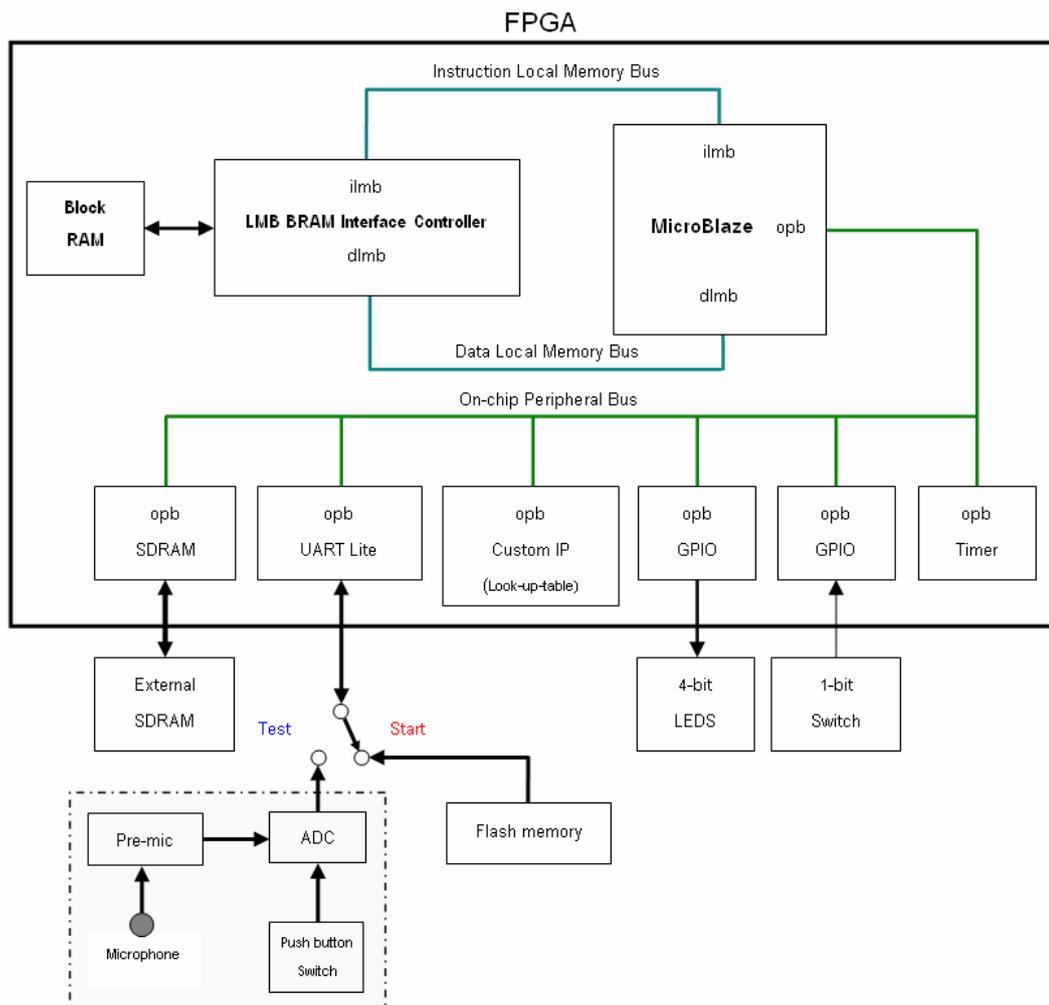
ในขั้นตอนการฝึกระบบเพื่อหาค่าถ่วงน้ำหนักของโครงข่ายประสาทเทียมจะต้องใช้ระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์ทำการหาค่าลักษณะสำคัญของเสียงพูดที่ปรับจำนวนให้เท่ากันแล้ว (M_f และ M_b) และบันทึกเอาไว้เป็นฐานข้อมูลชุดฝึก จากนั้นนำข้อมูลที่บันทึกไว้มาจัดเรียงเป็นชุดข้อมูลในชั้นอินพุตป้อนให้โครงข่ายประสาทเทียมทั้ง 2 โครงข่ายทำการเรียนรู้และปรับค่าถ่วงน้ำหนักจนกระทั่งค่าผิดพลาดน้อยกว่าค่าที่กำหนดไว้ จึงหยุดการฝึกและเก็บค่าถ่วงน้ำหนักทั้ง 2 ชุดไว้ภายในอุปกรณ์เก็บข้อมูล

ในการตรวจสอบการทำงานของระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA จะต้องมีการถ่ายโอนข้อมูล (ค่าถ่วงน้ำหนัก) จากอุปกรณ์เก็บข้อมูลเข้าไปในหน่วยความจำ (RAM) ของระบบเป็นขั้นตอนแรก จากนั้นส่งสัญญาณเสียง (ที่บันทึกไว้ในคอมพิวเตอร์) ให้ระบบบนอุปกรณ์ FPGA ทำการหาค่าลักษณะสำคัญของเสียงพูด (M_f และ M_b) ป้อนให้โครงข่ายประสาทเทียมทั้ง 2 โครงข่าย เพื่อคำนวณหาค่าผลลัพธ์ของแต่ละโครงข่ายโดยอาศัยค่าถ่วงน้ำหนักที่ได้จากขั้นตอนการฝึก เช่นเดียวกันหากส่งสัญญาณเสียงนั้นให้ระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์คำนวณหาค่าผลลัพธ์ของโครงข่ายจะได้ค่าผลลัพธ์ที่ใกล้เคียงกัน ซึ่งแสดงว่าระบบทำงานได้ถูกต้อง โดยที่ผลลัพธ์ของระบบรู้จำเสียงพูดบนเครื่องคอมพิวเตอร์จะมีความเที่ยงตรง (Double-precision floating point) สูงกว่าผลลัพธ์ของระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA (Single-precision floating point)

ในขั้นตอนการทดสอบระบบเพื่อนำไปใช้งานจริง เริ่มต้นด้วยการถ่ายโอนข้อมูล (ค่าถ่วงน้ำหนัก) จากอุปกรณ์เก็บข้อมูลเข้าไปในหน่วยความจำ (RAM) ของระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA จากนั้นเมื่อมีการกดสวิทช์ ระบบจะรับสัญญาณเสียงจากวงจร ADC เพื่อทำการดึงค่าลักษณะสำคัญและนำไปคำนวณหาค่าผลลัพธ์ของโครงข่าย แล้วตัดสินใจว่าสัญญาณเสียงที่รับเข้ามาตรงกับคำพูดคำใด

3.2.2 ระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA

ระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA ใช้ MicroBlaze เป็นหน่วยประมวลผลข้อมูล ซึ่งมีอุปกรณ์ต่อร่วม (Peripherals) ต่างๆ ดังนี้



รูปที่ 3.6 ระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA

- หน่วยความจำแบบ SDRAM ใช้สำหรับเก็บข้อมูล
- ช่องทางสื่อสารข้อมูลแบบอนุกรม (Serial port) ใช้รับ-ส่งข้อมูลกับเครื่องคอมพิวเตอร์และอุปกรณ์ต่างๆ
- LEDES (Output port) ใช้แสดงผลลัพธ์ของระบบ
- Switch (Input port) ใช้เลือกการแสดงผลการทำงานของระบบ

ในส่วนของ MicroBlaze ที่สร้างขึ้นภายในอุปกรณ์ FPGA สามารถกำหนดลักษณะพิเศษในการทำงานได้ (Configurable Features) ซึ่งลักษณะพิเศษที่เลือกใช้เพื่อเพิ่มประสิทธิภาพให้กับระบบ มีดังต่อไปนี้

- Barrel Shifter เป็นวงจรถิตจิตอลที่สามารถเลื่อนข้อมูลตามจำนวนบิตที่กำหนดได้ภายในการทำงาน 1 รอบสัญญาณนาฬิกา (single clock cycle)
- Floating-Point Unit (FPU) เป็นวงจรถิตพิเศษที่ออกแบบมาเพื่อปฏิบัติการทางคณิตศาสตร์ (การบวก ลบ คูณ หาร และเปรียบเทียบ) ของตัวเลขทศนิยม ซึ่งเชื่อมต่อโดยตรงกับโครงสร้างแบบ pipeline ของ MicroBlaze ทำให้การทำงานมีประสิทธิภาพสูง
- Integer Multiplier เป็นวงจรถิตคูณเลขจำนวนเต็ม ช่วยเพิ่มประสิทธิภาพของระบบ
- Pattern comparator เป็นกลุ่มคำสั่งที่ใช้ในการเปรียบเทียบค่าใน register

