สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ตู้ชุมสายโทรศัพท์อัตโนมัติ

นางสาวชนิตา ใจแสวง
นายอมร อมรศิลป์

ว/ท.
ส/153 ก1
2537
เลขหมู่.....
เลขทะเบียน.....
วัน.เดือน.ปี.....

b12554091

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

PRIVATE AUTOMATIC BRANCH EXCHANGE

(PABX)

Miss. Shanita Jaisaweang

Mr. Amorn Amornsilpa

A Special Project Submitted in Partial Fulfillment of the Requirment

of the Degree of Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

1994

หัวข้อโครงงานพิเศษ	ตู้ชุมสายโทรศัพท์อัตโนมัติ
โดย	นางสาวชนิตา ใจแสวง
	นายอมร อมรศิลป์
ภาควิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	อาจารย์.วิชิต ศิริโชติ
	อาจารย์.ธีรวัฒน์ ประกอบผล

ภาควิชาฟิสิกส์ประยุกต์

คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

อนุมัติให้นับโครงงานพิเศษฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

..... หัวหน้าภาควิชาฟิสิกส์ประยุกต์

(ผศ.ดร. ปรีชา เทียนสมประสงค์)

คณะกรรมการโครงงานพิเศษ

..... ประธานกรรมการ

(อาจารย์.รัชภาคย์ จิตต์อารี)

..... กรรมการ

(ผศ. สุวรรณ คูสำราญ)

..... กรรมการ

(อาจารย์.ธีรวัฒน์ ประกอบผล)

..... กรรมการ

(ผศ.ดร. วราวุฒิ เถาลัดดา)

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงงานพิเศษ	ตู้ชุมสายโทรศัพท์อัตโนมัติ
โดย	นางสาวชนิตา ใจแสวง
	นายอมร อมรศิลป์
อาจารย์ที่ปรึกษา	อาจารย์วิชิต ศิริโชติ
	อาจารย์ธีรวัฒน์ ประกอบผล
ภาควิชา	ฟิสิกส์ประยุกต์
ปีการศึกษา	2537

บทคัดย่อ

ตู้ชุมสายโทรศัพท์อัตโนมัติ เป็นอุปกรณ์ที่ช่วยอำนวยความสะดวกในการติดต่อสื่อสารระหว่างโทรศัพท์ภายนอกหรือโทรศัพท์ขององค์การโทรศัพท์กับโทรศัพท์ภายในสำนักงาน โดยไม่จำเป็นต้องผ่านโอเปอร์เรเตอร์ ตู้ชุมสายนี้สามารถขยายคู่สายโทรศัพท์ภายในจากสายโทรศัพท์ภายนอกได้เป็นจำนวนมาก ซึ่งจะช่วยให้การติดต่อสื่อสารภายในสำนักงานขนาดใหญ่เป็นไปได้อย่างสะดวก รวดเร็ว และช่วยประหยัดค่าใช้จ่ายอันเนื่องจากการติดต่อระหว่างภายในสำนักงานเดียวกันนั้นโดยไม่ต้องผ่านชุมสายขององค์การโทรศัพท์ ตู้ชุมสายโทรศัพท์อัตโนมัติที่นำเสนอในโครงงานนี้จะเป็นการเชื่อมต่อระหว่างสายโทรศัพท์ภายนอก 2 สาย กับสายโทรศัพท์ภายใน 4 สาย โดยมี SWITCH UNIT เป็นตัวเชื่อมต่อระหว่างสายภายในกับสายภายนอก ซึ่ง SWITCH UNIT นี้จะเป็นอนาลอกสวิทช์ที่ควบคุมโดยวงจรดิจิตอล เพื่อเชื่อมต่อกับสัญญาณเสียง และสัญญาณควบคุม และใช้ไมโครคอนโทลเลอร์ในการควบคุมการทำงานของระบบ

Special Project Title	Private Automatic Branch eXchange (PABX)
Name	Miss. Shanita Jaisaweang
	Mr. Amorn Amornsilpa
Special Project Advisor	Mr Wichit Sirichote
	Mr. Teerawat Prakobphol
Department	Applied Physics
Academic Year	1994

Abstract

Private Automatic Branch Exchange (PABX) is an automatic device used to connect external (Trunks) and internal lines without operators. The connections made by the device can be performed in numerous ways which , as a result , give rise to convenient communication services. In this project , SC11390 chip as an analog switch in the PABX is used to control connections between two external and four internal lines.

กิตติกรรมประกาศ

โครงงานพิเศษนี้ เสร็จสมบูรณ์ด้วยดีได้ ด้วยความช่วยเหลือจากบุคคลต่างๆดังนี้

คุณพ่อและคุณแม่	ที่เป็นแรงบันดาลใจ ทั้งส่งเสริมและสนับสนุนในทุกสิ่งทุกอย่าง
อ. วิชิต ศิริโชติ	ที่ให้คำปรึกษาในการทำโครงงานพิเศษ
อ. ศิริศักดิ์ เตชะทวีกุล	ที่ให้คำปรึกษาและความช่วยเหลือในการจัดซื้ออุปกรณ์
อ. ธีรศักดิ์ ประกอบผล	ที่ให้คำปรึกษาและเอื้อเฟื้ออุปกรณ์ต่างๆ

คณาจารย์ภาควิชาฟิสิกส์ประยุกต์ ที่อบรม ชี้แนะตลอดระยะเวลา 4 ปี

และขอขอบคุณเพื่อนๆทุกคนที่ให้กำลังใจ ช่วยเหลือในด้านต่างๆ มาตลอด

นางสาวชนิตา ใจแสวง

นายอมร อมรศิลป์

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการทำงาน	4
2.1 ระบบโทรศัพท์	4
2.1.1 การทำงานของโทรศัพท์เบื้องต้น	4
2.1.2 เครื่องส่งโทรศัพท์	4
2.1.3 เครื่องรับโทรศัพท์	5
2.1.4 ระบบชุมสายโทรศัพท์	6
2.1.5 ระบบสวิทชิ่งชนิดต่างๆ	8
2.1.6 ระบบโทรศัพท์ที่ใช้ในปัจจุบัน	8
2.1.7 ข้อเปรียบเทียบระหว่างระบบพัลซ์และระบบความถี่	9
2.1.8 สัญญาณโทรศัพท์มาตราฐาน	10
2.1.9 การทำงานของโทรศัพท์	13
2.2 ตู้ชุมสายโทรศัพท์อัตโนมัติ	16
2.2.1 ลักษณะการทำงานของระบบ PABX	16
บทที่ 3 หลักการทำงานและการออกแบบวงจรของตู้ชุมสายโทรศัพท์อัตโนมัติ	20
3.1 โครงสร้างทางด้านฮาร์ดแวร์	20
3.2 - การออกแบบวงจรในส่วนต่างๆของระบบ PABX	24
- แผนผังการทำงานของระบบ PABX	36
3.3 โครงสร้างทางด้านซอฟท์แวร์	40

บทที่ 4 ผลการทดลอง 42

บทที่ 5 สรุปผลการทดลองและแนวทางการพัฒนา 44

ภาคผนวก 45

SC11390 SWITCH UNIT 45

ANT-32 53

เอกสารอ้างอิง 55

ประวัติผู้เขียน 56

สารบัญตาราง

		หน้า
ตารางที่ 1	แสดงการแบ่งกลุ่มความถี่เรียงตามคีย์โทรศัพท์	9
ตารางที่ 2	แสดงการเซตโหมดการทำงานของระบบ PABX	18
ตารางที่ 3	แสดงพื้นที่ของหน่วยความจำของระบบ PABX	41
ตารางที่ 1-ก	แสดงความถี่ DTMF	49
ตารางที่ 2-ก	แสดงความถี่การเข้ารหัสและถอดรหัสของระบบ DTMF	50
ตารางที่ 3-ก	แสดงการเข้ารหัสของเสียงกริ่งโทรศัพท์	50

สารบัญรูป

		หน้า
รูปที่ 2-1	แสดงการทำงานของเครื่องส่งโทรศัพท์เบื้องต้น	4
รูปที่ 2-2	แสดงวงจรโทรศัพท์อย่างง่าย	5
รูปที่ 2-3	การต่อวงจรเมื่อมีโทรศัพท์ 2 เครื่อง	6
รูปที่ 2-4	การต่อวงจรเมื่อมีโทรศัพท์หลายเครื่อง	6
รูปที่ 2-5	การต่อวงจรเมื่อมีโทรศัพท์ในหลายท้องถิ่น	7
รูปที่ 2-6	แสดงสัญญาณ Dial Tone	10
รูปที่ 2-7	แสดงสัญญาณ Busy Tone	11
รูปที่ 2-8	แสดงสัญญาณ Ring Back Tone	11
รูปที่ 2-9	แสดงสัญญาณ Ringing Tone	12
รูปที่ 2-10	แสดงการทำงานของโทรศัพท์โดยทั่วไป	13
รูปที่ 2-11	แสดงการทำงานของโทรศัพท์โดยผ่านชุมสาย	14
รูปที่ 2-12	แสดงการทำงานของโทรศัพท์โดยผ่านชุมสายหลายชุมสาย	15
รูปที่ 2-13	แสดงตัวอย่างการทำงานของระบบ PABX	19
รูปที่ 3-1	วงจรควบคุมการทำงานของระบบ	25
รูปที่ 3-2	วงจรแหล่งจ่ายไฟ	26
รูปที่ 3-3	วงจรบัฟเฟอร์	27
รูปที่ 3-4	วงจรของ SWITCH UNIT	28
รูปที่ 3-5	วงจรกำเนิดเสียงโทรศัพท์	29
รูปที่ 3-6	วงจรกำเนิดเสียงสำหรับโทรศัพท์	30
รูปที่ 3-7-1	วงจรของสายโทรศัพท์สายนอกสายที่ 1	31
รูปที่ 3-7-2	วงจรของสายโทรศัพท์สายนอกสายที่ 2	32
รูปที่ 3-8-1	วงจรของสายโทรศัพท์สายในสายที่ 1	33
รูปที่ 3-8-2	วงจรของสายโทรศัพท์สายในสายที่ 2	34
รูปที่ 3-8-3	วงจรของสายโทรศัพท์สายในสายที่ 3	35
รูปที่ 3-8-4	วงจรของสายโทรศัพท์สายในสายที่ 4	36
รุปที่ 3-9	แสดงโครงงานพิเศษ PABX	44

รูปที่ ก-1	แสดงลักษณะของ SWITCH UNIT แบบ PLCC	46
รูปที่ ก-2	แสดงลักษณะของ SWITCH UNIT แบบ PQFP	47
รูปที่ ก-3	แสดงลักษณะของสัญญาณอินพุทและเอาท์พุทของเมตริกซ์	51
รูปที่ ก-4	แสดงถึงอินพุทและเอาท์พุทของเมตริกซ์	52
รูปที่ ก-5	แสดงส่วนประกอบต่างๆของบอร์ด ANT-32	54

บทที่ 1
บทนำ

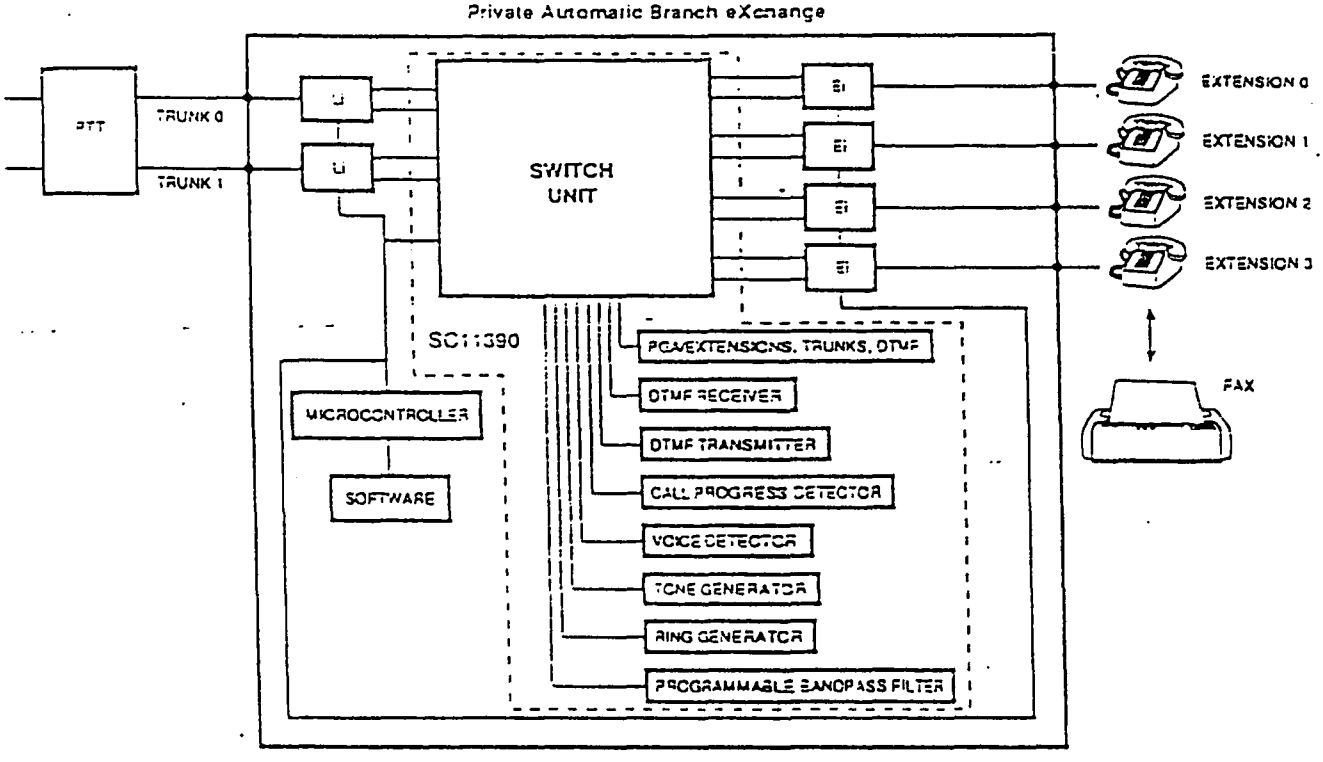
ตู้สาขาโทรศัพท์อัตโนมัติ (Private Automatic Branch Exchange ; PABX)

โทรศัพท์เป็นวิธีการติดต่อสื่อสารพื้นฐานในสังคมปัจจุบัน การใช้โทรศัพท์ในยุคแรกๆ นั้น ถ้าเราต้องการจะติดต่อจากบ้านไปที่สำนักงานก็จะต้องติดโทรศัพท์เครื่องหนึ่งไว้ที่สำนักงานอีกเครื่องหนึ่งติดไว้ที่บ้าน หากต้องการโทรศัพท์ไปยังร้านขายของก็ต้องมีโทรศัพท์อีกเครื่องหนึ่งต่อไปยังโทรศัพท์ที่ร้านขายของซึ่งเป็นวิธีที่ค่อนข้างยุ่งยากมาก จึงได้มีการนำเอาชุมสายโทรศัพท์ (Telephone Exchange)มาใช้ โดยโทรศัพท์ทุกเครื่องจะต่อเข้าชุมสาย ชุมสายแบบแรกนั้นจะมีพนักงานสลับสายหรือโอเปอเรเตอร์เป็นผู้ต่อให้ โดยรับโทรศัพท์จากผู้ที่เรียกเข้ามาแล้วสอบถามว่าต้องการต่อไปปลายทางที่ใด ก็จะใช้สายต่อซึ่งมีแจ็คอยู่ที่ปลายทั้งสองด้านต่อระหว่างต้นทางกับปลายทาง

เมื่อมีผู้ใช้บริการโทรศัพท์มากขึ้นก็เกิดปัญหาในแง่ของความรวดเร็วและความถูกต้อง จึงได้มีการสร้างเครื่องชุมสายโทรศัพท์แบบอัตโนมัติ (Automatic Switching System)ในระบบ step-by-step ขึ้นมา เมื่อผู้ใช้ยกหูโทรศัพท์ก็จะได้ยินสัญญาณให้หมุนผู้ใช้ก็จะหมุนหมายเลขที่ต้องการติดต่อแล้วเครื่องชุมสายก็จะต่อไปปลายทางให้ จากนั้นก็มีการพัฒนามาเรื่อยๆมาเป็น Crossbar และมาเป็นเครื่องชุมสายระบบดิจิตอล เอสพีซี (Digital Stroed Program Control) ซึ่งอยู่ในความดูแลขององค์การโทรศัพท์แห่งประเทศไทย

ในปัจจุบันโทรศัพท์มีความสำคัญมากในแง่ของธุรกิจ ในสำนักงานใหญ่ๆที่มีสถานที่มาก พนักงานจำนวนมาก หากมีการขอใช้โทรศัพท์ขององค์การโทรศัพท์จะทำให้เสียค่าใช้จ่ายมากและหากเป็นการติดต่อระหว่างภายในสำนักงานเองแล้วก็จะเป็นการสิ้นเปลืองมากขึ้น จึงได้มีการนำตู้สาขามาใช้และพัฒนามาเป็นตู้สาขาอัตโนมัติแบบดิจิตอล (Digital Private Automatic Branch Exchange, Digital PABX) โดยตู้สาขานี้เหมาะสำหรับสำนักงานที่มีขนาดใหญ่ มีพนักงานจำนวนมาก ซึ่งมีข้อดีคือ ลดค่าใช้จ่ายของสำนักงานที่เกิดจากการใช้โทรศัพท์ลงและยังสามารถควบคุมการใช้โทรศัพท์ทางไกลได้อีกด้วย

แผนผังระบบการทำงานของ PABX



หลักสำคัญของระบบ PABX คือ Switch Unit ซึ่งจะใช้ในการเชื่อมต่อสายโทรศัพท์ภายในกับสายโทรศัพท์ภายนอก และจะต้องมีการเชื่อมต่อกับอุปกรณ์ควบคุมต่างๆ เช่น ตัวรับและตัวส่งแบบ DTMF , ตัวตรวจจับการเรียกเข้า (Call Progress Detector) , ตัวตรวจจับเสียง (Voice Detector) , เครื่องกำเนิดความถี่ (Tone Generator) และ เครื่องกำเนิดเสียงโทรศัพท์ (Ring generator) เป็นต้น โดยที่อุปกรณ์ควบคุมเหล่านี้จำเป็นต้องมีการเชื่อมต่อกับสายภายในเพื่อที่จะรับ-ส่งข้อมูลจากสายใดสายหนึ่งไปยังสายที่ต้องการติดต่อได้

จุดประสงค์ของโครงงานพิเศษ

- เพื่อศึกษาการทำงานของตู้สาขาโทรศัพท์อัตโนมัติ
- เพื่อให้ผู้ใช้บริการสามารถติดต่อสื่อสารกันได้ตามที่ต้องการ โดยไม่จำเป็นต้องมีการส่งผ่านโอเปอเรเตอร์
- เพื่อลดค่าใช้จ่ายในการขอใช้บริการโทรศัพท์จากองค์การโทรศัพท์ เมื่อจำเป็นต้องใช้คู่สายโทรศัพท์จำนวนมาก
- เพื่อสามารถเชื่อมโยงโทรศัพท์ที่ติดตั้งอยู่ภายในสำนักงานได้โดยไม่ต้องผ่านชุมสายโทรศัพท์ภายนอก ซึ่งเป็นการช่วยลดค่าใช้จ่ายภายในสำนักงาน

ขั้นตอนการทำโครงงานพิเศษ

- ศึกษาทฤษฎีและหลักการทำงานของระบบโทรศัพท์
- ศึกษาทฤษฎีและหลักการทำงานของตู้สาขาโทรศัพท์อัตโนมัติ
- ศึกษาทฤษฎีและหลักการทำงานของ Switch Unit
- ศึกษาการทำงานของไมโครคอนโทลเลอร์
- ออกแบบวงจรการทำงานของตู้สาขาโทรศัพท์อัตโนมัติในแต่ละส่วน
- เขียนโปรแกรมควบคุมการทำงานของระบบตู้สาขาโทรศัพท์อัตโนมัติ

บทที่ 2
ทฤษฎีและหลักการทำงาน

2.1 ระบบโทรศัพท์

2.1.1 การทำงานของโทรศัพท์เบื้องต้น

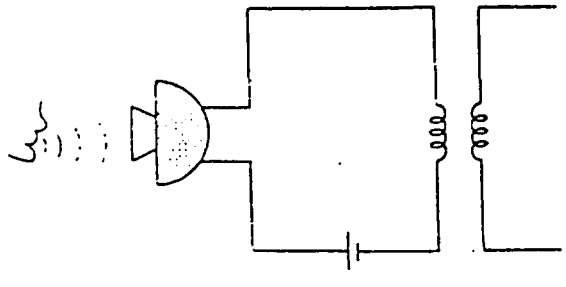
โทรศัพท์เป็นเครื่องมือทางไฟฟ้าที่ใช้ในการรับและส่งข้อมูลข่าวสาร ชุดโทรศัพท์จะประกอบด้วย ตัวส่ง ตัวรับ กระดิ่ง หน้าปัดหมุนหมายเลข และ hook-switch โทรศัพท์ทุกเครื่องใช้แรงดันในช่วง -42 ถึง -52 Vdc (typically -48 V)

2.1.2 เครื่องส่งโทรศัพท์

เครื่องส่งจะประกอบด้วยแผ่นไดอะแฟรมที่ยึดติดอยู่ระหว่างสารคาร์บอน เมื่อมีการพูดแรงดันของอากาศจะไปสั่นสะเทือนแผ่นไดอะแฟรม คาร์บอนจะเปลี่ยนค่าความต้านทานแปรไปตามค่าแรงอัดของโมเลกุลอากาศ ซึ่งเกิดจากเสียงพูดนั่นเอง จากรูปที่ 2-1 เมื่อต่อวงจรดังรูปจะทำให้สัญญาณที่ได้จากขดทุติยภูมิของหม้อแปลงเป็นแรงดันไฟสลับตามคลื่นเสียง สรุปได้ว่า

1. เสียงซึ่งเกิดจากการสั่นสะเทือนของอากาศ จะถูกเปลี่ยนไปเป็นการเปลี่ยนแปลงทางไฟฟ้า
2. ข่าวสาร ข้อมูล ส่งไปในรูปของไฟ ac ที่ควบอยู่บนแรงดันไฟ dc
3. กระแสไฟฟ้าที่เปลี่ยนแปลงไปมาจะแปรผันตามเสียงของผู้พูด

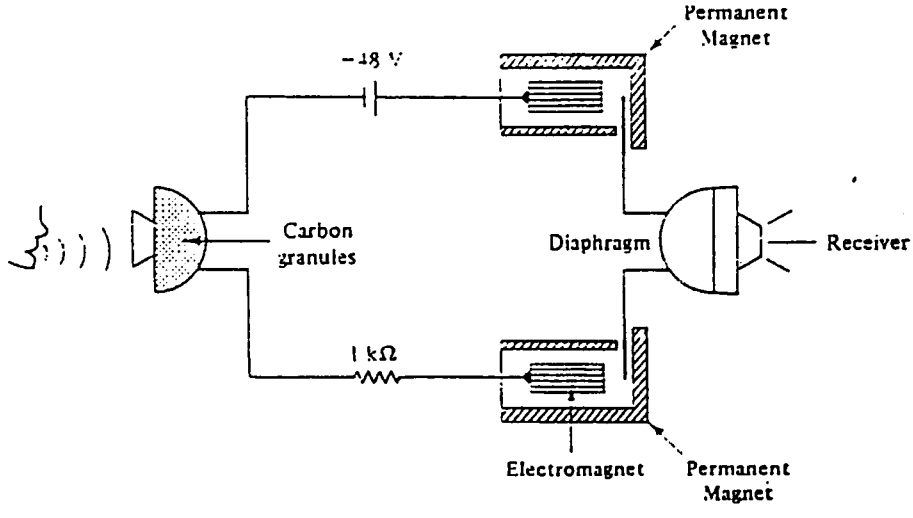
รูปที่ 2-1 แสดงการทำงานของเครื่องส่งโทรศัพท์เบื้องต้น



