บทที่ 4

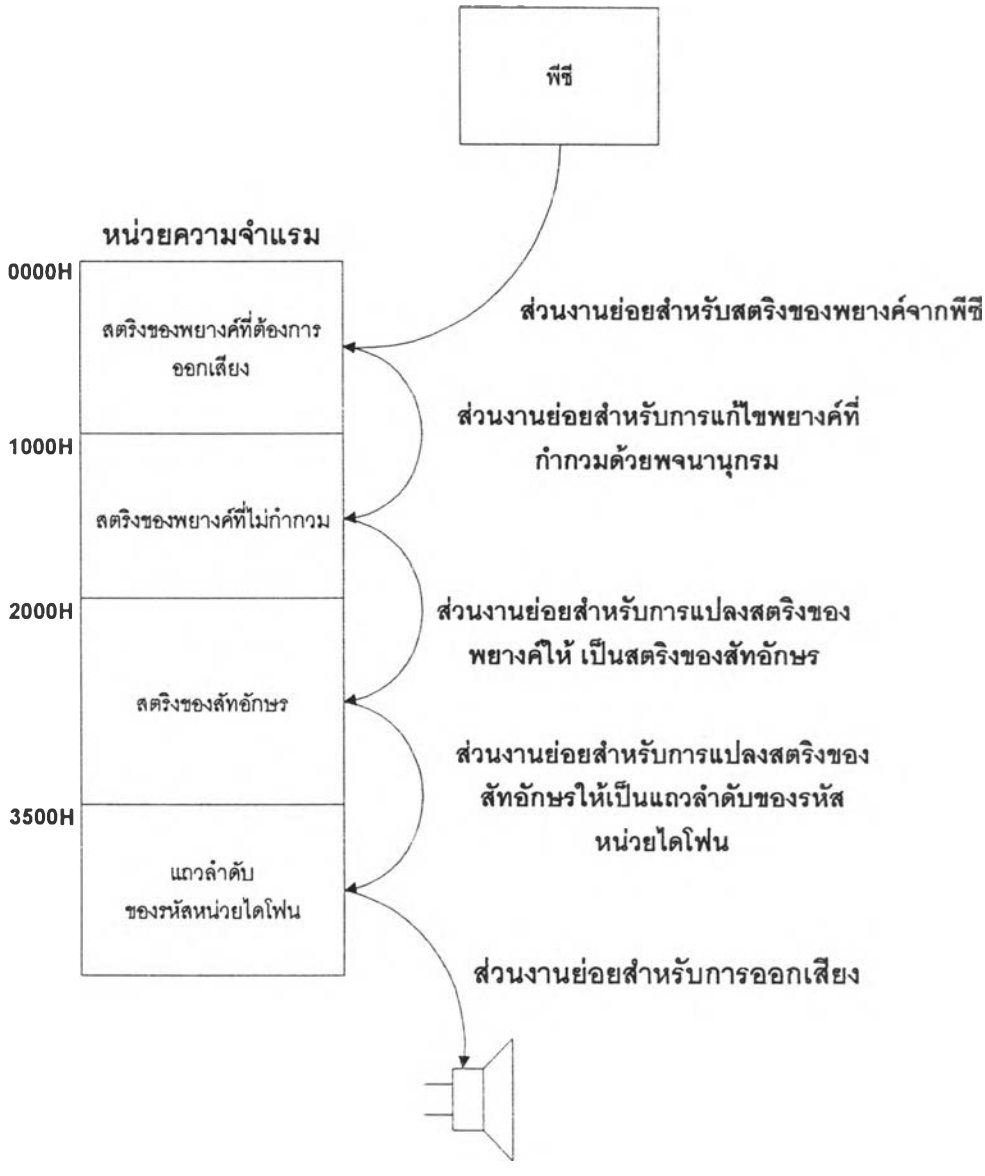
การพัฒนาซอฟต์แวร์ของเครื่องอ่านคำไทยพยางค์เดียวแบบปรับความเร็วได้

4.1 บทนำ

ซอฟต์แวร์ของเครื่องอ่านคำไทยพยางค์เดียวแบบปรับความเร็วได้นี้ถูกพัฒนาเป็นส่วนงานย่อย(module)ต่างๆ แต่ละส่วนสามารถทำงานได้เป็นอิสระ เพื่อให้สะดวกต่อการนำไปพัฒนาและแก้ปัญหา อีกทั้งยังช่วยให้เข้าใจการทำงานของระบบโดยรวมได้ง่าย เมื่อได้พัฒนาส่วนงานย่อยต่างๆเรียบร้อยแล้ว ก็จะนำส่วนต่างๆนั้นมาประกอบกันเป็นระบบที่สมบูรณ์ โดยในที่นี้จะทำการแบ่งการทำงานที่ต้องอาศัยซอฟต์แวร์ออกเป็น 5 ส่วนงานย่อยหลักที่สำคัญดังในรูปที่ 4.1

ลำดับขั้นตอนของการพัฒนานั้นไม่ได้เริ่มพัฒนาตามลำดับในรูปที่ 4.1 แต่จะเริ่มที่ส่วนของโปรแกรมย่อยควบคุมอุปกรณ์ต่างๆของระบบเป็นส่วนแรก เมื่อฮาร์ดแวร์ต่างๆทำงานได้ถูกต้องตามต้องการแล้วจึงค่อยพัฒนาส่วนของการออกเสียงเป็นขั้นตอนต่อไป โดยในขั้นนี้จะพัฒนาโปรแกรมให้ใช้กับสัญญาณเสียงที่ไม่ได้ถูกบีบอัด เมื่อสามารถทำให้เครื่องออกเสียงได้แล้วจึงพัฒนาส่วนงานย่อยการแยกสัทอักษรสากลให้เป็นหน่วยเสียงต่างๆและเชื่อมต่อกับส่วนงานย่อยของการออกเสียง จากนั้นจึงคอยพัฒนาส่วนงานย่อยของการออกเสียงต่อไปโดยได้เพิ่มความสามารถในการคลายสัญญาณเสียงที่ถูกบีบอัดได้

ขั้นต่อไปของการพัฒนาก็คือ การพัฒนาส่วนงานย่อยในส่วนของการแปลงพยางค์ไทยให้เป็นสัทอักษรสากล ในขั้นตอนนี้จะเป็นส่วนที่ยุ่งยากอีกส่วนงานย่อยหนึ่ง เนื่องจากจะต้องศึกษาหลัก ไวยากรณ์ไทยเพื่อที่จะนำมาสร้างเป็นขั้นตอนวิธี (algorithm) ในการแยกหน่วยต่างๆของพยางค์เดี่ยวให้ได้



รูปที่ 4.1 แสดงลำดับการทำงานของโปรแกรมและตำแหน่งที่อยู่ของข้อมูลในหน่วยความจำแรมของเครื่องอ่านคำไทยพยางค์เดี่ยวแบบปรับความเร็วได้

ขั้นตอนต่อไปก็จะเป็นการพัฒนาส่วนงานย่อยในส่วนของการตรวจสอบพยางค์พิเศษและพยางค์ยกเว้นต่างๆกับพจนานุกรมที่สร้างไว้สำหรับการตรวจสอบและแปลงให้สามารถใช้กับส่วนของการแยกหน่วยต่างๆของพยางค์ ท้ายสุดจะเป็นการพัฒนาส่วนงานย่อยของการออกเสียงต่อในด้านของการปรับความเร็วในการออกเสียง โดยในขั้นตอนนี้จะต้องเตรียมการหาตำแหน่งคาบของลูกคลื่นแต่ละลูกคลื่นในหน่วยเสียงทุกหน่วย เพื่อให้สามารถทำการบีบอัดสัญญาณแต่ละลูกคลื่น ได้ทำให้ขั้นตอนของการปรับความเร็วในการอ่านออกเสียงนั้น ทำได้โดยการข้ามการ

ออกเสียงของลูกคลื่นบางลูกไปทำให้สามารถพูดเร็วได้โดยที่เสียงที่ได้ยังเป็นเสียงเดิมไม่มีการเปลี่ยนความถี่ โดยในส่วนการเตรียมการหาตำแหน่งคาบของลูกคลื่นนี้ได้ใช้โปรแกรมกำหนดตำแหน่งพิตช์ (ณัฐกร ทับทอง,2538) ช่วยในการกำหนดตำแหน่งพิตช์ของแต่ละลูกคลื่น โดยเลือกค่าพารามิเตอร์ของโปรแกรมที่ใช้กับทุกหน่วยเสียงให้เหมือนกันแล้วสั่งให้ ทำงานวนจนกว่าจะครบทุกหน่วยเสียง แม้ว่าผลที่ได้อาจไม่แม่นยำเท่ากับการกำหนดตำแหน่งพิตช์ด้วยคน แต่ก็อยู่ในเกณฑ์ที่ยอมรับได้ จากนั้นจึงนำผลที่ได้ไปทำการบีบอัดสัญญาณ กระบวนการทั้งหมดนี้ทำเสร็จได้ภายในเวลาไม่ถึงครึ่งชั่วโมง ในที่นี้รวมเวลาการบีบอัดสัญญาณเสียงด้วย ซึ่งจะเร็วกว่าการทำด้วยคนมาก

4.2 ส่วนงานย่อยสำหรับรับสตริงของพยางค์จากพีซี

ส่วนนี้จะทำหน้าที่คอยรับและเก็บสตริงของพยางค์จากคอมพิวเตอร์ส่วนบุคคล(พีซี) โดยจะส่งอักษรบอกว่าพร้อม(R) ไปยังพีซีเพื่อบอกโปรแกรมประยุกต์ที่ติดต่ออยู่กับเครื่องให้ทราบว่าพร้อมที่จะทำงานแล้วให้ทำการส่งข้อมูลมาได้ โดยจะทำการส่งอักษรพร้อมนี้ทุกๆ 50 มิลลิวินาทีเพื่อทำการกระตุ้นให้โปรแกรมประยุกต์บนพีซีให้เริ่มทำการส่งข้อมูล ผังการทำงานของส่วนงานย่อยรับข้อมูลเป็นดังรูปที่ 4.2

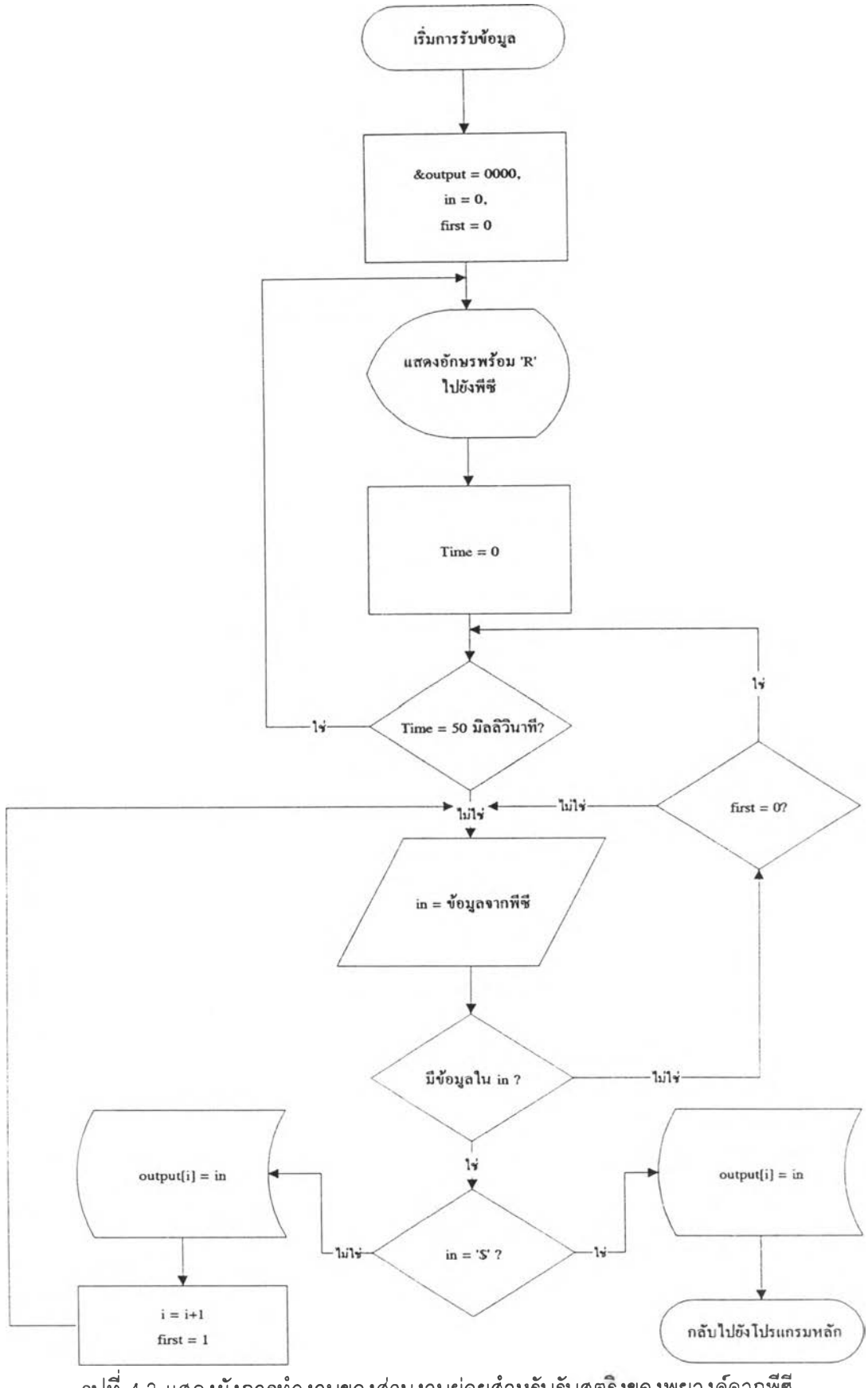
4.2.1 ลักษณะสตริงของพยางค์

ลักษณะของสตริงของพยางค์ที่ส่งมาให้ออกเสียงพูดนั้นประกอบไปด้วย 4 ส่วนที่สำคัญคือ

1. คำสั่งปรับความเร็วในการออกเสียงของพยางค์ คำสั่งนี้จะมีหรือไม่ก็ได้โดยถ้าไม่มีก็จะถือเอาความเร็วครั้งล่าสุดนั้นใช้ในการพูด ค่าโดยปริยายในตอนเปิดระบบมาครั้งแรกคือระดับปกติ(0) รูปแบบที่ใช้ก็คือจะส่งเครื่องหมาย '!' และตามด้วยระดับของความเร็วโดยเริ่มที่ 0 - 3 ขนาด 1 ไบต์ คำสั่งนี้จะต้องอยู่หน้าสุดและติดกับพยางค์เสมอ

2. คำสั่งสั่งให้เงียบชั่วขณะ โดยจะใช้เครื่องหมาย '*' ในการสั่งให้เงียบ ผู้ใช้สามารถส่งเครื่องหมาย '*' มาได้ไม่จำกัดโดย 1 ตัวจะหมายถึงการเงียบ 100 มิลลิวินาที

3. ส่วนของพยางค์ที่ต้องการจะออกเสียงในส่วนนี้ข้อมูลจะต้องเป็นคำอ่านไทยพยางค์เดี่ยว เว้นระหว่างคำด้วยช่องว่าง