การสร้างระบบรู้จำเสียงพูดภายใน MicroBlaze นั้นค่อนข้างยุ่งยากกว่าระบบที่สร้างจากโปรแกรม MATLAB เนื่องจากต้องคำนึงถึงพื้นที่ของหน่วยความจำซึ่งใช้เก็บข้อมูลที่นำมาประมวลผล โดยข้อมูลเป็นตัวเลขทศนิยมแบบ single precision (ข้อมูลขนาด 32 บิต) ดังนั้นเพื่อลดขนาดของทรัพยากรที่ต้องใช้ในระบบและเพิ่มความเร็วในการทำงาน จึงต้องเปลี่ยนจากการเขียนโปรแกรมภาษา C เพื่อคำนวณค่าโดยตรงเป็นการสร้างตาราง (Look-up Table) ที่มีข้อมูลจากการคำนวณไว้ก่อนแล้วใน 3 ขั้นตอน คือ

- การคำนวณค่าของฟังก์ชันหน้าต่างแบบแฮมมิง
- การคำนวณค่าของฟังก์ชัน \cos และ \sin สำหรับใช้คำนวณ FFT 256 จุด
- การคำนวณค่าของฟังก์ชันตัวกรองความถี่แบบเมลในแต่ละช่อง

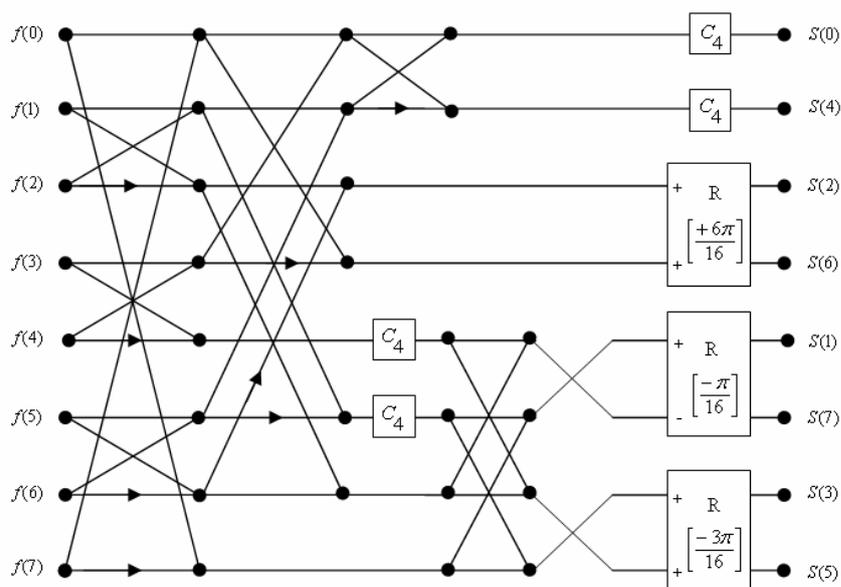
ซึ่งมีตัวอย่าง Look-up Table ดังนี้ (รายละเอียดภาษา VHDL ดูได้จากภาคผนวก ข)

```

READ_ADDR : process( ar_address ) is
begin
  case ar_address is
    -- hamming window
    when x"000" => ar_data_out(0) <= x"3da3d70a";
    when x"001" => ar_data_out(0) <= x"3da4203f";
    when x"002" => ar_data_out(0) <= x"3da4fbd3";
    :
    when x"199" => ar_data_out(0) <= x"3e37d027";
    when x"19a" => ar_data_out(0) <= x"3e25ec6b";
    when x"19b" => ar_data_out(0) <= x"3e1b3dba";
    when others => ar_data_out(0) <= (others => '0');
  end case;
end process READ_ADDR;

```

สำหรับ DCT เพื่อให้การคำนวณเร็วขึ้น จึงเขียนโปรแกรมภาษา C จากแผนภาพการคำนวณ Fast DCT 8 จุด ซึ่งมีรายละเอียดของโปรแกรมดังนี้



