บทที่ 4

ซอฟท์แวร์

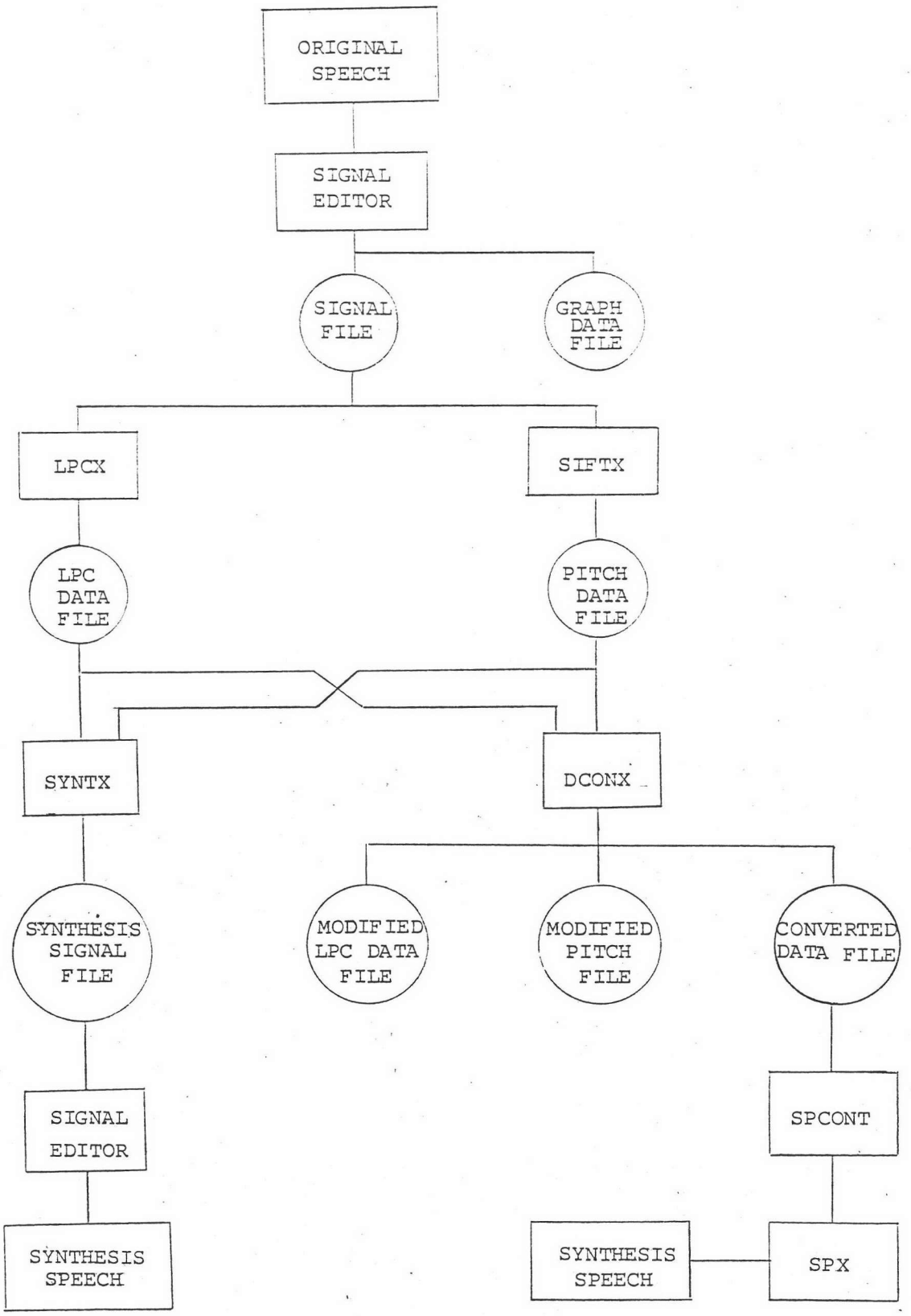
ซอฟท์แวร์ที่เกี่ยวข้องกับการวิเคราะห์และสังเคราะห์เสียงพูดทั้งหมดสามารถแบ่งได้เป็น 2 กลุ่ม คือ

1. โปรแกรมวิเคราะห์-สังเคราะห์เสียงในมาสเตอร์คอมพิวเตอร์ เป็นโปรแกรมที่ใช้งานในส่วนของการวิเคราะห์เสียงพูดเป็นหลัก ประกอบด้วย

- โปรแกรม SIGNAL EDITOR ทำหน้าที่ควบคุมการแปลงสัญญาณระหว่างสัญญาณอนาลอกกับสัญญาณเชิงเลขและแสดงสัญญาณสุ่มในรูปของกราฟ รวมทั้งจัดการเกี่ยวกับไฟล์ของสัญญาณเสียง
- โปรแกรม LPCX ทำหน้าที่คำนวนหาพารามิเตอร์ของฟิลเตอร์จากข้อมูลสัญญาณเสียง จำนวนออร์เดอร์ของฟิลเตอร์กำหนดได้ตั้งแต่ 2 ถึง 30
- โปรแกรม SIFTX ทำหน้าที่คำนวณหาคาบ (Pitch Period)ของสัญญาณเสียง
- โปรแกรม SYNTX ทำหน้าที่สังเคราะห์เสียงจากข้อมูลที่ได้จากโปรแกรม LPCX และโปรแกรม SIFTX มีไว้ทดสอบการสังเคราะห์เสียงด้วยการคำนวณในมาสเตอร์คอมพิวเตอร์

2. โปรแกรมที่เกี่ยวกับการสังเคราะห์เสียงด้วยภาคประมวลผลสัญญาณ ประกอบด้วย

- โปรแกรม DCONX ทำหน้าที่ดัดแปลงแก้ไขข้อมูลที่ได้จากโปรแกรม LPCX และโปรแกรม SIFTX แล้วทำการแปลง (Convert) ข้อมูลซึ่งเป็นตัวเลขชนิด Floating Point ให้เป็นตัวเลข Fixed Point ขนาด 16 บิท เพื่อใช้เป็นข้อมูลในการสังเคราะห์เสียงด้วยภาคประมวลผลสัญญาณ
- โปรแกรม SPCONT ทำหน้าที่ดาวน์โหลดโปรแกรมและข้อมูลจากมาสเตอร์คอม-พิวเตอร์ลงสู่ภาคประมวลผลสัญญาณ ตรวจสอบและแก้ไขข้อมูลโดยการอ่านและเขียนกับหน่วยความจำของภาคประมวลผลสัญญาณ ควบคุมการ RUN โปรแกรมและทำการติดต่อกับภาคประมวลผลสัญญาณขณะ RUN โปรแกรม
- โปรแกรม SPX เป็นโปรแกรมภาษาเครื่องของโปรเซสเซอร์ TMS 32010 ใช้ RUN ในภาคประมวลผลสัญญาณ ทำหน้าที่สังเคราะห์เสียงเป็นคำๆ ในลักษณะ Real-Time จากข้อมูลที่ได้จากการวิเคราะห์ และผ่านการแปลงข้อมูลด้วยโปรแกรม DCONX



รูป 4.1 โปรแกรมที่ใช้ในการวิเคราะห์ และสังเคราะห์เสียง

โปรแกรมที่กล่าวมาทั้งหมดยกเว้นโปรแกรม SPX เขียนด้วยภาษาเบสิกบนเครื่อง APPLE II และส่วนใหญ่ผ่านการคอมไพล์ด้วย Apple Soft Basic Compiler เพื่อเพิ่มความเร็วในการทำงาน บางหน้าที่ในโปรแกรมอาศัยการทำงานด้วยภาษาเครื่อง (ซีพียู.6502) เนื่องจากข้อจำกัดด้านความเร็วของภาษาเบสิก รูป 4.1 แสดงโปรแกรมต่างๆ ที่กล่าวมาซึ่งในการวิเคราะห์และสังเคราะห์เสียงแต่ละคำ จะมีขั้นตอนสรุปได้คือ

1. เสียงพูดถูกสุ่มมาเก็บไว้เป็นไฟล์ข้อมูลสัญญาณ Signal File โดยโปรแกรม SIGNAL EDITOR

2. นำสัญญาณเสียงพูดจากไฟล์ข้อมูลสัญญาณไปคำนวณหาพารามิเตอร์ของฟิลเตอร์ด้วยโปรแกรม LPCX และหาค่าคาบของเสียงด้วยโปรแกรม SIFTX

3. ในกรณีที่ต้องการทดสอบการสังเคราะห์เสียงโดยอาศัยการคำนวณในเครื่องมาสเตอร์คอมพิวเตอร์ สามารถทำได้ด้วยโปรแกรม SYNTX สัญญาณเสียงที่ได้จะอยู่ในไฟล์สัญญาณ Synthesis Signal File ซึ่งสามารถนำไปแปลงเป็นเสียงพูดได้ด้วยโปรแกรม SIGNAL EDITOR

4. นำข้อมูลพารามิเตอร์ของฟิลเตอร์ และข้อมูลคาบของเสียงที่ได้จากข้อ 2. มาดัดแปลงแก้ไขถ้าจำเป็นด้วยโปรแกรม DCONX ข้อมูลที่แก้ไขเรียบร้อยแล้วสามารถเก็บลงดิสค์ไฟล์ได้ ในกรณีของการสังเคราะห์เสียงด้วยภาคประมวลผลสัญญาณ ข้อมูลดังกล่าวจะต้องแปลงจากค่าตัวเลขชนิด Floating Point เป็นค่าตัวเลข Fixed Point ก่อนเก็บลงดิสค์ไฟล์