รูปที่ 4.2 แสดงผังการทำงานของส่วนงานย่อยสำหรับรับสตริงของพยางค์จากพีซี

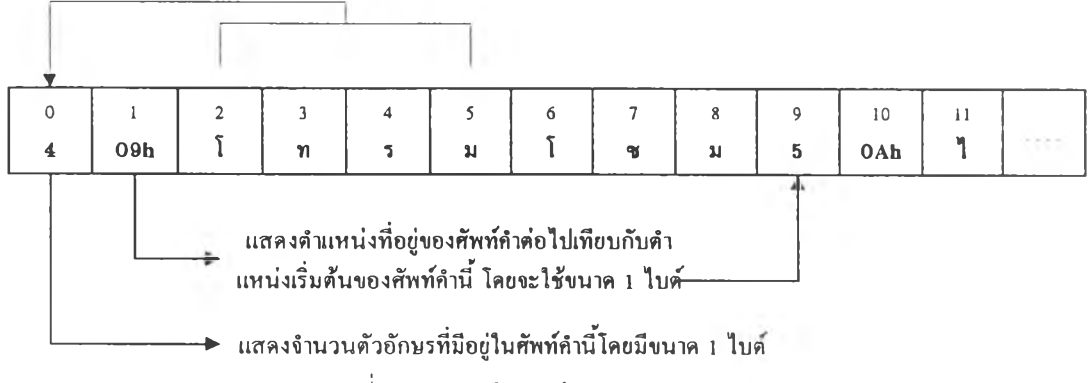
4. ส่วนปิดท้ายข้อมูล เป็นการบอกให้เครื่องอ่านคำไทยพยางค์เดี่ยวแบบปรับความเร็วได้ให้รู้ว่าหมดข้อมูลที่ต้องการจะออกเสียงแล้ว

ตัวอย่างเช่น สะ !1หวัด ดี ครับ ***** !3ยิน !2ดี ต้อน รับ$

4.3 <u>ส่วนงานย่อยสำหรับการแก้ไขพยางค์ที่กำกวมด้วยพจนานุกรม</u>

ส่วนงานย่อยนี้จะทำการตรวจสอบพยางค์ที่กำกวมต่างๆและแก้ไขให้ไม่กำกวมโดยใช้พจนานุกรม เช่นคำว่า 'เกียรติ' จะแก้ให้เป็น 'เกียด' หรือคำว่า 'โหน' ซึ่งจะกำกวมตรงที่ว่าพยัญชนะต้นจะเป็นพยัญชนะ 'ห' ตัวเดี่ยวหรือเป็น 'หน' ซึ่งเป็นอักษรนำและอักษรตามก็ได้ แต่ในความเป็นจริงคำนี้จะมีพยัญชนะต้นคือ 'ห' โดยมี 'น' เป็นตัวสะกด ดังนั้นเพื่อลดความกำกวมในวิทยานิพนธ์นี้ได้ใช้อักษร '.' เพื่อแยก 'ห' และ 'น' ออกจากกันอย่างชัดเจนเป็นดังนี้ 'ห.น'

โครงสร้างของพจนานุกรมเป็นดังรูปที่ 4.3

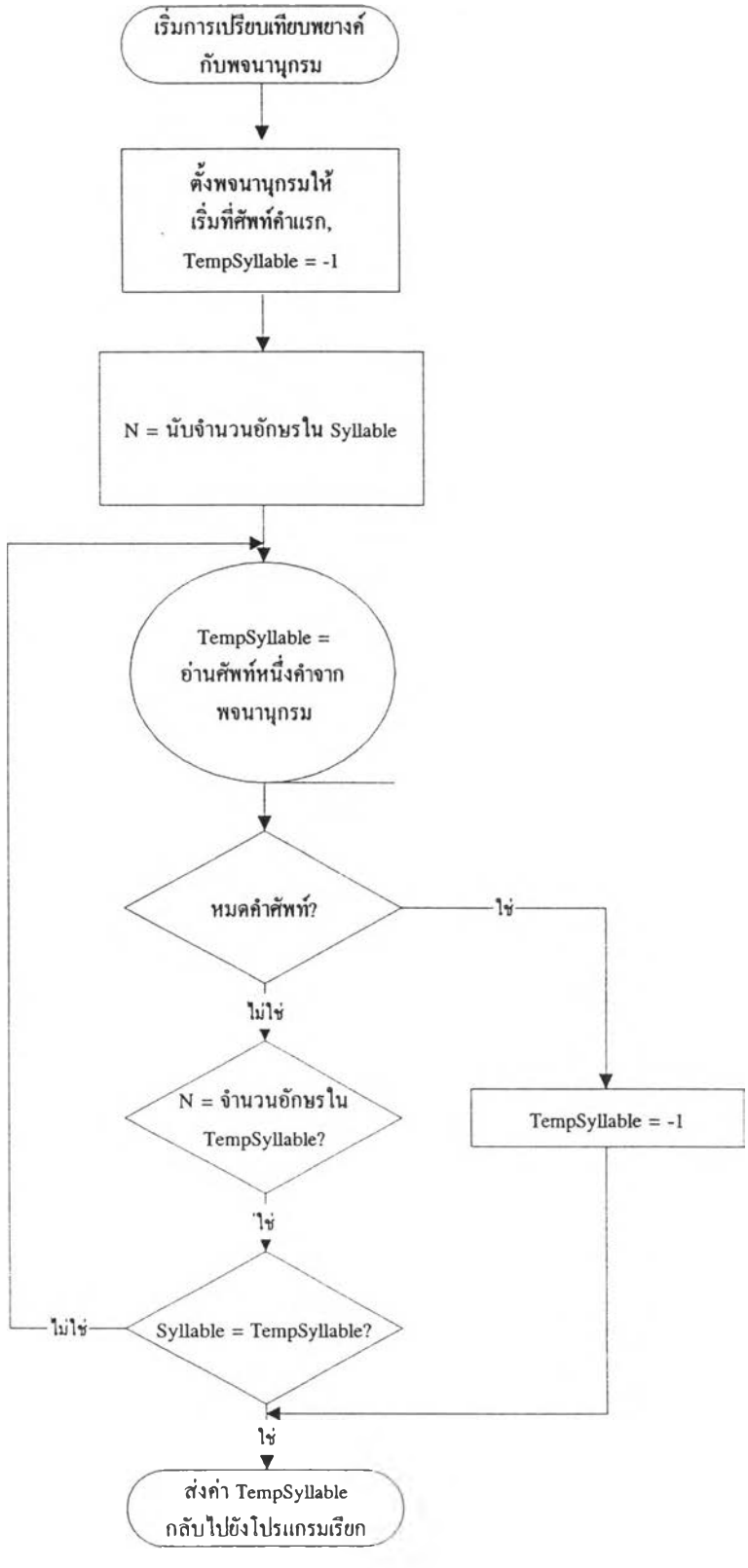


รูปที่ 4.3 แสดงโครงสร้างของพจนานุกรม

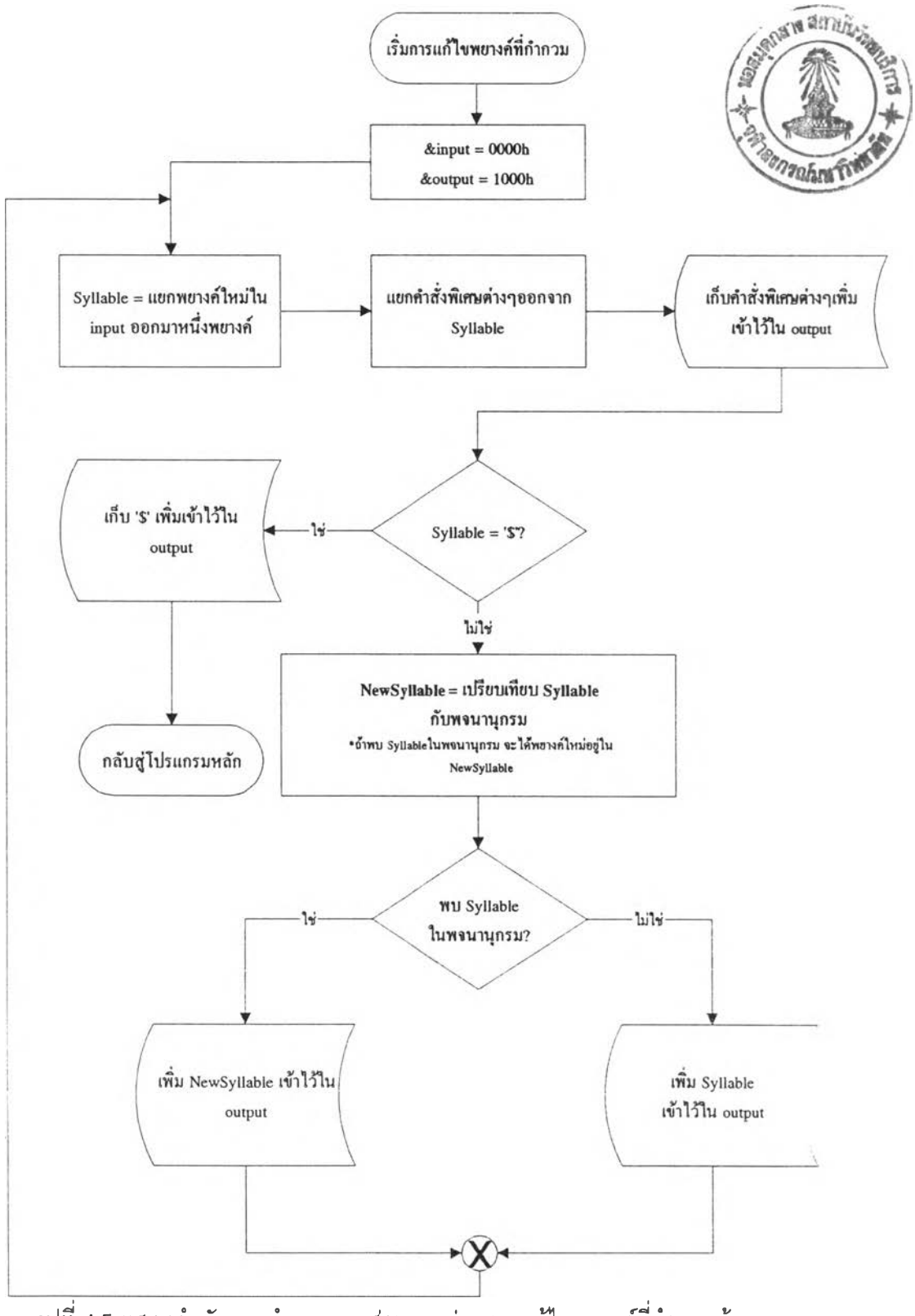
การใช้พจนานุกรม เพื่อสืบค้นพยางค์(syllable) ที่ต้องการ จะมีวิธีการดังแสดงในรูปที่ 4.4 กล่าวคือจะใช้จำนวนตัวอักษรในพยางค์ เป็นตัวช่วยในการเปรียบเทียบ โดยจะนำเอาจำนวนตัวอักษรพยางค์นั้นมาเปรียบเทียบกันก่อนว่าเท่าหรือไม่ ถ้าเท่าจึงค่อยทำการเปรียบเทียบพยางค์แบบตัวต่อตัวต่อไป แต่ถ้าไม่เท่าก็จะทำการอ่านคำศัพท์คำต่อไปมาเปรียบเทียบโดยการอาศัยข้อมูลใน 1 ไบต์ถัดมาซึ่งจะใช้บอกตำแหน่งที่อยู่ของคำศัพท์ถัดไป สำหรับในกรณีที่ผลการเปรียบเทียบแล้วพบว่า Syllable เป็นพยางค์เดียวกันกับศัพท์ ก็จะสามารถนำคำแปลไปใช้ได้โดยคำแปลจะอยู่ติดกับศัพท์คำนั้น สำหรับความยาวของคำแปลจะหาได้จาก

ความยาวของคำแปล = ข้อมูลของศัพท์ในไบต์ที่สอง - ข้อมูลของศัพท์ในไบต์แรก - 2

สำหรับการทำงานของส่วนงานย่อยนี้อยู่ในรูปที่ 4.5 ซึ่งจะมีส่วนงานย่อยย่อยที่สำคัญในการแก้ไขพยางค์ที่กำกวมด้วยพจนานุกรมก็คือส่วนงานย่อยย่อยของการเปรียบเทียบพยางค์กับพจนานุกรมดังกล่าวข้างต้น การเรียกใช้ส่วนงานย่อยนี้จะต้องทำการส่งข้อมูลหนึ่งพยางค์ไว้ในตัวแปร Syllable และส่วนงานย่อยการเปรียบเทียบพยางค์กับพจนานุกรมจะส่งค่ากลับเป็นพยางค์ใหม่ที่อยู่ในตัวแปร TempSyllable ไปให้กับส่วนงานย่อยที่เรียก ถ้าเกิดว่าค่าตัวแปร TempSyllable เป็น -1 จะหมายถึงว่าไม่พบพยางค์นั้นในพจนานุกรม นอกนั้นจะหมายถึงว่าพบพยางค์นั้นในพจนานุกรม



รูปที่ 4.4 แสดงการทำงานของส่วนงานย่อยย่อยการเปรียบเทียบพยางค์กับพจนานุกรม



รูปที่ 4.5 แสดงลำดับการทำงานของส่วนงานย่อยการแก้ไขพยางค์ที่กำกวมด้วยพจนานุกรม

4.4 ส่วนงานย่อยการแปลงสตริงของพยางค์ให้เป็นสตริงของสัทอักษร

สัทอักษรสากล (International Phonetic Alphabets) คือสัญลักษณ์สากลที่ใช้แทนการออกเสียงต่างๆ ซึ่งเมื่อนำไปแทนคำในภาษาใดด้วยสัทอักษรสากลแล้ว จะทำให้ผู้อื่นที่ไม่ทราบภาษานั้นสามารถเข้าใจและสามารถออกเสียงของคำนั้นได้ สาเหตุที่ต้องทำการแปลงเป็นสัทอักษรสากลก่อนนั้นเป็นเพราะว่าข้อมูลหน่วยเสียงที่เก็บไว้นั้น ได้เก็บไว้ในชื่อทางสัทอักษรสากล ทำให้การอ้างอิงของหน่วยเสียงต่างๆเหล่านั้นจะต้องอาศัยสัทอักษรสากลเป็นสื่อกลาง สำหรับลำดับการทำงานของส่วนงานย่อยนี้เป็นตามในรูปที่ 4.6

จากรูปที่ 4.6 ลำดับการทำงานของส่วนงานย่อยจะเริ่มจากการแยกพยางค์หนึ่งพยางค์ออกจากสตริงของพยางค์ จากนั้นก็จะทำการแยกและเก็บคำสั่งพิเศษต่างๆออกจากพยางค์ที่แยกได้นั้น จากนั้นจึงส่งพยางค์นั้นไปยังส่วนงานย่อยย่อยที่ทำการแยกพยางค์นั้นออกเป็นหน่วยต่างๆที่ประกอบเป็นพยางค์ตามหลักไวยากรณ์ไทย เมื่อแยกได้แล้วก็จะให้ผลลัพธ์นั้นแก่ส่วนงานย่อยย่อยถัดไปที่จะทำการหาสัทอักษรสากลที่จะแทนพยางค์นั้น โดยอาศัยหน่วยต่างๆที่แยกได้มาร่วมกันทำการแปลง สุดท้ายเมื่อหาสัทอักษรสากลได้แล้วก็จะนำผลนี้ไปเก็บในส่วนของข้อมูลออก

ในกระบวนการทำงานในส่วนนี้จะมีส่วนงานย่อยการทำงานที่สำคัญ 2 ส่วนดังนี้

4.4.1 ส่วนงานย่อยแยกพยางค์ไทยคำเดี่ยวออกเป็นหน่วยย่อยต่างๆตามหลักไวยากรณ์ในภาษาไทย(สมชาย, 2526)

หลักไวยากรณ์ภาษาไทยได้จำแนกอักษรในพยางค์ออกเป็น 4 หน่วยย่อยที่เกี่ยวข้องกับการออกเสียงดังนี้

1. พยัญชนะต้น
2. สระ
3. วรรณยุกต์
4. ตัวสะกด

ที่ต้องทำการแบ่งพยางค์ออกเป็นหน่วยย่อยๆก็เพราะว่าจะสามารถแทนพยางค์นั้นด้วยสัทอักษรสากลโดยการเทียบหน่วยต่างๆกับตารางที่ทำไว้สำหรับการแปลง ผลที่ได้จากการทำงานในส่วนนี้จะเป็นข้อมูลเข้าให้กับส่วนงานย่อยการหาสัทอักษรแทนพยางค์ต่อไป