รูปที่ 3.7 แผนภาพที่ใช้ในการคำนวณ Fast DCT 8 จุด

```

void dct(float *feature, float *dct_coef)
{
    register unsigned long i;

    float s07, s12, s34, s56, d12, d56, d07, d34, tmp1, tmp2, tmp3, tmp4;

    float C4 = 0.707106781186548;
    float C6 = 0.38268343236509;
    float S6 = 0.923879532511287;
    float C3 = 0.831469612302545;
    float S3 = 0.555570233019602;
    float C1 = 0.98078528040323;
    float S1 = 0.195090322016128;

    float half = 2;

    s07 = feature[0] + feature[7];
    s12 = feature[1] + feature[2];
    s34 = feature[3] + feature[4];
    s56 = feature[5] + feature[6];
    d12 = feature[1] - feature[2];
    d56 = feature[5] - feature[6];
    d07 = feature[0] - feature[7];
    d34 = feature[3] - feature[4];

    tmp1 = s07+s34;
    tmp2 = s12+s56;

    dct_coef[0] = ( tmp1+tmp2)*(C4) /half;
    dct_coef[4] = ( tmp1-tmp2)*(C4) /half;

    tmp1 = d12-d56;
    tmp2 = s07-s34;

    dct_coef[2] = ( tmp1*(C6) + tmp2*(S6) )/half;
    dct_coef[6] = ( tmp1*(-S6) + tmp2*(C6) )/half;

    tmp1 = (C4)*(s12-s56);
    tmp2 = (C4)*(d12+d56);

    tmp3 = d07-tmp1;
    tmp4 = d34-tmp2;

    dct_coef[3] = ( (C3)*tmp3 - (S3)*tmp4 )/half;
    dct_coef[5] = ( (S3)*tmp3 + (C3)*tmp4 )/half;

    tmp3 = d07+tmp1;
    tmp4 = d34+tmp2;

    dct_coef[1] = ( (C1)*tmp3 + (S1)*tmp4 )/half;
    dct_coef[7] = ( (S1)*tmp3 - (C1)*tmp4 )/half;
}

```

ส่วนการทำงานของระบบรู้จำเสียงพูดในขั้นตอนต่อไปนี้ออกแบบโดยการเขียนโปรแกรมภาษา C

- การปรับระดับสัญญาณ
- การเน้นล่องหน้า
- การเพิ่มค่า Delta ให้สัญญาณ
- การคำนวณค่าพลังงานของเสียง
- การคำนวณค่าระยะห่างแบบยูคลีเดียน
- การตรวจหาขอบเขตของคำ
- การคำนวณผลลัพธ์ของโครงข่ายประสาทเทียม

3.2.3 รายละเอียดของระบบรู้จำเสียงพูดที่ทำงานบนอุปกรณ์ FPGA

การทำงานของระบบรู้จำเสียงพูดบนอุปกรณ์ FPGA ในแต่ละขั้นตอน มีรายละเอียดของค่าตัวแปร (Parameters) ต่างๆ ดังนี้

- สัญญาณเสียงมีอัตราการสุ่มตัวอย่าง (Sampling rate) 10 kHz
- การเน้นล่องหน้าใช้ค่าสัมประสิทธิ์ตัวกรอง (a) เท่ากับ 0.95
- สัญญาณเสียงถูกแบ่งเป็นเฟรม เฟรมละ 236 ตัวอย่าง (ประมาณ 24 mSec) เพื่อหาค่าความแตกต่างของเฟรมรอบข้าง
- สัญญาณในแต่ละเฟรมเหลื่อมกัน 20 ตัวอย่าง
- Hamming window มีขนาด 256 ตัวอย่าง (ประมาณ 26 mSec)
- จำนวนจุดในการแปลง FFT เท่ากับ 256
- จำนวนช่องของตัวกรองความถี่แบบเมลเท่ากับ 8
- จำนวนสัมประสิทธิ์ Cepstrum ในแต่ละเฟรมเท่ากับ 8
- จำนวนสัมประสิทธิ์ Cepstrum หลังจากปรับจำนวนเพื่อป้อนให้โครงข่าย (จำนวนโหนดในชั้นอินพุตของโครงข่าย) เท่ากับ 125
- จำนวนชั้นซ่อนตัวของโครงข่ายเท่ากับ 3
- จำนวนโหนดในชั้นซ่อนตัวของโครงข่ายเท่ากับ 25, 18, 16
- จำนวนโหนดในชั้นเอาต์พุตของโครงข่ายเท่ากับจำนวนค่าที่ต้องการรู้จำเท่ากับ 7
- จำนวนโครงข่ายประสาทเทียมเท่ากับ 2