5. โปรแกรมที่ใช้สังเคราะห์เสียงในภาคประมวลผลสัญญาณหรือโปรแกรม SPX รวมทั้งข้อมูลของเสียงพูดจาก Converted Data File จะถูกดาวน์โหลดลงสู่หน่วยความจำของภาคประมวลผลสัญญาณ ด้วยโปรแกรม SPCONT

6. สั่งให้โปรแกรม SPX สังเคราะห์เสียงพูดเป็นคำๆ โดยคำศัพย์ที่ต้องการกำหนดได้จากโปรแกรม SPCONT

4.1 โปรแกรมวิเคราะห์-สังเคราะห์เสียงพูดในมาสเตอร์คอมพิวเตอร์

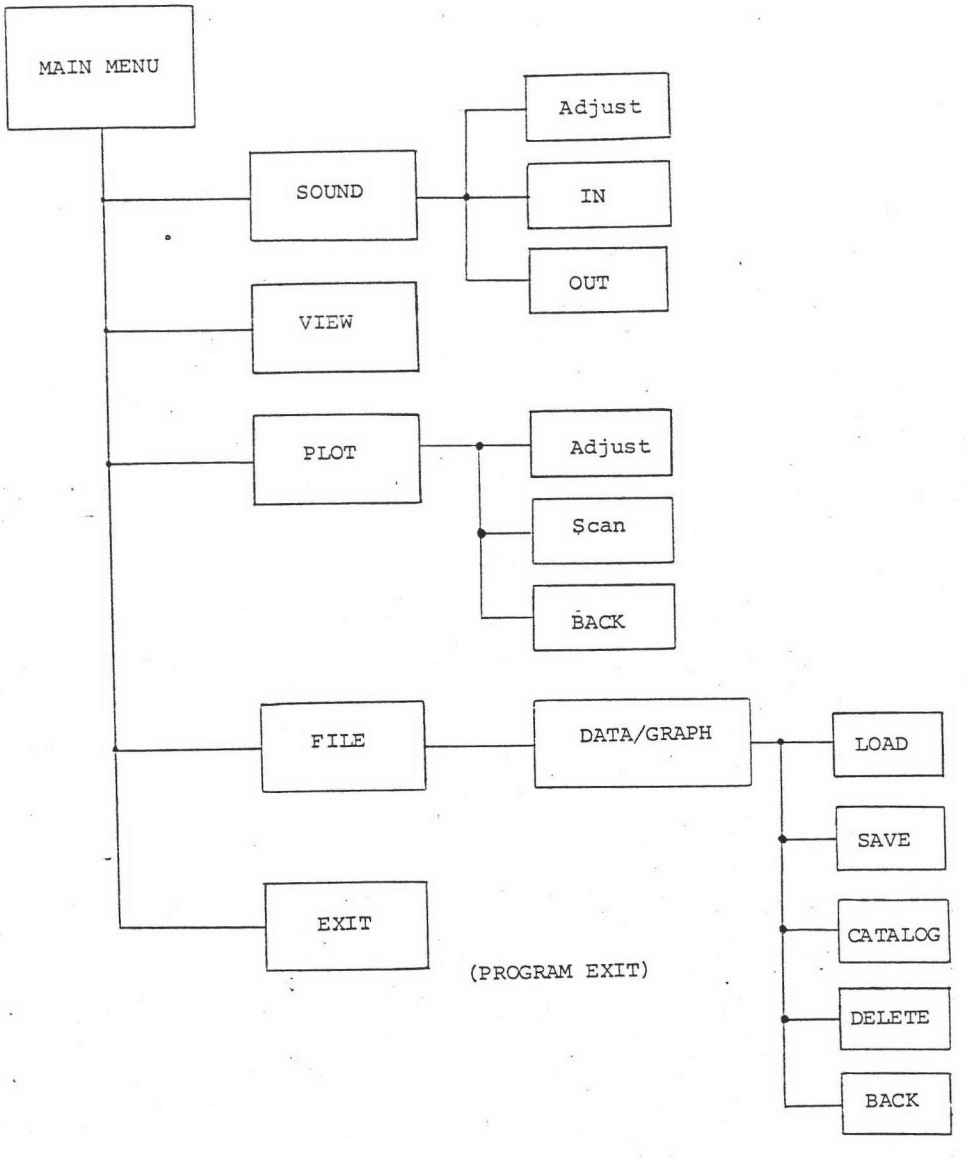
เป็นโปรแกรมที่ใช้งานในส่วนของการวิเคราะห์เป็นหลัก การคำนวณทั้งหมดทำในมาสเตอร์คอมพิวเตอร์ ประกอบด้วย

- โปรแกรม SIGNAL EDITOR
- โปรแกรม LPCX
- โปรแกรม SIFTX
- โปรแกรม SYNTX

ในหัวข้อนี้จะกล่าวถึงรายละเอียดของแต่ละโปรแกรม

4.1.1 โปรแกรม SIGNAL EDITOR

เป็นโปรแกรมทำหน้าที่ควบคุมการแปลงสัญญาณระหว่างสัญญาณอนาลอกกับสัญญาณเชิงเลข และแสดงสัญญาณสุ่มในรูปของกราฟ รวมทั้งจัดการเกี่ยวกับไฟล์ รูป 4.2 แสดงคำสั่งต่าง ๆ ในโปรแกรม การทำงานแบ่งเป็น 4 โหมด คือ 1. Sound 2. View 3. Plot และ 4. File



รูป 4.2 คำสั่งในโปรแกรม SIGNAL EDITOR

โหมด Sound มี 2 หน้าที่ คือ คำสั่ง In ทำหน้าที่ควบคุมการแปลงสัญญาณอนาลอกเป็นสัญญาณเชิงเลขแล้วบันทึกลงหน่วยความจำ คำสั่ง Out ทำหน้าที่ควบคุมการแปลงสัญญาณเชิงเลขจากหน่วยความจำเป็นสัญญาณอนาลอก คำสั่ง Adjust ทำหน้าที่กำหนดตำแหน่งในหน่วยความจำสำหรับการบันทึกหรือ Play Back สัญญาณเสียง กำหนดเบอร์สล็อตที่เป็นที่ตั้งของบอร์ดมาสเตอร์อินเตอร์เฟซ และกำหนดความถี่ในการสุ่มสัญญาณ (Sampling Frequency) ความถี่ในการสุ่มสัญญาณกำหนดได้ระหว่าง 7.5-17.8 kHz สัญญาณเสียงหนึ่งแซมเปิลใช้หน่วยความจำ 2 ไบท์ จำนวนแซมเปิลของเสียงที่บันทึกได้สูงสุดเท่ากับ 10240 แซมเปิล

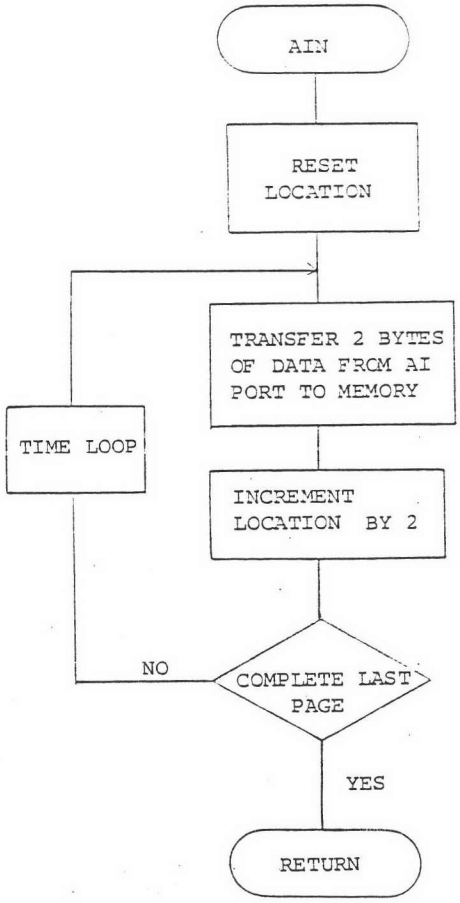
โหมด View ทำหน้าที่แสดงสัญญาณเสียงในหน่วยความจำตั้งแต่หน้าที่ 70 ถึงหน้า 145 ออกจอมอนิเตอร์ในรูปของกราฟ โหมดนี้ใช้ตรวจสอบสัญญาณเสียงตอนบันทึกเสียงหรือ ใช้ดูรูปคลื่นของคำพูดตลอดคำ

โหมด Plot ทำหน้าที่แสดงสัญญาณเสียงเป็นกราฟ โดยที่สามารถขยายแกนเวลาหรือดูรายละเอียดของรูปคลื่น และสามารถกวาดดูสัญญาณตั้งแต่ต้นคำถึงท้ายคำ คำสั่ง Adjust ทำหน้าที่กำหนดสเกลทั้งสองแกนของรูปกราฟ คำสั่ง Scan คือ คำสั่งสำหรับเลื่อนดูส่วนต่างๆ ของคำพูดโดยกำหนดแซมเปิลแรกที่จะนำมาแสดง คำสั่ง Back สำหรับกลับ Main Menu