2.1.3 เครื่องรับโทรศัพท์

การทำงานของตัวรับโทรศัพท์ จะทำงานตรงข้ามกับตัวส่ง ภายในตัวรับจะมีลักษณะเป็นแผ่นโลหะบางๆที่ยืดหยุ่นง่าย ซึ่งการเคลื่อนไหวจะควบคุมและขึ้นอยู่กับระหว่างแม่เหล็กสองชนิดคือ แม่เหล็กถาวร (permanent magnet) และแม่เหล็กไฟฟ้า (eletromagnet) ซึ่งได้จากขดลวดพันรอบแกนเหล็กชิ้นเล็กๆ เมื่อมีกระแสไหลผ่านขดลวดจะทำให้เกิดการสั่นสะเทือน ซึ่งเป็นผลจากกระแสในสายที่มาจากเครื่องส่ง การสั่นสะเทือนนี้ก็จะแปรตามไดอะแฟรมของเครื่องส่งหรือตามเสียงของคนที่พูดนั่นเอง การสั่นสะเทือนของไดอะแฟรมของตัวรับ (receiver) นี้ จะทำให้เกิดเสียงพูดเหมือนตัวเครื่องส่งจึงเกิดการติดต่อระหว่างผู้รับและผู้ส่งได้ นอกจากนี้ยังต้องมีตัวทำให้สัญญาณตัวส่งและตัวรับส่งไปในสายเดียวกันได้เรียกว่า hybrid ซึ่งจะทำหน้าที่เป็น balancing เพื่อป้องกันสัญญาณย้อนกลับเข้าไปในตัวรับซึ่งจะเป็นผลไม่ให้เกิดเสียงสะท้อน (echo)ได้

รูปที่ 2-2 แสดงวงจรโทรศัพท์อย่างง่าย

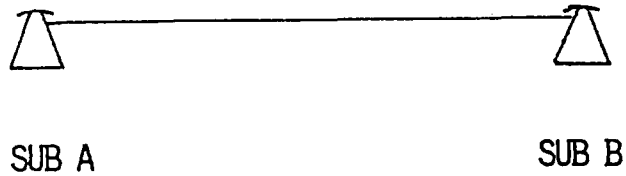


2.1.4 ระบบชุมสายโทรศัพท์

เหตุผลที่ต้องมีเลขหมายโทรศัพท์และชุมสายโทรศัพท์ สามารถอธิบายได้ดังนี้

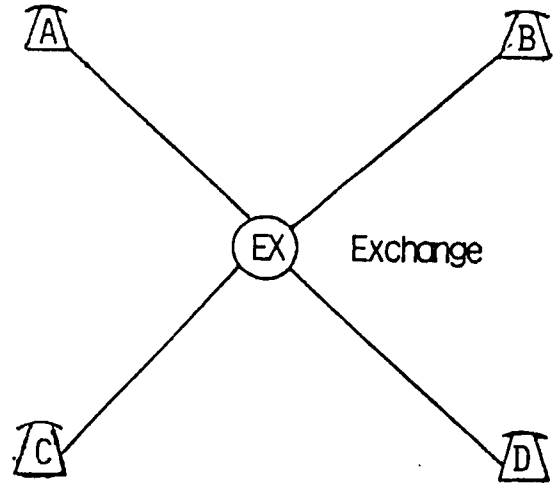
- เมื่อมีโทรศัพท์ 2 เครื่อง เราสามารถต่อเพื่อใช้งานสนทนากัน 2 คนได้ดังรูป

รูปที่ 2-3 การต่อวงจรเมื่อมีโทรศัพท์ 2 เครื่อง



- เมื่อมีโทรศัพท์เพิ่มจำนวนมากขึ้น ต้องมีสวิทช์คอยตัด-ต่อวงจรให้สนทนาได้เป็นคู่ๆ เช่น จากรูปที่ 4 ขณะที่ A พูดกับ B C จะไม่สามารถติดต่อกับ D ได้

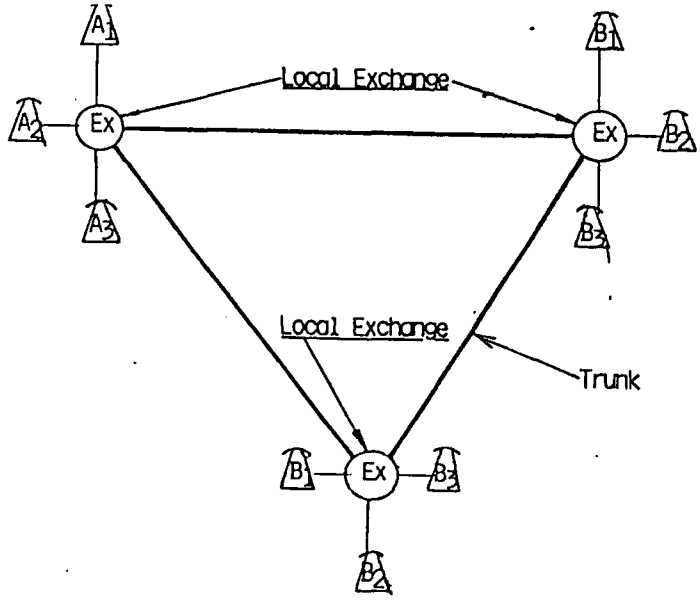
รูปที่ 2-4 การต่อวงจรเมื่อมีโทรศัพท์หลายเครื่อง



อุปกรณ์ที่คอยตัดต่อสัญญาณให้สมาชิก (subscriber) ติดต่อกันได้คือชุมสายโทรศัพท์ (exchange) สมาชิกแต่ละเครื่องจะมีรหัสชื่อที่กำหนดเป็นชื่อเรียกของตนเอง รหัสนี้เรียกว่า " เลขหมาย" (Telephone Number) สมมติ A ต้องการติดต่อกับ B เครื่อง A จะส่งสัญญาณที่ใช้แสดงถึงเลขหมายของ B พนักงานที่ชุมสายหรือสวิทช์อัตโนมัติจะรับรู้ความหมายของสัญญาณนี้ จะส่งสัญญาณกระดิ่งไปเรียก B เมื่อมีคนมารับสายที่ชุมสายจะต่อวงจรให้ A และ B สนทนากันได้ เพราะฉะนั้นเลขหมายคือชื่อเรียกเฉพาะของคู่สาย

- เมื่อมีการติดต่อกันในระยะไกลๆ เราสามารถนำชุมสายของแต่ละท้องถิ่นมาเชื่อมต่อกันได้ดังรูปที่ 2-5

รูปที่ 2-5 การต่อวงจรเมื่อมีโทรศัพท์ในหลายท้องถิ่น



จะต้องมีการกำหนดรหัสของแต่ละชุมสายเพิ่มขึ้นมา และถ้ามีการเชื่อมโยงเครือข่ายกันในระดับภูมิภาค ก็ต้องมีรหัสเพิ่มขึ้นมาอีกคือ รหัสทางไกล

ชุมสายโทรศัพท์สามารถแบ่งออกเป็น 2 ชนิดใหญ่ๆ ได้แก่

1. ชุมสายที่มีผู้เช่าต่อเข้าโดยตรง ได้แก่ ตู้สาขาและชุมสายท้องถิ่น

- ตู้สาขา (Private Branch Exchange) ได้แก่ PBX และ PABX ที่ถูกนำมาใช้ในสำนักงานฯลฯ เพื่อใช้ติดต่อกันภายในโดยไม่ต้องผ่านเครื่องชุมสายของตู้สาขา นอกจากนี้หมายเลขภายในยังสามารถต่อไปยังชุมสายท้องถิ่นเพื่อเรียกไปยังหมายเลขภายนอกได้ด้วย

- ชุมสายท้องถิ่น (Local Exchange) แบ่งออกเป็น 2 ชนิดคือ

1). Large urban exchanges หมายถึง ชุมสายท้องถิ่นซึ่งเป็น Switching centre สำหรับ traffic จากผู้เช่าภายในพื้นที่ของตัวเอง ชุมสายแบบนี้มีขนาดตั้งแต่ร้อยหมายเลขจนถึงไม่มีจำนวนจำกัด เหมาะสำหรับใช้ในเมืองและชุมชนขนาดใหญ่

2). Rural exchanges หมายถึง ชุมสายที่มีขนาดเล็กมีขนาดตั้งแต่สิบหมายเลขจนถึงพันๆหมายเลข มักใช้ในชุมชนขนาดเล็ก

2. ชุมสายต่อผ่าน (Transit Exchange)

เป็นชุมสายที่รับ traffic ระหว่างชุมสายกับชุมสาย โดยไม่มีผู้เช่าต่อตรงเข้ามายังชุมสาย การทำงานของเครื่องชุมสายต่อผ่านนี้จะต้องทำงานอย่างรวดเร็วเพราะการที่ผู้เรียกเรียกไปยังผู้ถูกเรียกนั้น อาจจะต้องผ่านชุมสายต่อผ่านหลายชุมสาย ชุมสายต่อผ่านแบ่งเป็น 2 ชนิดคือ ชุมสายต่อผ่านท้องถิ่น (Tandem Exchange) และ ชุมสายต่อผ่านทางไกล (Transit Exchange)

2.1.5 ระบบสวิทชิ่ง (Switching system) ชนิดต่างๆ

1. ระบบสวิทชิ่งด้วยมือ (Manual Switching System)

ในระบบสวิทชิ่งแบบนี้การต่อสายสื่อสารในแผงสวิทช์จะทำโดยใช้พนักงานโทรศัพท์เป็นผู้ต่อให้

2. ระบบสวิทชิ่งอัตโนมัติ (Automatic Switching System)

ระบบนี้จะทำงานตามกระแสพัลซ์ของหน้าปัทม์ที่ถูกเหนี่ยวนำให้เกิดขึ้น จากการหมุนหมายเลขบนหน้าปัทม์ของเครื่องโทรศัพท์ โดยจะจับตัวเลขทีละตัวด้วยพัลซ์ของหน้าปัทม์และต่อสายทีละสาย เรียกว่า วิธีการขั้นต่อขั้น (Step By Step)

3. ระบบสวิทชิ่งแบบควบคุมร่วม

เมื่อผู้พูดยกหูโทรศัพท์วงจรควบคุมจะได้รับสัญญาณการทำงานของอุปกรณ์วงจรของผู้เช่า ซึ่งติดตั้งที่ชุมสายเฉพาะสายของผู้เช่าทุกสายและทำให้ทราบหมายเลขของผู้เรียก นั่นคือผู้เรียกจะถูกต่อไปยังรีจิสเตอร์ แล้วรีจิสเตอร์จะส่งสัญญาณให้หมุนเพื่อรอรับหมายเลข เมื่อได้รับสัญญาณจากผู้เรียกแล้ว วงจรควบคุมจะค้นหาตำแหน่งของผู้ถูกเรียกทันที

4. ชุมสายโทรศัพท์ระบบดิจิตอล (SPC:Store Program Control)

ชุมสายแบบนี้จะมีการนำเอาเทคโนโลยีทางคอมพิวเตอร์มาใช้เรียกว่าระบบ SPC ซึ่งมีการทำงานของระบบ duplex เข้ามาใช้

2.1.6 ระบบโทรศัพท์ที่ใช้ในปัจจุบัน

ระบบโทรศัพท์ที่ใช้อยู่ในปัจจุบันมีอยู่ 2 ระบบ คือ

1. โทรศัพท์แบบพัลซ์ เป็นระบบโทรศัพท์ที่ส่งสัญญาณหมายเลขเป็นพัลซ์ โดยที่จำนวนพัลซ์จะเท่ากับจำนวนหมายเลขที่หมุนหรือกด เช่น ถ้าหมุนหมายเลข 5 ก็จะมีพัลซ์ออกมา 5 ลูก ซึ่งโทรศัพท์แบบนี้กำลังจะเลิกใช้แล้ว

2. โทรศัพท์แบบความถี่ (DTMF) เป็นโทรศัพท์ที่นิยมใช้กันในปัจจุบัน โทรศัพท์แบบนี้จะส่งความถี่ 2 ความถี่ไปพร้อมกัน เพื่อแทนหมายเลขหนึ่งหมายเสข ซึ่งมีตารางความถี่ ตามหมายเลขดังตารางที่ 1

ความถี่สูง	Column 1	Column 2	Column 3	Column 4
	1209 Hz	1339 Hz	1477 Hz	1633 Hz
ความถี่ต่ำ				
Row1 679 Hz	1	2	3	A
Row2 770 Hz	4	5	6	B
Row3 852 Hz	7	8	9	C
Row4 941 Hz	*	0	#	D

ตารางที่ 1 แสดงการแบ่งกลุ่มความถี่ (DTMF) เรียงตามคีย์ของโทรศัพท์

โดยทั่วไปมี 12 ปุ่มแบ่งเป็น 4 แถว และ 3 คอลัมภ์ แต่สามารถเพิ่มขึ้นได้อีกถึง 16 ปุ่ม โดยเพิ่มคอลัมภ์ที่ 4 ขึ้นมาอีก ความถี่ที่ใช้แต่ละแถวและคอลัมภ์นั้นจะต่างความถี่กัน ทั้ง 4 แถวเรียกว่า "กลุ่มความถี่ต่ำ" 3 หรือ 4 คอลัมภ์เรียกว่า "กลุ่มความถี่สูง"

ตัวอย่างเช่น ถ้าเรากดหมายเลข 4 ก็จะมีความถี่ออกมา 2 ความถี่ คือ 770 Hz และ 1209 Hz เป็นต้น

2.1.7 ข้อเปรียบเทียบระหว่างระบบพัลซ์และระบบความถี่

- ระบบพัลซ์ ใช้เวลาในการหมุนมาก และระยะเวลาการหมุนแต่ละหลัก (digit) ต้องห่างกันอย่างน้อย 0.5 วินาที เพื่อให้ขุมสายสามารถแยกแยะตัวเลขได้

- ระบบความถี่ ใช้เวลาในการกดปุ่มน้อยกว่า ระยะเวลาการกดแต่ละหลัก (digit) จะใช้เวลาอย่างน้อยประมาณ 0.25 วินาที ระยะเวลาห่างกันแต่ละหลักต้องมากกว่า 0.25-0.5 วินาที

ความถี่ที่ใช้ในระบบ DTMF นั้น ทุกความถี่จะอยู่ในย่านความถี่เสียงและเป็นความถี่ IN-BAND คือความถี่เสียงในย่านที่กำหนดให้ใช้ในระบบโทรศัพท์ คือ สามารถส่งไปตามสายโทรศัพท์ได้ดี โดยมีการสูญเสียน้อยคือในย่าน 300 Hz - 3.4 kHz

นอกจากนี้ระบบ DTMF ยังสามารถนำไปใช้ในการควบคุมพิเศษต่างๆได้ เช่น ระบบการสั่งงานทางโทรศัพท์ ระบบส่งรหัสทางโทรศัพท์ ฯลฯ ในขณะที่ระบบพัลซ์ทำไม่ได้ เนื่องจากถ้าหมุนหน้าปัดเกิดการเบรคของหน้าสัมผัสแล้ว ช่องสัญญาณที่ติดต่อจะขาดทันที

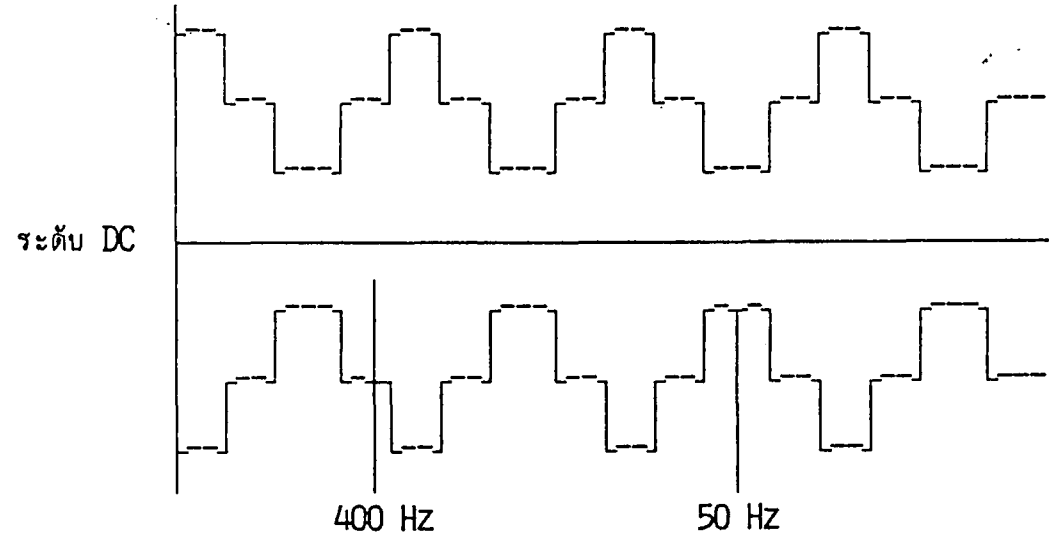
2.1.8 สัญญาณโทรศัพท์มาตรฐาน (subscriber signalling)

สัญญาณโทรศัพท์ที่เกิดขึ้นกับเครื่องโทรศัพท์ในสภาวะต่างกัน จะมีลักษณะของสัญญาณแตกต่างกัน และเพื่อให้โทรศัพท์ทั่วไปสามารถใช้งานร่วมกันได้ จึงมีการจัดมาตรฐานขึ้นโดยให้โทรศัพท์ทั่วโลกใช้มาตราฐานเดียวกันหมด อาจเรียกได้ว่า subscriber signal คือ เป็นสัญญาณที่ชุมสายโทรศัพท์จะแจ้งให้สมาชิกทราบว่า ควรจะปฏิบัติอย่างไรเมื่อได้ยินสัญญาณนั้น

โดยทั่วไปจะแบ่งลักษณะสัญญาณได้ดังต่อไปนี้

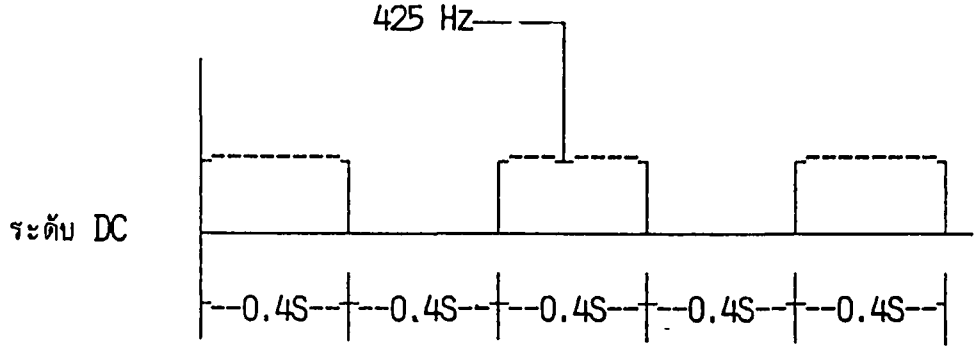
1. สัญญาณให้หมุน (Dial Tone) CCITT กำหนดให้ลักษณะสัญญาณเป็นเสียงต่อเนื่องกัน ตั้งแต่ความถี่ 400 Hz ถึง 450 Hz ผสมกับความถี่แบบ AM จะเป็นการบอกให้ผู้เรียก (Calling) ทราบว่าขณะนี้ชุมสายโทรศัพท์นั้นสามารถพร้อมที่จะรับรู้การหมุนเลขหมายเพื่อเรียกออกได้แล้ว สามารถเรียกสายติดต่อออกได้ เวลาที่ยกหูโทรศัพท์ชุมสายจะส่ง dial tone ให้เป็นเวลา 5 วินาที ถ้าเกินเวลานี้ line subscriber จะกลายเป็น line-lockout เพื่อปลด register และ code receiver ไปใช้งานกับ line subscriber รายอื่นต่อไป ขณะนี้เราจึงไม่สามาราถเรียกออกได้ ถ้าจะเรียกออกต้องวางหู (hook-on) และยกหู (hook-off) ก็จะได้ (dial tone) ใหม่

รูปที่ 2-6 แสดงสัญญาณ Dial Tone



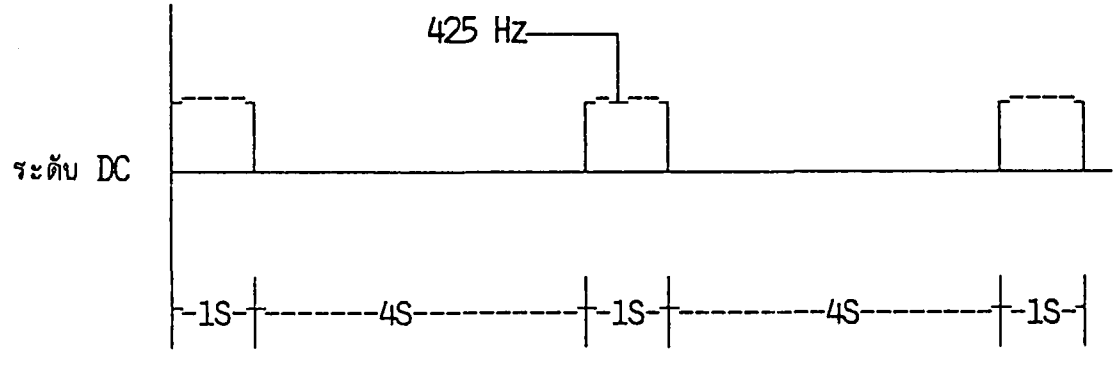
2. สัญญาณไม่ว่าง (Busy Tone) ลักษณะของสัญญาณจะดัง-หยุด-ดัง-หยุด เป็นช่วงๆ เพื่อบอกให้ทราบว่า เราไม่สามารถติดต่อกับปลายทางที่เราหมุนเลขหมายนั้นได้ อาจเป็นเพราะว่ากำลังสนทนาอยู่หรือเกิดขัดข้องในการติดต่อ เราต้องวางหูก่อนแล้วยกใหม่ เมื่อได้ยิน dial tone จึงค่อยลองโทรออกใหม่ ลักษณะของสัญญาณคือเป็นความถี่ 400 Hz (เหมือนกับ Dial Tone) แต่ดัง 0.4 วินาที หยุด 0.4 วินาที ดังแสดงในรูปที่ 2-7

รูปที่ 2-7 แสดงสัญญาณ Busy Tone



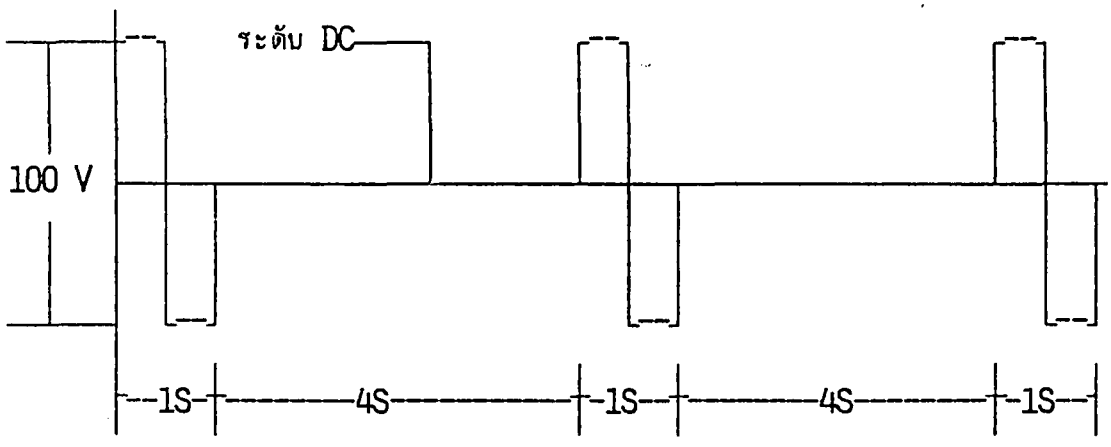
3. สัญญาณเรียกกลับ (Ring Back Tone) เมื่อเราทำการเรียกออกแล้ว ชุมสายได้ต่อสายปลายทางให้เราเป็นผลสำเร็จ ชุมสายจะส่งสัญญาณไปเรียกทางผู้รับ (called) คือทำให้กริ่งโทรศัพท์ของผู้รับดังขึ้น พร้อมๆกันนั้นทางด้านเราซึ่งเรียกออกจะมีสัญญาณเรียกกลับเป็นจังหวะพร้อมๆกับสัญญาณกระดิ่ง โดยเสียงนี้จะมีความถี่ 400 Hz ดังนาน 1 วินาที หยุดเงียบ 4 วินาที และชุมสายจะตัดเครื่องเป็น Busy ถ้าภายใน 1.5 นาที ไม่มีคนรับ