สำหรับปัญหาในขั้นตอนของการแยกหน่วยย่อยต่างๆในพยางค์เดี่ยวก็คือการที่พยัญชนะบางตัวสามารถเป็นได้ทั้งพยัญชนะต้นเดี่ยว,สระ,และตัวสะกดได้แก่ 'ร,ว,ย' ส่วน

'อ' เป็นได้เฉพาะพยัญชนะต้นและสระเท่านั้นโดยแต่ละกรณีจะมีตำแหน่งการปรากฏในพยางค์ที่ไม่เหมือนกัน ทำให้การเกิดความยุ่งยากในขั้นการแยกหน่วย เช่น 'ว' โดยปกติจะเป็นพยัญชนะต้นและตัวสะกด แต่ถ้าต้องการจะเป็นสระ ก็จะเป็นได้สระเดียวคือ สระอัว อย่างในคำว่า 'กวน' ในขณะที่ 'ย' ถ้าจะเป็นสระจะต้องเป็นสระผสมเท่านั้นคำว่า 'เมีย' เหล่านี้เป็นต้น กระบวนการแยกพยางค์จะประกอบไปด้วยการทำงาน 2 ส่วนคือ

1. ส่วนของขั้นตอนวิธีหลักในการแยกหน่วยของพยางค์ไทยคำเดี่ยว
2. ส่วนของขั้นตอนวิธีการตรวจสอบและแก้ไข

แต่ทั้งหมดนั้นตั้งอยู่บนสมมุติฐานที่ว่าพยางค์ที่เข้ามาตรวจสอบนั้นจะต้องถูกต้องตามหลักไวยากรณ์เสมอ ถ้าพยางค์ที่เข้ามาผิดไวยากรณ์หรือเป็นคำหลายพยางค์แล้วการแปลงก็จะผิดทำให้อ่านออกเสียงผิดไปหรือไม่ก็จะไม่ออกเสียง

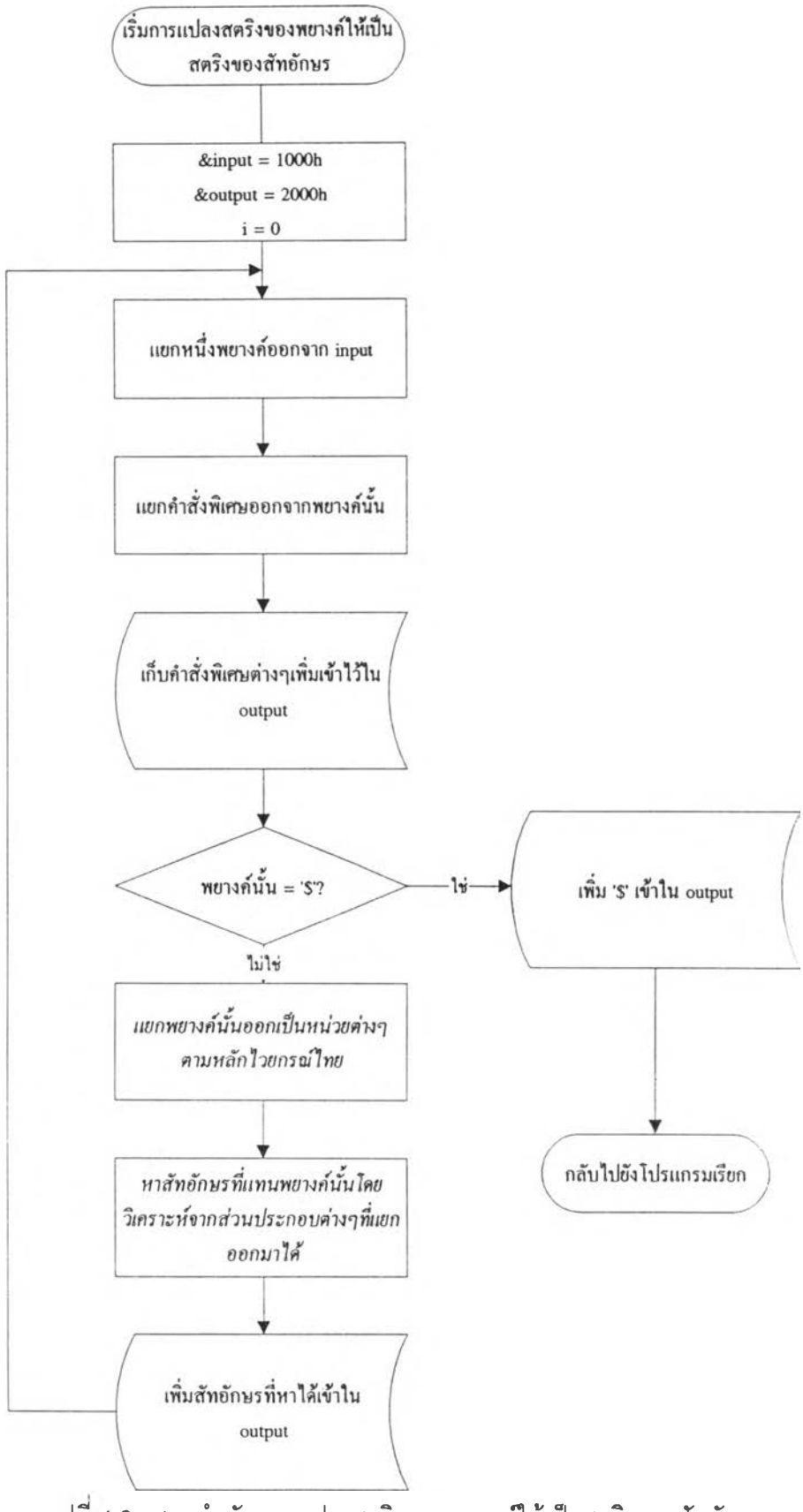
4.4.1.1 ขั้นตอนวิธีการแยกหน่วยพยางค์ที่เป็นพยางค์ไทยคำเดียวออกเป็นหน่วยย่อย

เพื่อให้เข้าใจการทำงานในขั้นตอนวิธีนี้ จะขออธิบายหลักที่ใช้ในการสร้างขั้นตอนวิธีและคำที่ใช้เรียกต่างๆ ในขั้นตอนวิธี

1. สระนำ ได้แก่ "เ,แ,โ,ใ,ไ," ซึ่งจะต้องมาก่อนทุกหน่วยและจะต้องอยู่นำหน้าพยัญชนะต้นเสมอ

2. พยัญชนะต้นจะต้องมาก่อนหน่วยอื่นทุกหน่วย ยกเว้นในกรณีที่มีสระนำ พยัญชนะต้นนี้สามารถเป็นอักษรควบกล้ำหรืออักษรนำกับอักษรตามได้ ตัวอย่างของอักษรควบกล้ำคือ "กว,กล,กร,คร,ขว,ฯลฯ" ตัวอย่างของอักษรนำกับอักษรตามก็คือ "หน,หร,หล,หว,อย," เนื่องจากมีพยัญชนะบางตัวที่สามารถเป็นได้ทั้งพยัญชนะต้น,สระหรือตัวสะกดได้ดังนั้นในส่วนของอักษรในพยางค์ที่เป็นพยัญชนะจะแบ่งออกเป็น 5 ส่วนคือ

- พยัญชนะทุกตัวยกเว้นพยัญชนะ "อ,ย,ว,ร"
- พยัญชนะ อ
- พยัญชนะ ย
- พยัญชนะ ว
- พยัญชนะ ร



รูปที่ 4.6 แสดงลำดับการแปลงสตริงของพยางค์ให้เป็นสตริงของสัทอักษร

3. วรรณยุกต์ ถ้ามีจะต้องอยู่ก่อนตัวสะกดเสมอและต้องตามหลังพยัญชนะต้นหรือตามหลังสระตามบางสระที่ใช้เขียนอยู่บนพยัญชนะต้นดังนี้ สระอิ,สระอี,สระอึ, สระอือ และไม้หันอากาศ

4. สระตาม สระตามในที่นี้คือ "ะ,า,อิ,อี,อึ,อือ,อุ,อู,อำ" ถ้ามีจะต้องอยู่หลังพยัญชนะต้นและอยู่ก่อนตัวสะกดเสมอ

5. สระลดรูปต่างๆ ถ้ามีจะต้องอยู่หลังพยัญชนะต้นเสมอและอยู่ก่อนตัวสะกดเช่น ไม้ไต่คู้,รอ หัน(รร)

6. ตัวสะกด ถ้ามีจะเป็นหน่วยที่อยู่หลังที่สุดยกเว้นจะมีตัวการันต์ในพยางค์

7. ตัวการันต์ ถ้ามีจะต้องอยู่หลังและติดกับตัวสะกด และจะต้องไม่ออกเสียงตัวสะกดนั้น

สำหรับขั้นตอนวิธีที่สร้างจะเป็นดังในรูปที่ 4.7 ในขั้นตอนวิธีนี้จะเริ่มต้นที่รับข้อมูลเข้าเป็นพยางค์คำไทยคำเดี่ยวใน Input และจะมีตัวแปรสำคัญอยู่ 4 ตัวแปรที่จะไว้เก็บหน่วยต่างๆ โดยจะมีขนาดของตัวแปรดังนี้

1. พยัญชนะต้น ขนาด 2 ไบต์เพราะโดยปกติพยัญชนะต้นจะมีมากสุดไม่เกิน 2 ตัว
2. สระ ขนาด 4 ไบต์เพราะในภาษาไทยจำนวนตัวอักษรในสระที่มีจำนวนอักษรมากที่สุดจะไม่เกิน 4 ตัวอักษร
3. ตัวสะกด ขนาด 1 ไบต์เพราะตามหลักในภาษาไทย ตัวสะกดจะมีไม่เกิน 1 ตัว
4. วรรณยุกต์ ขนาด 1 ไบต์เพราะตามหลักในภาษาไทย วรรณยุกต์จะมีไม่เกิน 1 ตัว

จากนั้นจะทำการกวาดตรวจข้อมูลใน Input ที่ละตัวจากซ้ายไปขวาว่าอยู่ในกรณีใดใน 10 กรณี เมื่อพบว่าอยู่ในกรณีใดแล้ว ก็จะทำการเรียกส่วนงานย่อยในแต่ละกรณีนั้นทำการตรวจสอบคุณสมบัติของข้อมูลตัวนั้นว่าควรจะทำหน้าที่อะไรในพยางค์นั้น จากนั้นถ้าพบว่าข้อมูลตัวนั้นทำหน้าที่เป็นหน่วยใดในพยางค์แล้วก็จะทำการเก็บข้อมูลนั้นเข้าไว้ในตัวแปรของแต่ละหน่วยนั้นเพื่อใช้สำหรับการแปลงหน่วยต่างๆเหล่านี้เป็นสัทอักษรสากลต่อไป และถ้าไม่สามารถระบุได้ว่าทำหน้าที่อะไรก็แสดงว่าผิดไวยากรณ์ภาษาไทย และสำหรับการทำงานในแต่ละส่วนงานย่อยในรูปที่ 4.7 จะทำหน้าที่ดังต่อไปนี้

รูปที่ 4.7 แสดงขั้นตอนวิธีที่ใช้แยกหน่วยต่างๆของพยางค์ที่เป็นคำไทยพยางค์เดี่ยว

4.4.1.1.1 ส่วนงานย่อยของพยัญชนะ

ส่วนงานย่อยนี้จะถือว่าพยัญชนะที่ได้นั้นไม่รวมพยัญชนะ "อ,ย,ว,ร" เมื่อเป็นเช่นนี้ทำให้ไม่ต้องกังวลว่าพยัญชนะตัวที่มาตรวจสอบนี้จะสามารถเป็นสระได้หรือไม่ ดังนั้นความเป็นไปได้ที่พยัญชนะตัวนี้จะทำหน้าอะไรในพยางค์ก็จะเหลือเพียงแค่ พยัญชนะต้นหรือไม่ก็จะเป็นตัวสะกด สำหรับการที่จะเป็นพยัญชนะต้นนั้น พยัญชนะในกลุ่มนี้สามารถที่จะเป็นคำควบกล้ำหรืออักษรนำกับอักษรตามได้ โดยจะต้องมีการตรวจสอบกับตารางซึ่งจะมีการบันทึกเก็บเอาพยัญชนะต้นที่เป็นคำควบกล้ำหรืออักษรนำกับอักษรตามเอาไว้ทั้งหมดซึ่งจะมี 35 คู่ดังนี้ ปร,ปฤ,ตร,ตฤ,กร,กฤ,ปล,กล,กว,พร,พฤ,ภร,ภฤ,ทร,ทฤ,คร,ขร,พล.ผล,คล,ขล,คว,ขว,สร,ศร,หง,หม,หร,หว,หญ,หย,หน,หล,จร สำหรับขั้นตอนวิธีการตรวจดังกล่าวนี้สามารถแสดงได้ด้วยภาษาที่ดัดแปลงจากภาษาซีดังนี้

if ไม่มี พยัญชนะต้น
then { ใส่ Input[i] เป็นพยัญชนะต้นตัวที่หนึ่ง และ ดูว่า Input[i+1] สามารถใส่เป็นตัวควบกล้ำหรืออักษรตามได้ไหม ถ้าได้ก็ให้ใส่ Input[i+1] เป็นพยัญชนะต้นตัวที่สอง} ตย. เช่นคำว่า กราน จะได้ว่าพยัญชนะต้นเป็นอักษรควบกล้ำ
else if ไม่มีตัวสะกด และ มีสระ
then { ใส่ Input[i] ให้เป็นตัวสะกด } ตย. เช่นคำว่า กราน พยัญชนะ 'น' จะไม่สามารถเป็นพยัญชนะต้นได้เพราะว่ามีพยัญชนะต้น 'กร'อยู่ก่อนแล้ว ดังนั้นที่จะเป็นได้ก็คือ ตัวสะกด
else if ไม่มีตัวสะกดและไม่มีสระและ Input[i] เป็นตัวสะกดในแม่กน,กม,กง,กด
then { ใส่ Input[i] ให้เป็นตัวสะกดและให้สระเป็นสระโอะ}
ตย. เช่นคำว่า กล จะได้เป็น ก + สระโอะ+ล
else { return Error เพราะว่าไม่สามารถหาว่าInput[i] ทำหน้าที่อะไรในพยางค์}
end if
return Ok

4.4.1.1.2 ส่วนงานย่อยของพยัญชนะ 'อ'

'อ' นี้เป็นได้ทั้งพยัญชนะต้นเดี่ยวหรืออักษรนำหรือสระก็ได้ แต่ไม่สามารถเป็นตัวสะกดหรือคำควบกล้ำ ในกรณีเป็นอักษรนำนั้น 'อ' สามารถที่จะนำ 'ย' ได้เพียงตัวเดียวและจะมีใช้อยู่เพียง 4 คำเท่านั้นคือ "อย่า อยู่ อย่าง อยาก" สำหรับเงื่อนไข

ในการที่จะเป็นสระนั้น 'อ' สามารถเป็นสระเดียวหรือสระประสมก็ได้ เช่น "ดอง,เหลือ,เถอะ" ขั้นตอนวิธีในส่วนนี้เป็นดังนี้

if ไม่มีพยัญชนะต้น
then
{ ใส่ Input[i] เป็นพยัญชนะต้นตัวที่ 1 และ ดูว่า Input[i+1] สามารถใส่เป็นอักษรตามได้ไหม ถ้าได้ก็ให้ใส่ Input[i+1] เป็นพยัญชนะต้นตัวที่ 2) ตย. เช่นคำว่า **อย่า**
else if อ ต้องผ่านส่วนงานย่อยการตรวจสอบลำดับของสระและวรรณยุกต์และยังไม่มีตัวสะกด
{ เพิ่ม อ เข้าไปในสระ } ตย. เช่นคำว่า เรื่อง
else
{ return Error เนื่องเพราะไม่สามารถเป็นสระหรือพยัญชนะต้นได้ }
end if
return Ok

4.4.1.1.3 ส่วนงานย่อยของพยัญชนะ 'ย'