โหมด File ทำหน้าที่ย้ายข้อมูลระหว่างหน่วยความจำกับดิสค์ไฟล์ การทำงานมี 2 ลักษณะ คือ ย้ายข้อมูลที่เป็นสัญญาณเสียง และย้ายข้อมูลของกราฟ คำสั่ง Data และ Graph ทำหน้าที่เลือกชนิดของข้อมูลว่าเป็นสัญญาณเสียง หรือเป็นข้อมูลของเส้นกราฟตามลำดับ คำสั่ง Load จะทำหน้าที่อ่านข้อมูลจากดิสค์ไฟล์มาเก็บในหน่วยความจำ คำสั่ง Save ทำหน้าที่ย้ายข้อมูลเสียงจากหน่วยความจำไปยังดิสค์ไฟล์ ก่อนทำการ Save ข้อมูลทั้งสองชนิดสามารถเขียนคำอธิบายหรือ Comment ได้ 230 ตัวอักษร ซึ่งคำอธิบายนี้ใช้แสดงหลังคำสั่ง Load คำสั่ง Catalog ใช้แสดงรายชื่อไฟล์ในดิสค์ คำสั่ง Delete ใช้ลบไฟล์ในดิสค์ และคำสั่ง Back ใช้สำหรับกลับไป Main Menu ทั้งข้อมูลสัญญาณเสียงและข้อมูลของเส้นกราฟเก็บในรูปของ Binary File ตาราง 4.1 แสดงตำแหน่งในหน่วยความจำที่ใช้เก็บข้อมูลของโปรแกรม SIGNAL EDITOR

LOCATION	LENGTH (BYTE)	FUNCTION
7680-7951	272	Graph Data Buffer
7960-8191	232	Graph Comment Text
8192-16383	8192	Apple II HI-RES Graphic Area
16384-16617	232	Signal Data Comment Text
16640	1	Value Of Low Data Page (U1)
16642	1	Value of High Data Page (U2)
16643	1	Value of Sampling Rate No.(SR)
16896-37375	20480	Signal Data Area
17920-37375	19456	Signal Data Area Default

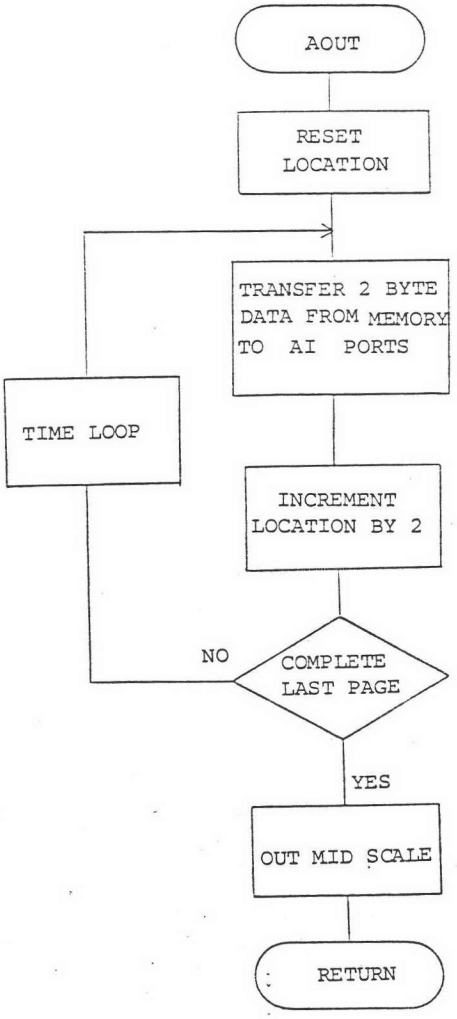
ตาราง 4.1 ตำแหน่งในหน่วยความจำที่ใช้เก็บข้อมูลของโปรแกรม SIGNAL EDITOR



LOCATION	FUNCTION
07	PAGE COUNT
768	LAST PAGE
769	TIME LOOP COUNT
781	LOW BYTE AI PORT POINTER
787	HIGH BYTE AI PORT POINTER
778	PROGRAM START

รูป 4.3 แผนผังการทำงานของโปรแกรมย่อย AIN

โปรแกรมย่อยควบคุมการแปลงสัญญาณระหว่างสัญญาณอนาลอกกับสัญญาณเชิงเลข เป็นโปรแกรมภาษาเครื่อง แบ่งเป็น 2 โปรแกรมคือ AIN และ ACUT โปรแกรมย่อย AIN ทำหน้าที่ควบคุมการแปลงสัญญาณอนาลอกเป็นสัญญาณเชิงเลข คำสั่ง In ในโหมด SOUND เรียกโปรแกรมย่อย AIN ซึ่งจะทำงานตามแผนผังในรูป 4.3 โดยก่อนการทำงานจะต้องกำหนดค่าต่างๆ ดังแสดงไว้ในรูป ด้วยคำสั่ง POKE ก่อนการทำงานทุกครั้งต้องกำหนด Page Count เท่ากับหน้าแรกในหน่วยความจำที่ใช้เก็บสัญญาณเสียง หน้าสุดท้าย Last Page กำหนดหน้าสุดท้ายในหน่วยความจำสำหรับเก็บข้อมูลเสียง TIME LOOP กำหนดตามความถี่ในการสุ่ม มีค่าระหว่าง 1-20 AI PORT POINTER คือค่าแอดเดรส 8 บิทล่างของ AI DATA PORT มีค่าเท่ากับ 140 + 16 x N สำหรับ Low-byte โดย N เท่ากับเบอร์สล็อตที่เป็นที่ตั้งของบอร์ดมาสเตอร์อินเตอร์เฟซ สำหรับ High byte บวกค่าดังกล่าวด้วย 1



LOCATION	FUNCTION
07	PAGE COUNT
768	LAST PAGE
769	TIME LOOP COUNT
833, 874	LOW BYTE AI PORT POINTER
839, 877	HIGH BYTE AI PORT POINTER
828	PROGRAM START

รูป 4.4 แผนผังการทำงานของโปรแกรมย่อย AOUT

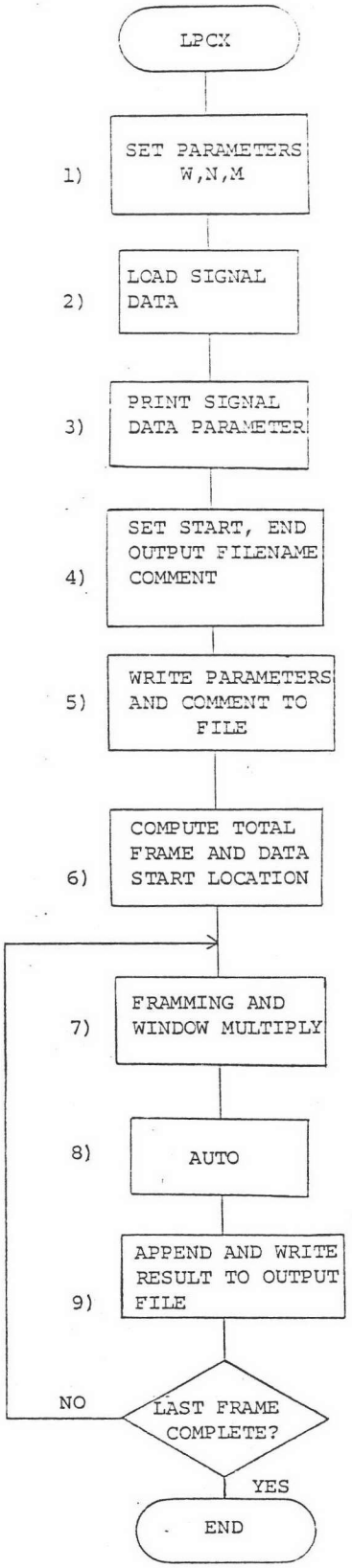
โปรแกรมย่อย AOUT ทำหน้าที่ควบคุมการแปลงสัญญาณเชิงเลขเป็นสัญญาณอนาลอก คำสั่ง Out ในโหมด Sound เรียกโปรแกรมย่อย AOUT ซึ่งจะทำงานตามแผนผังในรูป 4.4 เช่นเดียวกับโปรแกรมย่อย AIN ก่อนการทำงานต้องมีการกำหนดค่าต่างๆ ตามตารางในรูป

4.1.2 โปรแกรม LPCX

เป็นโปรแกรมคำนวณค่าสัมประสิทธิ์และค่าผิดพลาดยกกำลังสองในแต่ละเฟรม โดยอาศัยโปรแกรมย่อย AUTO ในบทที่ 2 การทำงานโปรแกรมจะนำสัญญาณมาแบ่งเป็นเฟรมๆ ในเฟรมหนึ่งจะกำหนดให้มีจำนวนสัญญาณได้สูงสุด 512 แซมเปิล จำนวนสัญญาณในแต่ละเฟรมใช้ตัวแปรชื่อ W ระยะห่างระหว่างเฟรมเรียกว่า Analysis Frame Length ใช้ตัวแปรชื่อ N เป็นตัวกำหนดว่าแซมเปิลแรกในเฟรมปัจจุบันอยู่ห่างจากแซมเปิลแรกของเฟรมก่อนหน้าเป็นจำนวนกี่แซมเปิล การคำนวณจะทำทีละเฟรมจนครบสัญญาณเสียงทั้งคำ จำนวนออร์เดอร์ของฟิลเตอร์ใช้ตัวแปรชื่อ M มีค่าระหว่าง 2 ถึง 30 การทำงานเริ่มจาก

1. กำหนดพารามิเตอร์ในการคำนวณ พารามิเตอร์ประกอบด้วยค่าของ W,N,M
2. อ่านข้อมูลสัญญาณเสียงของคำที่ต้องการจากดิสค์ไฟล์ลงหน่วยความจำ
3. แสดงตำแหน่งในหน่วยความจำของสัญญาณเสียง และความถี่ในการสุ่ม
4. กำหนดแซมเปิลแรกและแซมเปิลสุดท้ายในการคำนวณ ชื่อของเอาท์พุทไฟล์และคำบรรยาย
5. เขียนข้อมูลเกี่ยวกับการคำนวณและคำบรรยายลงเอาท์พุทไฟล์
6. คำนวณจำนวนเฟรมทั้งหมดและตำแหน่งในหน่วยความจำของสัญญาณเสียงแซมเปิลแรกที่ใช้ในการคำนวณ
7. คำนวณตำแหน่งในหน่วยความจำของแซมเปิลเริ่มต้นของเฟรมและนำสัญญาณจำนวนเท่ากับ Window ตั้งแต่แซมเปิลแรกของเฟรมคูณด้วยฟังก์ชัน Hamming Window
8. เรียกโปรแกรมย่อย AUTO
9. เขียนผลลัพธ์ของการคำนวณลงไฟล์ต่อท้ายข้อมูลของเฟรมที่แล้ว จากนั้นโปรแกรมจะกลับไปทำข้อ 7. ถึง 9. จนครบทุกเฟรม