รูปที่ 2-8 แสดงสัญญาณเป็น Ring Back Tone



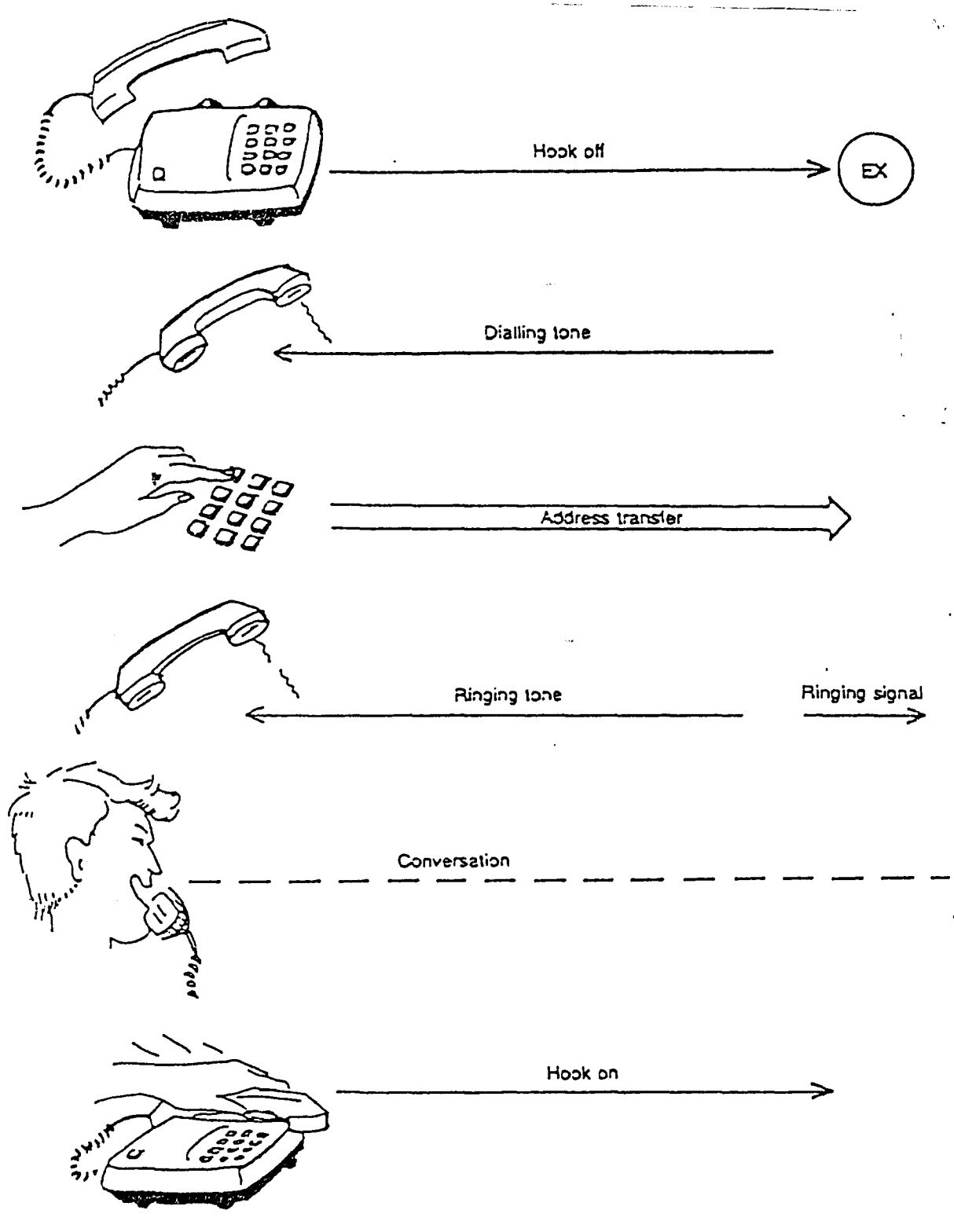
4. สัญญาณกระดิ่ง (Ringing Tone) เป็นสัญญาณไฟฟ้าที่ขับให้กระดิ่งในเครื่องโทรศัพท์ดัง เพื่อให้ทราบว่ามีคนโทรศัพท์มาหา เนื่องจากเป็นสัญญาณใช้ขับให้กระดิ่งดัง จึงต้องใช้แรงดันสูงพอสมควรคือ อยู่ในช่วง 90-150 Vpp (มักใช้ 100 V) ความถี่ประมาณ 20-60 Hz จังหวะการดัง-หยุด จะพร้อมๆกับ Ring Back Tone ของผู้ที่โทรศัพท์เข้ามาคือดัง 1 วินาที หยุดเงียบ 4 วินาที

รูปที่ 2-9 แสดงสัญญาณ Ringing Tone

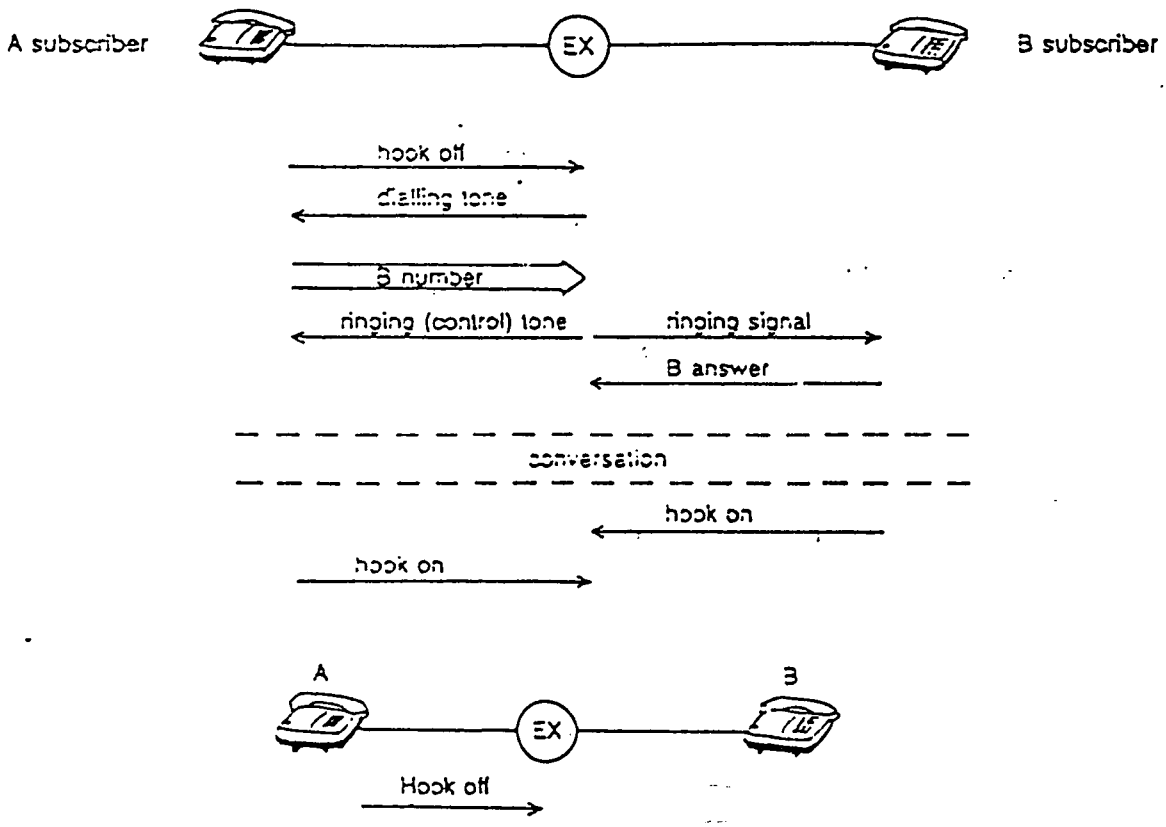


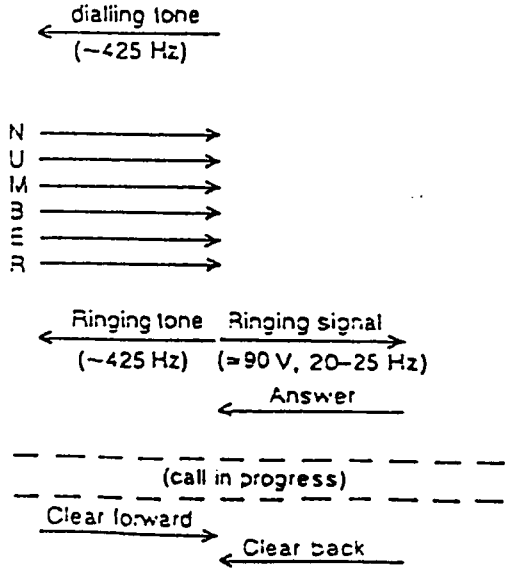
2.1.9 การทำงานของโทรศัพท์

รูปที่ 2-10 แสดงการทำงานของโทรศัพท์โดยทั่วไป

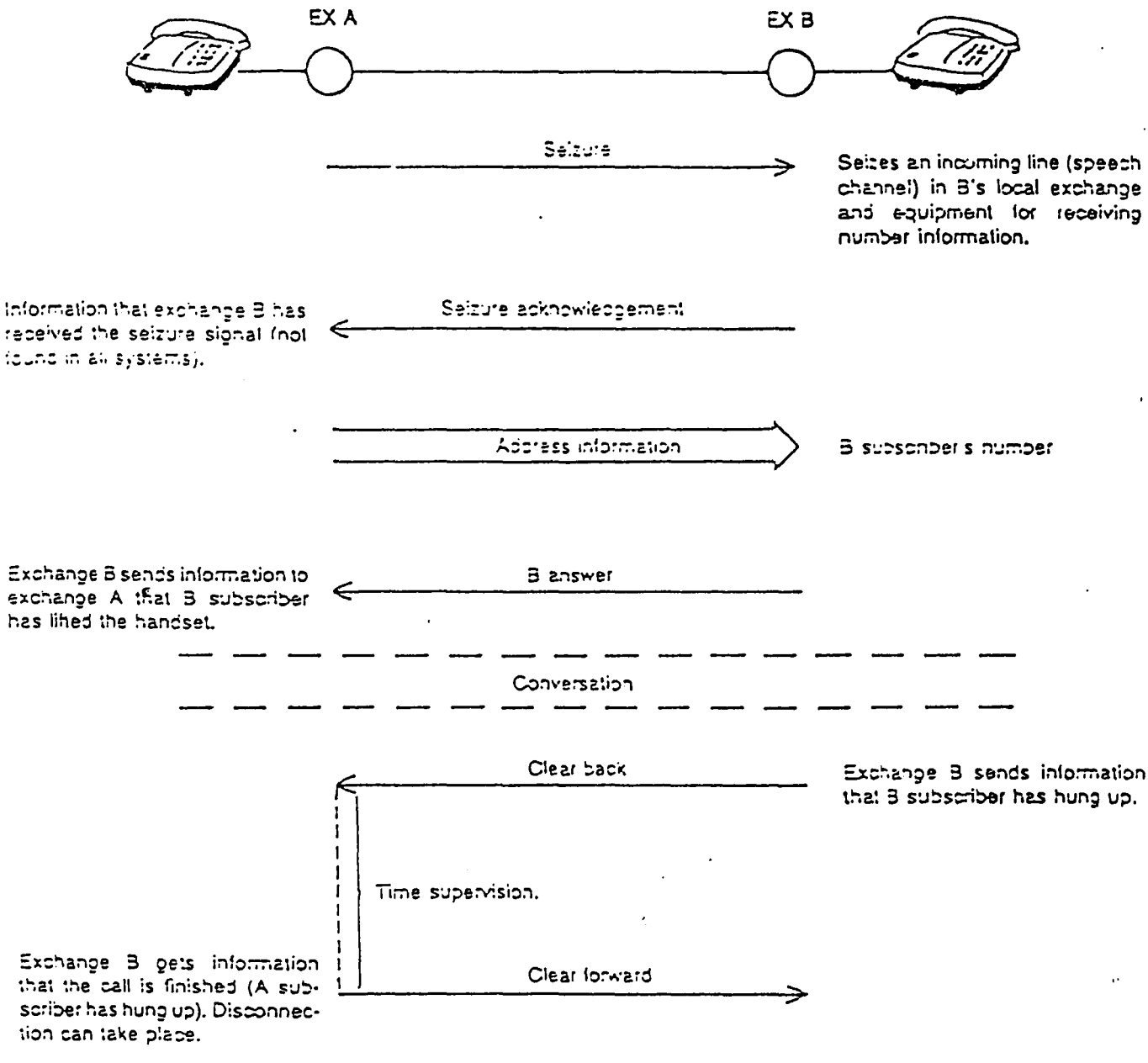


รูปที่ 2-11 แสดงการติดต่อระหว่างโทรศัพท์โดยผ่านชุมสาย





รูปที่ 2-12 แสดงการติดต่อระหว่างโทรศัพท์ โดยผ่านชุมสายหลายชุมสาย



2.2 ตู้ชุมสายโทรศัพท์อัตโนมัติ (Private Automatic Branch Exchange ; PABX)

ตู้ชุมสายโทรศัพท์อัตโนมัติ คือเครื่องมือที่ใช้ขยายคู่สายโทรศัพท์ให้มากขึ้น จะทำหน้าที่ในการเชื่อมต่อโทรศัพท์สายนอกกับโทรศัพท์สายในเข้าด้วยกัน ซึ่งอาจจะมีโอเปอเรเตอร์หรือไม่มีโอเปอเรเตอร์ควบคุมอยู่ ในลักษณะของการทำงานทางด้านผู้ใช้จะเป็นฝ่ายควบคุมเองเป็นส่วนใหญ่ ทำให้อำนวยความสะดวกในการสื่อสารให้ดียิ่งขึ้นระหว่างบุคคลในโรงงาน หรือ บริษัท เป็นต้น ดังนั้นถ้าตู้ชุมสายโทรศัพท์อัตโนมัติมีขีดความสามารถสูงเท่าใด ก็จะอำนวยความสะดวกและความคล่องตัวในการติดต่อสื่อสารมากขึ้นเท่านั้น

2.2.1 ลักษณะการทำงานของระบบ PABX

1).การเรียกออก (Dialling)

- การเรียกภายใน (Intercom function) การโทรศัพท์จากเครื่องภายในไปยังเครื่องภายในด้วยกันทำได้โดยการกดหมายเลขภายใน เช่น 11,12,13 หรือ 14

- การเรียกภายนอก (External dialling) การโทรศัพท์จากสายภายในไปยังสายภายนอกทำได้โดยการกด "0" แล้วตามหมายเลขของโทรศัพท์ภายนอกที่ต้องการจะติดต่อ

2). การเรียกเข้า (Incoming calls)

การโทรศัพท์เข้าไปยังสายภายในนั้นสามารถสังเกตุได้จากเสียงกริ่งโทรศัพท์ ซึ่งถ้าเป็นการโทรไปยังสายภายนอกเสียงกริ่งโทรศัพท์ที่ได้ยินจะดังยาวหนึ่งครั้งแล้วหยุดหนึ่งครั้งเป็นช่วงๆ ถ้าเป็นการโทรศัพท์ไปยังสายภายในเสียงกริ่งโทรศัพท์ที่ได้ยินจะดังสั้นๆ 2 ครั้งติดๆกันแล้วหยุดเรื่อยๆไป

- สัญญาณเตือนเมื่อสายถูกใช้งานอยู่ (Busy / call waiting) ถ้ามีเสียงเรียกจากสายภายนอกในขณะที่สายภายในยังมีการใช้งานอยู่ ที่สายภายในจะได้ยินสัญญาณเตือนเป็นเสียงดังสั้นๆ เพื่อบอกว่ามีการเรียกเข้าจากภายนอก และเมื่อกด hookflash จะเป็นการต่อสายภายในกับสายภายนอกที่กำลังเรียกเข้า

- การโอนสายจากสายภายในที่ไม่มีผู้รับ (Unattended call pickup) ถ้ามีการโทรเข้าแต่ไม่มีผู้รับสาย ผู้อื่นสามารถรับสายที่กำลังโทรเข้าได้ โดยการยกหูโทรศัพท์ขึ้นกด hookflash แล้วกด "7" สายที่โทรเข้าจะติดต่อกับสายภายในนั้นทันที

3). การโอนสาย (Call hold / transfer)

การโอนสายไปยังสายภายนอกหรือสายภายในทำได้โดยการถือหูโทรศัพท์ แล้วทำการกด hookflash ตามด้วยเลขหมายของเครื่องที่ต้องการจะติดต่อ เมื่อได้ยินสัญญาณการเรียกกลับแล้วจึงวางหูโทรศัพท์ลง

- การประชุมทางโทรศัพท์ (Conferencing) ถ้าในขณะที่มีการสนทนาทางโทรศัพท์อยู่แล้วผู้ใช้ต้องการโทรศัพท์ไปยังผู้อื่นอีก ซึ่งจะเป็นการใช้โทรศัพท์ร่วมกันสามสายหรือการประชุมทางโทรศัพท์นั้น ทำได้โดยการกด hookflash แล้วกดหมายเลข "8" ตามด้วยหมายเลขปลายทาง สายภายในนี้จะถูกสร้างขึ้นเป็นการประชุมทางโทรศัพท์

- สายภายนอกไม่สามารถติดต่อกับการประชุมภายใน (Trunk disconnection from conference) จากการประชุมกันระหว่างสายภายใน สายภายนอกที่อยู่บนสายการเรียกประชุมสามารถตัดออกจากการประชุมได้โดยการกด "9" ระบบ PABX จะเลือกตัดสายนอกที่ใช้งานอยู่ แต่ถ้าไม่มีการเรียกเข้าจากสายภายนอก ระบบ PABX ก็จะทำการตัดสายนอกออกจากระบบในขณะที่มีการประชุมทางโทรศัพท์ของสายภายใน

4). ลักษณะของสายภายนอก (Trunk configuration)

โหมดการส่งของสายภายนอกทั้ง 2 สายจะสามารถเซตได้โดยเป็นอิสระต่อกัน โดย - ถ้าต้องการให้สายภายนอกทำงานในระบบหมุน ให้กด "31" หรือ "32" โดย 1 หรือ 2 เป็นเลขหมายของสายภายนอกตามลำดับ

- ถ้าต้องการให้สายภายนอกทำงานในระบบกด (DTMF) ให้เซต "41" หรือ "42" โดย 1 หรือ 2 เป็นหมายเลขของสายภายนอกตามลำดับ

5). การตรวจจับสัญญาณแฟกซ์โดยอัตโนมัติ (Automatic fax detection)

ระบบ PABX สามารถตรวจจับ CNG tone ของสัญญาณแฟกซ์ที่เข้ามาของสายภายนอกหรือทั้งสองสายภายนอก และสายภายในได้ โดย

ถ้าต้องการให้มีการตรวจจับสัญญาณแฟกซ์ให้กดหมายเลข "51"

ถ้าไม่ต้องการให้มีการตรวจจับสัญญาณแฟกซ์ให้กดหมายเลข "50"

การเลือกสายภายนอกที่จะทำการรับแฟกซ์ทำได้โดยการกดหมายเลข "58X"

โดย X เป็น
- 0 เมื่อ reset the fax detection
- 1 เมื่อให้สายภายนอกสายที่ 1 รับแฟกซ์
- 2 เมื่อให้สายภายนอกสายที่ 2 รับแฟกซ์
- 3 เมื่อให้สายภายนอกทั้ง 2 สายรับแฟกซ์

การเลือกสายภายในที่จะทำการรับแฟกซ์ ทำได้โดยการกดหมายเลข "59XX" โดย XX คือ เลขหมายของสายภายใน

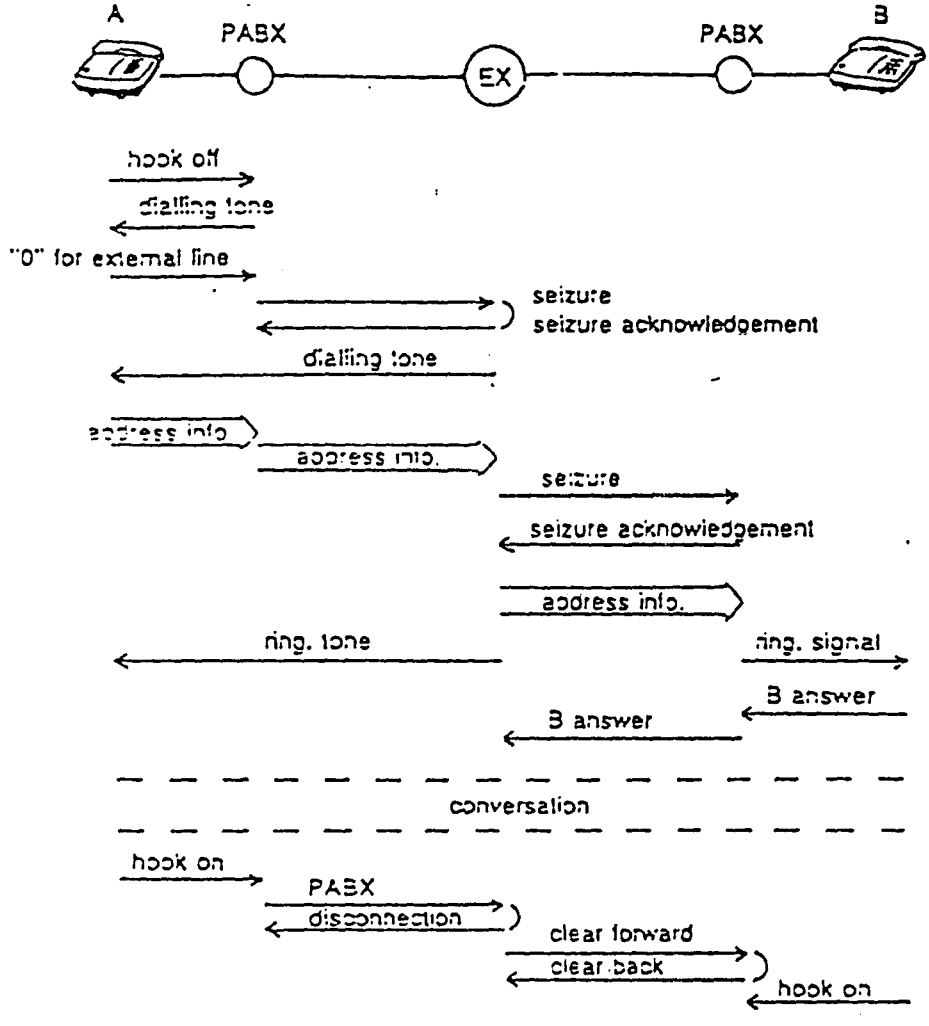
6). การเรียกโดยผ่านทางโอเปอเรเตอร์ (Secretary filter)

การเรียกเข้าของทุกสายสามารถทำให้เข้าสู่สายภายในเพียงสายเดียวได้ เพื่อใช้สำหรับผ่านทางเลขานุการหรือโอเปอเรเตอร์ ซึ่งทำได้โดยการกดหมายเลข "61" แล้วเลือกหมายเลขภายในโดยกด "69XX" เมื่อ XX เป็นหมายเลขของสายภายใน แต่ถ้าต้องการให้มีการเรียกเข้าทุกสายภายในให้กดหมายเลข "60"

ตารางที่ 2 แสดงการเซตโหมดการทำงานของระบบ PABX

FUNCTION	OPTIONS	KEYS
Dialling	Internal External	11,12,13,14 0
Call pickup		Hookflash 7
Hold	 Transfer Toggle	Hookflash Replace handset Hookflash
Conference	 Trunk disconnect	8 9
Trunk config	Disable Pulse DTMF	30,40 31,32 41,42
Fax detect	Disable Enable	50 51
Fax trunk detect	Reset Trunk 1 Trunk 2 Both	580 581 582 583
Fax select	Ext 1 Ext 2 Ext 3 Ext 4	5911 5912 5913 5914
Secretary filter	Disable Enable	60 61
Secretary select	Ext 1 Ext 2 Ext 3 Ext 4	6911 6912 6913 6914

รูปที่ 2-13 แสดงตัวอย่างการทำงานของระบบ PABX



บทที่ 3
หลักการทำงานของตู้ชุมสายโทรศัพท์อัตโนมัติ

ตู้ชุมสายโทรศัพท์อัตโนมัติที่นำเสนอนี้จะประกอบด้วยส่วนที่สำคัญ คือ

- ส่วนของฮาร์ดแวร์ จะเป็นส่วนของวงจรทั้งหมดของระบบตู้สาขาโทรศัพท์อัตโนมัติ
- ส่วนของซอฟท์แวร์ จะเป็นส่วนที่เกี่ยวข้องกับโปรแกรมควบคุมการทำงานต่างๆ ของตู้สาขาโทรศัพท์อัตโนมัติให้สามารถทำงานได้ตรงตามคุณสมบัติที่กำหนดไว้ข้างต้น