'ย' นี้เป็นได้ทั้งพยัญชนะต้นหรือสระประสมหรือตัวสะกด แต่ไม่สามารถเป็นคำควบกล้ำและอักษรนำได้ กรณีที่เป็นสระประสมนั้นได้เพียง 2 สระดังนี้คือ สระเอียและสระเอียะ ขั้นตอนวิธีที่ใช้ในส่วนนี้จะเป็นดังนี้

if ไม่มีพยัญชนะต้น
then { ใส่ Input[i] เป็น พยัญชนะต้น } ตย. เช่นคำว่า **ย**าย
else if ย ต้องผ่านส่วนงานย่อยการตรวจสอบลำดับของสระและวรรณยุกต์
then { เพิ่ม ย เข้าในสระ } ตย. เช่นคำว่า เสี**ย** ซึ่งจะทำหน้าที่เป็นสระเอีย
else if ไม่มีตัวสะกด
then { ใส่เป็นตัวสะกด } ตย. เช่นคำว่า ยา**ย**
else { return Error }
end if
return Ok

4.4.1.1.4 ส่วนงานย่อยของพยัญชนะ 'ว'

'ว' นี้เป็นได้ทั้งพยัญชนะต้นหรือสระหรือตัวสะกดได้ แต่ไม่สามารถเป็นอักษรตัวแรกในคำควบกล้ำหรืออักษรนำได้ กรณีที่เป็นสระ 'ว' สามารถเป็นได้สระ 'อัว' มีรูปสระ 2 รูปคือรูปสระเดี่ยวเช่น 'กวน' และรูปสระประสมคือ 'อัว' ขั้นตอนวิธีที่ใช้ในส่วนนี้จะเป็นดังนี้

if ไม่มีพยัญชนะต้น
then { ใส่ Input[i] เป็น พยัญชนะ ต้น } ตย. เช่นคำว่า วาน
else if ว ต้องผ่านส่วนงานย่อยการตรวจสอบลำดับของสระและวรรณยุกต์
then { ใส่ Input[i] เป็นสระ }ตย. เช่นคำว่า กวน,วัว
else if ไม่มีตัวสะกด
then { ใส่ Input[i] เป็นตัวสะกด }ตย. เช่นคำว่า กาว
else { return Error }
end if
return Ok

4.4.1.1.5 ส่วนงานย่อยของพยัญชนะ 'ร'

'ร' นี้เป็นได้ทั้งพยัญชนะต้นหรือสระหรือตัวสะกดได้ แต่ไม่สามารถเป็นอักษรคำแรกในคำควบกล้ำหรืออักษรนำ แต่ถ้าไม่สามารถเป็นพยัญชนะต้นได้ตัวมันก็สามารถที่จะเป็นสระได้ โดยมีเงื่อนไขว่า 'ร' จะเป็นสระได้เพียงอย่างเดียวคือ สระอะ ในรูปของ 'รร' เท่านั้นและจะต้องไม่มีสระอื่นหรือวรรณยุกต์อยู่ก่อนหน้านี้ และถ้าไม่มีสระอยู่ก่อนหน้านี้แล้ว ก็ยังจะต้องตรวจสอบดูว่ายังมีพยัญชนะ 'ร' ตัวอื่นอยู่ในส่วนของพยัญชนะต้นตัวที่สองซึ่งเกิดจากส่วนงานย่อยที่ 4.4.1.1.1 ที่มีการตรวจดูคำควบกล้ำหรือเป็นอักษรนำอักษรตามหรือจะมีอยู่ในหน่วยของตัวสะกด และถ้าพบว่ามีพยัญชนะ 'ร' อยู่ก่อนแล้วในหน่วยใดหนวยหนึ่งตามที่กล่าวมา ก็จะต้องทำการย้ายพยัญชนะ 'ร' ออกจากหน่วยพยางค์นั้นมาเป็นสระ 'รร' แทน แต่ถ้าไม่พบว่ามีพยัญชนะ 'ร' มาก่อน ก็ถึงค่อยมาดูว่ามีตัวสะกดหรือยังถ้ายังก็ให้เป็นตัวสะกด ตัวอย่างเช่นคำว่า 'วรรณ' เมื่อพยางค์นี้ผ่านเข้าสู่ขั้นตอนวิธีการนี้แล้วจนกระทั่งถึงพยัญชนะ 'ร' ตัวแรกซึ่งขั้นตอนวิธีนี้จะให้เป็นตัวสะกด เนื่องจากไม่พบพยัญชนะ 'ร' ในหน่วยพยัญชนะต้นหรือในหน่วยตัวสะกด และหน่วยตัวสะกดยังว่างอยู่ ดังนั้นจึงให้เป็นตัวสะกดได้ แต่พอเจอตัว 'ร' ตัวที่สอง ก็จะพบว่ามีตัว 'ร' มาก่อนหน้านี้และอยู่ที่หน่วยของตัวสะกด ทำให้สามารถที่จะเป็นสระได้ แต่จะต้องไม่มีสระมาก่อน ซึ่งก็ไม่มีสระมาก่อนจริงๆทำให้สามารถเป็นสระได้ ดังนั้นจึง

ต้องย้ายหน่วยของตัวสะกดมาเป็นสระคู่กับตัว 'ร' ใหม่นี้ หลังจากนั้นเมื่อขั้นตอนวิธีพบกับพยัญชนะ 'ณ' ก็จะเข้ากรณีของ 4.4.1.1 ซึ่งก็จะว่าได้เป็นตัวสะกดต่อไป ขั้นตอนวิธีที่ใช้ในส่วนนี้จะเป็นดังนี้

if ไม่มีพยัญชนะต้น
then { ใส่ Input[i] เป็นพยัญชนะต้น }
else if ไม่มีสระ และไม่มีวรรณยุกต์
then if มีพยัญชนะต้นตัวที่สองเป็น ร และไม่มีตัวสะกด
then{ ใส่ รร เป็นสระและตัดพยัญชนะต้นตัวที่ 2 ที่เป็น ร ทิ้ง} ตย. เช่นคำว่า กรรณ เมื่อได้ตัว ร ตัวที่ 2 เข้ามาในส่วนงานย่อยนี้ก็จะพบว่าหน่วยพยัญชนะต้นในตอนนี้จะเป็น **กร** อยู่ดังนั้นจะต้องทำการย้ายตัว ร ก่อนหน้าที่อยู่ในพยัญชนะต้นมาเป็นสระ **รร**
else if มีพยัญชนะต้น หนึ่งตัวและมีตัวสะกดเป็น ร
then { ใส่ รร เป็นสระ และตัดตัวสะกดที่เป็น ร ทิ้ง } ตย. เช่นคำว่า วรรณ เมื่อตัว ร ตัวที่ 2 เข้ามาในส่วนงานย่อยนี้จะพบว่าพยัญชนะต้นที่มีอยู่คือ **ว** และตัวสะกดคือ **ร** ดังนั้นจะต้องทำการย้ายตัว ร ในหน่วยของตัวสะกดมาเป็นสระ **รร**
else if ไม่มีตัวสะกด
then { ใส่ Input[i] เป็นตัวสะกด } ตย. เช่นคำว่า วรรณ ซึ่งตัว ร ตัวที่ 1 นี้ไม่สามารถเป็นพยัญชนะต้นได้ เพราะว่ามีตัว ว เป็นพยัญชนะต้นอยู่ก่อน แต่ในส่วนของตัวสะกดยังว่างอยู่ดังนั้นจึงสามารถที่จะเป็นตัวสะกดได้ ดังนั้นจึงให้เป็นตัวสะกด
else{ return Error}
else { return Error }
end if
return Ok

4.4.1.1.6 ส่วนงานย่อยของสระนำ

สระนำเป็นสระที่ทำให้ขั้นตอนวิธีในส่วนงานย่อยนี้ง่ายกว่าส่วนงานย่อยอื่นที่ผ่านมาดังนี้

if ยังไม่มีหน่วยอื่นมาก่อน

then { ใส่ Input[i] เป็น สระ }

else { return Error }

end if

return Ok

4.4.1.1.7 ส่วนงานย่อยของสระตาม

สระที่เป็นสระตามได้จะต้องมีพยัญชนะต้นอยู่ก่อนแล้วและยังไม่มีตัวสะกด ขั้นตอนวิธีที่ใช้ในส่วนนี้จะเป็นดังนี้

if มี พยัญชนะต้น และ ไม่มีตัวสะกดและผ่านส่วนงานย่อยการตรวจสอบลำดับของสระและวรรณยุกต์

*then { เพิ่ม Input[i] เข้าเป็นสระได้ }*ตย. เช่นคำว่า เกาะ

else { return Error }

end if

return

4.4.1.1.8 ส่วนงานย่อยของวรรณยุกต์

เงื่อนไขของการที่จะเป็นวรรณยุกต์ได้คือจะต้องมีพยัญชนะต้นและต้องไม่มีตัวสะกดและจะมีวรรณยุกต์ซ้ำมากกว่า 1 ตัวไม่ได้ ขั้นตอนวิธีที่ใช้ในส่วนนี้จะเป็นดังนี้

if มี พยัญชนะต้น และไม่มีตัวสะกดและผ่านส่วนงานย่อยการตรวจสอบลำดับของสระและวรรณยุกต์

then { ใส่ Input[i] เป็นวรรณยุกต์ }

else { return Error }

end if

return Ok

4.4.1.1.9 ส่วนงานย่อยของการันต์

ตามปกติตัวสะกดสามารถมีได้เพียง 1 ตัว ดังนั้นตัวการันต์จะอยู่เหนือพยัญชนะในตำแหน่งของตัวสะกดที่ไม่ต้องการจะออกเสียงโดยทั่วไปคำที่มีตัวการันต์เป็นคำยืมมาจากภาษาต่างประเทศ เช่นคำว่า 'เบิร์ด' ซึ่งมีตัวสะกด 2 ตัวแต่จะไม่ออกเสียงในตัว 'ร' กล่าวคือจะอ่านเหมือนกับคำว่า 'เบิด' ดังนั้นเงื่อนไขของส่วนงานย่อยนี้ก็คือถ้า

พบตัวการันต์ก็จะทำการดูว่ามีตัวสะกดอยู่ก่อนแล้วหรือยัง ถ้ายังก็จะถือว่าตำแหน่งของการันต์อยู่ผิดตำแหน่งทำให้คำนั้นผิดไวยากรณ์ แต่ถ้ามีตัวสะกดอยู่แล้ว ก็ให้ตัดตัวสะกดตัวนั้นทิ้งไป สำหรับขั้นตอนวิธีเป็นดังนี้

```
if มีตัวสะกด
    then { ให้ตัดตัวสะกดทิ้ง }
    else { return Error }
end if
return Ok
```

4.4.1.1.10 อักษร '.'

กรณีนี้เป็นอักษรพิเศษที่จะไม่มีการดำเนินการใดๆกับอักษรนี้ และไม่ถือว่าผิดไวยากรณ์ด้วย ดังนั้นให้ข้ามข้อมูลเข้าตัวนี้ไปยังข้อมูลตัวถัดไป

4.4.1.1.11 ส่วนงานย่อยการตรวจสอบลำดับของสระและวรรณยุกต์

ส่วนงานย่อยนี้ใช้สำหรับตรวจสอบลำดับของสระและวรรณยุกต์ กล่าวคือในแต่ละเสียงสระเช่น สระเอีย สระอะ สระอัวะ ฯลฯ อาจจะประกอบได้ด้วยรูปเขียนของสระย่อยหลายตัวประกอบกันหรือมีเพียงรูปเขียนของสระย่อยตัวเดียว ดังนั้นเมื่อเสียงสระใดมีรูปเขียนของสระย่อยๆหลายตัวประกอบกันก็จะต้องมีลำดับของการประกอบรวมกัน ซึ่งถ้าลำดับผิดก็อาจจะหมายถึงเสียงสระอื่นหรือไม่ก็เขียนพยางค์นั้นผิดไวยากรณ์ไทย และนอกจากนั้นตำแหน่งของวรรณยุกต์ที่ปรากฏในพยางค์ก็จะมีตำแหน่งที่สัมพันธ์กันกับรูปเขียนของสระที่แน่นอน โดยที่เสียงสระต่างๆมีอยู่เพียง 31 แบบซึ่งไม่ถือว่ามาก ดังนั้นจึงจะใช้วิธีการบันทึกลักษณะของลำดับสระย่อยกับวรรณยุกต์ของทุกรูปเขียนสระไว้ในตารางที่ชื่อว่าตารางการตรวจลำดับของสระและวรรณยุกต์เพื่อเอาไว้ตรวจสอบว่าสระย่อยหรือวรรณยุกต์ที่ได้นั้นถูกหลักไวยากรณ์ ข้อมูลของตารางอยู่ในตารางที่ 4.1 นี้

◌ะ	◌า	◌ิ◌	◌ี◌	◌ึ◌	◌ื◌	◌ั◌	◌ุ◌
◌ู◌	เ◌	แ◌	โ◌	ใ◌	ไ◌	◌็◌	◌์◌
◌อ	◌ว	◌ัว	เ◌ะ	แ◌ะ	โ◌ะ	เ◌ี◌	เ◌ย
เ◌อะ	เ◌็	◌็อ	เ◌าะ	เ◌ียะ	เ◌ีอะ	ไ◌ย	

ตารางที่ 4.1 แสดงข้อมูลในตารางตรวจลำดับของสระและวรรณยุกต์

ขั้นตอนการตรวจลำดับของสระและวรรณยุกต์นี้จะมีการสร้างตัวแปรทดสอบที่มีขนาด 5 ไบต์ไว้สำหรับเก็บสระและวรรณยุกต์ตั้งแต่เริ่มแยกหน่วยของแต่ละพยางค์ เมื่อมีการเพิ่มสระหรือวรรณยุกต์เข้าในหน่วยของทั้งสองนี้ จะต้องเพิ่มเข้าในตัวแปรทดสอบด้วย จากนั้นก็เอาตัวแปรทดสอบนี้ไปเปรียบเทียบแบบตำแหน่งต่อตำแหน่งกับตารางตรวจลำดับของสระและวรรณยุกต์ โดยจะเปรียบเทียบเท่าที่จำนวนข้อมูลในตัวแปรทดสอบนั้นมีอยู่เท่านั้นว่าสามารถที่จะเป็นส่วนประกอบส่วนหนึ่งของสระต่างๆที่ถูกบันทึกไว้ในตารางตรวจลำดับของสระและวรรณยุกต์หรือไม่ ถ้าไม่พบก็แสดงว่าสระและวรรณยุกต์ที่ตรวจสอบนั้นมีลำดับไม่ถูกต้อง ตารางที่ 4.1 จะใช้อักษร 'T' แทนวรรณยุกต์ทั้ง 5 รูปซึ่งรวมทั้งรูปวรรณยุกต์สามัญซึ่งจะไม่มีรูปเขียน ดังนั้นเมื่อเปรียบเทียบไล่ตำแหน่งมาจนพบกับอักษร 'T' นี้ในตาราง ก็จะหมายความว่าที่ตำแหน่งนี้ของตัวแปรทดสอบจะมีวรรณยุกต์หรือไม่มีก็ได้ ถ้ามีวรรณยุกต์ก็ให้เลื่อนตำแหน่งเปรียบเทียบไปยังตำแหน่งถัดไปทั้งตำแหน่งของตารางและของตัวแปรทดสอบ แต่ถ้าไม่มีก็ให้เลื่อนเฉพาะตำแหน่งของตารางเท่านั้น

4.4.1.2 ส่วนงานย่อยตรวจสอบและแก้ไข

1. พยางค์ที่มีอักษรเพียง 2 ตัวและสามารถที่จะเป็นคำควบกล้ำหรืออักษรนำได้เช่นคำว่า*กร,หน* ซึ่งจะทำให้ได้ผลลัพธ์คือ

พยัญชนะต้น	*กร,หน*
สระ	ไม่มี,ไม่มี
ตัวสะกด	ไม่มี,ไม่มี
วรรณยุกต์	ไม่มี,ไม่มี

วิธีแก้ก็คือ จะใช้การตรวจดูส่วนต่างๆที่ได้หลังจากผ่านขั้นตอนวิธีว่าเป็นดังกรณีนี้หรือไม่ถ้าเป็นก็จะทำการเปลี่ยนให้ถูกต้องคือถ้าลงท้ายด้วย '*ร*' จะได้สระเป็นสระออ สำหรับตัวอักษรอื่นจะได้สระเป็นสระโอะ จากนั้นให้ย้ายพยัญชนะตัวสุดท้ายนั้นมาเป็นตัวสะกด