ผลลัพธ์จากการคำนวณทุกเฟรมจะเก็บไว้ในไฟล์ข้อมูลแอลพีซี (LPC DATA FILE) ในการใช้งานโปรแกรม LPCX จะผ่านการคอมไพล์ได้โปรแกรมชื่อ LPX



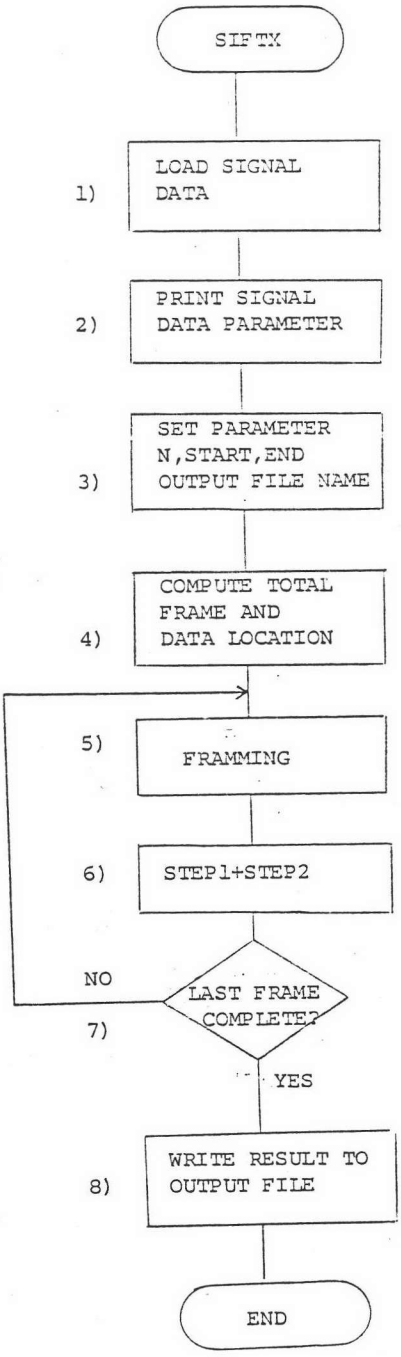
รูป 4.5 แสดงแผนผังการทำงานของโปรแกรม LPCX

4.1.3 โปรแกรม SIFTX

โปรแกรมวิเคราะห์หาคาบของสัญญาณเสียงพูดอาศัยโปรแกรมย่อย SIFT ในบทที่ 2 สัญญาณเสียงของคำที่ต้องการวิเคราะห์จะถูกแบ่งเป็นเฟรมๆ เฟรมหนึ่งมีจำนวนสัญญาณดงที่เท่ากับ 400 แซมเปิล ระยะห่างระหว่างเฟรม (ตัวแปรชื่อ N) กำหนดได้ซึ่งโดยทั่วไปจะให้เท่ากับค่า Analysis Frame Length ที่ใช้กับโปรแกรม LPCX การคำนวณจะทำทีละเฟรมจนครบสัญญาณเสียงทั้งคำ การทำงานเริ่มจาก

1. โหลดข้อมูลสัญญาณเสียงของคำที่ต้องการจากดิสค์ไฟล์ลงหน่วยความจำ
2. แสดงตำแหน่งในหน่วยความจำของสัญญาณเสียง ความถี่ในการสุ่มของสัญญาณ
3. กำหนดค่า N แซมเปิลแรกและแซมเปิลสุดท้ายในการคำนวณ และกำหนดชื่อของเอาท์พุทไฟล์
4. คำนวณจำนวนเฟรมทั้งหมด และตำแหน่งในหน่วยความจำของแซมเปิลแรกที่ใช้คำนวณ
5. คำนวณตำแหน่งในหน่วยความจำของแซมเปิลแรกของเฟรม
6. ทำการคำนวณตามโปรแกรมย่อย SIFT
7. ทำการคำนวณตามหัวข้อ 5. และ 6. จนครบทุกเฟรม
8. เขียนคำตอบของทุกเฟรมลงเอาท์พุทไฟล์

ค่าของคาบที่คำนวณได้แต่ละเฟรมจะถูกเก็บในตัวแปร จากนั้นเมื่อคำนวณเสร็จทุกเฟรมแล้วจึงเขียนผลลัพธ์ลงไฟล์เอาท์พุทที่เก็บข้อมูลคาบของเสียง (Pitch Data File) การใช้งานโปรแกรม SIFTX จะผ่านการคอมไพล์ได้โปรแกรมชื่อ SFX

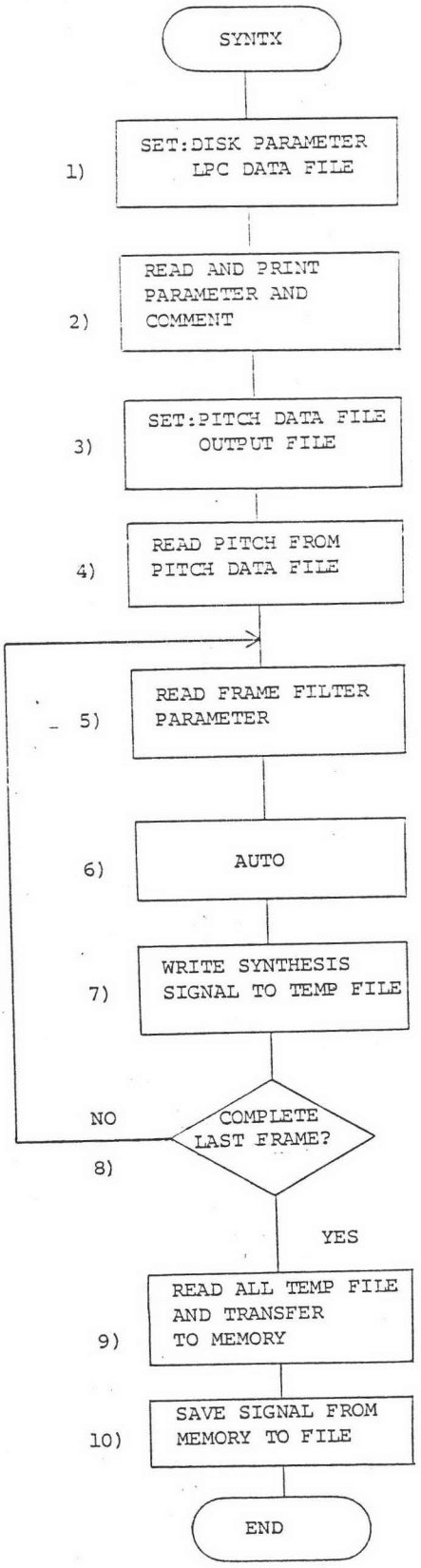


รูป 4.6 แผนผังการทำงานของโปรแกรม SIFTX

4.1.4 โปรแกรม SYNTX

ทำหน้าที่สังเคราะห์เสียงจากข้อมูลพารามิเตอร์ของฟิลเตอร์จากไฟล์ LPC Data File และข้อมูลคาบของเสียงจากไฟล์ Pitch Data File การสังเคราะห์เสียงกระทำทีละเฟรมโดยอาศัยการทำงานของโปรแกรมย่อย SYNT ในบทที่ 2 สัญญาณเสียงที่สังเคราะห์ได้จะถูกเก็บลงไฟล์ชั่วคราวชื่อ S.TEMP การสังเคราะห์เสียงจะกระทำไปจนครบทุกเฟรมหรือได้สัญญาณเสียงครบคำพูด จากนั้นจะย้ายสัญญาณเสียงตลอดคำพูดจากไฟล์ชั่วคราวมาที่หน่วยความจำให้ได้ข้อมูลในลักษณะเดียวกัน สัญญาณเสียงในโปรแกรม SIGNAL EDITOR จากนั้นย้ายข้อมูลสัญญาณเสียงจากหน่วยความจำไปเก็บในไฟล์สัญญาณเสียงสังเคราะห์ (Synthesis Signal File) ลำดับการทำงานเริ่มจาก