3.1 โครงสร้างทางด้านฮาร์ดแวร์

โครงสร้างทางด้านฮาร์ดแวร์นี้จะเป็นในส่วนของวงจรต่างๆรวมถึงส่วนที่สำคัญของระบบ PABX ก็คือ Switch Unit ด้วย ทางด้านฮาร์ดแวร์ที่จะกล่าวถึงนี้แบ่งเป็น

1. ส่วนของ Microcontroller และ Memory
2. Analog ground buffer
3. เครื่องกำเนิดเสียงเพลงและความถี่ (Music and tone generation)
4. แหล่งจ่ายไฟ (Power supply)
5. การเชื่อมต่อสายโทรศัพท์ภายใน (Extension line interfaces)
6. การเชื่อมต่อสายโทรศัพท์ภายนอก (Trunk line interfaces)
7. การเชื่อมต่อกับ chip SC11390 (SC11390 connections)
8. วงจรขยายเสียงโทรศัพท์ (Ring amplifier)

3.1.1 ส่วนประมวลผลกลาง (Microcontroller) และ หน่วยความจำ (Memory)

Microcontroller ของระบบ PABX นี้คือ 80C32 ซึ่งจะทำงานที่ความถี่ 11.0592 MHz โดยทำงานเป็นหน่วยประมวลผลกลางของระบบ PABX โดย 27C256 ทำหน้าที่เป็น EPROM และ 62256 (RAM) จะทำงานเป็นหน่วยความจำโดย address bus ของหน่วยความจำนี้จะถูกถอดรหัสโดย 16V8 PAL และใช้ 74HC573 เป็นตัว latch ข้อมูลซึ่งจะถูกควบคุมจากสัญญาณ ALE ของ Microcontroller โดย PAL จะเป็นตัวกำหนดการทำงานของสายต่างๆในวงจร ซึ่งในส่วนของ Microcontroller นี้จะทำหน้าที่เป็น

1). Watchdog โดยมีการใช้ TL7705 เพื่อให้วงจร reset ในขณะเริ่มทำงานและใช้เป็น continuous processor watchdog ซึ่งจะช่วยให้มีการ reset อย่างสมบูรณ์และป้องกันความผิดพลาดของ firmware โดยจะใช้ขา WD ของ TL7705 เพื่อ pull up กับตัวต้านทาน 2.2 kΩ เพื่อขับ large capacitive load

2). Serial EEPROM โดย Serial EEPROM ในระบบ PABX นี้คือ MC9366 ซึ่งจะใช้ในการเก็บหมายเลขและลักษณะของระบบ โดยที่ MC9366 นี้จะถูกควบคุมจาก port pins บน Microcontroller

3). RS232 ในการเชื่อมต่อ RS232 กับเครื่อง PC นั้นจะใช้ MAX232 ในการเชื่อมต่อซึ่ง RS232 นี้จะใช้ไฟเลี้ยงเพียง 5 โวลต์ โดย RS232 จะควบคุมสัญญาณ RTS และ CTS

3.1.2 Analog ground buffer

analog reference voltage จะถูกนำมาเป็นบัฟเฟอร์เพื่อ support ให้กับวงจร analog ส่วน analog ground reference จะถูกนำมาขยายเพื่อให้แรงดันที่ป้อนเข้าไปเลี้ยงวงจรมีความเสถียร (stable bias voltage)

3.1.3 เครื่องกำเนิดความถี่และเสียงเพลงสำหรับโทรศัพท์ (Music input and Tone generation)

ที่ audio input จะรับสัญญาณจากเครื่องขยายเสียงและนำสัญญาณที่ออกจากลำโพงซ้ายและขวามารวมกัน ระดับความดังของเสียงเพลงจะปรับได้โดยใช้ตัวต้านทานปรับค่าได้ (R12) ส่วนสัญญานให้หมุนและสัญญาณสายไม่ว่างจะเกิดขึ้นของการสั่นรอบๆ NE5514 ซึ่งเป็น quad op-amp เพื่อเซตให้เกิดสัญญาณรูป sine wave ความถี่ 400 Hz ซึ่งระดับความดังของเสียงที่ได้ยินจะปรับได้โดยการใช้ตัวต้านทานปรับค่าได้ (R16)

3.1.4 แหล่งจ่ายไฟ (Power supply)

ระบบ PABX นี้จะทำงานโดยใช้แหล่งจ่ายไฟตรง 24 โวลต์ (1 A) โดยที่แรงดัน 24 โวลต์นี้จะเป็นตัวจ่ายไฟตรงให้กับ hybrid amplifier , รีเลย์ และวงจรเสียงโทรศัพท์ ซึ่งแรงดัน 24 โวลต์นี้จะถูกส่งผ่านไปยัง 7805 และ 78L05 เพื่อปรับแรงดันให้เหลือเพียง 5 โวลต์เพื่อจ่ายให้กับวงจรดิจิตอลและวงจรอนาลอก

การสร้างแรงดัน 50 โวลต์สำหรับสายโทรศัพท์นั้นจะใช้ power supply แบบ switch mode โดยใช้ NE5532 (dual op-amp) เป็นตัวกำเนิดแรงดัน 50 โวลต์จากแรงดัน 24 โวลต์ของแหล่งจ่ายไฟ ซึ่งแรงดัน 50 โวลต์นี้จะมีกระแสต่ำและได้ถูกออกแบบมาเพื่อใช้เป็นแรงดันอ้างอิงของโทรศัพท์

3.1.5 การเชื่อมต่อสายโทรศัพท์ภายใน (Extension line interfaces)

การเชื่อมต่อสายภายในเข้ากับชิพ SC11390 นั้นจะใช้ dual op-amp hybrid โดยวงจร hybrid นี้ จะใช้แรงดัน 24 V และ 0 V เพื่อสร้างเสียงเรียกที่เพียงพอสำหรับสายโทรศัพท์ โดยเอาท์พุทอิมพิแดนซ์เท่ากับ 600 Ωแต่สามารถควบคุมได้โดยใช้ jumper 11 และ jumper 18 ส่วนสัญญาณ audio จะผ่านไปยังรีเลย์และผ่านไปยังสาย โดยรีเลย์นี้จะทำการเลือกให้สัญญาณ audio หรือสัญญาณเสียงโทรศัพท์ที่มีแรงดันสูงผ่านไปยังสาย โดยที่รีเลย์จะถูกควบคุมจาก microprocessor ผ่านทาง 82C55

3.1.6 การเชื่อมต่อสายโทรศัพท์ภายนอก (Trunk line interfaces)

การเชื่อมต่อสายโทรศัพท์ภายนอกหรือ Trunk เข้ากับชิพ SC11390 นั้นคล้ายกับการเชื่อมต่อสายโทรศัพท์ภายในคือ ผู้ใช้สามารถทำการปรับค่าอิมพิแดนซ์ของสายให้เหมาะสมได้โดยใช้ jumper โดยในส่วนนี้จะมีซีเนอร์ไดโอดเป็นตัวป้องกันไฟกระชาก (spike) จากสายภายนอก ซึ่งสายภายนอกนี้จะถูกแยกไว้โดยใช้หม้อแปลง

หม้อแปลงจะช่วยให้มี distortion ต่ำ โดยที่ "wet" transformer จะถูกออกแบบมาเพื่อควบคุมกระแสDC ที่ไหลอยู่บนสาย หรือมิฉะนั้นแล้วอาจมีการใช้ "dry" transformer ซึ่งขนาดเล็กกว่าแต่การใช้หม้อแปลงแบบนี้จำเป็นจะต้องใช้ไฟกระแสตรงในการเลี้ยงวงจรและมักมีราคาแพงกว่า

ในส่วนของ hook switch โดยทั่วไปแล้วจะใช้เป็นรีเลย์มาตรฐานซึ่งจะถูกควบคุมจาก microprocessor ผ่านทาง 82C55 และในการตรวจจับสัญญาณเสียงโทรศัพท์นั้นจะมีการใช้ตัวกรองและออปโตคัปเปิ้ล(filter and opto coupler)

3.1.7 การเชื่อมต่อกับ SC11390 (SC11390 connectiohs)

ส่วนนี้จะเป็นส่วนที่สำคัญของระบบ PABX โดยหลักที่สำคัญคือ Chip SC11390 ซึ่งจะใช้ไฟเลี้ยง 5 โวลต์ดิจิตอล(VDD) จากแหล่งจ่ายไฟ (power supply) ซึ่งในระบบ PABX ที่นำ

เสนอนี้จะมีการต่อของสายภายนอก 2 สายกับสายภายใน 4 สาย โดยสัญญาณต่างๆของ SC11390 ที่เกี่ยวข้องกับระบบ PABX มีดังนี้

- Xin เป็นแหล่งกำเนิดความถี่อินพุท ซึ่งต้องใช้คริสตอลความถี่ 4.579 MHz ต่อระหว่าง Xin กับ Vss

- AVD เป็นจ่ายไฟแบบอนาล็อกซึ่งมีแรงดัน 5 โวลต์

- VSS จะเป็นแรงดัน 0 โวลต์ดิจิตอลที่ป้อนให้กับ SC11390

- AGND เป็นแหล่งกำเนิดแรงดันภายใน 1.75 โวลต์ซึ่งจะใช้ป้อนให้กับออปแอมป์ภายนอก และจะถูกใช้เป็นบัฟเฟอร์โดย analog ground buffer

- ER จะเป็นการ reset วงจรภายนอกซึ่งจะทำงานที่ลอจิก "1"

- CT1 และ CT2 เป็นตัวตรวจจับสัญญาณการเรียกเข้าสำหรับ SC11390 ซึ่งจะตรวจจับสัญญาณให้หมุน, สัญญาณสายไม่ว่างและให้มีการเรียกเข้าได้อย่างถูกต้อง

- CPFO1 เป็นขาเอาท์พุทของการเรียกเข้าบน CT1

- TGR จะเป็นขาอินพุทสำหรับให้กำเนิดความถี่ของสัญญาณสายไม่ว่างและสัญญาณให้หมุน

- RINGEROUT จะเป็นสัญญาณเอาท์พุทรูป sine wave หรือ square wave เพื่อขับให้กับวงจรขยายเสียงโทรศัพท์ (Ring Amplifier)

- EPSR เป็นสัญญาณอินพุทของ audio สำหรับเสียงเพลงในขณะที่ถือหูโทรศัพท์รอ

- CPD1 และ CPD2 เป็น TTL เอาท์พุทของตัวตรวจจับความถี่ ซึ่งสัญญาณที่ได้รับบน CT1 และ CT2 จะถูกกรองและแปลงเป็นสัญญาณรูป square wave ดังนั้นสัญญาณ TTL ที่วัดได้จาก microprocessor จะเป็นค่าความถี่

- TTG และ TRG เป็นขาที่ควบคุมอัตราขยายของการรับและการส่งจากสายภายนอกตามลำดับ ซึ่งอัตราการขยายของการรับ-ส่งนี้จะถูกควบคุมโดยตัวต้านทานปรับค่าได้

- TD เป็นขาเอาท์พุทของตัวกรองระดับความถี่ที่โปรแกรมได้ (programmable bandpass filter) ซึ่งจะถูกควบคุมจาก microprocessor เพื่อตรวจจับสัญญาณแฟกซ์ที่เข้ามาโดยทำการวัดความถี่ของสัญญาณที่ได้รับผ่านทางตัวกรอง

- ส่วนของสายภายในของชิพที่ไม่มีการใช้งาน (DRSR,DRST,TSR,TST,RCR และ RCT) นั้น จะทำการต่อลงกราวด์ (ground) ซึ่งหากมีการขยายสายภายในออกไปอีกก็จะมีการนำสายเหล่านี้มาใช้งาน

โดย 82C55 เป็นตัว mop up สัญญาณดิจิตอล อินพุท/เอาท์พุท จำนวนหนึ่งในระบบ ซึ่งจะนำมาใช้ในการต่อกับ address bus ของ microcontroller, วงจรตรวจจับการใช้งาน, ring relays, ตัวกำเนิดความถี่ (tone generator) และตัวตรวจจับการเรียกเข้า (call progress detectors)

3.1.8 วงจรขยายเสียงโทรศัพท์ (Ring amplifier)

วงจรขยายเสียงโทรศัพท์จะสร้างสัญญาณรูป sine wave ที่มีแรงดันสูงเพื่อทำการป้อนให้กับเสียงเรียกของโทรศัพท์ ซึ่งสัญญาณเสียงเรียกของโทรศัพท์สามารถผลิตได้ 2 แบบ โดยขึ้นอยู่กับความถี่ที่ต้องการใช้งานในแต่ละประเทศ

ตัวกำเนิดสัญญาณรูป sine wave ใน chip SC11390 จะผลิตความถี่ 16.67Hz ,25 Hz หรือ 50 Hz โดยสัญญาณรูป sine wave นี้จะเป็นตัวขับให้ LM2877 ขยายสัญญาณขึ้นซึ่งจะต่อกับวงจรบริดจ์ โดยที่วงจรขยายสัญญาณนี้จะทำการขับแรงดัน 12 โวลต์ผ่านหม้อแปลงให้เป็น 110 โวลต์ ซึ่งหม้อแปลงนี้จะมีความสามารถในการทำงานเกินจากช่วงความถี่ที่กำหนดไว้โดยมี distrotion ต่ำ และแรงดันเอาท์พุทจะเท่ากับ 60-70 โวลต์เพื่อป้อนให้กับโทรศัพท์ ซึ่งสามารถปรับค่าได้โดยใช้ตัวต้านทานปรับค่าได้

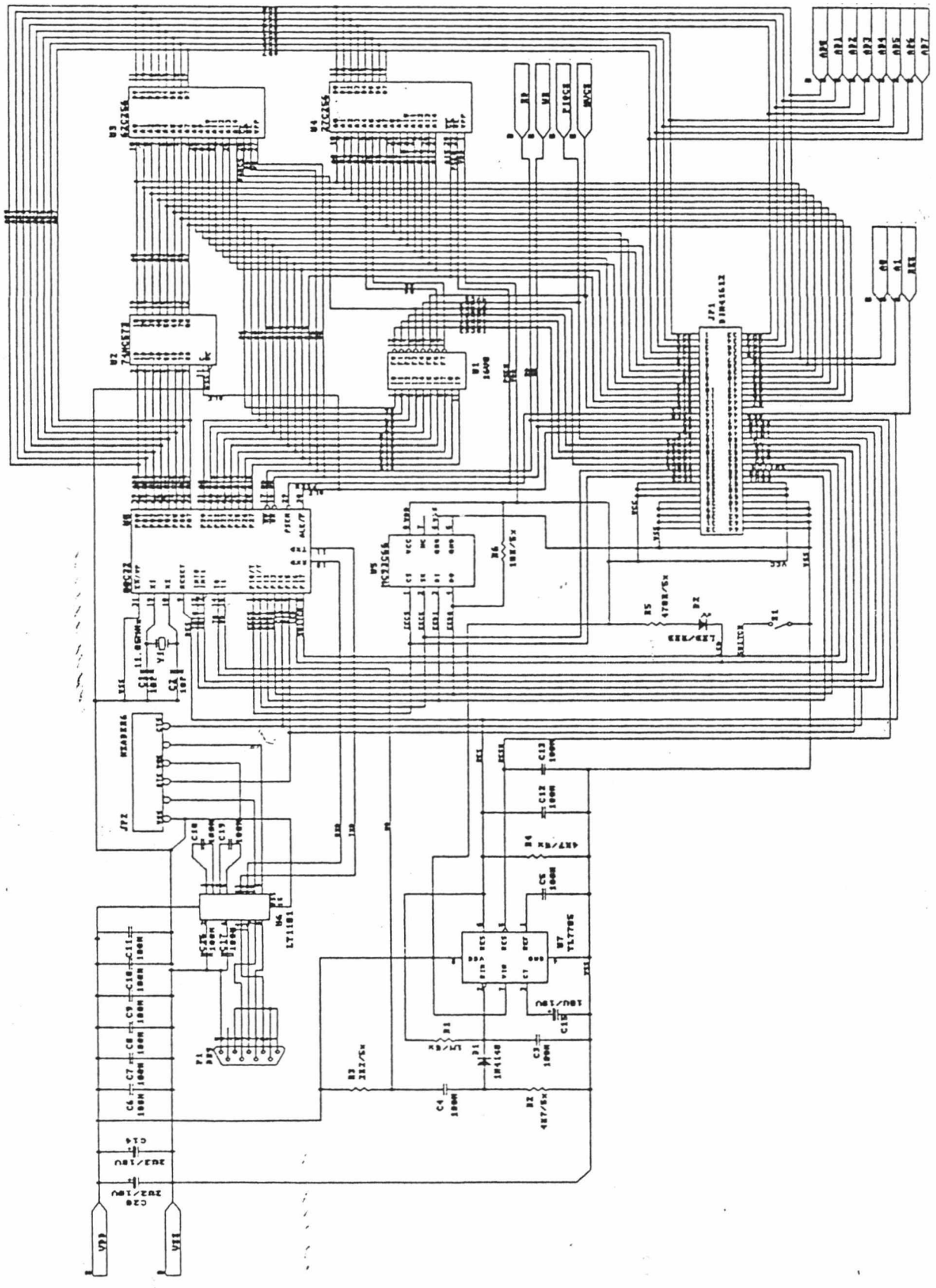
3.2 การออกแบบวงจรในส่วนต่าง ๆของระบบ PABX

ในการออกแบบวงจรของระบบ PABX ได้ทำการออกแบบวงจรโดยคำนึงถึงลักษณะการทำงานที่แตกต่างกัน โดยทำการแบ่งวงจรตามส่วนการทำงานต่างๆ ดังนี้

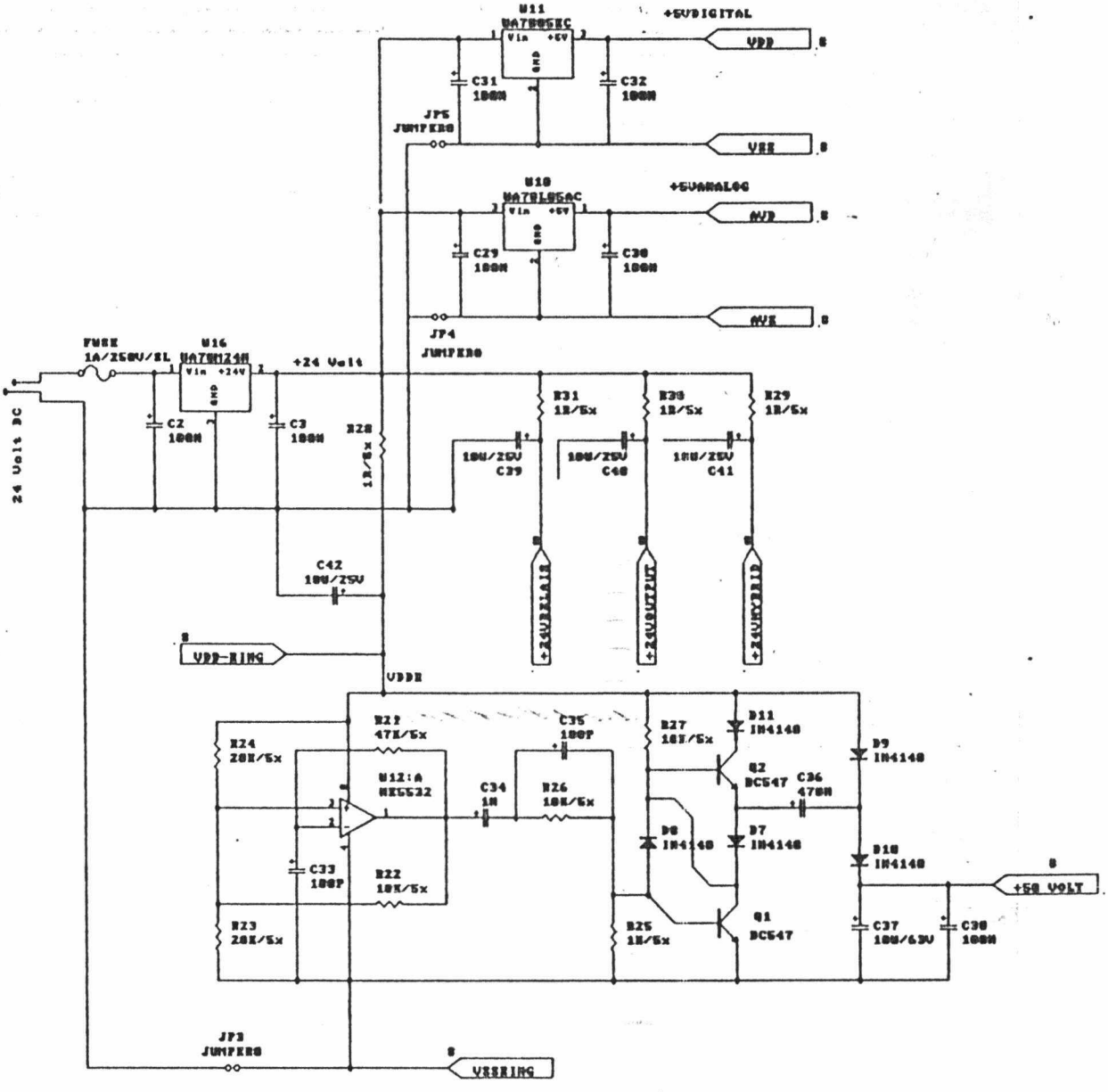
1. วงจรควบคุมการทำงานของระบบ (Process)
2. วงจรแหล่งจ่ายไฟทั้งระบบ (Supply)
3. วงจรบัฟเฟอร์ (Analog ground buffer)
4. วงจรของ SWITCH UNIT
5. วงจรกำเนิดเสียงโทรศัพท์ (Ring Amplifier)
6. วงจรกำเนิดเสียงสำหรับโทรศัพท์ (Audio)
7. วงจรของสายโทรศัพท์ภายนอก (Trunks)
8. วงจรของสายโทรศัพท์ภายใน (Extensions)

ซึ่งวงจรที่กล่าวมาแล้วนั้นจะแสดงดังรูป

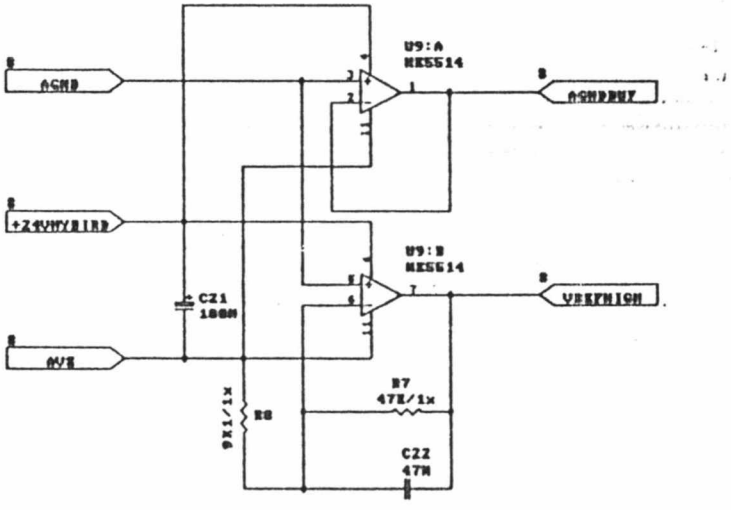
รูปที่ 3-1. วงจรควบคุมการทำงานของระบบ (Process)