2. พยางค์ที่มีสระเป็น '*รร*' และไม่มีตัวสะกด พยางค์ในพวกนี้จะออกเสียงแบบมี 'น' เป็นตัวสะกดแต่ลดรูปเอาไว้ ดังนั้นทางแก้ก็คือจะต้องทำการเติม 'น' เข้าเป็นตัวสะกด

4.4.2 ส่วนการหาสัทอักษรจากหน่วยต่างๆที่ได้จากการแยกหน่วยพยางค์ไทย

เมื่อได้หน่วยต่างๆของพยางค์ไทยคำเดี่ยวแล้ว ขั้นตอนต่อไปก็จะทำการแปลงหน่วยต่างๆเหล่านั้นให้เป็นสัทอักษรสากล โดยการใช้ตารางที่เก็บค่าสัทอักษรสากลเหล่านั้นไว้ให้ใช้แปลงตามหน่วยของพยางค์ที่แยกมาได้ ดังในตารางที่ 4.2 ถึง 4.4(Luksaneeyanawin, 1993)ต่อไปนี้

พยัญชนะ	สัทอักษร	พยัญชนะ	สัทอักษร	พยัญชนะ	สัทอักษร
ป	p	พ,ภ,ผ	ph	บ	b
ม	m	ฟ,ฝ	f	ว	w
ต,ฏ	t	ท , ธ , ฑ , ฒ , ฌ,ฐ	th	ด,ฎ,ฒ	d
น,ณ	m	ซ,ศ,ษ,ส	s	ร,ฤ	r
ล,ฬ	l	จ	c	ช,ฌ,ฉ	ch
ย,ญ	j	ก	k	ค,ฆ,ข	kh
ง	ng	อ	a	ฮ,ห	h
ปร,ปฤ	pr	ตร,ตฤ	tr	กร,กฤ	kr
ปล	pl	กล	kl	กว	kw
พร,พฤ,ภร,ภฤ	phr	ทร,ทฤ	thr	คร,ขร	khr
พล,ผล	phl	คล,ขล	khl	คว,ขว	khw
สร,ศร,ทร**	s	หง	ng	หม	m
หร	r	หว	w	หญ,หย	j
หน	n	หล	l	อย	j
จร	c				

* ปกติเสียง 'ฑ' จะออกเสียงเป็น 'th' เช่นคำว่า มณฑา แต่จะมีบางคำจะออกเสียงเป็น 'd' เช่นคำว่า มณฑป

** ปกติเสียง 'ทร' จะออกเสียงเป็น 'thr' เช่นคำว่า โทร แต่จะมีบางคำจะออกเสียงเป็น 's' เช่นคำว่า โทรม ซึ่งมีอยู่ไม่กี่คำ

ตารางที่ 4.2 แสดงสัทอักษรของพยัญชนะต้นต่างๆ

พยัญชนะ	สัทอักษร	พยัญชนะ	สัทอักษร	พยัญชนะ	สัทอักษร
ด,ฎ,ฎ,ต,ท,ธ, ฒ,ฑ,ฒ,ฐ,จ,ว, ช,ศ,ษ,ส	t	บ,พ,ภ,ฟ,ป	p	ก,ต,ฎ,ข	k
น,ณ,ร,ล,ฬ,ญ	n	ม	m	ง	ng
ว	w	ย	j		

ตารางที่ 4.3 แสดงสัทอักษรของตัวสะกดต่างๆ

สระ	สัทอักษร	สระ	สัทอักษร	สระ	สัทอักษร
อิ	i	อี	ii	อึ	v
อื	vv	อุ	u	อู	uu
เอะ	e	เอ	ee	เออะ,เอิ_*	q
เออ,เอิ_	qq	โอะ	o	โอ	oo
แอะ	x	แอ	xx	อะ,อั,รร	a
อา	aa	เอาะ	@	_อ	@@
เอียะ	ia	เอีย	iia	เอือะ	va
เอือ	vva	อ้วะ	ua	อัว	uua
อำ	am	ไอ,ใอ	aj	เอา	aw
เอ็	e	อ็	@@	อ็อ_	@
เ_ย,เ_อ	qq	ไ_ย	aj	_ว	uua

* ปกติสระ 'เอิ' จะออกเสียงเป็น 'qq' เช่นคำว่า เกิน แต่จะมีคำว่ายกเว้นที่จะออกเสียงเป็น 'q' คือคำว่า เงิน

ตารางที่ 4.4 แสดงสัทอักษรของสระต่างๆ

สำหรับสัทอักษรของวรรณยุกต์นั้นมีเพียงแค่ 5 รูปคือ 0-4 ซึ่งจะแทนเสียงสามัญ,เสียงเอก,เสียงโท,เสียงตรี,เสียงจัตวา ตามลำดับซึ่งแม้จะมีไม่มากแต่เป็นส่วนที่ยุ่งยากที่สุด เนื่องจากพยางค์ในภาษาไทยนั้นมักมีรูปวรรณยุกต์ไม่ตรงกับเสียงทำให้ไม่สามารถที่จะแปลงวรรณยุกต์ที่แยกมาได้โดยการเทียบกับตารางได้โดยตรง แต่สามารถที่จะใช้หลักไตรยางค์และหลักคำเป็น-คำตาย(สมชาย, 2526) ช่วยในการแยกแยะ จากหลักดังกล่าวสามารถสรุปได้ว่า

จะใช้หลักไตรยางค์ในการแยกกลุ่มของพยัญชนะต้นออกจากกันเป็นกลุ่มๆ จากนั้นในแต่ละกลุ่มของพยัญชนะต้นจะสามารถแบ่งย่อยได้อีกโดยจะใช้กฎของคำเป็น-คำตายในการแยกกลุ่มย่อย เมื่อได้กลุ่มย่อยในแต่ละกลุ่มของพยัญชนะแล้วก็จะใช้รูปวรรณยุกต์ที่ปรากฏในพยางค์นั้นมาเป็นตัวพิจารณาว่าจะมีเสียงวรรณยุกต์อย่างไรต่อไป

ดังนั้นเพื่อให้ขั้นตอนในการแปลงหน่วยวรรณยุกต์ให้เป็นสัทอักษรนั้นสามารถพัฒนาเป็นโปรแกรมได้ง่าย ก็จะต้องแยกให้ได้ว่าพยัญชนะที่มีนั้นจัดอยู่ในกลุ่มใด และสระที่ได้มานั้นเป็นสระเสียงสั้นหรือเสียงยาว และสุดท้ายตัวสะกดนั้นเป็นตัวสะกดในกลุ่มใด โดยการแทนหน่วยต่างๆเหล่านั้นด้วยค่าของตัวเลขที่แตกต่างกัน เพื่อที่จะได้ใช้เป็นตัวชี้ในการแยกกลุ่มต่อไปในโปรแกรม การที่จะเปลี่ยนหน่วยต่างๆเหล่านั้นเป็นตัวชี้จะทำได้โดยการเอาหน่วยต่างๆไปเทียบกับตารางคุณสมบัติเพื่อที่จะหาว่าหน่วยต่างๆนั้นมีค่าตัวชี้อย่างไรดังต่อไปนี้

การแบ่งกลุ่มของพยัญชนะต้นนั้นจากกฎของไตรยางค์จะพบว่าสามารถแบ่งกลุ่มได้ 3 กลุ่มใหญ่ๆและในกลุ่มของอักษรต่ำจะมีการแบ่งกลุ่มของอักษรต่ำออกเป็น 3 กลุ่มย่อยดังต่อไปนี้

1. อักษรกลาง พยัญชนะในกลุ่มนี้มี ก,จ,ฎ,ฏ,ด,ต,บ,ป,อ และให้ค่าคุณสมบัติบัติของอักษรในกลุ่มนี้เป็น 0
2. อักษรสูง พยัญชนะในกลุ่มนี้มี ข,ฃ,ฉ,ฐ,ถ,ผ,ฝ,ส,ศ,ษ,ห และให้ค่าคุณสมบัติบัติของอักษรในกลุ่มนี้เป็น 10
3. อักษรต่ำ
   - 3.1 อักษรต่ำคู่กับอักษรสูง พยัญชนะในกลุ่มนี้มี ค,ต,ฆ,ช,ฌ,ซ,ฑ,ฒ,ธ,ท,พ,ภ,ฟ,ฮ และให้ค่าคุณสมบัติในอักษรกลุ่มนี้เป็น 20
   - 3.2 อักษรต่ำเดี่ยวที่มีอักษรกลางนำ พยัญชนะในกลุ่มนี้มี ย เพียงตัวเดียวและใช้อักษรกลางคือ อ นำหน้า และให้ค่าคุณสมบัติในอักษรกลุ่มนี้เป็น 30
   - 3.3 อักษรต่ำเดี่ยวที่มีอักษรสูงนำ พยัญชนะในกลุ่มนี้มี ง,ญ,ณ,น,ม,ร,ล,ว,ฬ โดยจะมี

อักษรสูงคือ ห นำ และให้ค่าคุณสมบัติในอักษรกลุ่มนี้เป็น 40

แต่เนื่องจากในกลุ่มอักษรต่ำเดี่ยวที่มีอักษรนำทั้งหลายนั้น จะมีเสียงวรรณยุกต์บางเสียงที่ไม่จำเป็นต้องมีอักษรนำช่วยในการออกเสียงปรากฏร่วมอยู่ด้วย ดังนั้นเพื่อไม่ให้เกิดปัญหากับการวิเคราะห์ทางซอฟต์แวร์จึงต้องแบ่งเพิ่มอีก 1 กลุ่มคือ

3.4 อักษรต่ำเดี่ยวไม่มีอักษรนำ พยัญชนะในกลุ่มนี้ได้แก่พยัญชนะในข้อที่ 3.2,3.3 และให้ค่าคุณสมบัติในอักษรกลุ่มนี้เป็น 50

และจากกฎของคำเป็นคำตายจึงแบ่งกลุ่มของสระออกได้เป็น 2 ส่วน คือจะบอกว่าเป็นสระเสียงสั้นหรือเสียงยาวดังต่อไปนี้

1. สระเสียงสั้น และให้ค่าคุณสมบัติในสระนี้เป็น 0
2. สระเสียงยาว และให้ค่าคุณสมบัติในสระนี้เป็น 2

สำหรับตารางที่จะแสดงว่าสระใดเป็นเสียงสั้นหรือสระเสียงยาวนั้นจะอยู่ในตารางที่ 4.5

สระ	ค่าคุณสมบัติ	สระ	ค่าคุณสมบัติ	สระ	ค่าคุณสมบัติ
อิ	0	อี	2	อึ	0
อึ	2	อุ	0	อู	2
เอะ	0	เอ	2	เออะ, เอิ	0
เออ	2	โอะ	0	โอ	2
แอะ	0	แอ	2	อะ,อั, รร	0
อา	2	เอาะ	0	_อ	2
เอียะ	0	เอีย	0	เออะ	0
เอือ	2	อ้วะ	0	อัว	2
อำ	2	ไอ,ใอ	2	เอา	2
เอ็	0	อ็	2	อ็อ_	2
เ_ย	2	ไ_ย	2	_ว	2

ตารางที่ 4.5 แสดงค่าคุณสมบัติของสระต่างๆ

และคุณสมบัติของตัวสะกดก็จะใช้บอกว่าเป็นตัวสะกดอยู่ในกลุ่มใด โดยการจัดเป็นกลุ่มได้ 3 กลุ่ม ดังต่อไปนี้

1. ไม่มีตัวสะกด ให้ค่าคุณสมบัติเป็น 0
2. กลุ่มของตัวสะกดแม่กง,กน,กม,เกย,เกอวและให้ค่าคุณสมบัติเป็น 1
3. กลุ่มของตัวสะกดแม่กด,กบ,กก และให้ค่าคุณสมบัติเป็น 2

สุดท้ายคือค่าคุณสมบัติของวรรณยุกต์ซึ่งจะเป็น 0-4 ตามแต่รูปที่ได้มา เมื่อได้คุณสมบัติของทั้งหมดแล้วจะนำมาใช้เป็นตัวชี้ว่าพยางค์นั้นควรจะผันเสียงวรรณยุกต์อย่างไร โดยมีลำดับการใช้ตัวชี้ดังนี้

1. ใช้คุณสมบัติของพยัญชนะเป็นตัวชี้ในลำดับแรกเพื่อแยกพยัญชนะต้นออกจากกัน

2. หลังจากทราบกลุ่มของพยัญชนะต้นแล้ว ต่อไปก็จะใช้คุณสมบัติของคำเป็นคำตายเป็นตัวชี้ ตารางที่ 4.6 เป็นการแสดงวิธีหาคำเป็นคำตายโดยใช้คุณสมบัติของ

สระและตัวสะกด ซึ่งจะเห็นว่าเมื่อรวมค่าคุณสมบัติของตัวสะกดเข้ากับสระจะได้ค่าของคุณสมบัติรวมที่สามารใช้เป็นตัวชี้เพื่อแยกแยะว่าพยางค์นั้นเป็นคำเป็นหรือคำตาย

คุณสมบัติของตัวสะกด	คุณสมบัติของสระ	ผลรวมของคุณสมบัติของตัวสะกดและสระเพื่อใช้เป็นตัวชี้ในโปรแกรม (att_sum)	ผลที่ได้
0	0	0	คำตายรัสสะ
0	2	2	คำเป็น
1	0	1	คำตายรัสสะ
1	2	3	คำตายทีฆะ
2	0	2	คำเป็น
2	2	4	คำเป็น