1. กำหนดดิสค์สำหรับเก็บสัญญาณเสียงสังเคราะห์ และกำหนดดิสค์สำหรับไฟล์ชั่วคราว (S.TEMP) รวมทั้งชื่อไฟล์ข้อมูลพารามิเตอร์ของฟิลเตอร์
2. อ่านพารามิเตอร์ W,N,M รวมทั้งคำบรรยาย จากไฟล์ข้อมูลแล้วแสดงออกจอมอนิเตอร์
3. กำหนดชื่อไฟล์ข้อมูลคาบของสียงและชื่อไฟล์สัญญาณเสียงสังเคราะห์ซึ่งเป็นเอาท์พุทของโปรแกรม
4. อ่านข้อมูลคาบของเสียงทุกเฟรมจากไฟล์เก็บไว้ในตัวแปรชุด (Array Variable) ไว้สำหรับคำนวณตลอดคำพูด
5. อ่านค่าพารามิเตอร์ของฟิลเตอร์ ซึ่งประกอบด้วยสัมประสิทธิ์พาร์คอร์จำนวน M ตัว และค่าผิดพลาดยกกำลังสองรวมของแต่ละเฟรมนั้นๆ
6. การคำนวณเพื่อสังเคราะห์เสียงในหนึ่งเฟรมตามการทำงานของโปรแกรมย่อย SYNT
7. เขียนผลลัพธ์ซึ่งเป็นสัญญาณเสียงสังเคราะห์ลงไฟล์ชั่วคราวชื่อ S.TEMP I เมื่อ I คือเลขลำดับของเฟรม
8. ทำงานตาม ข้อ 5. ถึง 7. จนครบทุกเฟรม
9. อ่านสัญญาณเสียงสังเคราะห์จากไฟล์ชั่วคราว แปลงค่าที่คำนวณได้ ซึ่งเป็นตัวเลขชนิด Floating Point ให้เป็นเลข Integer ซึ่งอยู่ในลักษณะเดียวกับสัญญาณเสียงที่แปลงมาจาก A/D Converter เก็บค่าที่ได้ในหน่วยความจำในลักษณะเดียวกับโปรแกรม SIGNAL EDITOR
10. ย้ายสัญญาณเสียงสังเคราะห์จากหน่วยความจำลงสู่ไฟล์สัญญาณเสียงสังเคราะห์ สัญญาณเสียงที่ได้จะอยู่ในไฟล์สัญญาณเสียงสังเคราะห์ซึ่งสามารถนำไปแปลงเป็นเสียงพูดเพื่อทดสอบโดยการฟังได้ด้วยโปรแกรม SIGNAL EDITOR ในการใช้งานโปรแกรม SYNTX จะถูกคอมไพล์ได้โปรแกรมชื่อ SYX



รูป 4.7 แผนผังการทำงานของโปรแกรม SYNTX

4.2 โปรแกรมที่เกี่ยวกับการสังเคราะห์เสียงด้วยภาคประมวลผลสัญญาณ

ประกอบด้วย - โปรแกรม DCONX
- โปรแกรม SPCONT
- โปรแกรม SPX

ในหัวข้อนี้จะกล่าวถึงรายละเอียดของแต่ละโปรแกรม

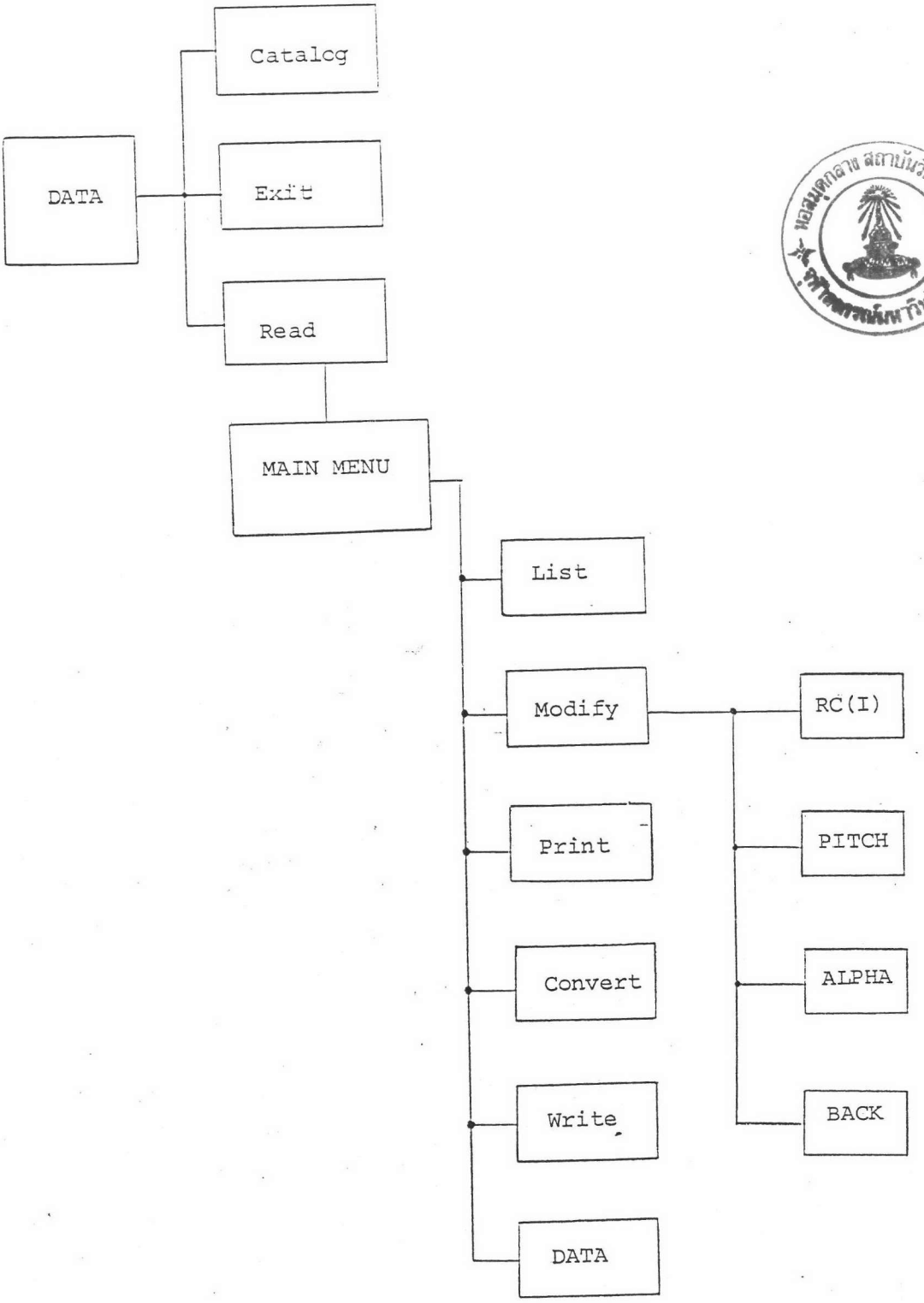
4.2.1 โปรแกรม DCONX

ทำหน้าที่ คือ

1. ดัดแปลงแก้ไขข้อมูลในไฟล์พารามิเตอร์ของฟิลเตอร์ (LPC Data File) และข้อมูลในไฟล์ข้อมูลคาบของเสียง (Pitch Data File)
2. คำนวณเพื่อเตรียมข้อมูลและแปลงข้อมูลซึ่งเป็นตัวเลขชนิด Floating Point ให้เป็นตัวเลขชนิด Fixed Point

รูป 4.8 แสดงคำสั่งในโปรแกรม DCONX เมื่อเริ่มทำงาน โปรแกรมจะเข้าสู่โหมด Data โดยอัตโนมัติ โหมด Data ทำหน้าที่อ่านข้อมูลจากไฟล์ประกอบด้วย 3 คำสั่ง คือ คำสั่ง Catalog แสดงรายการของไฟล์ในดิสค์ คำสั่ง Exit ออกจากโปรแกรม คำสั่ง Read ทำหน้าที่อ่านข้อมูลโดยการกำหนดชื่อของไฟล์ข้อมูลทั้งสอง หลังจากที่โปรแกรมอ่านข้อมูลจากไฟล์ทั้งสองเสร็จแล้ว โปรแกรมจะเข้าสู่ Main Menu ใน Main Menu จะประกอบด้วย 6 คำสั่ง คือ

1. คำสั่ง List ทำหน้าที่แสดงข้อมูลพารามิเตอร์ของฟิลเตอร์และข้อมูลคาบของเสียงซึ่งใช้ในการสังเคราะห์เสียงในเฟรมต่างๆ จากการกำหนดเฟรมแรกและเฟรมสุดท้าย
2. คำสั่ง Modify ทำหน้าที่แก้ไขข้อมูลที่ใช้ในการสังเคราะห์เสียงในแต่ละเฟรม โดยสามารถเลือกข้อมูลที่จะแก้ไขว่าเป็น สัมประสิทธิ์พาร์คอร์ RC(I) คาบของเสียง (Pitch) หรือ ค่าผิดพลาดยกกำลังสองรวม ALPHA คำสั่ง Back สำหรับกลับไป Main Menu
3. คำสั่ง Print พิมพ์ข้อมูลลักษณะเดียวกับข้อ 1. ด้วยเครื่องพิมพ์
4. คำสั่ง Convert ทำการคำนวณเพื่อเตรียมข้อมูลให้สอดคล้องกับการสังเคราะห์เสียงด้วยโปรแกรม SPX หลังจากนั้นทำการแปลงข้อมูลซึ่งเป็นตัวเลขชนิด Floating Point ให้เป็นตัวเลขชนิด Fixed Point (หรือตัวเลขจำนวนเต็มขนาด 16บิท) การแปลงข้อมูลแตกต่างกันตามพารามิเตอร์แต่ละตัว คือ



รูป 4.8 คำสั่งในโปรแกรม DCONX

- สัมประสิทธิ์พาร์คอร์ ซึ่งเป็นตัวเลขทศนิยมมีค่าระหว่าง -1 ถึง +1 แปลงให้เป็นเลขจำนวนเต็มขนาด 16 บิท มีค่าระหว่าง -32767 ถึง +32767

- ค่าคาบของเสียงซึ่งเป็นตัวเลข Real ในภาษาเบสิค จะมีค่าเท่ากับ 0 (กรณีเสียงเป็นเสียงไม่ก้อง) และเลข Real ระหว่างประมาณ 25 ถึง 155 ซึ่งได้จากกรรมวิธี SIFT การแปลงข้อมูล คือ เปลี่ยนจากเลข Realเป็นเลขจำนวนเต็ม หรือใช้ฟังก์ชัน INT (X)