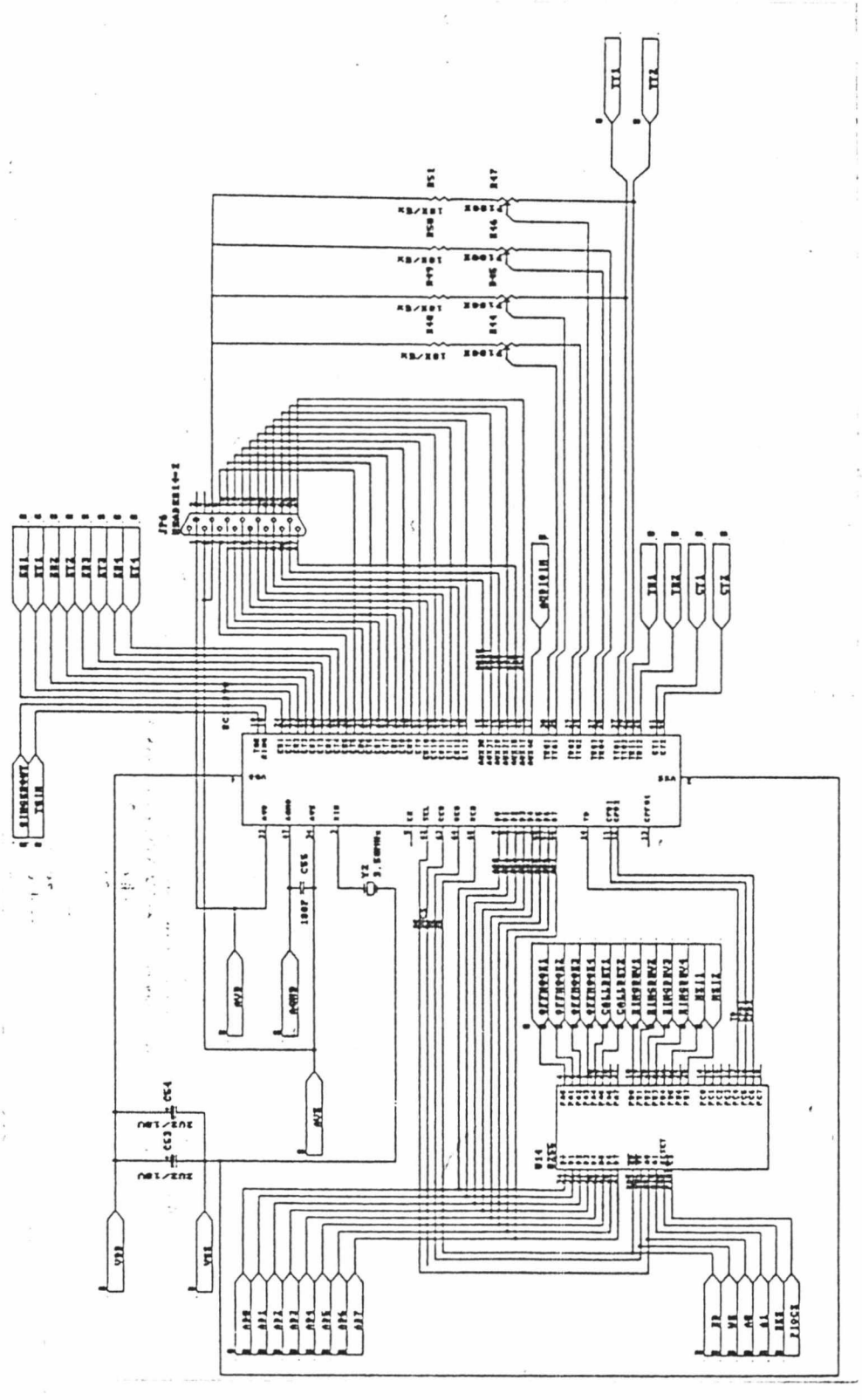
รูปที่ 3-2 วงจรแหล่งจ่ายไฟทั้งระบบ (Supply)



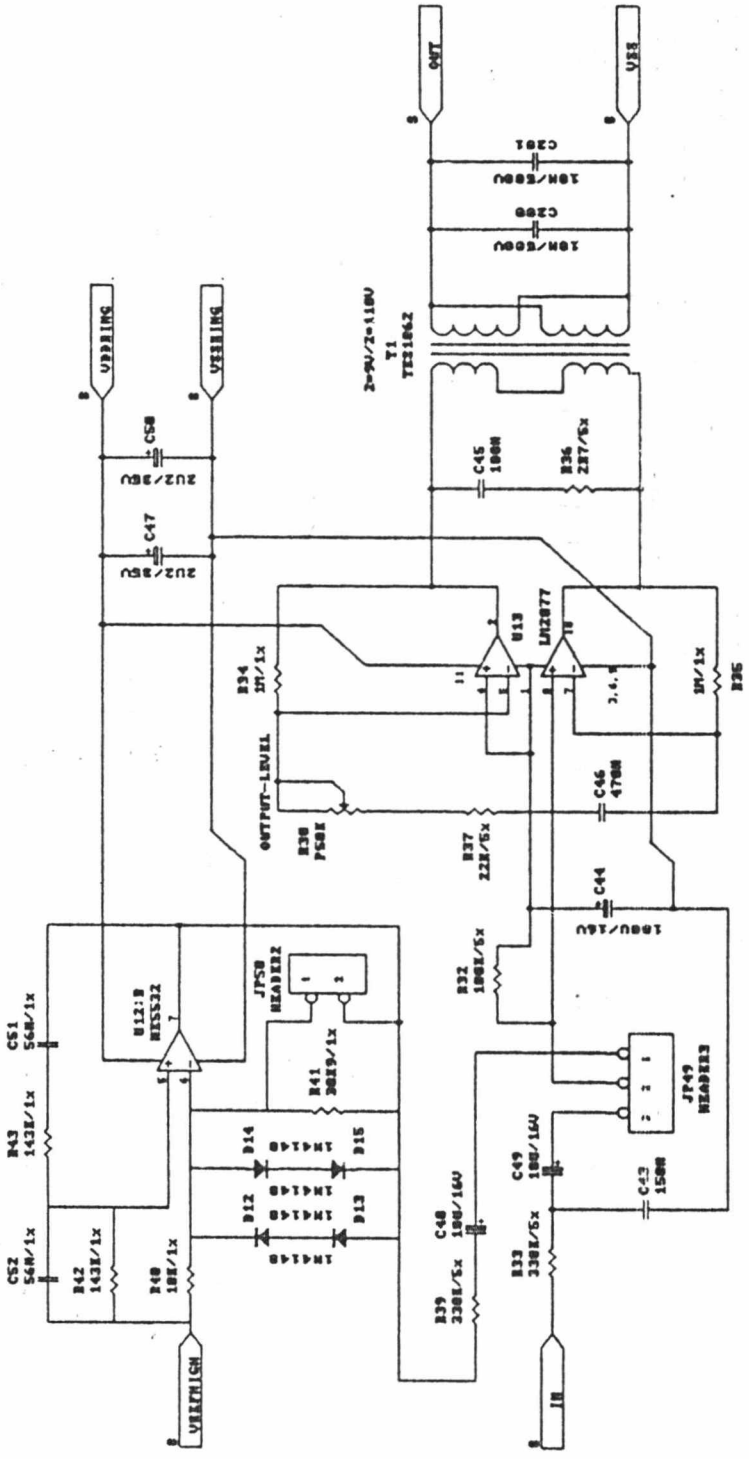
รูปที่ 3-3 วงจรบัฟเฟอร์ (Analog ground buffer)



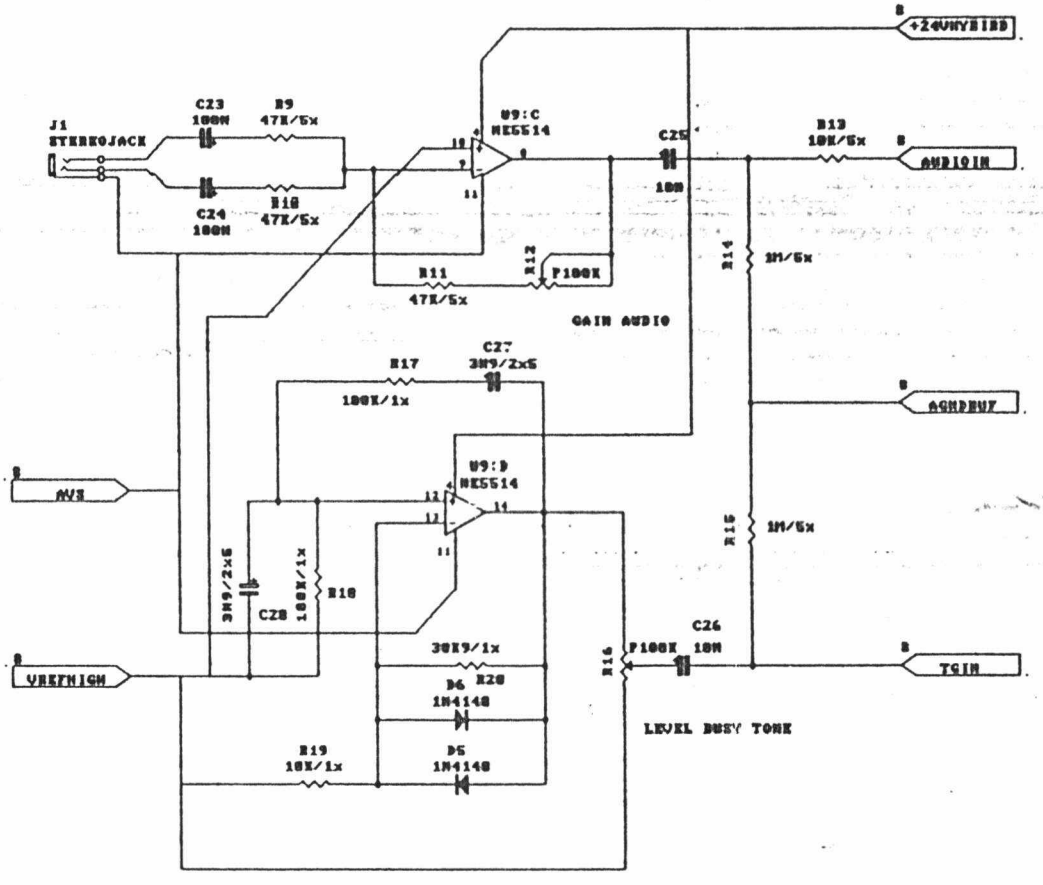
รูปที่ 3-4. วงจรของ SWITCH UNIT



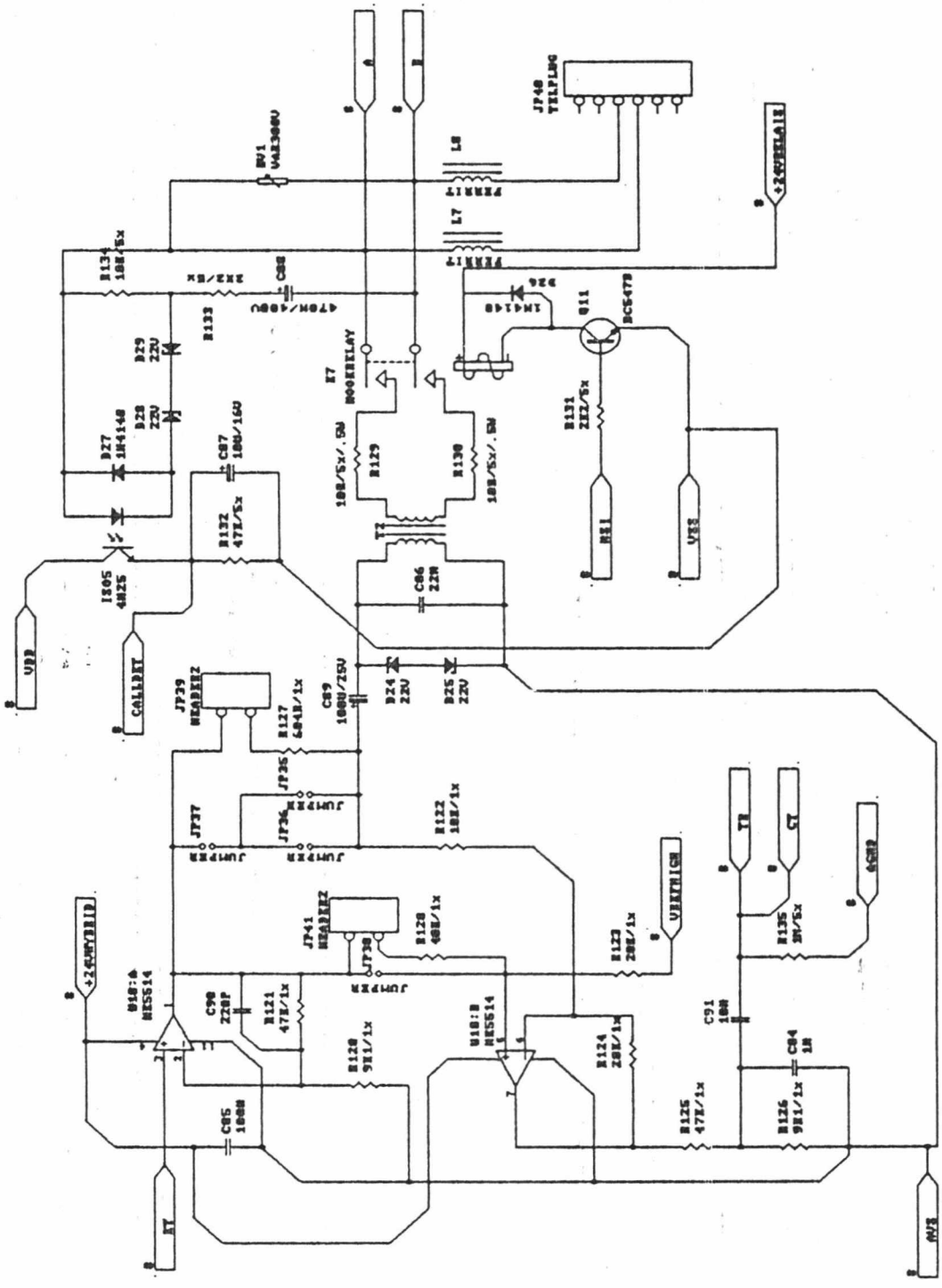
รูปที่ 3-5. วงจรกำเนิดเสียงโทรศัพท์ (Ring Amplifier)



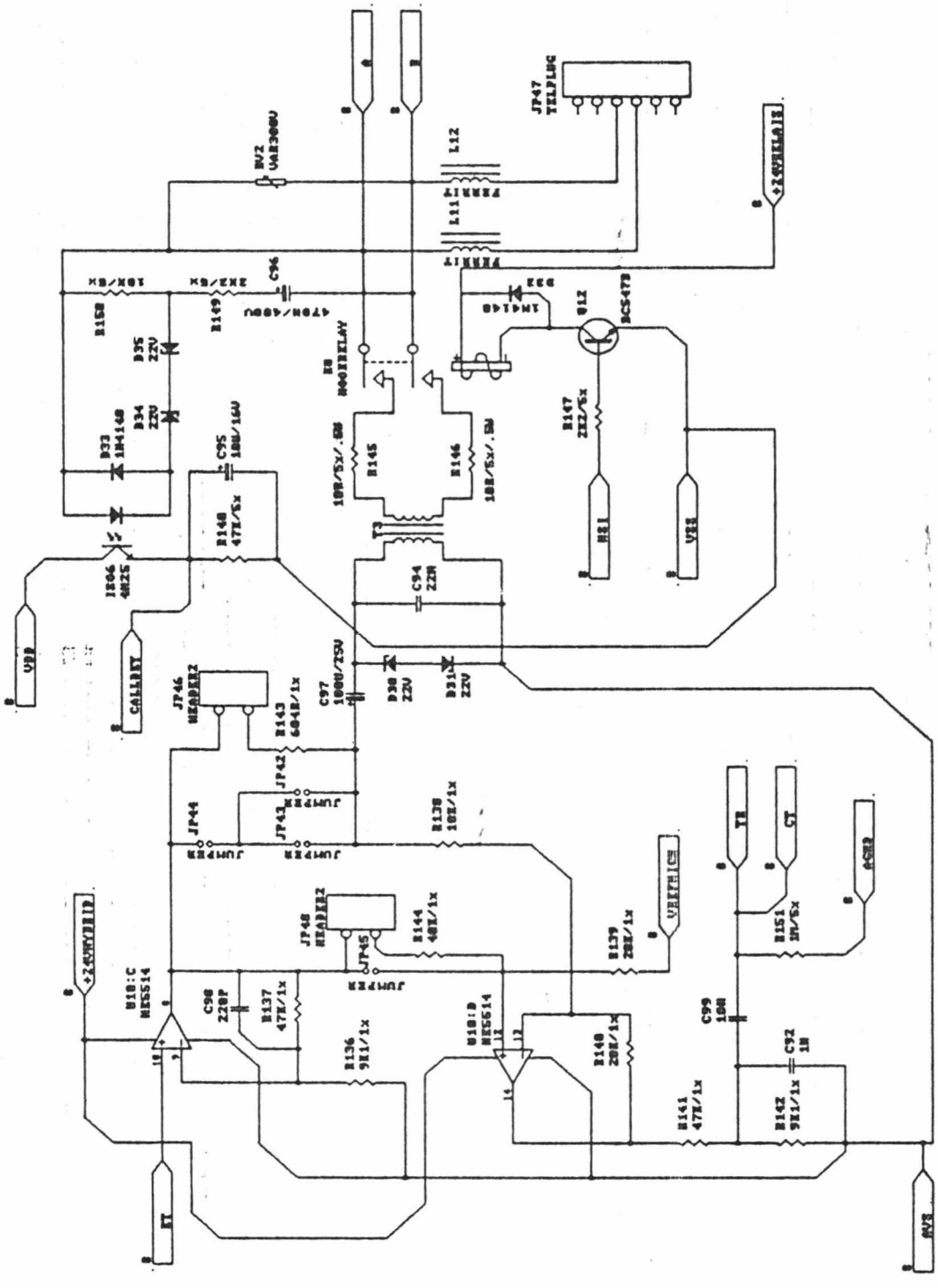
รูปที่ 3-6. วงจรกำเนิดเสียงสำหรับโทรศัพท์ (Audio)



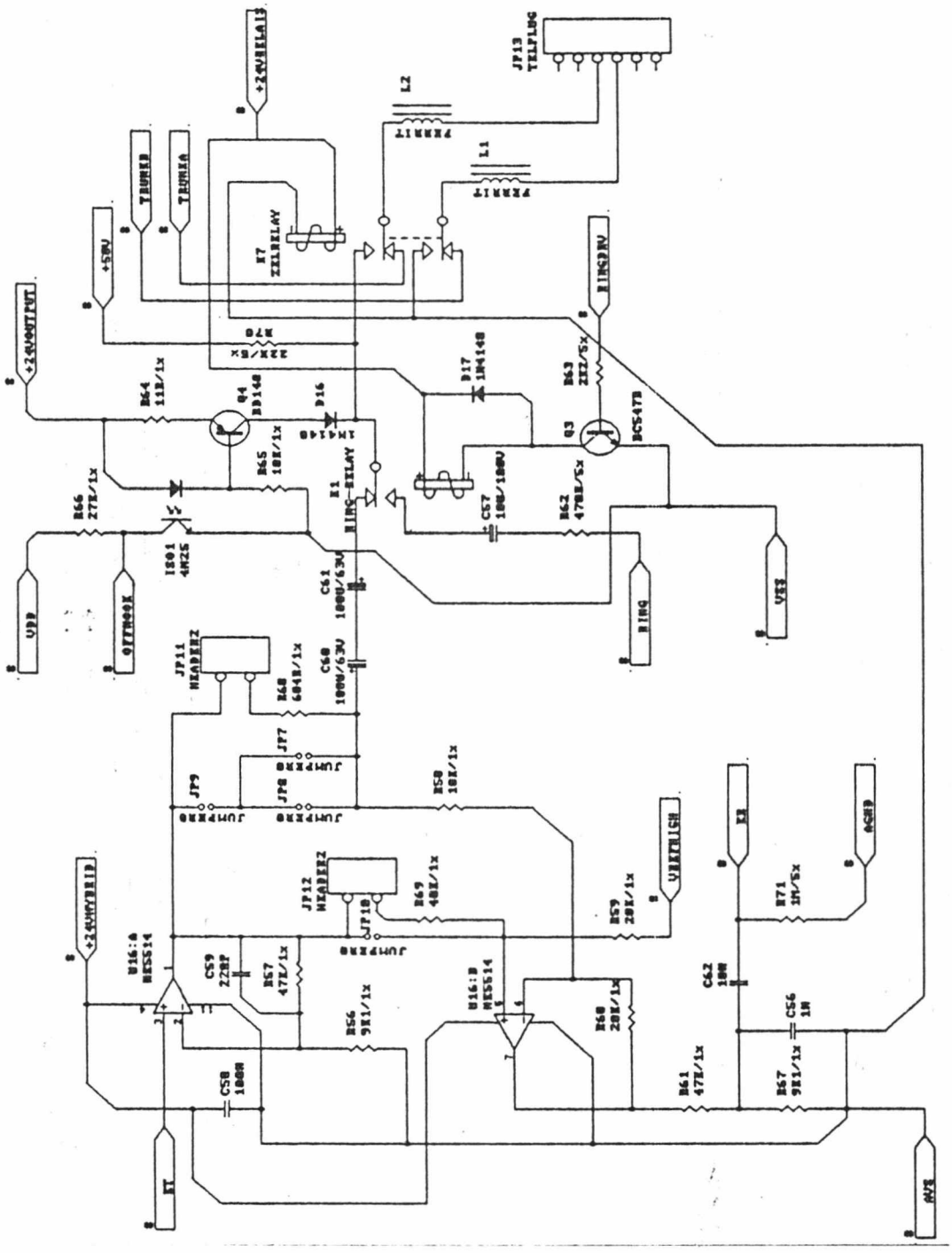
รูปที่ 3-7-1. วงจรของสายโทรศัพท์ภายนอกสายที่ 1 (Trunk 1)



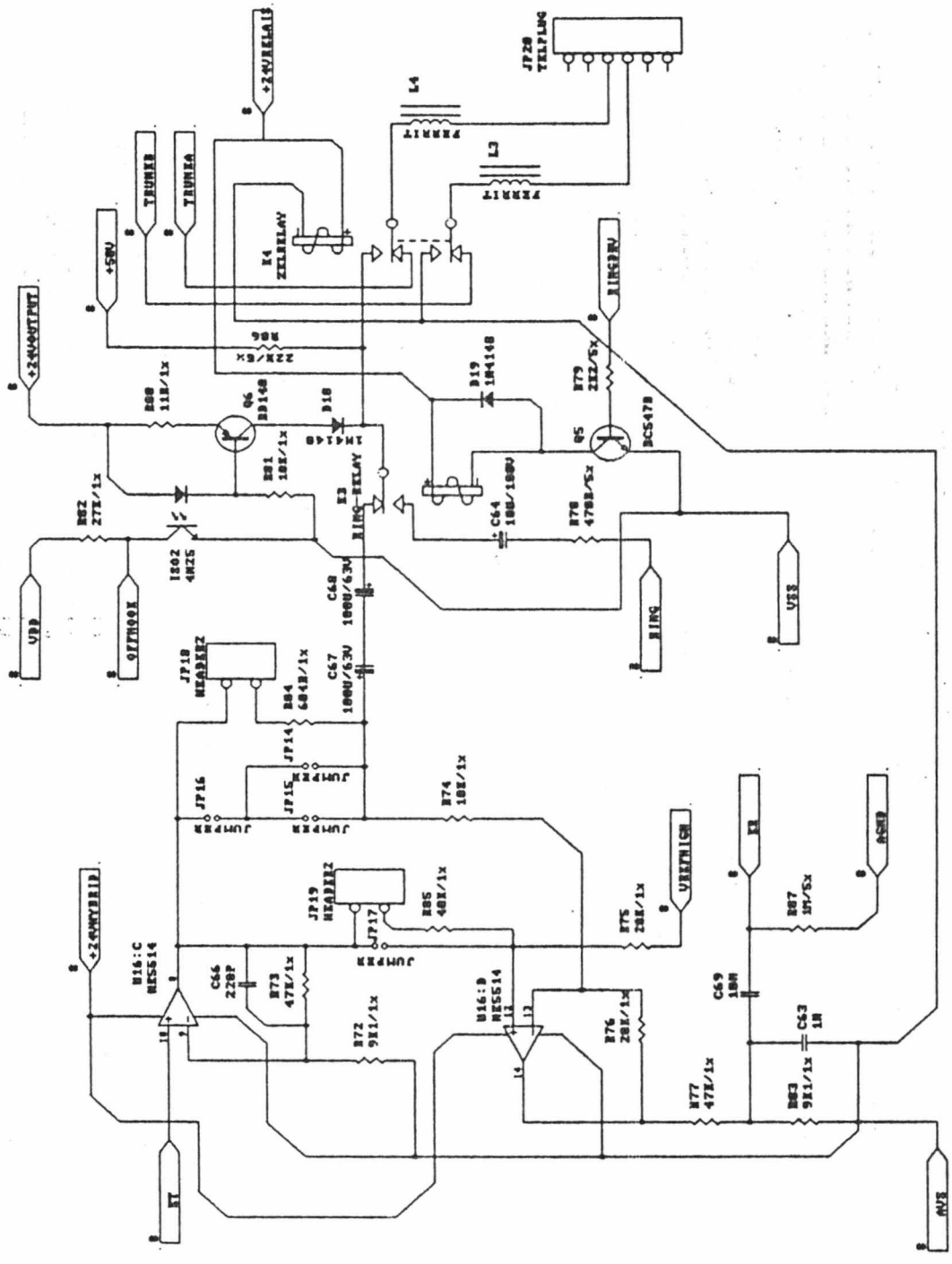
รูปที่ 3-7-2 วงจรของสายโทรศัพท์ภายนอกสายที่ 2 (Trunk 2)