ตารางที่ 4.6 แสดงการหาคำเป็นคำตายเพื่อเป็นตัวชี้ในโปรแกรม

3. ตัวชี้สุดท้ายจะใช้คุณสมบัติของหน่วยวรรณยุกต์เป็นตัวชี้

สำหรับขั้นตอนวิธีการหาสัทอักษรของวรรณยุกต์ที่จะอธิบายต่อไปนี้เพื่อให้ง่ายต่อความเข้าใจจะใช้ชื่อเรียกกลุ่มของพยัญชนะแทนที่จะใช้เป็นตัวชี้ที่เป็นตัวเลข และจะใช้คำเป็นคำตายแทนค่าตัวชี้ดังในตารางที่ 4.6 เป็นดังนี้

```
switch type(พยัญชนะต้น):
{
        case กลาง:
                switch (att_sum)
                {
                        case คำเป็น : วรรณยุกต์ = ตามที่ได้รับมา;
                        case คำตาย:  swicth (วรรณยุกต์)
                                        {
                                                case สามัญ: วรรณยุกต์ = เอก
                                                case เอก:    Error
                                                default :    ตามรูปที่ได้รับมา
```

```
                                        }// จบส่วนของอักษรกลางคำตาย
                    }// จบส่วนของอักษรกลาง
case สูง:
                    switch (att_sum)
                    {
                            case คำเป็น : switch (วรรณยุกต์)
                                        {
                                                case สามัญ : วรรณยุกต์ = จัตวา
                                                case ตรี,จัตวา: Error
                                                default: วรรณยุกต์คงเดิม
                                        } // จบอักษรสูงคำเป็น
                            case คำตาย:  swicth (วรรณยุกต์)
                                        {
                                                case สามัญ: วรรณยุกต์ = เอก
                                                case โท :   วรรณยุกต์คงเดิม
                                                default :   Error
                                        }// จบส่วนของอักษรสูงคำตาย
                    }// จบส่วนของอักษรสูง
case ต่ำคู่ :
                    switch (att_sum)
                    {
                            case คำเป็น : switch (วรรณยุกต์)
                                        {
                                                case สามัญ : วรรณยุกต์ = สามัญ
                                                case เอก,โท: วรรณยุกต์ = เพิ่มขึ้น 1 เสียง
                                                default: Error
                                        } // จบอักษรกลางคำเป็น
                            case คำตาย-รัสสะ:  swicth (วรรณยุกต์)
                                        {
                                                case สามัญ: วรรณยุกต์ = ตรี
                                                case เอก :  วรรณยุกต์ = โท
                                                case จัตวา : วรรณยุกต์คงเดิม
                                                default :   Error
                                        }// จบส่วนของอักษรต่ำคู่คำตาย รัสสะ
                                case คำตาย-ทีฆะ:
```

```
                                        switch (วรรณยุกต์)
                                        {
                                                case สามัญ : วรรณยุกต์ = โท
                                                case โท     : วรรณยุกต์ = ตรี
                                                case จัตวา  : วรรณยุกต์ = จัตวา
                                                default: Error
                                        } // จบอักษรต่ำคำคู่คำตายที่มะ
                        }// จบส่วนอักษรต่ำคู่
case ต่ำเดี่ยวกลางนำ :
                switch (att_sum)
                        {
                                case คำเป็น : switch (วรรณยุกต์)
                                                {
                                                        case  สามัญ,ตรี : Error
                                                        default: วรรณยุกต์คงเดิม
                                                } // จบอักษรต่ำเดี่ยวกลางนำคำเป็น
                                case คำตาย:  swicth (วรรณยุกต์)
                                                {
                                                        case สามัญ: วรรณยุกต์ = เอก
                                                        case เอก :   Error
                                                        default :    วรรณยุกต์คงเดิม
                                                }// จบส่วนของคำตายอักษรต่ำเดี่ยวกลางนำ
                                }//จบส่วนของ อักษรต่ำเดี่ยวกลางนำ

case ต่ำเดียวสูงนำ :
                switch (att_sum)
                        {
                                case คำเป็น : switch (วรรณยุกต์)
                                                {
                                                        case สามัญ: วรรณยุกต์ = จัตวา
                                                        case  ตรี : Error
                                                        default: วรรณยุกต์คงเดิม
                                                } // จบอักษรต่ำเดี่ยวสูงนำคำเป็น
                                case คำตาย:  swicth (วรรณยุกต์)
                                                {
```

```
                                        case สามัญ: วรรณยุกต์ = เอก
                                        case เอก,ตรี :   Error
                                        default :     วรรณยุกต์คงเดิม
                                    }// จบส่วนของคำตายอักษรต่ำเดี่ยวสูงนำ
                            }//จบส่วนของอักษรต่ำเดี่ยวสูงนำ
        case ต่ำเดี่ยวไม่มีอักษรนำ :
                switch (att_sum)
                        {
                            case คำเป็น : switch (วรรณยุกต์)
                                        {
                                            case  สามัญ : วรรณยุกต์ = สามัญ
                                            case เอก,โท : เพิ่มเสียง 1 ขั้น
                                            default: Error
                                        } // จบอักษรต่ำเดี่ยวไม่มีอะไรนำ

                            case คำตาย-รัสสะ:  swicth (วรรณยุกต์)
                                        {
                                            case สามัญ: วรรณยุกต์ = ตรี
                                            case เอก :   วรรณยุกต์ = โท
                                            default :     Error
                                        }// จบส่วนของคำตาย รัสสะต่ำเดี่ยวไม่มีนำ
                                case คำตาย-ทีฆะ:
                                        switch (วรรณยุกต์)
                                        {
                                            case สามัญ : วรรณยุกต์ = โท
                                            case โท      : วรรณยุกต์ = ตรี
                                            default: Error
                                        } // จบอักษรต่ำคำเดี่ยวคำตายทีฆะ
                            }//จบส่วนของอักษรต่ำเดี่ยวไม่นำ
} // จบขั้นตอนวิธี
```

4.5 ส่วนงานย่อยของการแปลงสตริงของสัทอักษรสากลไปเป็นชุดของรหัสหน่วยคู่เสียง

ในส่วนงานย่อยของการแปลงสัทอักษรสากลไปเป็นหน่วยคู่เสียงนี้ ข้อมูลที่เข้ามาจะเป็นสตริงของข้อมูลที่ประกอบด้วยส่วนของคำสั่งพิเศษร่วมกับสัทอักษรสากลดังรูปที่ 4.8 นี้

k	a	a	n	0	!	1	*	*	n	a	a	2	

รูปที่ 4.8 แสดงตัวอย่างของสตริงข้อมูลเข้าที่จะนำมาแปลงเป็นหน่วยคู่เสียง

จากรูปที่ 4.8 จะเห็นว่าข้อมูลเข้านี้ไม่มีช่องว่างคั่นระหว่างพยางค์แต่จะใช้ตัวเลขที่แสดงเสียงวรรณยุกต์เป็นตัวคั่นแทน และจะมีข้อมูลที่เป็นคำสั่งพิเศษสำหรับในการปรับความเร็วของพยางค์ที่จะพูดในที่นี้จะเห็นว่ามี '!1' เป็นคำสั่งปรับความเร็วเป็นระดับที่สอง และสัญลักษณ์ '**' ไว้สำหรับการบอกให้เงียบก่อนที่ออกเสียงคำว่า 'หน้า' ในส่วนงานย่อยนี้จะมีการทำงานที่สำคัญอยู่ 2 ส่วนคือ

1. ส่วนของการแยกสัทอักษรสากลออกเป็นชื่อของหน่วยคู่เสียงจำนวน 2 หน่วย ซึ่งชื่อของหน่วยคู่เสียงนี้จะใช้สัญลักษณ์ทางสัทอักษรสากล

2. ส่วนของการหาว่าชื่อของหน่วยคู่เสียงที่แยกได้นั้นมีอยู่ในส่วนที่เก็บหน่วยความจำเสียงหรือไม่ และถ้ามีก็จะให้ผลลัพธ์เป็นรหัสหน่วยคู่เสียง

4.5.1 ส่วนงานย่อยการแยกสัทอักษรสากลออกเป็นชื่อของหน่วยคู่เสียง

หน้าที่สำคัญของส่วนงานย่อยนี้คือจะทำการแยกสัทอักษรสากลที่ใช้แทนพยางค์ให้ออกเป็น 2 ส่วน ซึ่งจะเป็นชื่อของหน่วยคู่เสียงต้นและหน่วยคู่เสียงท้ายสำหรับใช้ในการออกเสียงพยางค์ใดๆ โดยจะมีความสัมพันธ์ระหว่างหน่วยคู่เสียงทั้งสองด้วยเสียงสระดังนี้ คือ

$$XyZ = Xy_1 + yZ$$

โดยที่ X จะแทนสัทอักษรของพยัญชนะต้น

y จะแทนสัทอักษรของสระ

y_1 จะแทนบางส่วนของสัทอักษรของสระ

Z จะแทนสัทอักษรของตัวสะกดและวรรณยุกต์

สำหรับลักษณะของสัทอักษรของหน่วยคู่เสียงต้นที่จะใช้คู่กับสระต่างๆเป็นดังตารางที่ 4.7

สัทอักษรของสระต่างๆ (y)	ลักษณะของสัทอักษรของหน่วยคู่เสียงต้นที่ใช้กับสระต่างๆตามลำดับ (Xy_1)
i , i i , v , v v , u , u u , e , e e , q , q q , o,oo,x,xx,a,aa,@,@@	Xi,Xi,Xv,Xv,Xu,Xu,Xe,Xe,Xq,Xq, Xo,Xo,Xx,X,Xa,Xa,X@,X@
ia,iia,va,vva,ua,uua	Xia,Xia,Xva,Xva,Xua,Xua

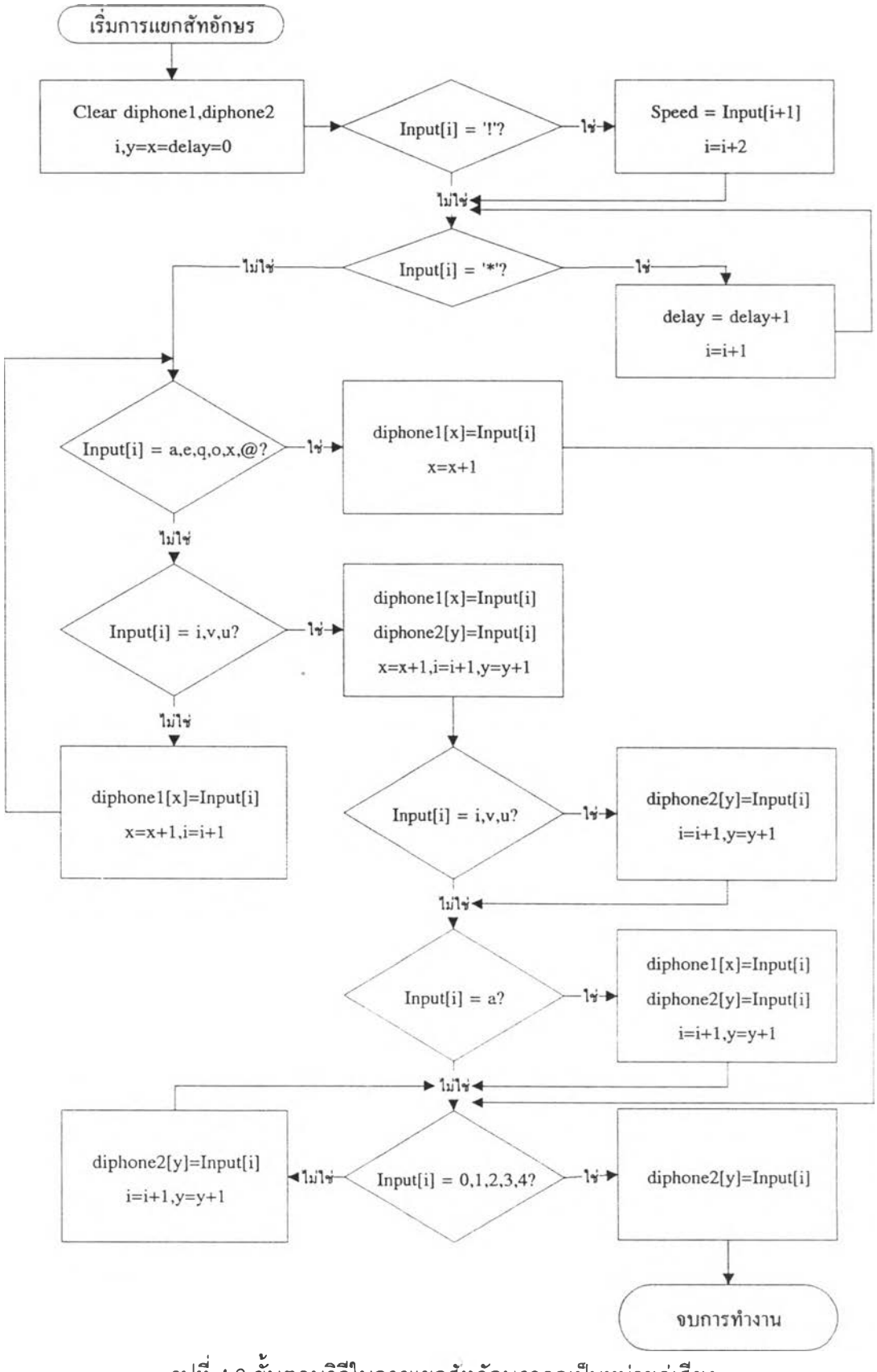
ตารางที่ 4.7 แสดงสัทอักษรของหน่วยคู่เสียงต้นที่สัมพันธ์กันกับสัทอักษรของสระต่างๆ

ในการพัฒนาซอฟต์แวร์ของส่วนงานย่อยนี้จะอาศัยข้อมูลจากตารางที่ 4.7 ในการแยกหน่วยคู่เสียงทั้งสองออกจากกัน โดยจะมีการจัดแบ่งกลุ่มสัทอักษรของสระที่ดูแล้วว่ากำกวมออกจากสระที่ไม่กำกวม คำว่าสัทอักษรของสระที่กำกวมในที่หมายถึงว่าตัวอักษรของสัทอักษรตัวแรกสามารถที่จะเป็นอักษรของสัทอักษรตัวแรกของสระจากทั้งสองกลุ่มในตารางที่ 4.7 ได้ ซึ่งก็คือสัทอักษร i , u ,v ดังนั้นเมื่อจัดแบ่งกลุ่มใหม่แล้วจะได้ผลเป็นดังในตารางที่ 4.8

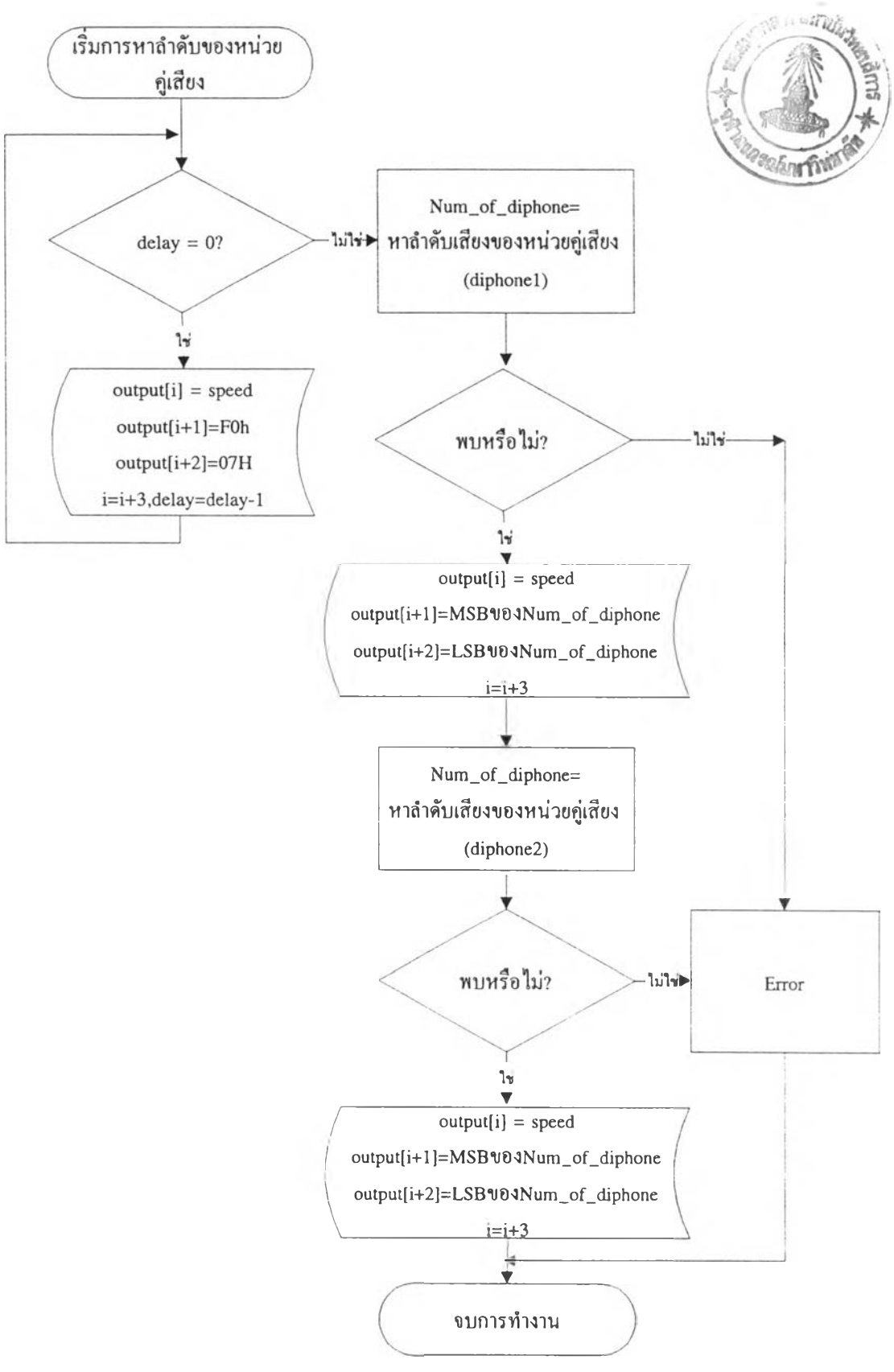
สัทอักษรของสระต่างๆ (y)	ลักษณะของสัทอักษรของส่วนคู่เสียงต้นที่ใช้กับสระต่างๆ ตามลำดับ (Xy_1)
e,ee,q,qq,o,oo,x,xx,a,aa,@,@@	Xe,Xe,Xq,Xq,Xo,Xo,Xx,Xx,Xa,Xa,X@,X@
ia,iia,va,vva,ua,uua	Xia,Xia,Xva,Xva,Xua,Xua
i,ii,v,vv,u,uu	Xi,Xi,Xv,Xv,Xu,Xu