- ค่าผิดพลาดยกกำลังสองรวม หรือ ALPHA จะถูกเปลี่ยนเป็น Gain ตามสมการ (2.81) และสมการ (2.83) ในลักษณะเดียวกับโปรแกรมย่อย SYNT ข้อมูลที่ได้จะมี 2 ตัว ในลักษณะเดียวกันคือ AM และ AU ส่วนค่า NC จะอยู่ที่ค่า AM กรณีที่คาบมีค่าเท่ากับ 0 จากนั้นแปลงข้อมูลเป็นเลขจำนวนเต็ม ระหว่าง 0 ถึง +32767 โดยมีการนอร์มัลไลซ์ค่าระหว่างเฟรมตลอดคำพูด

ข้อมูลทั้ง 3 ชนิดที่กล่าวหลังจากแปลงแล้ว จะถูกจัดอยู่รวมกันในหน่วยความจำในลักษณะตามรูป 4.9 ซึ่งเมื่อแปลงครบทุกเฟรมแล้ว โปรแกรมจะย้ายข้อมูลในหน่วยความจำไปเก็บใน Binary File การจัดข้อมูลด้วยวิธีนี้เพื่อให้สอดคล้องกับการสังเคราะห์เสียงด้วยโปรแกรม SPX

5. คำสั่ง Write บันทึกข้อมูลที่ทำการแก้ไขแล้วในข้อ 2. ลงดิสค์ไฟล์
6. คำสั่ง Data กลับไปโหมด Data

ในการใช้งานโปรแกรม DCONX จะถูกคอมไพล์เป็นโปรแกรม DCX

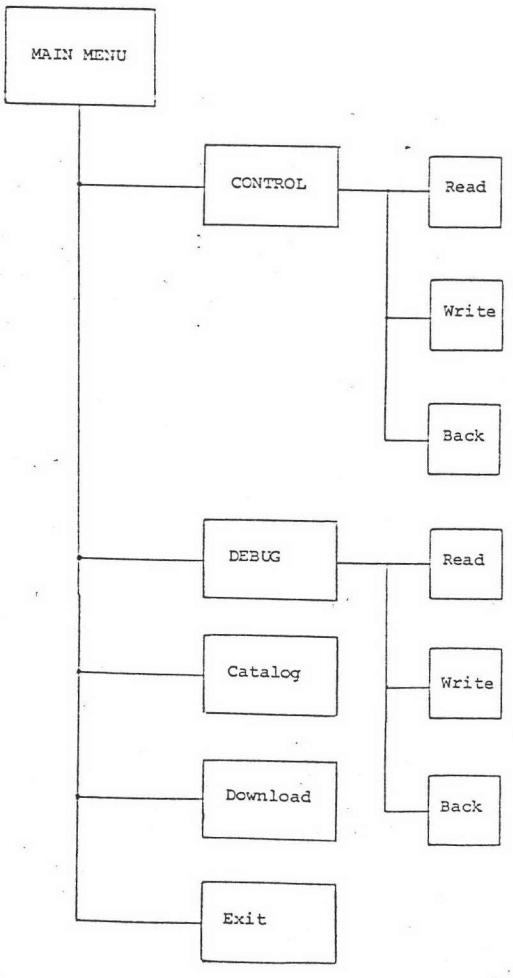
WORD	DATA	
0	Number of Frame (K)	
1	Number of Sample per Frame (N)	HEAD DATA
2	Number of Filter Order (M)	
4	GAIN (AM)	
5	PITCH	
6	AU	
7	RC(1)	FIRST FRAME
8	RC(2)	
¦	¦	
¦	¦	
¦	¦	
¦	¦	
¦	¦	
¦	RC(M)	
	¦	
	¦	
	¦	
	¦	

รูป 4.9 การจัดเรียงข้อมูลที่ทำการแปลงแล้ว

4.2.2 โปรแกรม SPCONT

หรือโปรแกรม Signal Processor Control ทำหน้าที่

- ดาวน์โหลดโปรแกรมและข้อมูลจากมาสเตอร์คอมพิวเตอร์ลงสู่ภาคประมวลผลสัญญาณ
- แก้ไขข้อมูลในหน่วยความจำของภาคประมวลผลสัญญาณ การทำงานส่วนนี้เรียกว่าโหมด Debug
- ควบคุมการทำงานและรับส่งข้อมูลระหว่างภาคประมวลผลสัญญาณ การทำงานในส่วนนี้เรียกว่าโหมด Control



รูป 4.10 คำสั่งในโปรแกรม SPCONT

เมื่อเริ่มทำงาน โปรแกรมจะให้กำหนดเบอร์สล็อตซึ่งเป็นที่อยู่ของบอร์ดมาสเตอร์อินเตอร์เฟซ จากนั้นโปรแกรมจะเข้าสู่ Main Menu รูป 4.10 แสดงคำสั่งต่างๆ ในโปรแกรม SPCONT คำสั่งใน Main Menu ประกอบด้วย

1. คำสั่ง Control เพื่อเข้าสู่โหมด Control โปรแกรมจะเตรียมพร้อมติดต่อพอร์ทต่างๆ ภายใน 16 พอร์ทผ่านมาสเตอร์อินเตอร์เฟซดังที่กล่าวในหัวข้อ 3.1 โหมด Control ประกอบด้วย คำสั่ง Read ทำหน้าที่แสดงข้อมูล 8 บิทที่อ่านได้จากพอร์ทที่กำหนด ค่าที่แสดงเป็นเลขฐานสิบ คำสั่ง Write ทำหน้าที่เขียนข้อมูลซึ่งเป็นตัวเลข 0 ถึง 255 ไปยังพอร์ทที่กำหนด คำสั่ง Back กลับสู่ Main Menu

2. คำสั่ง Debug เพื่อเข้าสู่โหมด Debug สำหรับอ่านหรือเขียนกับหน่วยความจำหรืออินพุท-เอาท์พุทพอร์ทของภาคประมวลผลสัญญาณ แต่ก่อนจะเข้าสู่โหมด Debug ต้องกำหนดการทำงานของภาคประมวลผลสัญญาณ โดยใช้คำสั่ง Write ในโหมด Control เขียนข้อมูลไปที่พอร์ท 0 หรือรีจีสเตอร์ควบคุม SP Control Register โหมด Debug ประกอบด้วย คำสั่ง Read ทำหน้าที่อ่านข้อมูลและแสดงออกจอมอนิเตอร์ตามจำนวนเวิร์ดและตำแหน่งที่กำหนด คำสั่ง Write ทำหน้าที่เขียนข้อมูล 16 บิทไปยังหน่วยความจำหรือพอร์ทตามตำแหน่งที่กำหนด ตัวเลขต่างๆ ในการอ่านหรือเขียนหรือกำหนดในโหมด Debug ใช้เลขฐานสิบหกทั้งสิ้น

3. คำสั่ง Catalog ทำหน้าที่แสดงรายการของไฟล์ในดิสค์

4. คำสั่ง Download ทำหน้าที่ดาวน์โหลดข้อมูลจากไฟล์ลงสู่หน่วยความจำในภาคประมวลผลสัญญาณ ใช้งานโดยกำหนดชื่อไฟล์ข้อมูลและตำแหน่งเริ่มต้น (เลขฐานสิบหก) ในหน่วยความจำสำหรับข้อมูล

5. คำสั่ง Exit สำหรับออกจากโปรแกรม

ในการใช้งานโปรแกรม SPCONT จะผ่านการคอมไพล์ได้เป็นโปรแกรมชื่อ SPCX

4.2.3 โปรแกรม SPX

เป็นโปรแกรมสังเคราะห์เสียงพูดเป็นคำๆ สำหรับการทดลองสังเคราะห์เสียงด้วยภาคประมวลผลสัญญาณ โดยโปรแกรมและข้อมูลของเสียงพูดทั้งหมดจะอยู่ในหน่วยความจำของภาคประมวลผลสัญญาณ โปรแกรม SPX เขียนด้วยภาษาแอสเซมบลีของซีพียู TMS 32010 และผ่านการแอสเซมเบิลด้วยโปรแกรม TASSM ของบริษัท Computalker ประเทศสหรัฐอเมริกา ผลที่ได้คือโคดภาษาเครื่องชนิด Absolute อยู่ในไฟล์ ในการใช้งานโปรแกรม SPCONT ทำการดาวน์โหลดโปรแกรม SPX ลงสู่หน่วยความจำของภาคประมวลผลสัญญาณ รวมทั้งข้อมูลที่ใช้สำหรับการสังเคราะห์เสียงที่ผ่านการแปลงข้อมูลด้วยโปรแกรม DCONX แล้ว นอกจากนั้นข้อมูลที่จำเป็นต้องทำการดาวน์โหลดด้วย คือ ตารางสัญญาณเสียงรบกวน (Noise Table) ข้อมูลของเสียงพูดแต่ละคำที่ได้จากโปรแกรม DCONX จะถูกนำมาจัดเรียงต่อกัน โดยช่วงต้นของกลุ่มข้อมูลนี้จะเป็นตารางบอกตำแหน่งในหน่วยความจำแรกของข้อมูลเสียงแต่ละคำเรียกว่า " ตารางคำศัพท์" รูป 4.11 แสดงการจัดข้อมูลเสียงหลายๆ คำมาต่อกัน หน่วยความจำของภาคประมวลผลสัญญาณสามารถเก็บข้อมูลเสียงได้ประมาณ 10-12 คำ (เนื้อที่ 3.25k เวิร์ด) การทำงานของโปรแกรม SPX คล้ายคลึงกับโปรแกรม SYNTX ที่แตกต่างกัน คือตัดการคำนวณเรื่อง Gain และใช้ตารางสัญญาณเสียงรบกวนแทนฟังก์ชันผลิต Random Number เหตุผลที่ทำเช่นนี้เพราะต้องการตัดทอนการคำนวณลงให้เหลือเท่าที่จำเป็นและไม่ให้มีการหารตัวเลขที่ตัวหารไม่ใช่จำนวนเท่าของ 2 ในโปรแกรม การใช้ตารางสัญญาณเสียงรบกวน นอกจากจะช่วยลดการคำนวณแล้วยังสามารถช่วยในการทดลองผลิตเสียงด้วยเสียงรบกวนแบบต่างๆ