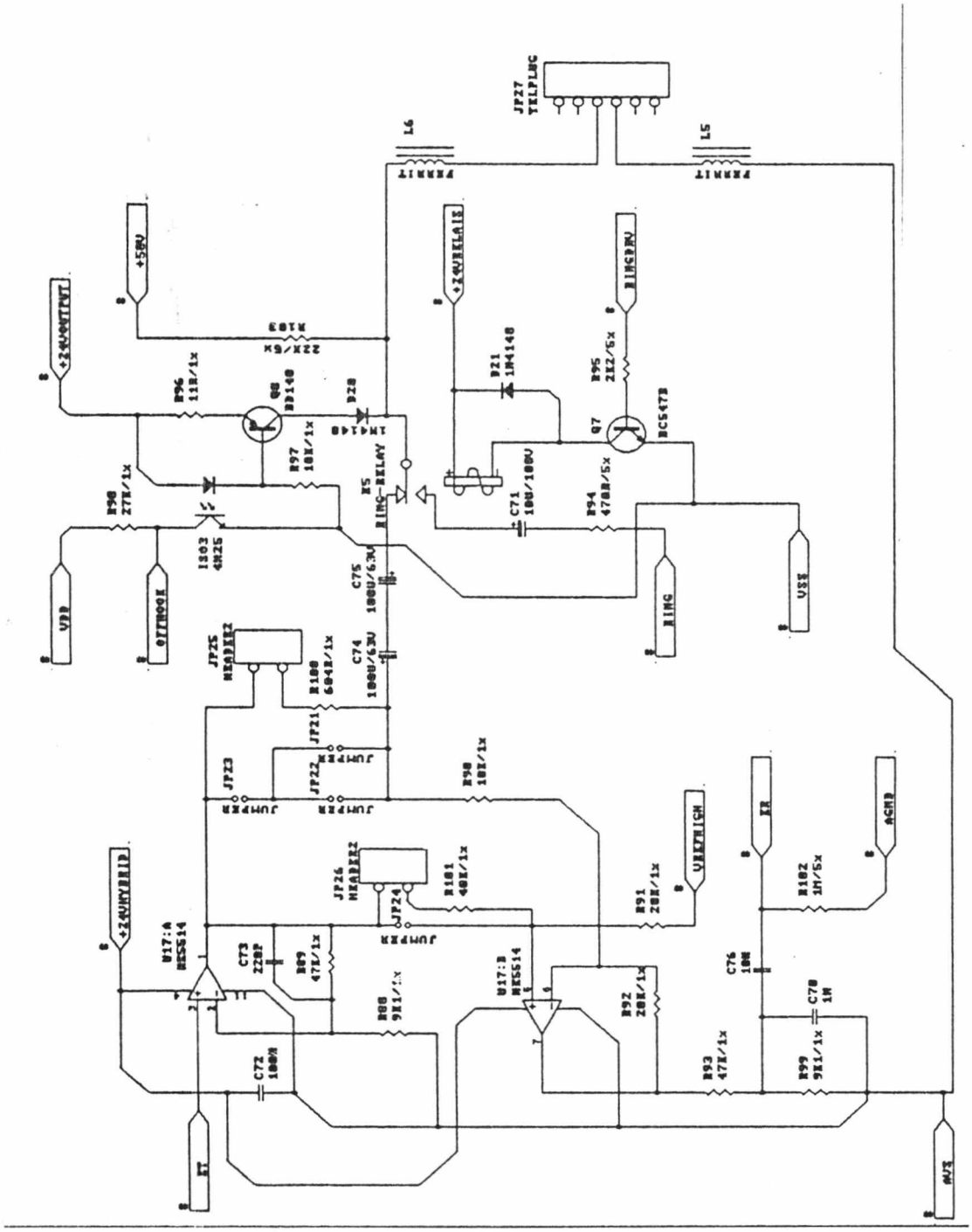
รูปที่ 3-8-1 วงจรของสายโทรศัพท์ภายในสายที่ 1 (Extension 1)



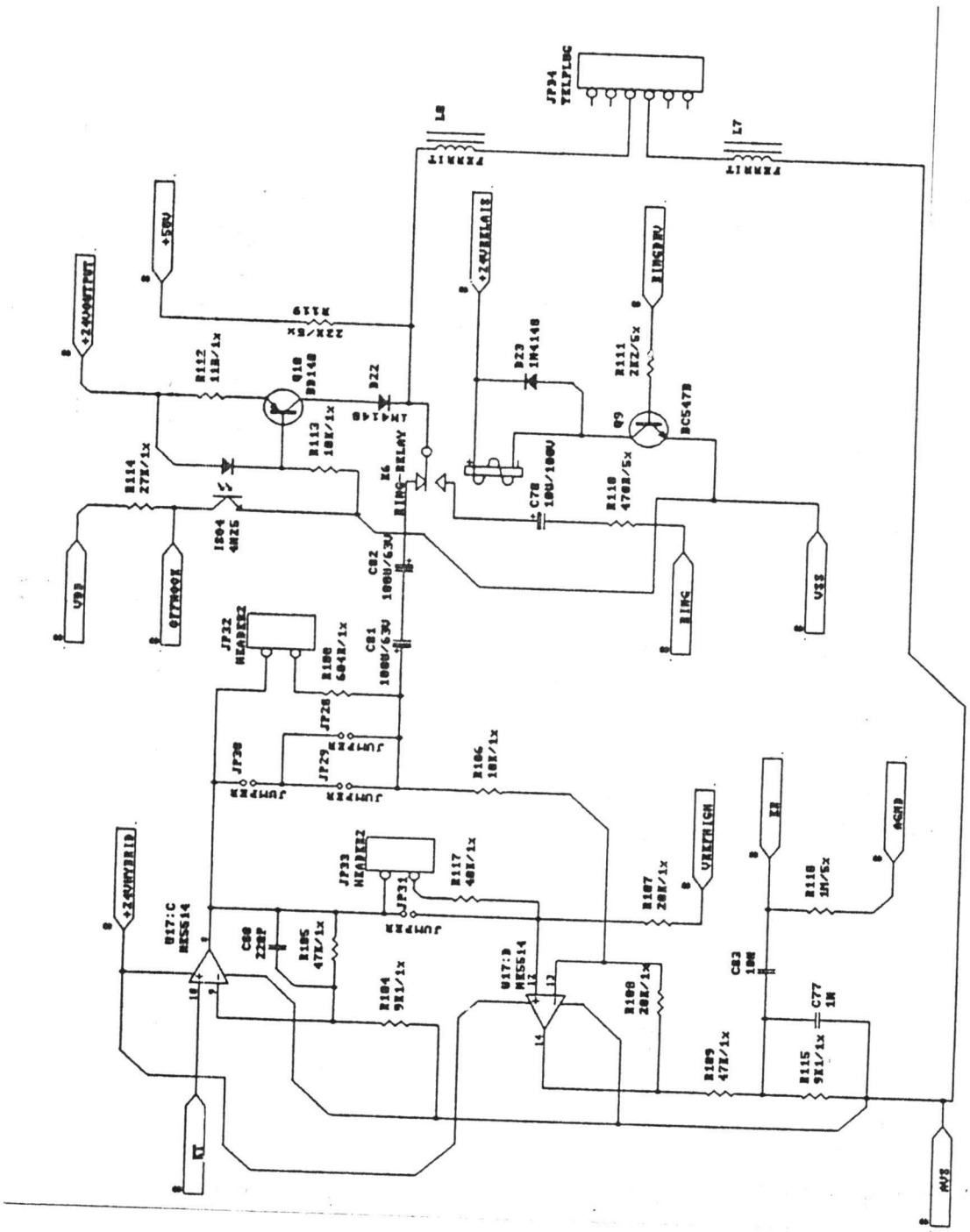
รูปที่ 3-8-2 วงจรของสายโทรศัพท์ภายในสายที่ 2 (Extension 2)



รูปที่ 3-8-3 วงจรของสายโทรศัพท์ภายในสายที่ 3 (Extension 3)



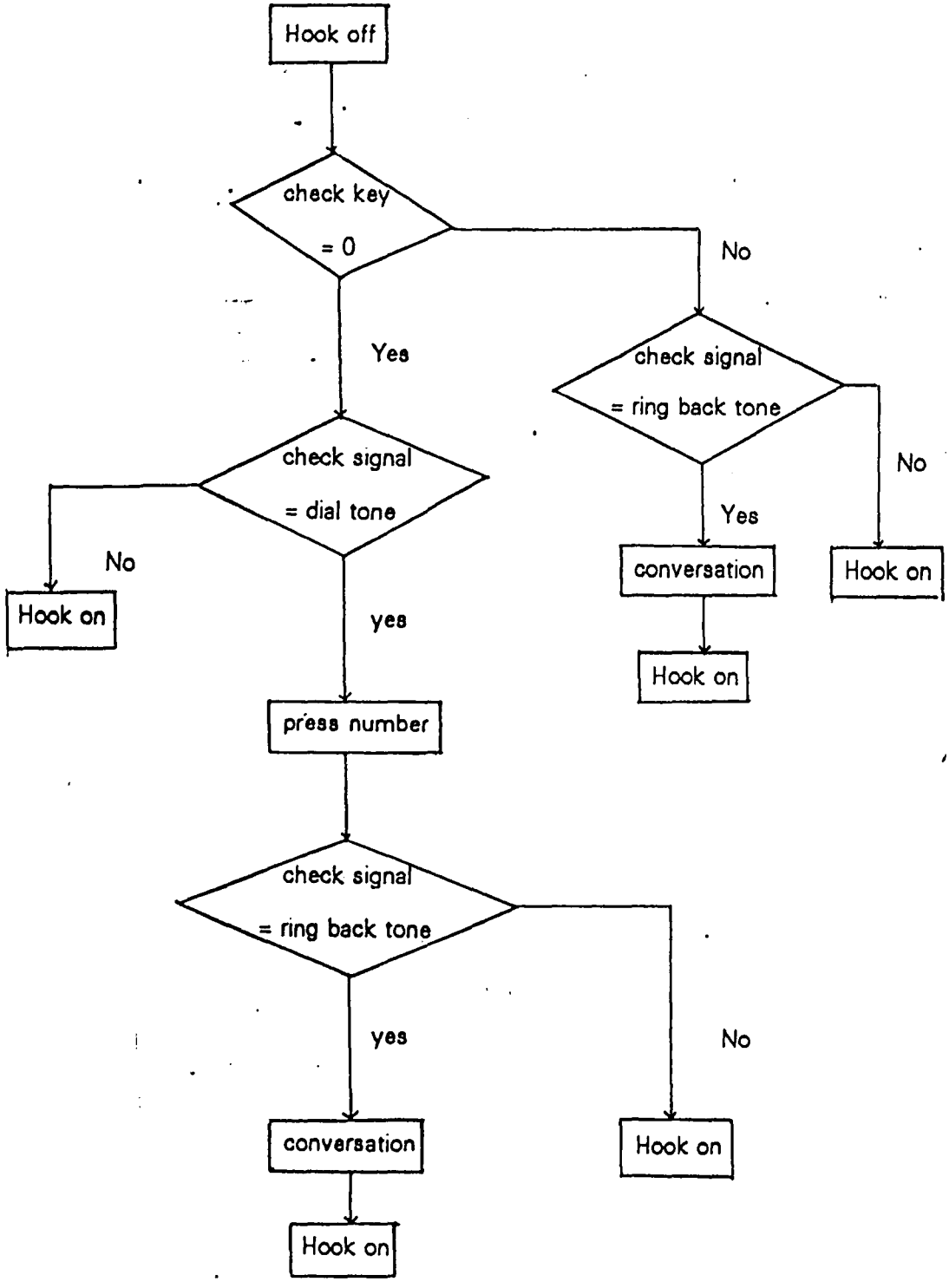
รูปที่ 3-8-4 วงจรของสายโทรศัพท์ภายในสายที่ 4 (Extension 4)



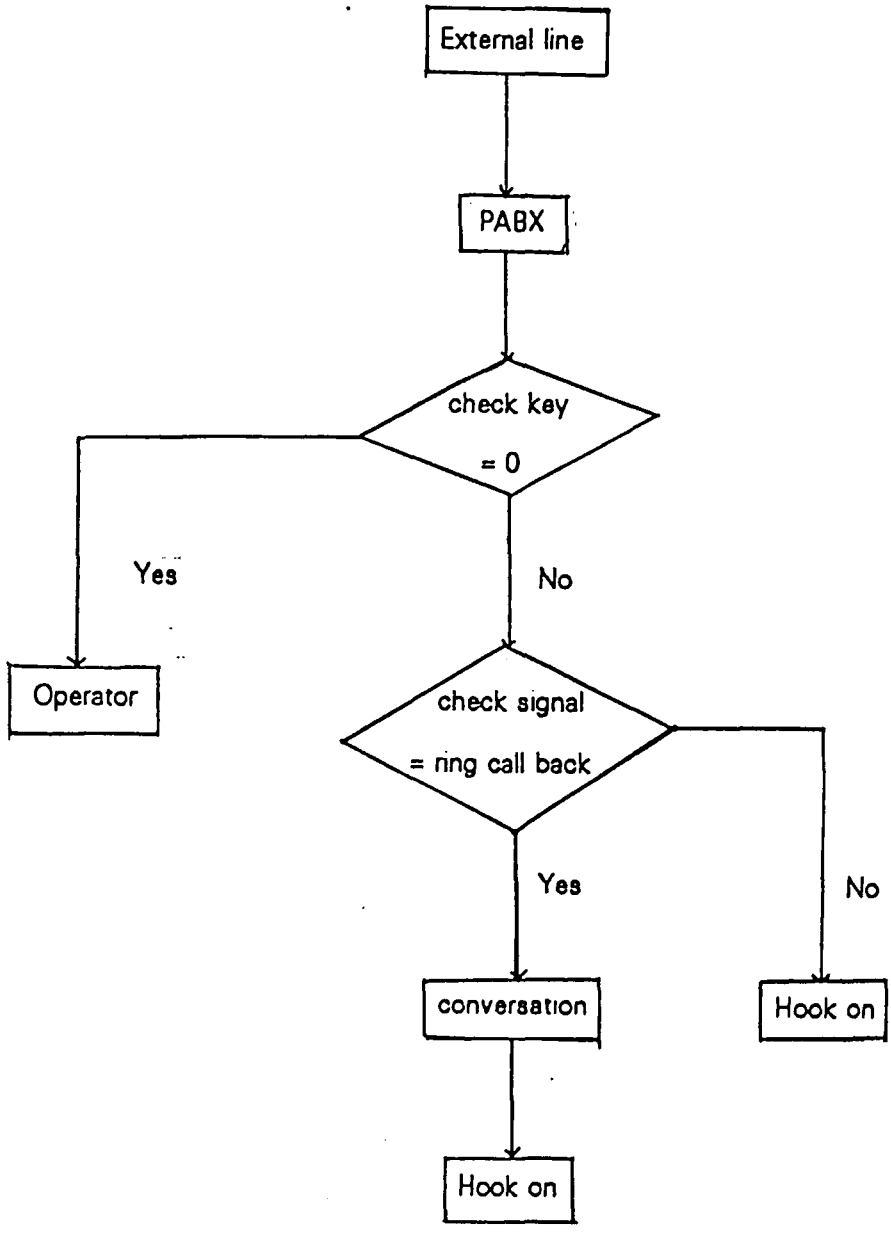
แผนผังการทำงานของระบบ PABX

ในส่วนนี้จะแบ่งลักษณะการทำงานของระบบ PABX ออกเป็นประเภทต่างๆดังนี้

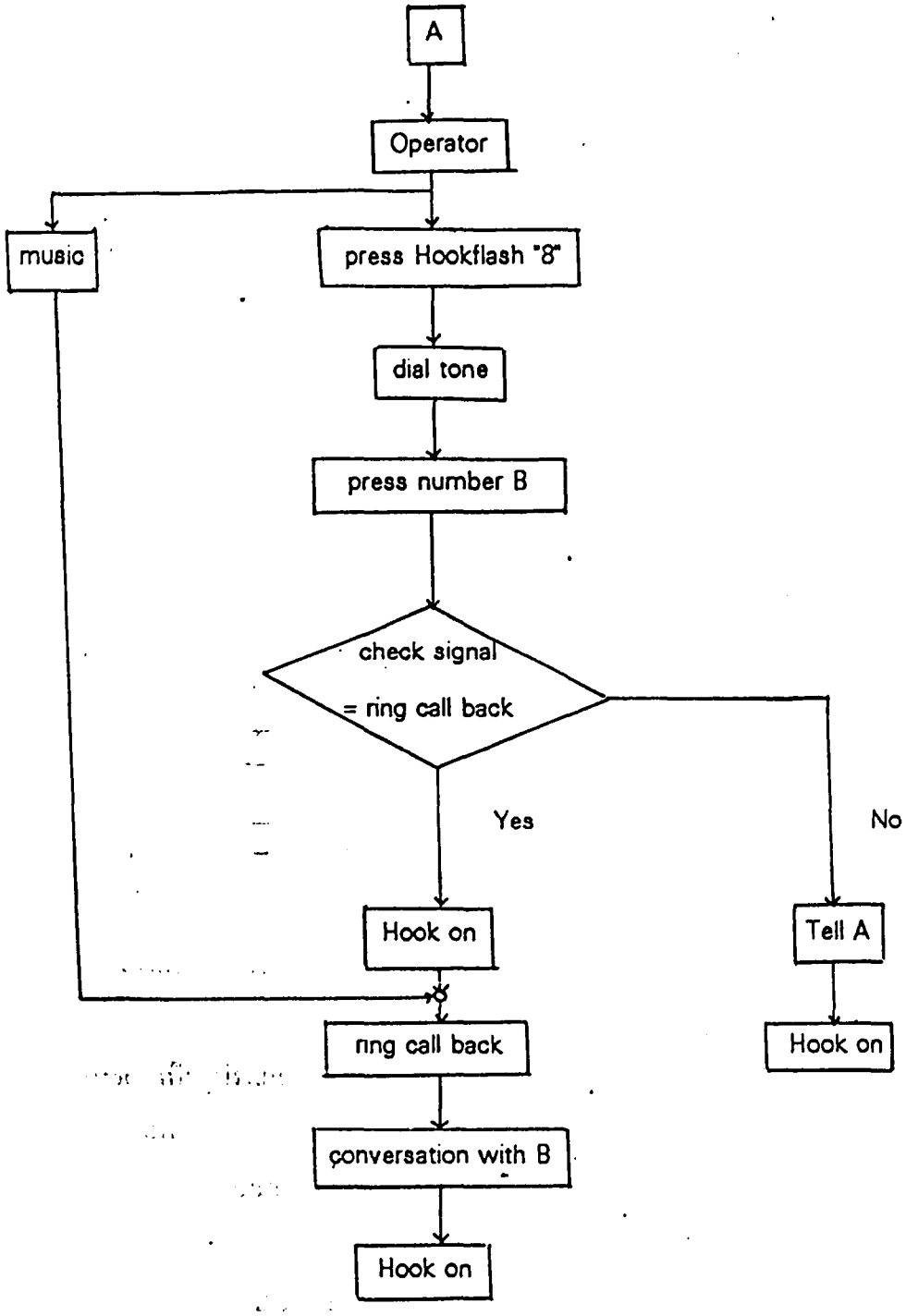
- การเรียกออก



- การเรียกเข้า


External line
PABX
check key
= 0
Yes
No
Operator
check signal
= ring call back
Yes
No
conversation
Hook on
Hook on

- การโอนสาย


A
Operator
music
press Hookflash "8"
dial tone
press number B
check signal
= ring call back
Yes
No
Hook on
Tell A
Hook on
ring call back
conversation with B
Hook on

3.3 โครงสร้างทางด้านซอฟท์แวร์

ระบบ PABX จำเป็นต้องใช้ซอฟท์แวร์เพื่อทำการควบคุมให้ฮาร์ดแวร์ทำงานตามต้องการ โดยที่ซอฟท์แวร์จะทำการควบคุมอุปกรณ์ภายนอกที่ต่ออยู่กับชิพ SC11390 ให้สามารถรับข้อมูลตามแต่ชนิดของอุปกรณ์จากระบบ ซึ่งข้อมูลนี้จะประกอบไปด้วย สภาวะของสาย (เช่น เปิดหรือปิด), ข้อมูลในการเชื่อมต่อ, ข้อมูลของการเรียกเข้า (เช่น สายไม่ว่าง,ระดับความถี่ของเสียงที่เรียกเข้า) และอื่นๆ ซึ่งข้อมูลที่ได้รับเหล่านี้จะถูกนำมาประมวลผลและซอฟท์แวร์ จะทำงานตามผลที่ได้รับโดยจะควบคุมอุปกรณ์ต่างๆต่อไป

ซอฟท์แวร์ของระบบ ประกอบด้วย 2 ส่วนคือ

1. ซอฟแวร์สำหรับระบบการจัดการ (Operating system software) ประกอบด้วย

1). ส่วนที่ทำงานตามคำสั่งที่ได้จากซอฟแวร์สำหรับการใช้งาน (Application software) ซึ่งฟังก์ชั่นเหล่านี้แบ่งตามกลุ่มของฮาร์ดแวร์ได้ 4 กลุ่มดังนี้

- SC11390
- PIO
- RS232
- EEPROM

2). ส่วนที่ทำการป้อนข้อมูลไปยังซอฟท์แวร์สำหรับการใช้งาน (Application software) ทุกๆ 1 ms ในขณะที่ระบบการจัดการกำลังป้อนข้อมูลเพื่อทำงานตามคำสั่งแล้ว ยังสามารถทำงานอื่นได้ด้วย

2. ซอฟท์แวร์สำหรับการใช้งาน (Application software) เป็นโปรแกรมที่กำหนดการทำงานของระบบ ซึ่งประกอบด้วย

- main ; เป็นโปรแกรมสำหรับเริ่มการทำงานของระบบการจัดการและการใช้งาน (Operating system and Application program)
- applicat ; เป็นโปรแกรมหลักของ Application software
- action ; เป็นโปรแกรมย่อยสำหรับการทำงานที่สภาวะต่างๆ
- reaction ; เป็นโปรแกรมย่อยสำหรับการทำงานตามคำสั่งที่สภาวะต่างๆ
- function ; เป็นโปรแกรมย่อยเพื่ออำนวยความสะดวกในการเขียนโปรแกรม
- interrup ; เป็นโปรแกรมย่อยเพื่อการตอบสนองการรบกวนของระบบ

ในระบบจะมีการแบ่งซอฟท์แวร์อย่างเป็นสัดส่วน แต่ยังมีการทำงานร่วมกันระหว่าง Operating system, Application software และส่วนของฮาร์ดแวร์ โดยที่ส่วนสำคัญของซอฟท์แวร์ของระบบคือ 80C32 microprocessor ขณะที่ระบบ PABX ประกอบด้วย SC11390 และ อุปกรณ์ภายนอกอื่นๆ

ตารางแสดงตำแหน่งของหน่วยความจำในระบบ PABX

ADDRESS	CS-SIGNAL	PERIPHERAL	REGISTER
0000-7FFF	RAMCS	EXTERNAL RAM	
8000 8001 800F	HVCS	SC11390	ADDRESS REGISTER DATA REGISTER
8010 8011 8012 8013 801F	PIOCS	82C55	PORT A REGISTER PORT B REGISTER PORT C REGISTER CONFIGURATION REGISTER
8020 802F	IOCS1	CSIO1	
8030 803F	IOCS2	CSIO2	
8040 804F	IOCS3	CSIO3	