ตารางที่ 4.8 แสดงการจัดแบ่งกลุ่มของสัทอักษรของสระที่สัมพันธ์กันกับสัทอักษรของหน่วยคู่เสียงต้น

สำหรับขั้นตอนวิธีของการแยกจะเป็นดังในรูปที่ 4.9 ซึ่งส่วนคำสั่งพิเศษต่างๆ จะถูกแยกออกก่อนที่จะเข้าสู่การแยกหน่วยคู่เสียงคือ คำสั่งความเร็วจะใช้ตัวแปร speed เอาไว้เก็บค่าความเร็วของพยางค์ต่างๆ โดยให้ค่าเริ่มต้นเมื่อเปิดเครื่องมีความเร็วเป็นระดับปกติ และจะยังคงใช้ค่าความเร็วนี้ต่อไปจนกว่าจะมีคำสั่งเปลี่ยนความเร็ว เมื่อเปลี่ยนความเร็วแล้วพยางค์ต่อไปนับแต่เปลี่ยนความเร็วก็จะใช้ความเร็วใหม่นี้ จนกระทั่งพบคำสั่งเปลี่ยนความเร็วอีกครั้ง



รูปที่ 4.9 ขั้นตอนวิธีในการแยกสัทอักษรออกเป็นหน่วยคู่เสียง



รูปที่ 4.10 แสดงการทำงานของส่วนงานย่อยการหาลำดับที่ของหน่วยคู่เสียง

สำหรับคำสั่งเงียบจะใช้ตัวแปร delay เอาไว้นับว่ามีคำสั่งให้เงียบอยู่กี่หน่วยโดยหนึ่งหน่วยจะทำการเงียบประมาณ 100 มิลลิวินาทีรายละเอียดจะอยู่ในส่วนการหาลำดับหน่วยคู่เสียงต่อไป

4.5.2 ส่วนงานย่อยการหาลำดับที่ของหน่วยคู่เสียงต่าง ๆ

ส่วนงานย่อยนี้มีแผนผังการทำงานดังแสดงในรูปที่ 4.10 โดยจะรับข้อมูลจากส่วนงานย่อยของการแยกสัทอักษร ข้อมูลที่จำเป็นได้แก่ ความเร็วของการพูดของพยางค์นั้น จำนวนความเงียบ และชื่อหน่วยคู่เสียงทั้ง 2 หน่วย ส่วนงานย่อยนี้จะให้ผลลัพธ์เป็นชุดของรหัสรหัสหน่วยคู่เสียง ที่มีโครงสร้างดังรูปที่ 4.11

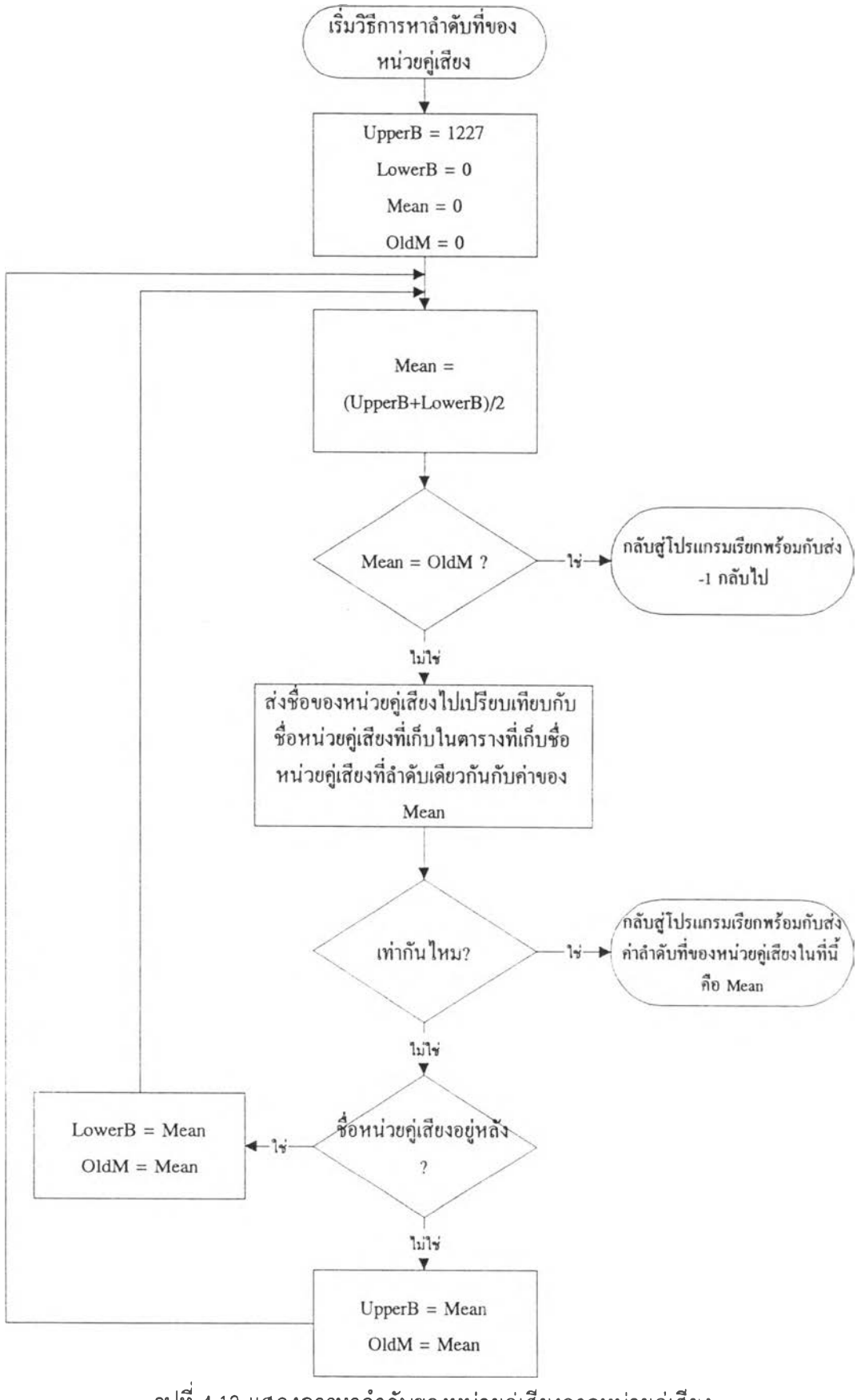
ความเร็วในการออกเสียง (1ไบต์)	ลำดับที่ของหน่วยคู่เสียง (2ไบต์)

รูปที่ 4.11 โครงสร้างของรหัสหน่วยคู่เสียง 1 ชุด

จากรูปที่ 4.11 รหัสหน่วยคู่เสียง 1 ชุดจะมีขนาด 3 ไบต์ โดยที่ไบต์แรกจะเก็บค่าความเร็วในการออกเสียงของพยางค์นั้น 2 ไบต์ถัดมาจะเก็บลำดับที่ของหน่วยคู่เสียงที่ได้รับมา ข้อมูลเหล่านี้จะเรียงกันออกมาเป็นชุดๆติดกัน

ขั้นตอนการหาลำดับที่ของหน่วยคู่เสียงของส่วนงานย่อยนี้จะเริ่มตรวจสอบว่ามีข้อมูลของหน่วยความเงียบหรือไม่ ถ้ามีก็จะทำการเก็บเป็นรหัสหน่วยคู่เสียง 1 ชุดที่มีความเร็วเป็นตามที่ได้รับและมีค่าลำดับเป็น F007h ลงในที่เก็บข้อมูลและจะทำเช่นนี้เท่ากับจำนวนของหน่วยความเงียบที่ได้รับมา ลำดับที่ F007h นี้เป็นลำดับที่ไม่มีหน่วยคู่เสียงใดใช้ ดังนั้นเมื่อส่วนงานย่อยการออกเสียงเจอลำดับหน่วยคู่เสียงนี้ก็จะหมายความว่าให้ทำการเงียบ จากนั้นขั้นตอนต่อไปก็จะเริ่มทำการหาว่าชื่อของหน่วยคู่เสียงที่รับเข้ามานั้นจะมีเก็บอยู่ในหน่วยความจำเสียงหรือไม่ โดยใช้การเปรียบเทียบแบบไบเซ็กชั่น (bisection) ระหว่างชื่อของหน่วยคู่เสียงกับตารางที่เก็บชื่อของหน่วยคู่เสียงทั้งหมด วิธีการหาลำดับของหน่วยคู่เสียงด้วยวิธีไบเซ็กชั่นจะอยู่ในรูปที่ 4.12 และถ้ามีก็จะให้ผลลัพธ์เป็นลำดับของหน่วยคู่เสียงที่อยู่ในส่วนที่เก็บหน่วยความจำเสียง

จากรูปที่ 4.12 ข้อมูลเข้าที่จะรับมาตรวจสอบหาลำดับของหน่วยคู่เสียงนั้นจะเป็นชื่อของหน่วยคู่เสียงแต่ละหน่วย โดยถ้าพบว่ามีหน่วยคู่เสียงในตารางที่เก็บชื่อของหน่วยคู่เสียง ส่วนงานย่อยก็จะส่งค่า Mean ซึ่งเป็นลำดับของหน่วยคู่เสียงของหน่วยคู่เสียงในตารางกลับมา และถ้าไม่พบก็จะส่ง -1 กลับมาให้โปรแกรมเรียก



รูปที่ 4.12 แสดงการหาลำดับของหน่วยคู่เสียงจากหน่วยคู่เสียง

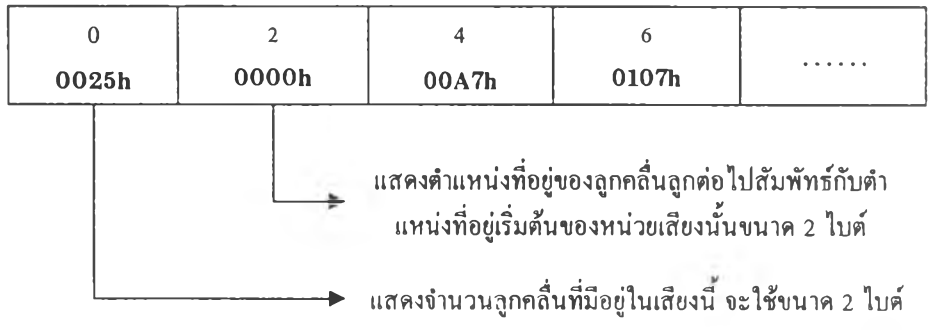
4.6 ส่วนงานย่อยของการออกเสียง

ในส่วนงานย่อยของการออกเสียงนี้จะมีการทำงานที่สำคัญอยู่ 2 ส่วนคือ

1. ส่วนของการควบคุมการออกเสียง
2. ส่วนของคลายสัญญาณเสียงและการออกเสียง

4.6.1 ส่วนของการควบคุมการออกเสียง

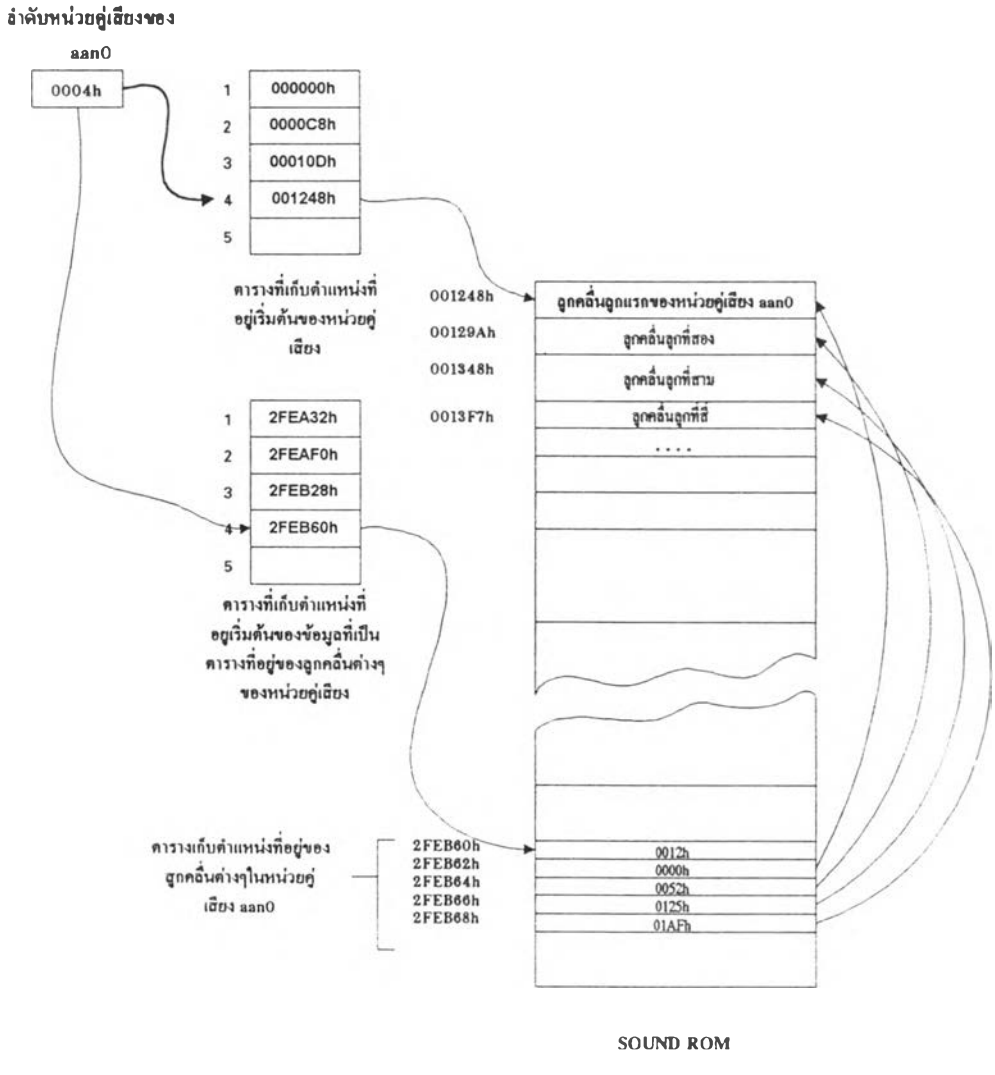
ในส่วนนี้จะทำการหาตำแหน่งที่อยู่ของหน่วยคู่เสียงที่ต้องการให้ออกเสียง โดยอาศัยข้อมูลที่เป็นลำดับที่หน่วยคู่เสียงในรหัสหน่วยคู่เสียงมาเป็นดรรชนีชี้หาตำแหน่งที่อยู่เริ่มต้นของหน่วยคู่เสียงและตำแหน่งที่อยู่เริ่มต้นของตารางที่เก็บตำแหน่งที่อยู่ของลูกคลื่นต่างๆของแต่ละหน่วยคู่เสียงดังในหัวข้อที่ 3.4.4 ของบทที่ 3 ทำให้ทราบถึงตำแหน่งที่อยู่เริ่มต้นของหน่วยคู่เสียงนั้นรวมทั้งตำแหน่งที่อยู่ของลูกคลื่นทุกลูกคลื่นในหน่วยคู่เสียงนั้น ซึ่งโครงสร้างของตารางที่เก็บตำแหน่งที่อยู่ของลูกคลื่นต่างๆของแต่ละหน่วยคู่เสียงจะเป็นดังรูปที่ 4.13