ตามโครงสร้างของซีพียู TMS 32010 โปรแกรมจะทำงานได้เร็วที่สุดก็ต่อเมื่อย้ายค่าที่เกี่ยวข้องในการคำนวณจากภายนอกตัวซีพียู ซึ่งอาจจะมาจากหน่วยความจำของภาคประมวลผลสัญญาณหรือจากอินพุทพอร์ทมาอยู่ใน RAM ซึ่งอยู่ภายในตัวซีพียูที่เรียกว่า แรมข้อมูล (Data RAM) ตัวเลขและค่าคงที่ต่างๆ ของโปรแกรมจะถูกย้ายจากหน่วยความจำมาอยู่ในดาต้าแรมด้วยคำสั่ง TBLR หรือ Table Read ก่อนการคำนวณ

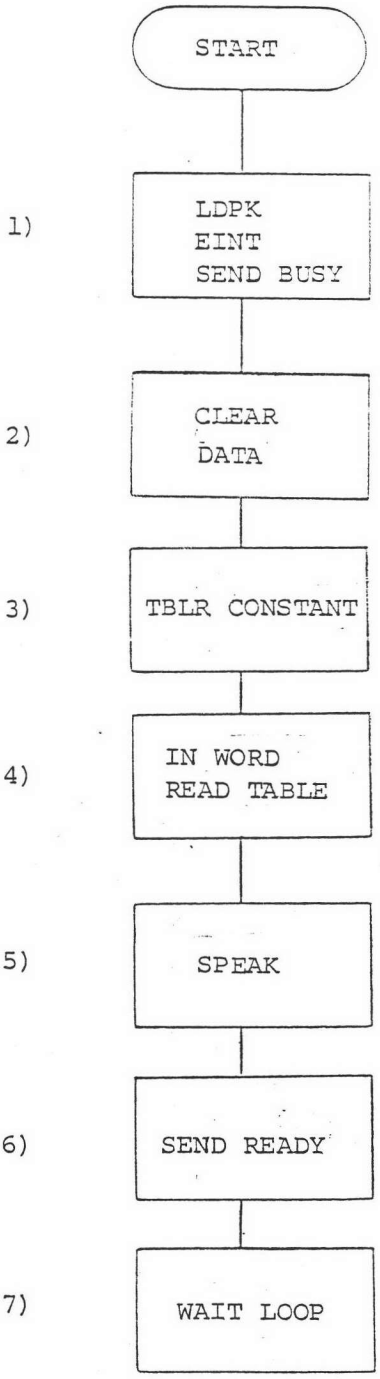
ตำแหน่ง ในหน่วยความจำ	ข้อมูล	หมายเหตุ
300H	XXXH	
301H	YYYH	ตารางคำศัพย์
¦	¦	
¦	¦	
XXXH	K	
¦	¦	ข้อมูลเสียงคำแรก *
¦	¦	
¦	¦	
YYYH	K	
¦	¦	ข้อมูลเสียงคำที่สอง *
¦	¦	
¦	¦	
	¦	
	¦	* ดูรูป 4.9 ในหัวข้อ 4.2.1
	¦	
	¦	

รูป 4.11 การจัดข้อมูลเสียงพูดสำหรับโปรแกรม SPX

โปรแกรม SPX สามารถทำการสังเคราะห์เสียงในลักษณะ Real-Time ความถี่ในการสุ่มสัญญาณเท่ากับ 10 kHz จำนวนออร์เดอร์ของฟิลเตอร์สูงสุดที่กำหนดได้เท่ากับ 15 จำนวนแซมเปิลในหนึ่งเฟรมไม่เกิน 256 แซมเปิล การคำนวณมีความละเอียด 16 บิท คำนวณในลักษณะของเลขจำนวนเต็ม คำตอบที่ได้ถูกหารให้เหลือ 12 บิทและมีการเลื่อนให้มีค่า Offset เพื่อให้สามารถป้อนสู่ D/A Converter ในภาคอนาลอกอินเตอร์เฟซได้ทันที การทำงานของโปรแกรม SPX สามารถอธิบายเป็นส่วนๆ ได้ดังนี้

1. <u>โปรแกรมหลัก (MAIN PROGRAM)</u> รูป 4.12 แสดงแผนผังการทำงานของโปรแกรมหลัก ขั้นตอนการทำงานมีดังนี้

1. คำสั่งเกี่ยวกับการเริ่มต้นทำงาน เช่น เลือกหน้าของดาต้าแรม (LDPK) และคำสั่ง Enable อินเตอร์รัพท์ (EINT) รวมทั้งส่งรหัส Busy ออกมาที่พอร์ทข้อมูล เพื่อให้มาสเตอร์คอมพิวเตอร์รู้ว่าการสังเคราะห์เสียงกำลังดำเนินอยู่
2. เคลียร์ข้อมูลในแรมข้อมูล ซึ่งหมายถึงให้ตัวแปรต่างๆ มีค่าเท่ากับ 0
3. ย้ายค่าคงที่ของโปรแกรมจากหน่วยความจำมาอยู่ในแรมข้อมูล
4. อ่านโคดคำพูดที่ต้องการจากพอร์ทข้อมูล แล้วเปลี่ยนให้เป็นตำแหน่งในหน่วยความจำโดยการอ่านตารางคำศัพท์ ตำแหน่งดังกล่าวจะเป็นตำแหน่งแรกของข้อมูลเสียงคำที่ต้องการ
5. เข้าสู่ส่วนของโปรแกรมสำหรับออกเสียงพูด หรือ SPEAK
6. ส่งรหัส Ready ออกพอร์ทข้อมูลเพื่อให้มาสเตอร์คอมพิวเตอร์รู้ว่า พูดจบแล้ว
7. รอการสั่งการเพื่อพูดคำต่อไป



รูป 4.12 แผนผังการทำงานของโปรแกรมหลัก (MAIN PROGRAM)

2. ส่วนของโปรแกรมสำหรับออกเสียงพูด (SPEAK) รูป 4.13 แสดงแผนผังการทำงานส่วนของโปรแกรมสำหรับออกเสียงพูด ขั้นตอนการทำงานมีดังนี้

1. จากตำแหน่งในหน่วยความจำของข้อมูลเสียงพูด (ข้อ 4 ในโปรแกรมหลัก) อ่านจำนวนเฟรม K จำนวนแซมเปิลในหนึ่งเฟรม N และจำนวนออร์เดอร์ของฟิลเตอร์ M ข้อมูลเหล่านี้อยู่ในส่วนต้นของข้อมูลเสียงหรือเรียกว่า HEAD DATA คำนวณตำแหน่งในหน่วยความจำของข้อมูลเสียงในเฟรมแรก จากนั้นคำนวณค่าที่ใช้ในการหน่วงเวลาของ TIMER 1 เพื่อให้ได้ความถี่ในการสุ่มเท่ากับ 10 kHz