ตารางที่ 3 จะแสดงถึงพื้นที่ของหน่วยความจำของระบบ PABX

โดย คอลัมภ์ที่ 1 จะแสดงถึงตำแหน่งของ address

คอลัมภ์ที่ 2 จะแสดงถึงสัญญาณ "chip-select" ซึ่งจะทำงานที่ address นั้นๆที่ลอจิก "0"

คอลัมภ์ที่ 3 จะแสดงถึงอุปกรณ์ภายนอกที่นำมาใช้งานที่ address นั้น

คอลัมภ์ที่ 4 จะแสดงถึงรีจิสเตอร์ซึ่งถูกเลือกให้ทำงานตาม address ที่กำหนด

บทที่ 4
ผลการทดลอง

การทดสอบตัวเอง (Selftest Facility)

ระบบ PABX ที่นำเสนอนี้สามารถทดสอบการทำงานของตัวมันเองได้โดยการสับสวิทช์ข้างหลอด LED ไปในทิศตรงข้ามกับ LED และต่อ RS232 ของระบบ PABX เข้ากับเครื่องคอมพิวเตอร์ แล้วใช้โปรแกรม Procomm หรือ Telix ทดสอบการทำงานของฮาร์ดแวร์ของระบบ โดยทำการเซตที่เครื่องคอมพิวเตอร์ ดังนี้

- 9600 Baud
- 8 Bit
- 1 Stopbit
- No Parity

และส่วนของฮาร์ดแวร์ให้ทำการเซตดังนี้

- สายภายนอกสายที่ 1 ต่อกับ สายภายในสายที่ 3
- สายภายนอกสายที่ 2 ต่อกับ สายภายในสายที่ 4
- Ring-amplitude ให้หมุนไป 180 องศาจากซ้ายสุด
- Tone-amplitude ให้หมุนไป 90 องศาจากซ้ายสุด
- Audio-amplitude ให้หมุนไปซ้ายสุด
- TT1,TT2,TR1 และ TR2 หมุนไปซ้ายสุด
- JP2 ให้ต่อขา 2-3,5-6
- JP49 ให้ต่อขา 1-2
- JP50 ให้เอาขา jumper ออก
- JP11,JP12,JP18,JP19,JP25,JP26,JP32,JP33,JP39,JP41,JP46 และ JP48 ให้เอาขา jumper ใส่เข้าไป

ผลการทดลอง

จากการทดสอบระบบ PABX ในการใช้งานจริง ระบบ PABX ที่นำเสนอจะสามารถทำงานได้ดังนี้

หมายเลข	การทำงาน
0	สายภายในต้องการติดต่อกับสายภายนอก เมื่อ 0 แล้ว ตามด้วยเลขหมายภายนอก
11-14	เป็นการติดต่อระหว่างสายภายในด้วยกันเอง โดยหมายเลข 11 , 12 , 13 และ 14 เป็นหมายเลขของสายภายใน

บทที่ 5
สรุปผลการทดลองและแนวทางการพัฒนา

โครงงานพิเศษที่น

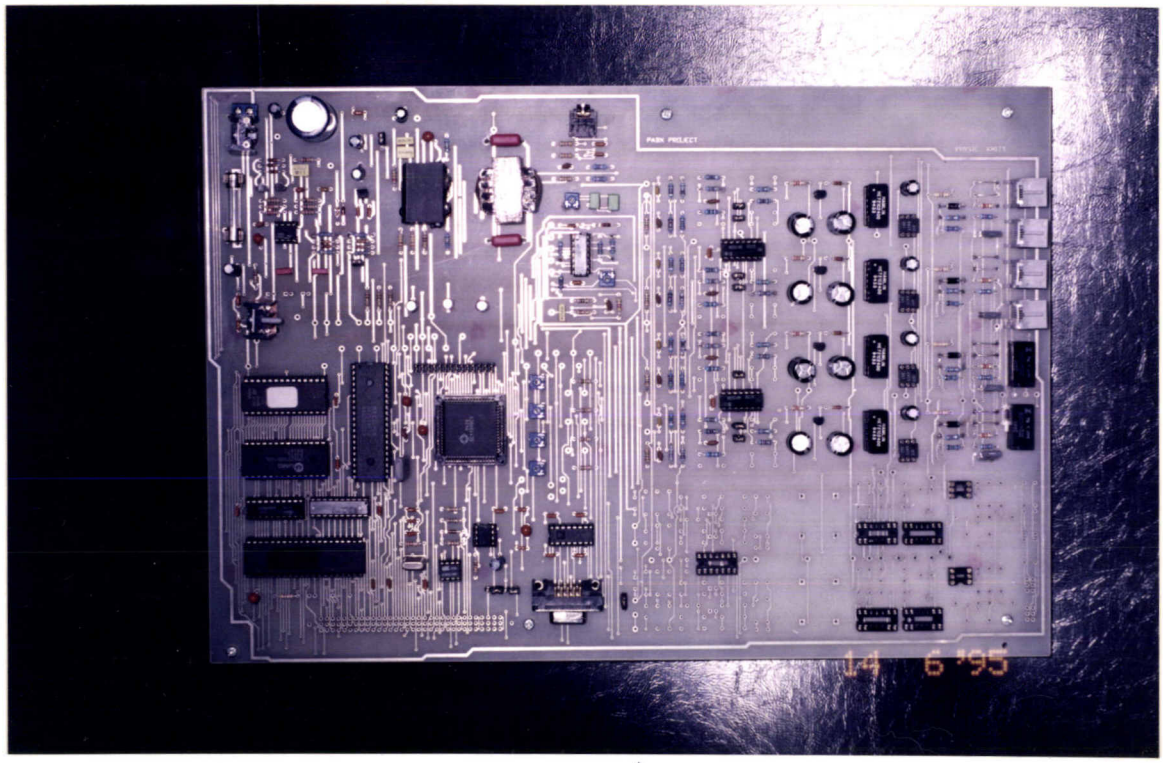
การเรียกออกและการเรียกเข้า ส่วนในกรณีของการรับส่งสัญญาณแฟกซ์นั้นยังไม่สามารถทำงานได้ เนื่องจากขาดอุปกรณ์

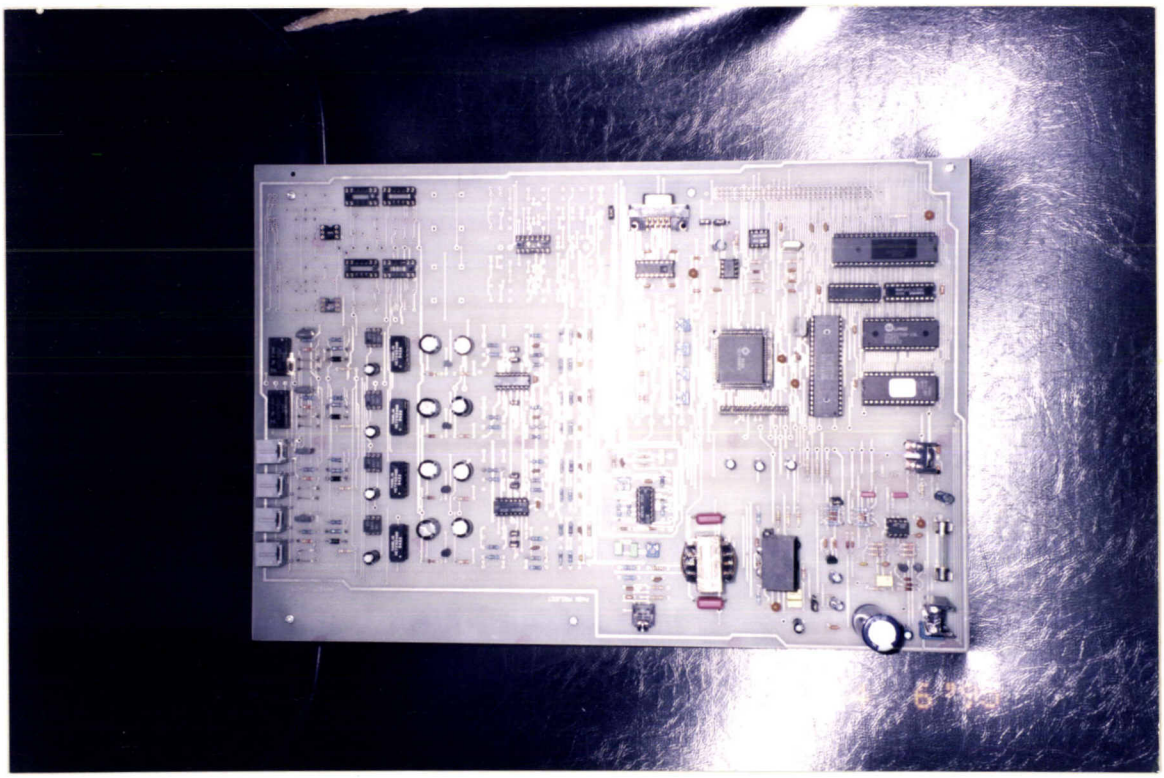
โครงงานนี้สามารถที่จะนำไปพัฒนา เพื่อเพิ่มความสามารถในการทำงานให้มากขึ้นได้ดังต่อไปนี้

1. การเพิ่มจำนวนคู่สายโทรศัพท์ให้มากขึ้น ซึ่งอาจจะทำได้โดยการเพิ่มสายภายนอกหรือสายขององค์การโทรศัพท์ และทำการขยายคู่สายภายในให้มากขึ้น

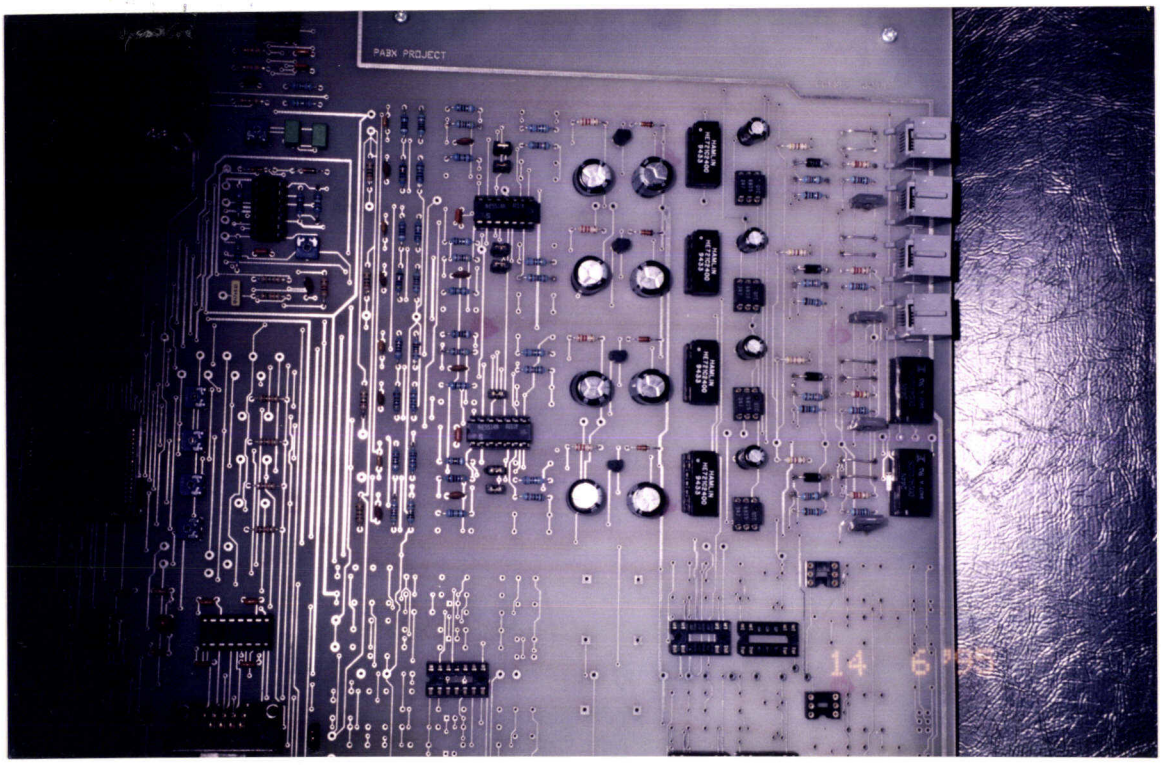
2. การพัฒนาระบบในส่วนของการรับ-ส่งสัญญาณแฟกซ์

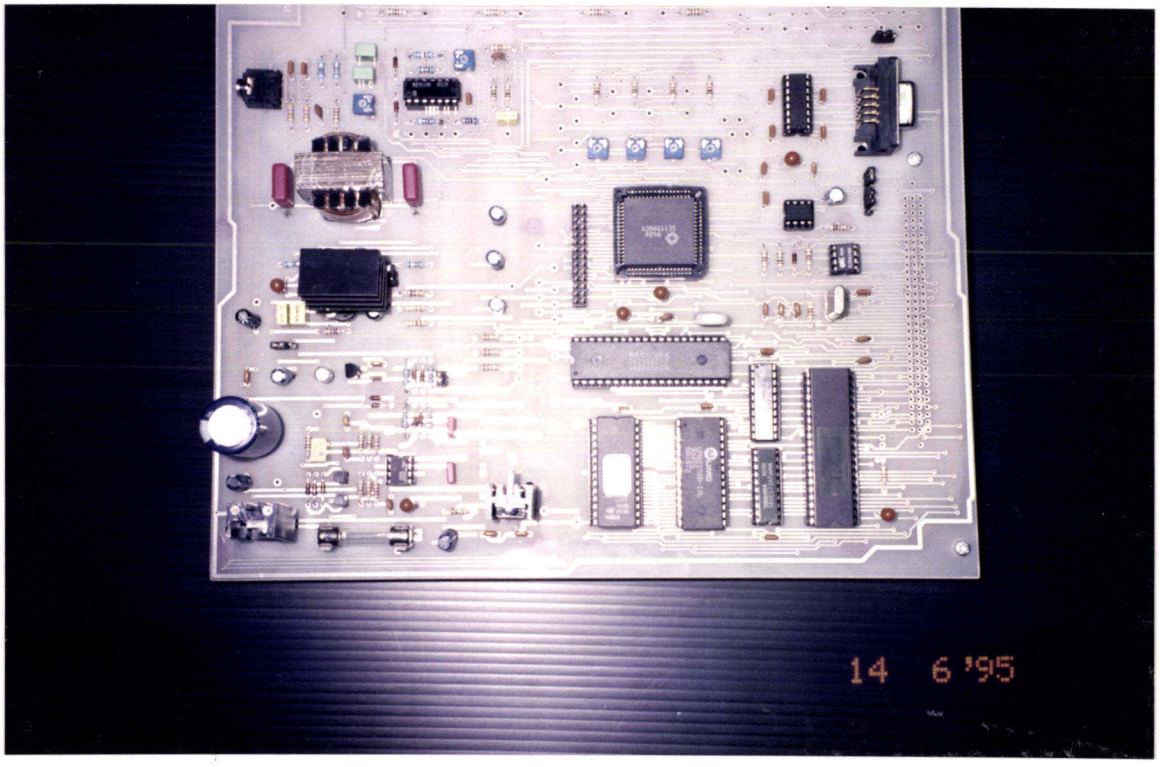
3. การพัฒนาในส่วนของการประชุมทางโทรศัพท์ โดยในขณะที่มีการประชุมทางโทรศัพท์ของสายภายในแล้ว สายภายนอกยังสามารถเรียกไปยังสายภายในสายอื่นที่ไม่ติดสายการประชุมได้





รูปแสดงผลงานของระบบชุมสายโทรศัพท์อัตโนมัติ PABX





รูปแสดงผลงานของระบบชุมสายโทรศัพท์อัตโนมัติ PABX

ภาคผนวก

SC11390 SWITCH UNIT

ลักษณะโดยทั่วไป

Switch Unit SC11390 (PABX Chip) เป็นวงจรรวมชนิด CMOS ของระบบ PABX โดยใช้ไฟเลี้ยง 5 Volts ซึ่งมีความสามารถในการขยายสายโทรศัพท์ภายใน 12 สายจากสายภายนอก หรือสายขององค์การโทรศัพท์ 5 สาย ซึ่งชิพนี้จะประกอบด้วยวงจรการรับ-ส่ง, วงจรการเรียกเข้า (call progress) สำหรับผสมเสียงพูดกับสัญญาณควบคุมเพื่อใช้ในระบบของPABX ส่วนประกอบสำคัญของ SC11390 คือ เป็น analog switch matrices แบบ 20x23 ports และ 4x4 ports ซึ่งสามารถควบคุมได้โดยการใช้ software นอกจากนี้ภายในชิพยังประกอบด้วยตัวส่งแบบ DTMF สำหรับส่งสัญญาณ 2 ตัว และ ตัวรับแบบ DTMF สำหรับตรวจจับระดับความถี่ (tone) 2 ตัว สำหรับเครื่องกำเนิดเสียงกริ่งโทรศัพท์บนชิพนั้น จะให้กำเนิดสัญญาณคลื่น sine หรือ square ซึ่งจะถูกขยายให้เกิดเป็นเสียงกริ่งโทรศัพท์ขึ้นที่สายภายใน ส่วนการเชื่อมต่อทางดิจิตอลในชิพนั้นจะติดต่อผ่านทาง address bus, data bus และ control bus 8 bits ของ microprocessor

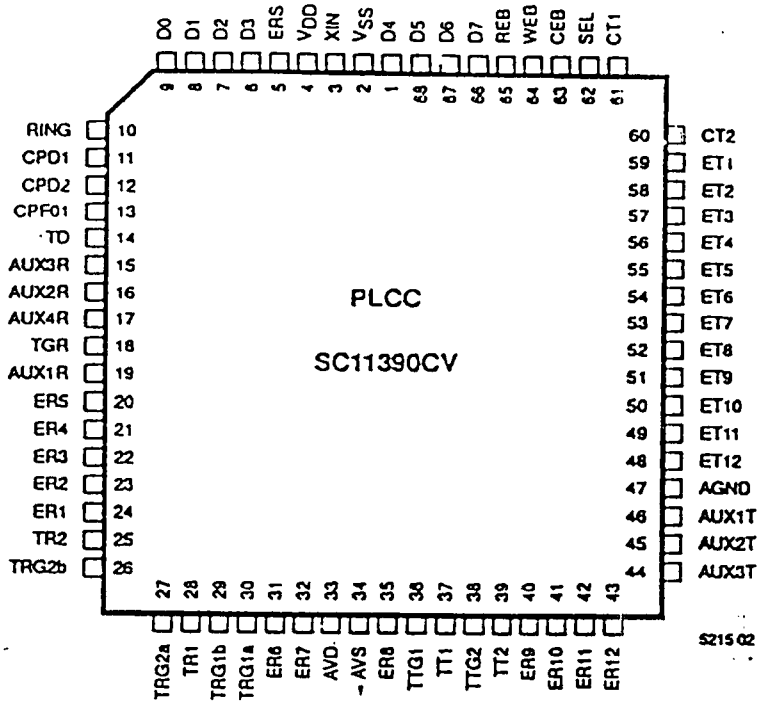
คุณสมบัติของ SC11390

- เป็นบัฟเฟอร์ขนาด 20x23 matrix
- ขยายสายภายในได้ 12 สายจากสายภายนอก 5 สาย
- มีตัวรับและตัวส่งแบบ DTMF 2 ชุด
- มีโปรแกรมสำหรับตรวจจับการเรียกเข้า 2 ชุด
- การโปรแกรมความถี่ของเสียงโทรศัพท์ 3 ระดับ
- ใช้สัญญาณดิจิตอลในการควบคุม
- การเชื่อมต่อพอร์ทของระบบ microprocessor เป็นแบบขนาน
- สามารถทำการประชุมทางโทรศัพท์ได้และมีอุปกรณ์ตรวจจับแฟกซ์อัตโนมัติ
- มีอุปกรณ์ช่วยในการเชื่อมต่อกับเทปหรือไอซีเก็บเสียง
- มีลักษณะเป็นแบบ PLCC 68 ขา หรือ แบบ PQFP 80 ขา
- เป็นระบบ CMOS ที่ใช้ไฟเลี้ยงวงจรเพียง 5 โวลต์

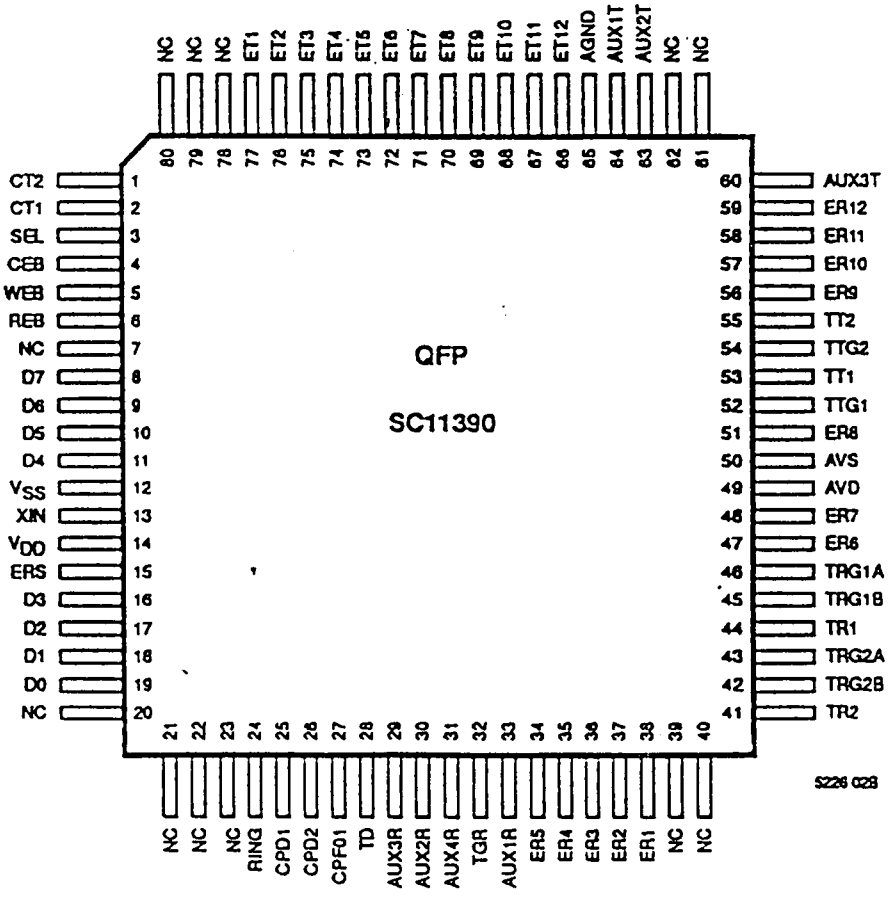
หลักสำคัญของระบบ PABX คือเป็น switch unit ซึ่งสามารถใช้เชื่อมต่อสายภายใน 12 สายจากสายภายนอก 5 สาย นอกจากนี้ switch unit ยังมีการเชื่อมต่อกับอุปกรณ์ควบคุมต่างๆที่สามารถควบคุมได้โดยการใช้โปรแกรม เช่น ตัวรับและตัวส่งแบบ DTMF (DTMF transmitters and receivers), ตัวตรวจจับสัญญาณเรียกเข้า (call progress detectors), ตัวตรวจจับเสียง (voice detector), เครื่องกำเนิดเสียงโทรศัพท์ (tone generator), เครื่องกำเนิดความถี่ และตัวกรองระดับความถี่ที่โปรแกรมได้ ซึ่งอุปกรณ์ควบคุมเหล่านี้จำเป็นต้องมีสวิทช์ระหว่างสายเพื่อที่จะส่งข้อมูลจากสายหนึ่งสายใดไปยังสายที่ต้องการได้

ลักษณะของ Switch Unit แบบต่างๆ

รูปที่ ก-1 แสดงลักษณะของ SWITCH UNIT แบบ PLCC



รูปที่ ก-2 แสดงลักษณะของ SWITCH UNIT แบบ PQFP



ในการควบคุมสัญญาณเรียกเข้าและสัญญาณเรียกออก ระบบ PABX จำเป็นจะต้องมีการเชื่อมต่อสายต่างๆ ซึ่งการเชื่อมต่อสายเหล่านี้จะมีการควบคุมสภาวะของสาย (เช่น เปิด หรือ ปิด) และสนับสนุนการทำงานแบบ hybrid ซึ่งการทำงานแบบ hybrid จะทำการแยกสัญญาณที่อยู่บนสายซึ่งเป็นผลรวมของสัญญาณเรียกเข้าและสัญญาณเรียกออก ให้เป็นสองสัญญาณโดยเฉพาะ ซึ่งสัญญาณรับและสัญญาณส่งนี้จะสามารถควบคุมได้โดยเป็นอิสระต่อกัน