รูปที่ 4.13 แสดงโครงสร้างข้อมูลของตารางที่เก็บตำแหน่งที่อยู่ของแต่ละลูกคลื่นในหนึ่งหน่วยคูเสียง

โครงสร้างของตารางที่เก็บตำแหน่งที่อยู่ของลูกคลื่นต่างๆในหน่วยคู่เสียงใดๆจะประกอบด้วยข้อมูลอยู่ 2 ส่วน คือส่วนแรกที่ไว้ใช้บอกว่ามีลูกคลื่นอยู่กี่ลูกคลื่นในหน่วยคู่เสียงนี้ ขนาดของข้อมูลที่ใช้ส่วนนี้ใช้ 2 ไบต์แรกของตาราง และส่วนที่ 2 จะเป็นแถวลำดับของข้อมูลที่ใช้บอกตำแหน่งที่อยู่ของลูกคลื่นต่างๆในหน่วยคู่เสียงเมื่อคิดสัมพัทธ์กับตำแหน่งที่อยู่เริ่มต้นของหน่วยคู่เสียงนี้ ซึ่งจะใช้ขนาดข้อมูล 2 ไบต์ต่อหนึ่งลูกคลื่นเรียงต่อกันไป ดังนั้นถ้าต้องการที่จะออกเสียงลูกคลื่นใดก็จะต้องทำการโหลดค่าตำแหน่งที่อยู่ของลูกคลื่นที่ต้องการนั้นจากตารางที่เก็บตำแหน่งที่อยู่ของลูกคลื่นแล้วนำมาบวกเข้ากับตำแหน่งที่อยู่เริ่มต้นของหน่วยคู่เสียงนั้น เมื่อได้ตำแหน่งที่อยู่ใหม่แล้วก็จะต้องโหลดตำแหน่งที่อยู่ของหน่วยคู่เสียงใหม่นี้ให้แก่ส่วนควบคุมตำแหน่งที่อยู่ของหน่วยคู่เสียงต่อไป เพื่อเลื่อนตำแหน่งที่อยู่ของหน่วยคู่

เสียงนั้นให้ไปอยู่ตำแหน่งเริ่มต้นของลูกคลื่นใหม่ตามที่ต้องการ จากนั้นก็จะทำการสั่งให้ส่วนของการคลายสัญญาณเสียงและการออกเสียงทำงาน สำหรับรูปแบบการอ้างอิงตำแหน่งที่อยู่ของหน่วยคู่เสียงจะเป็นดังในรูปที่ 4.14 ในส่วนของขนาดของตารางที่เก็บตำแหน่งที่อยู่ของลูกคลื่นของทุกหน่วยคู่เสียงที่ใช้ในงานจริงจะมีขนาด 96 กิโลไบต์



รูปที่ 4.14 แสดงการเข้าถึงตำแหน่งที่อยู่ของหน่วยคู่เสียง aan0 ตามลูกคลื่นต่างๆ

สำหรับการควบคุมความเร็วนั้นจะใช้ข้อมูลในรหัสของหน่วยคู่เสียงที่เก็บความเร็วในการออกเสียงเพื่อสำหรับการเลือกลูกคลื่นที่ต้องการออกเสียง และเนื่องจากเครื่องอ่านคำไทยพยางค์เดี่ยวแบบปรับความเร็วได้นั้น สามารถที่จะปรับความเร็วได้ 4 ระดับ ดังนั้นในการควบคุมการออกเสียงจะพิจารณาการเลือกลูกคลื่นเป็นชุดๆ ชุดละ 4 ลูกคลื่นเท่านั้นดังนี้

1. ความเร็วระดับปกติ ส่วนนี้จะออกเสียงทุกลูกคลื่นเรียงต่อกันไม่มีเว้น

2. ความเร็วระดับหนึ่ง ส่วนนี้จะออกเสียงลูกคลื่น 3 ลูกติดกันและจะเว้นไม่ออกเสียง 1 ลูกคลื่น

3. ความเร็วระดับสอง ส่วนนี้จะออกเสียงลูกคลื่น 2 ลูกติดกันและจะเว้นไม่ออกเสียง 2 ลูกคลื่น

4. ความเร็วระดับสาม ส่วนนี้จะออกเสียงลูกคลื่น 1 ลูกและจะเว้นไม่ออกเสียง 3 ลูกคลื่น

4.6.2 ส่วนของการคลายสัญญาณเสียงและการออกเสียง

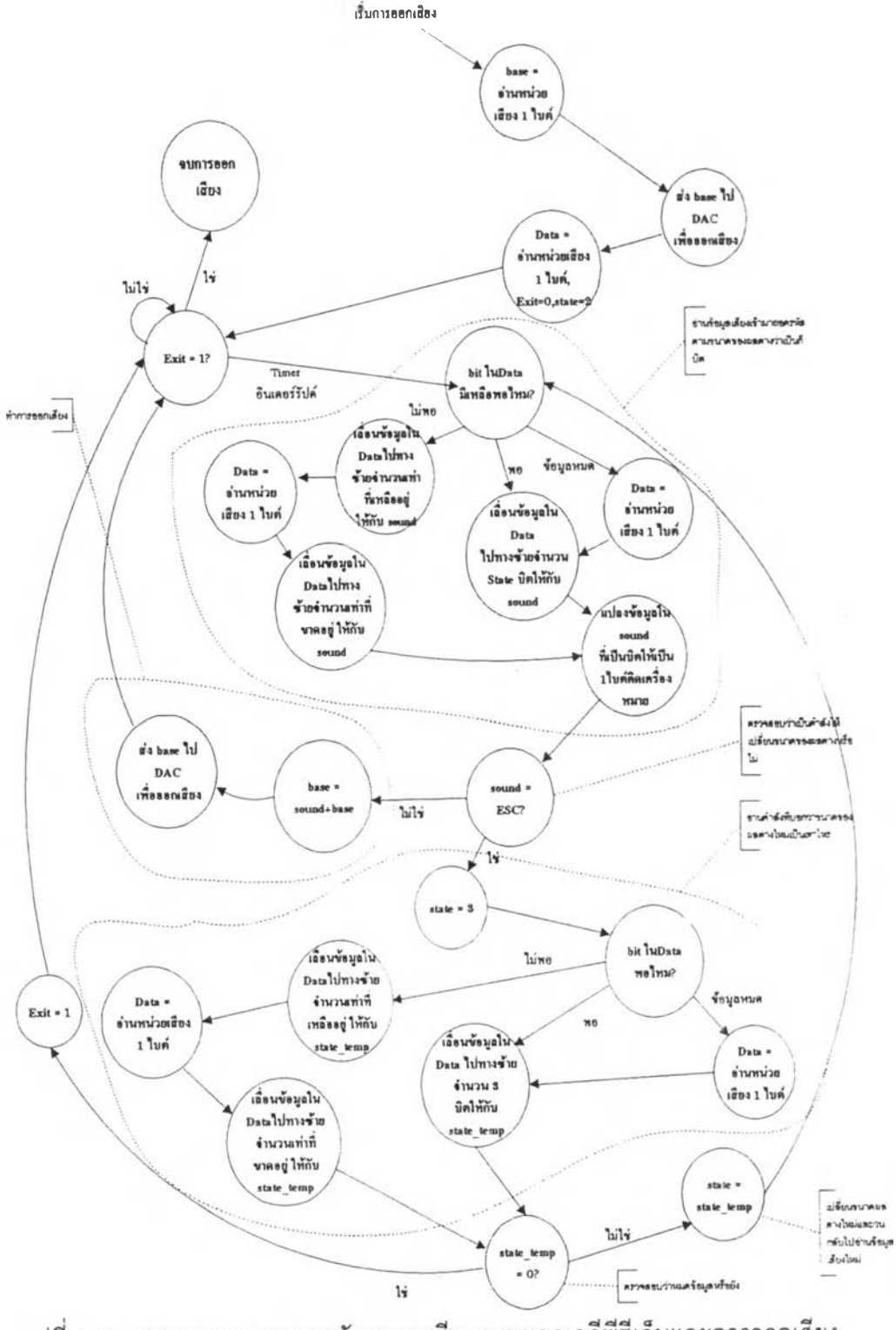
การทำงานหลักในส่วนนี้ก็คือจะต้องคลายสัญญาณเสียงที่ถูกบีบอัดด้วยวิธีแอลเอดีพีซีเอ็ม และส่งข้อมูลของหน่วยคู่เสียงที่คลายได้ไปยังส่วนการแปลงสัญญาณดิจิตอลเป็นแอนะลอกเพื่อให้เกิดเสียง

การคลายสัญญาณเสียงและการออกเสียงนี้จะถือว่าตำแหน่งเริ่มต้นของข้อมูลที่จะนำมาคลายสัญญาณนั้นเป็นตำแหน่งเดียวกับที่ส่วนควบคุมตำแหน่งที่อยู่ของหน่วยคู่เสียงชี้อยู่ จากนั้นก็จะทำการคลายสัญญาณและออกเสียงจนกระทั่งพบคำสั่งว่าหมดข้อมูลจึงเลิกการทำงาน สำหรับลำดับของการทำงานในส่วนนี้เป็นดังในรูปที่ 4.15

จากรูปที่ 4.15 จะมีการใช้ชื่อของตัวแปรต่างๆดังนี้ ตัวแปร state ใช้แทนขนาดของผลต่าง, ตัวแปร base ไว้เก็บค่าสัญญาณเสียงจุดปัจจุบัน, ตัวแปร Data ไว้สำหรับการเก็บข้อมูล 1 ไบต์ที่อ่านได้จากหน่วยเก็บเสียงเพื่อรอการแยกข้อมูลเป็นบิตๆตามจำนวนของ state ที่เป็นอยู่ในการคลายสัญญาณเสียง

ในการคลายสัญญาณเสียงนี้นั้นจะเริ่มเตรียมข้อมูลต่างๆที่จำเป็นให้พร้อม โดยการอ่านข้อมูลเข้ามา 1 ไบต์เพราะเป็นสัญญาณเสียงจุดแรกที่ไม่ถูกการบีบอัดสัญญาณแล้วส่งออกไปแปลงเป็นแอนะลอก จากนั้นก็จะอ่านข้อมูล 1 ไบต์ถัดมาเพื่อเตรียมสำหรับการคลายสัญญาณเสียง ถึงตอนนี้ระบบจะรอการอินเตอร์รัปต์จากตัวนับเวลาที่ตั้งเวลาไว้ให้อินเตอร์รัปต์ทุกๆ 16 กิโลเฮิรตท์ ส่วนงานย่อยการคลายสัญญาณจะเริ่มด้วยการทดสอบจำนวนบิตในตัวแปร Data นั้นมีเหลือพอที่จะทำการเลื่อนข้อมูลของ Data ไปทางซ้ายให้กับตัวแปร sound ตามจำนวนของค่า state ไหม ซึ่งก็จะมีทางที่เป็นไปได้อยู่ 3 ทางก็คือ ข้อมูลหมด ให้ทำการอ่านข้อมูลเข้ามาใหม่ 1 ไบต์และค่อยทำการเลื่อนข้อมูล ทางที่สองคือข้อมูลไม่หมดแต่มีไม่พอ

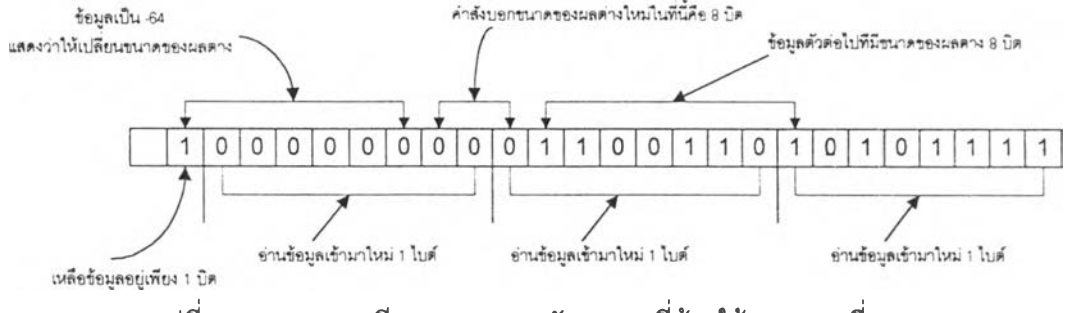
ให้ทำการเลื่อนข้อมูลไปทางซ้ายเท่าที่มีอยู่จนหมด และทำการอ่านข้อมูลใหม่เข้ามาเพื่อเลื่อนข้อมูลต่อจนกระทั่งครบ



รูปที่ 4.15 แสดงสเตตการคลายสัญญาณเสียงแบบแอลเอดีพีซีเอ็มและการออกเสียง

ทางสุดท้ายคือข้อมูลมีเหลือพอที่จะเลื่อนก็จะเลื่อนข้อมูลได้เลย จากนั้นก็จะทำการแปลงสัญญาณในตัวแปร sound ที่อยู่ในรูปของบิตให้เป็นข้อมูลในรูปของไบต์ โดยมีการคิดเครื่องหมายตามขั้นของขนาดผลต่าง และก็ทดสอบว่าข้อมูลในตัวแปร sound นี้เท่ากับคำสั่งให้เปลี่ยนขนาดของผลต่างหรือไม่ ถ้าไม่ใช่ก็จะทำการบวกค่าของ sound กับตัวแปร base เพื่อจะได้สัญญาณจุดใหม่และทำการส่งให้ตัวแปลงสัญญาณดิจิตอลเป็นแอนาล็อกทำการออกเสียงต่อไป แต่ถ้าตัวแปร sound เป็นคำสั่งการเปลี่ยนแปลงขนาดของผลต่าง ก็จะทำการอ่านข้อมูลถัดไปอีก 3 บิตมาเป็นคำสั่งบอกขนาดของผลต่างใหม่ วิธีการอ่านข้อมูลอีก 3 บิตจะทำได้โดยตั้งขนาดของผลต่างให้เป็น 3 บิตเสียก่อน จากนั้นจึงทำการเลื่อนข้อมูล 3 บิตเข้ามาเพื่อนำมาเป็นค่าขนาดของผลต่าง โดยที่จะต้องตรวจสอบว่าข้อมูล 3 บิตที่รับมานั้นเป็น 1 หรือไม่ ถ้าใช่ก็จะสั่งให้เลิกการทำงาน แต่ถ้าไม่ใช่ 1 ก็จะเปลี่ยนค่าของ state ให้เป็นค่าที่ได้รับมา และจะย้อนการทำงานไปยังจุดที่ทำการทดสอบจำนวนบิตข้อมูลที่เหลืออยู่พอที่จะเลื่อนไหมในตอนต้นเพื่อทำการคลายสัญญาณเสียงจุดต่อไปให้ได้ตามขนาดของผลต่างใหม่

สำหรับการทดสอบว่าโปรแกรมการคลายสัญญาณนี้สามารถทำงานได้ทันหรือไม่ การทดสอบจะพิจารณาจำนวนสัญญาณนาฬิกาที่ต้องใช้ในกรณีที่ต้องใช้เวลาที่มากที่สุดที่จะเกิดขึ้นได้ คือ กรณีที่ขนาดของผลต่างเป็น 7 บิต และต้องเปลี่ยนขนาดของผลต่างเป็น 8 บิตโดยที่มีข้อมูลเหลืออยู่เพียง 1 บิตเท่านั้นดังรูปที่ 4.16 ทำให้ต้องอ่านข้อมูลเข้าถึง 3 ครั้งจึงจะได้ข้อมูลครบถ้วน ผลที่ได้ก็คือจะใช้จำนวนลูกสัญญาณนาฬิกาเท่ากับ 1432 ลูก ซึ่งใกล้เคียงกับขนาดจำนวนลูกสัญญาณนาฬิกาที่ถูกจำกัดเนื่องเพราะความถี่ในการสร้างสัญญาณเสียง 16 กิโลเฮิรตท์คือ 1500 ลูก ทำให้เหลือจำนวนลูกสัญญาณนาฬิกาเพียง 68 ลูกหรือประมาณ 16 คำสั่งเท่านั้น



รูปที่ 4.16 แสดงกรณีของการคลายสัญญาณที่ต้องใช้เวลามากที่สุด