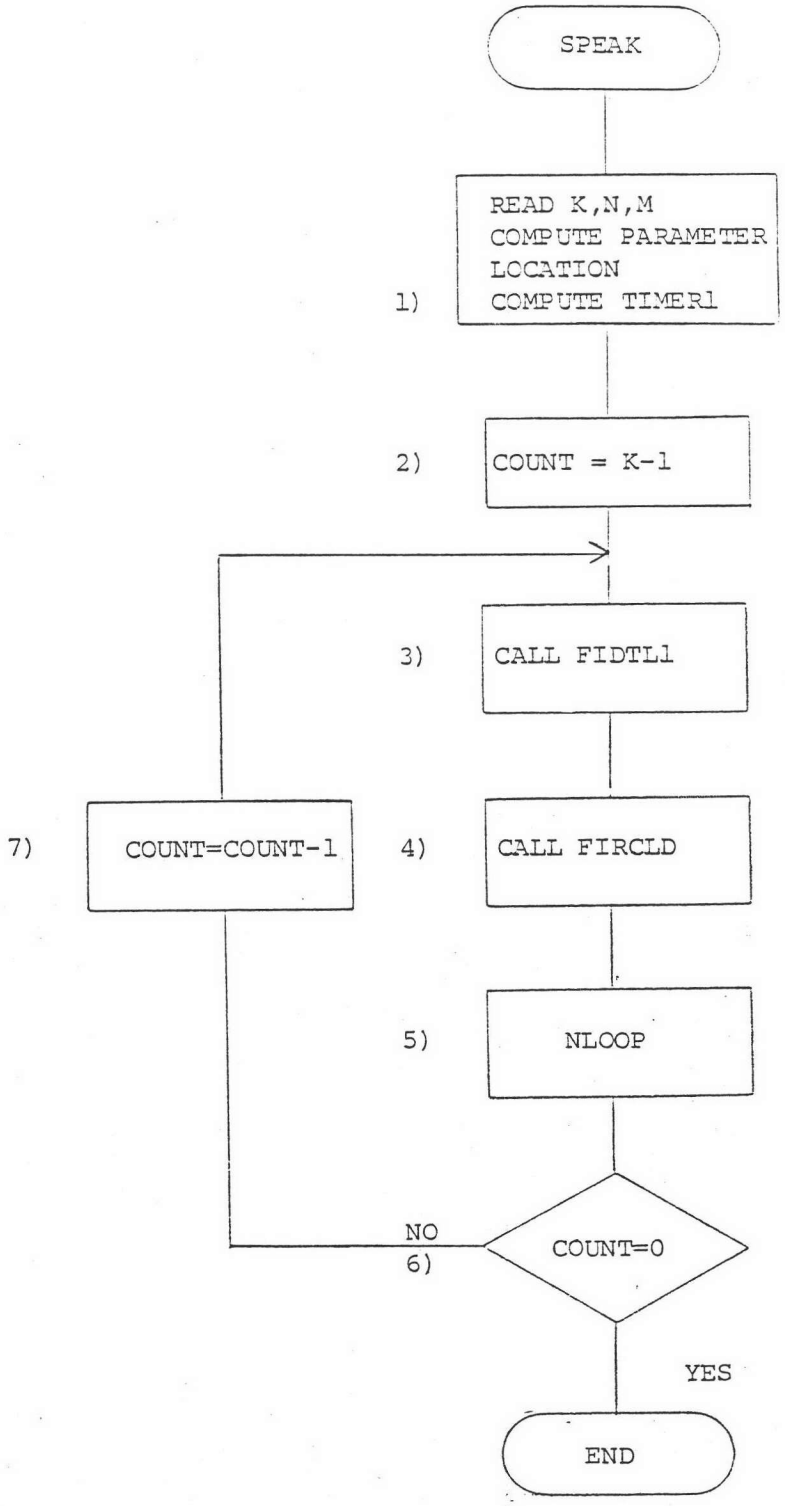
2. เริ่มการนับจำนวนเฟรม

3. คำนวณหาตำแหน่งในหน่วยความจำข้อมูลเสียงเฟรมปัจจุบัน จากนั้นคำนวณ Gain ของสัญญาณ Excitation โดยการคูณกับค่าคงที่ในโปรแกรม ส่วนนี้ใช้สำหรับการทดลองเพื่อกำหนดให้ขนาดของสัญญาณไม่เกิด Over Flow และทดลองเกี่ยวกับอัตราส่วนขนาดของสัญญาณเสียงรบกวนกับพัลส์ ทั้งหมดนี้อยู่ในโปรแกรมย่อย FIDTL1

4. อ่านค่าสัมประสิทธิ์พาร์คอร์ของเฟรมจนครบจำนวนออร์เดอร์โดยโปรแกรมย่อย FIRCLD

5. เข้าสู่ส่วนของโปรแกรม NLOOP เพื่อคำนวณสัญญาณเสียงแต่ละแซมเปิล

6. ถ้าจำนวนเฟรมที่คำนวณมาแล้วยังไม่ครบคำพูดให้กลับไปเริ่มคำนวณในข้อ 3. จนกว่าจะครบจำนวนเฟรม หรือออกเสียงจนครบคำพูด



รูป 4.13 แผนผังการทำงานส่วนของโปรแกรมสำหรับออกเสียงพูด (SPEAK)

3. ส่วนของโปรแกรมสำหรับคำนวณเสียงแต่ละแซมเปิล (NLOOP)

ขั้นตอนการทำงานมีดังนี้

1. เริ่มนับจำนวนแซมเปิล

2. ตรวจสอบว่าเสียงเป็นเสียงก้องหรือไม่ก้องจากค่าของคาบ ถ้าคาบมีค่าเป็น 0 คือเสียงเป็นเสียงไม่ก้อง ไปทำงานต่อข้อ 3.

3. ผลิตเสียงสัญญาณพัลส์ที่มีช่วงห่างเท่ากับคาบของเสียง ขนาดของพัลส์ได้จากการคำนวณข้อ 3. ใน SPEAK

4. เรียกโปรแกรมย่อย LATFIL ซึ่งทำหน้าที่คำนวณฟิลเตอร์สังเคราะห์ การทำงานเหมือนกับโปรแกรมย่อย TWCMUL ในหัวข้อ 2.5 หลังจากคำนวณเสร็จปรับค่าที่ได้ให้เป็นข้อมูล 12 บิทและส่งออกไปยัง D/A Converter ด้วยคำสั่ง Out Port 7

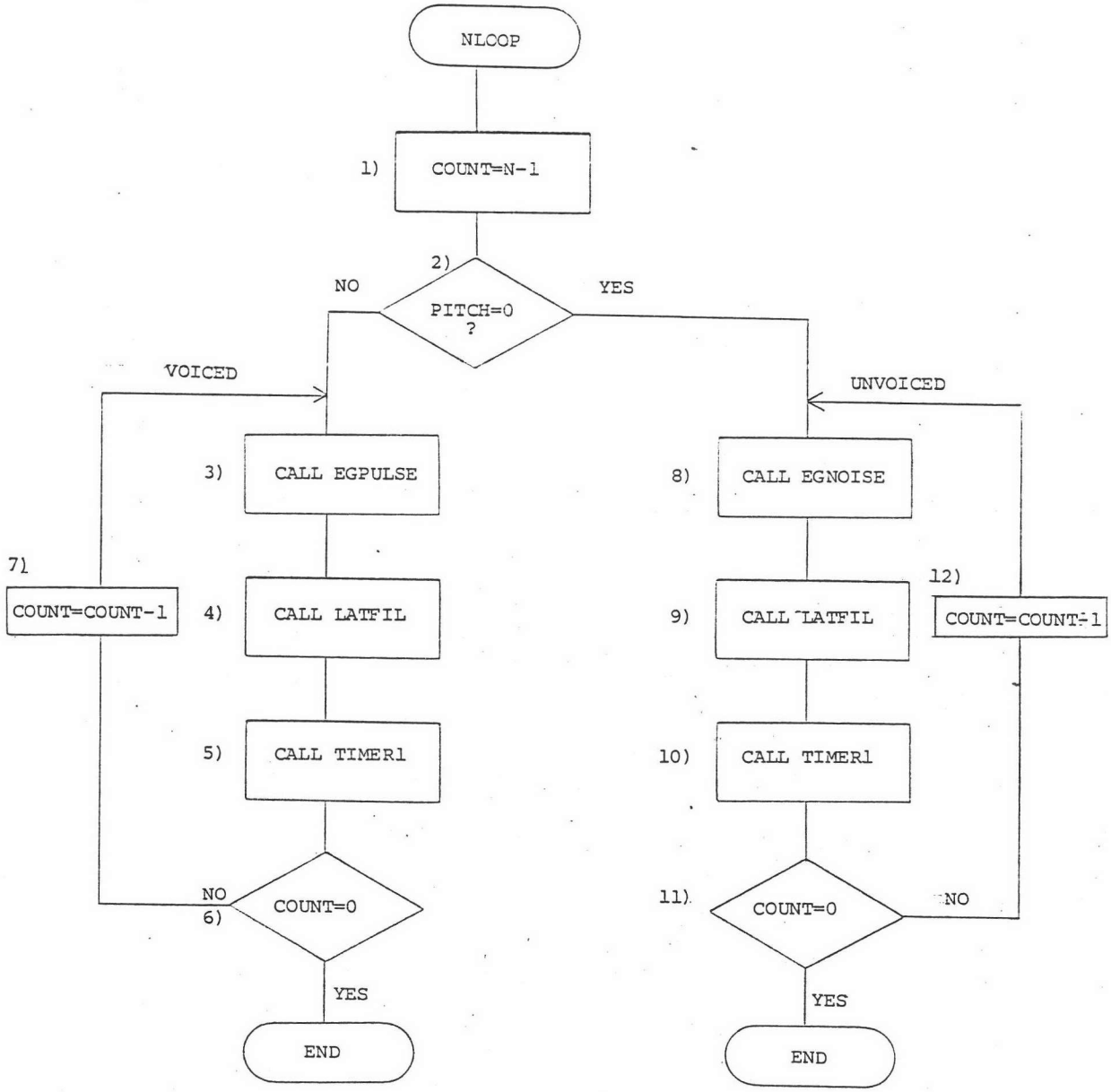
5. หน่วงเวลาให้ได้ค่าจากการคำนวณแต่ละแซมเปิลทุกๆ 100 μS

6.-7. คำนวณสัญญาณแต่ละแซมเปิลจากข้อ 3. จนครบเฟรม

8. ผลิตสัญญาณเสียงรบกวนโดยอ่านตารางสัญญาณเสียงรบกวน การอ่านตารางมีการเลื่อนไปทุกๆ เฟรม ขนาดของสัญญาณได้จากการคำนวณข้อ 3 ใน SPEAK

9.-12. ทำงานเช่นเดียวกับข้อ 4.-7.

การสั่งการทำงานจากมาสเตอร์คอมพิวเตอร์กระทำได้โดยอ่านข้อมูลจากพอร์ทข้อมูลแล้วตรวจสอบว่าเป็นรหัส Busy หรือ Ready ถ้าเป็นรหัส Ready ก็สามารถส่งรหัสของคำพูดที่ต้องการซึ่งเป็นตัวเลข 1 ถึง 10 หรืออื่นๆ ตามจำนวนคำศัพท์ที่มีอยู่ไปที่พอร์ทข้อมูล แล้วส่งพัลส์อินเตอร์รัพท์หรือสัญญาณรีเซทไปยังภาคประมวลผลสัญญาณ ซึ่งภาคประมวลผลสัญญาณจะออกเสียงคำพูดนั้น



รูป 4.14 แผนผังการทำงานส่วนของโปรแกรมสำหรับคำนวณสัญญาณเสียง (NLOOP)