ตารางแสดงขาต่างๆภายใน Switch Unit

PIN NAME	PIN NUMBER PLCC	PIN NUMBER QFP	DESCRIPTION
A_{GND}	47	65	Analog Ground supply terminal (bias output)
AUX1R	19	33	Auxiliary 1 receive
AUX2R	16	30	Auxiliary 2 receive
AUX3R	15	29	Auxiliary 3 receive
AUX4R	17	31	Auxiliary 4 receive
AUX1T	46	64	Auxiliary 1 transmit
AUX2T	45	63	Auxiliary 2 transmit
AUX3T	44	60	Auxiliary 3 transmit
AVD	33	49	Positive power supply for analog circuits
AVS	34	50	Negative power supply for analog circuits (0 Volt)
CEB	63	4	Chip Enable Bar (active low)
CPD1, CPD2	11, 12	25, 26	Call Progress Detector outputs (TTL totem-pole output from zero crossing detector)
CPF01	13	27	Output CP filter1 (analog output)
CT1–CT2	61–60	2–1	Control Tone input trunk (analog input, direct input to tone switch matrix
D0–D7	9–6, 1, 68–66	19–16, 11–8	Bidirectional databus bits 0–7
ER1–ER12	24–20, 31–32, 35, 40–43	38–34, 47–48, 51, 56–59	Extension Receive (analog input)
ERS	5	15	External Reset (active high)
ET1–ET12	59–48	77–66	Extention Transmit (analog output)
NC	—	7, 20-23, 39–40 61–62, 78–80	No Connect
REB	65	6	Read Enable Bar (active low)
RING	10	24	Ringer output
SEL	62	3	Register Select (low for address, high for data)
TD	14	28	Tone Detector output (TTL output from zero crossing detector)
TGR	18	32	Tone Generator Receive (analog input)
TR1, TR2	28, 25	44, 41	Trunk Receive (non-inverting input)
TRG1a	30	46	Trunk Receive Gain adjust 1a (inverting input)
TRG1b	29	45	Trunk Receive Gain adjust 1b (output)
TRG2a	27	43	Trunk Receive Gain adjust 2a (inverting input)
TRG2b	26	42	Trunk Receive Gain adjust 2b (output)
TT1, TT2	37, 39	53, 55	Trunk Transmit Output (transmitter output)
TTG1, TTG2	36, 38	52, 54	Trunk Transmit Gain adjust (inverting input)
V_{DD}	4	14	Positive power supply for digital circuits
V_{SS}	2	12	Negative power supply for digital circuits (0 Volt)
WEB	64	5	Write Enable Bar (active low)
XIN	3	13	Input of the oscillator or crystal connection to V_{SS}

ฟังก์ชันการทำงานภายใน Switch Unit

- ตัวส่งแบบ DTMF

ตัวส่งแบบ DTMF (DTMFT) นี้มีความสามารถในการให้กำเนิดความถี่มาตราฐานได้ 16 คู่ความถี่ ดังตารางที่ 1-ก

ตารางที่ 1-ก แสดงความถี่ DTMF

CTONE RTONE	High frequency group (column) 1209	1336	1477	1633
low 697 Hz	1	2	3	A
freq 770 Hz	4	5	6	B
group 852 Hz	7	8	9	C
row 941 Hz	*	0	#	D

ซึ่งรูปคลื่นแบบ sine สำหรับความถี่แต่ละความถี่นั้นจะประกอบขึ้นจากอัตราส่วนของแอมพลิจูดระหว่างกลุ่มความถี่สูงกับความถี่ต่ำ

จากสัญญาณ DTMF เมื่อทำการถอดรหัสแล้วจะแสดงไว้ในตารางที่ 3 ซึ่งจะเป็นข้อมูลที่ทำการส่งเป็นรีจิสเตอร์ 68 หรือ 69 ตัว โดยโหมดความถี่เดียวจะสามารถเลือกความถี่ได้ จากกลุ่มความถี่ต่ำหรือกลุ่มความถี่สูง ซึ่งจะถูกนำมา generated โดย RTONE และ CTONE บิต ของรีจิสเตอร์ 68 หรือ 69

- ตัวรับแบบ DTMF

ตัวรับแบบ DTMF (DTMF) จะทำการตรวจจับคู่ความถี่ในขณะนั้นจากตัวเลขที่กดลงบนคีย์ของโทรศัพท์ซึ่งจะแสดงในตารางที่ 2-ก สัญญาณอินพุทแบบอนาลอกจะมาจากเมตริกซ์หรือสัญญาณใดสัญญาณหนึ่งของ CT1 หรือ CT2

ตัวรับแบบ DTMFR จะให้กำเนิดสัญญาณควบคุม 2 สัญญาณเพื่อทำการตรวจเช็คสัญญาณที่ผิดปกติระหว่างการทำงาน

ตารางที่ 2-ก ตารางแสดงการเข้ารหัสและถอดรหัสของระบบ DTMF

(CTONE = 1 , RTONE = 1)

F-low	F-high	DGT	D3	D2	D1	D0
697	1209	1	0	0	0	1
697	1336	2	0	0	1	0
697	1477	3	0	0	1	1
770	1209	4	0	1	0	0
770	1336	5	0	1	0	1
770	1477	6	0	1	1	0
852	1209	7	0	1	1	1
852	1336	8	1	0	0	0
852	1477	9	1	0	0	1
941	1336	0	1	0	1	0
941	1209	*	1	0	1	1
941	1477	#	1	1	0	0
697	1633	A	1	1	0	1
770	1633	B	1	1	1	0
852	1633	C	1	1	1	1
941	1633	D	0	0	0	0

- เครื่องกำเนิดความถี่ของเสียงกริ่งโทรศัพท์

วงจรกำเนิดเสียงกริ่งโทรศัพท์ที่โปรแกรมได้นั้นจะให้กำเนิดสัญญาณรูป sine wave หรือ square wave ได้ 3 ระดับความถี่ สัญญาณรูป sine wave สำหรับความถี่ที่ได้จะถูกถอดรหัสดังตารางที่ 3-ก

ตารางที่ 3-ก แสดงการเข้ารหัสของเสียงกริ่งโทรศัพท์

F0 (Hz)	D1	D0
Power down	0	0
16 2/3	0	1
25	1	0
50	1	1

ซึ่งจะต้องมีการกำหนดที่แอดเดรสที่ 94 ว่าสัญญาณกริ่งโทรศัพท์เป็นสัญญาณรูปใด

ถ้า D2 = 0 แสดงว่าเป็นสัญญาณรูป sine wave

D2 = 1 แสดงว่าเป็นสัญญาณรูป square wave

โดยที่จังหวะของเสียงกริ่งโทรศัพท์นี้จะถูกควบคุมจาก microprocessor

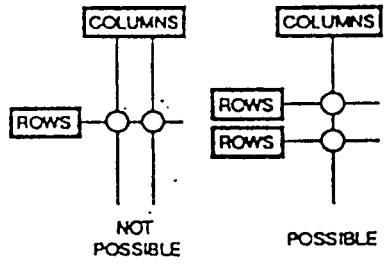
- วงจรไฮบริดจ์ (Hybrid)

สายโทรศัพท์ที่ใช้เพียง 2 สายนั้นสัญญาณที่อยู่บนสายจะเป็นผลรวมของสัญญาณส่งและสัญญาณรับ SC11390 จึงมีการวงจรไฮบริดจ์ภายนอกในการเชื่อมต่อกับอุปกรณ์ต่างๆและให้สายภายในกับสายภายนอกมีการแยกสัญญาณส่งและสัญญาณรับออกจากกัน ซึ่งสามารถควบคุมได้โดยเป็นอิสระต่อกัน

- เมตริกซ์ (Matrix)

SC11390 จะใช้สวิทซ์เมตริกซ์ขนาด 20 แถว 23 คอลัมภ์ซึ่งเท่ากับ 460 สวิทซ์ โดยแถวจะเป็นสัญญาณอินพุทของเมตริกซ์และคอลัมภ์เป็นสัญญาณเอาท์พุทของเมตริกซ์ซึ่งจะแสดงดังรูปที่ 1-ก

รูปที่ ก-3



- Matrix Programming

ในการต่ออินพุทกับเอาท์พุทนั้น อันดับแรกแถวของเอาท์พุทจะต้องทำการเลือกแอดเดรสและคอลัมภ์จะทำการเลือกข้อมูลที่จะเขียนลงในแอดเดรสนั้น

- เครื่องกำเนิดความถี่ (Tone generator)

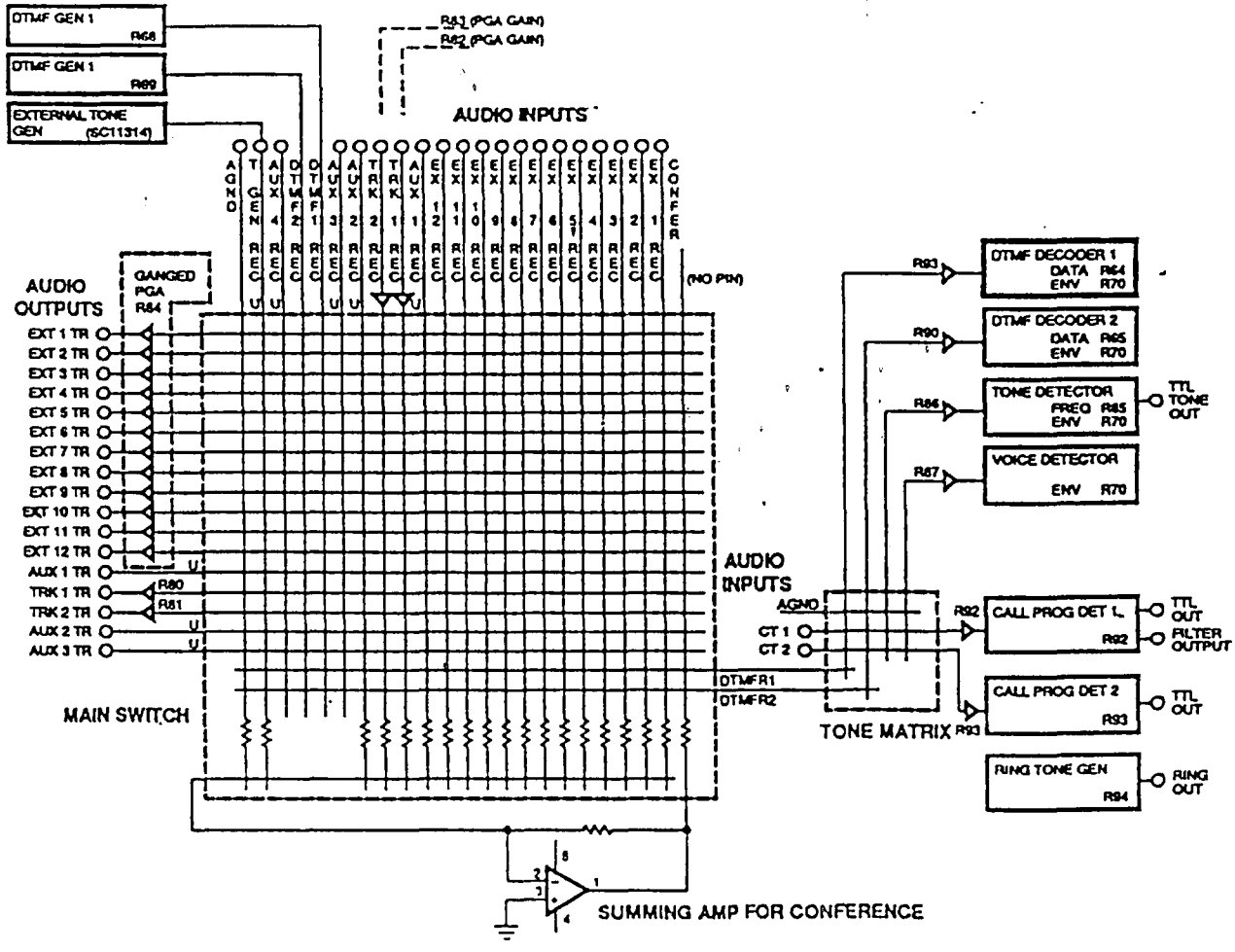
C1 เป็นสัญญาณอินพุทจากเครื่องกำเนิดความถี่ขา TGR ซึ่งสัญญาณนี้จะนำมาใช้สำหรับสัญญาณให้หมุนและสัญญาณสายไม่ว่างของโทรศัพท์

- การประชุม (Conferencing)

แถวที่ 19 (R19) จะมีเอาท์พุทเป็นผลรวมของสัญญาณเลือกของอินพุท ผลรวมของสัญญาณนี้จะต่อกับ C22 (conferencing column) ซึ่งในกรณีนี้ matrix จะใช้ในการประชุมทางโทรศัพท์

SC11390 จะใช้ extra 4x4 "tone matrix"เพื่อต่อกับอินพุทของ DTMFR, Tone Detector (TD) และ Voice Detector (VD)

รูปที่ ก-4 แสดงถึงอินพุทและเอาท์พุทของเมตริกซ์



หลังจากทำการรีเซต คอลัมภ์ที่ 0 จะต่อกับทุกๆแถว ซึ่งจะเป็นเพียงคอลัมภ์เดียวที่ต่อกับทุกแถวยกเว้นคอลัมภ์ที่ 1

- Voice Detector (VD)

ระดับของสัญญาณอินพุทของ VD สามารถปรับได้โดยใช้ PGA (แอดเดรส 87) ซึ่ง PGA จะเป็นผลมาจาก zero crossing detector (exceed) เพื่อแปลงสัญญาณเป็น square wave VD จะถูกใช้เป็น energy detector สำหรับตรวจสอบการทำงานของอินพุท

ANT-32

ANT-32 เป็นบอร์ดไมโครคอนโทลเลอร์ที่นำไปใช้งานในลักษณะ EMBEDDED CONTROLLER คือใช้ในงานควบคุมอัตโนมัติในระดับ 8 บิทโดยเฉพาะ ซึ่งจะถูกติดตั้งอยู่ในเครื่องมือ เครื่องจักรกล เครื่องใช้ไฟฟ้า รวมทั้งระบบอัตโนมัติต่างๆ บอร์ดนี้สามารถใช้กับ ชิพไมโครคอนโทลเลอร์ตระกูล MCS-51 ของ INTEL ซึ่งประกอบไปด้วย CPU เบอร์ต่างๆได้แก่ 8031, 8051, 8032, 8052, 8751, 8752 และ 8052 AHBASIC

ในบอร์ดนี้จะประกอบไปด้วยวงจรในส่วนของ Watchdog Timer, Battery Backup และ Power Fail Detector จะใช้ชิพ MAX691 ส่วนวงจร Real Time Clock จะใช้ชิพ DS1202

คุณสมบัติของบอร์ด ANT-32

- เป็นบอร์ดคอนโทรลใช้กับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 (8031/8032) ใช้ CPU เบอร์ 80C32 ทำงานที่ความถี่สัญญาณนาฬิกา 11.0592 MHz

- ใช้งานหน่วยความจำบนบอร์ดได้ 3 ตำแหน่งคือ

U2 เป็นหน่วยความจำโปรแกรม (PROGRAM MEMORY)ใช้กับ EPROM ขนาด 8-23 KByte เบอร์ 2764, 27128 หรือ 27256

U3 เป็นหน่วยความจำข้อมูล (DATA MEMORY) ใช้กับ RAM ขนาด 8 KByte เบอร์ 6264 หรือ 32 KByte เบอร์ 62256 สามารถแบคอัพข้อมูลได้โดยใช้แบตเตอรี่ลิเธียม

U4 เป็นหน่วยความจำโปรแกรมและข้อมูล (PROGRAM AND DATA MEMORY) ใช้กับ EPROM, RAM หรือ EEPROM ขนาด 8-32 KByte โดยใช้ EPROM เบอร์ 2764, 27256 ใช้ RAM เบอร์ 6264, 62256 หรือ EEPROM เบอร์ 2864(A), 28256(A)

- มีพอร์ต I/O เบอร์ 8255 จำนวน 2 ตัว (48 บิต) สำหรับต่อไปใช้งานภายนอก

- มีพอร์ต LCD สำหรับการต่อใช้งานกับ LCD แบบ DOT MATRIX

- มีวงจร SERIAL INTERFACE DRIVER RS232 ด้วยชิพเบอร์ MAX232 สำหรับการต่อเข้ากับเครื่องไมโครคอมพิวเตอร์

- มีวงจร Watchdog Timer, Power up/down ด้วยชิพเบอร์ MAX691

- มีวงจร RTC (Real Time Clock) ใช้ชิพเบอร์ DS1202

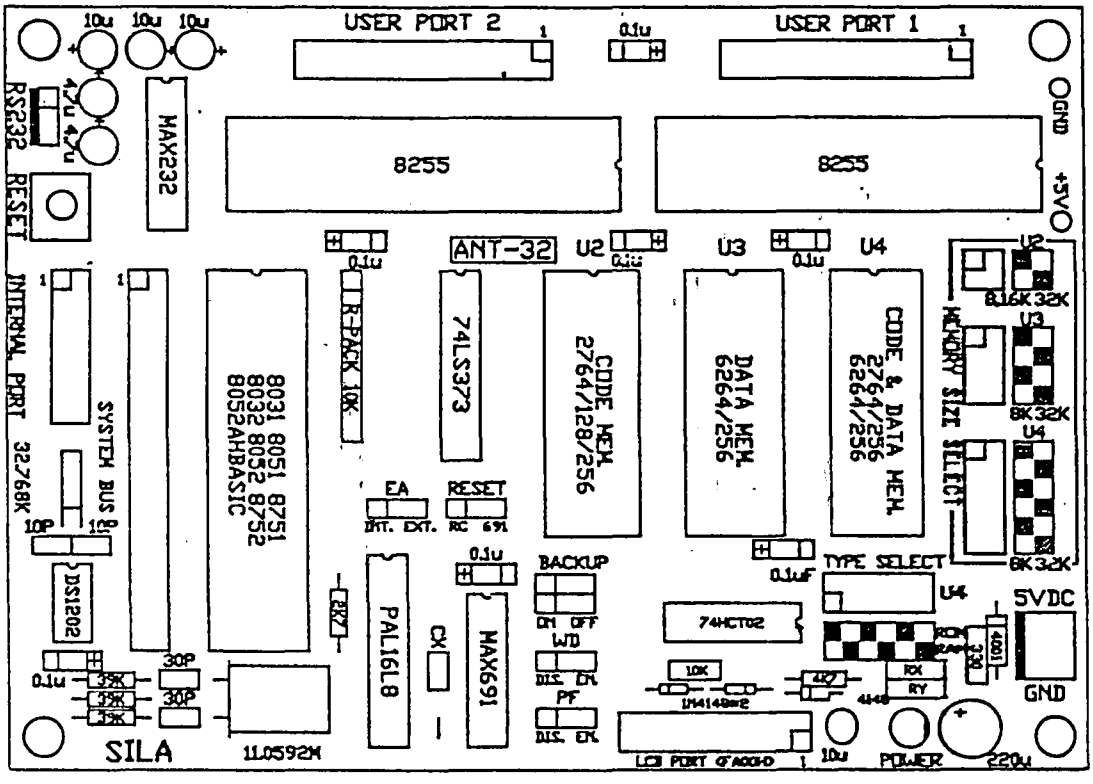
- มีคอนเนคเตอร์สำหรับ PORT1 ของไมโครคอนโทลเลอร์โดยเฉพาะ

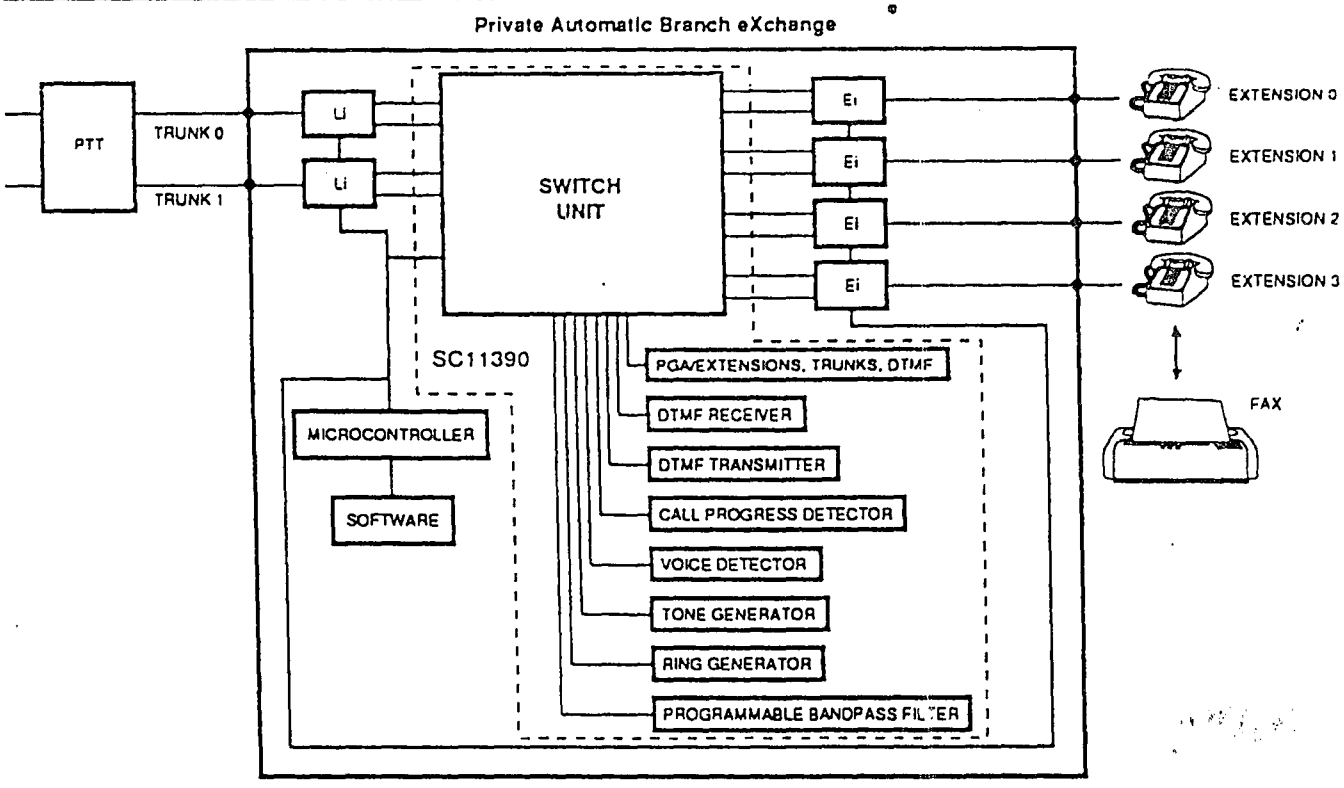
- มีคอนเนคเตอร์สำหรับ SYSTEM BUS ทำให้ขยายระบบได้ง่าย และสามารถใช้กับบอร์ดขยายต่างๆที่จะมีขึ้นในอนาคต

- สามารถเลือกเบอร์และชนิดหน่วยความจำ หรือกำหนดคุณสมบัติต่างๆของบอร์ดได้ด้วย jumper

- สามารถพัฒนาโปรแกรมได้ทั้ง ภาษาเบสิกและแอสเซมบลี้

รูปที่ ก-5 แสดงส่วนประกอบต่างๆของบอร์ด ANT 32





NOTE 1: Li = Line interface; Ei = Extension interface

Figure 1-1. A Typical PABX System - Based on SC11390

The heart of a PABX system is the switch unit which is able to connect internal and external phone lines in numerous ways. Figure 1-1 shows a typical PABX configuration. Besides the interconnections between the phone lines, the switch unit must also connect control devices like programmable gain controllers, DTMF receivers and transmitters, call progress detectors, voice detectors, tone generators, ring generators and programmable bandpass filters to the lines. All these control devices need to be able to be switched to the lines in order to get information from them or to perform an action on a particular line.

In order to control the incoming and outgoing signals, a PABX system needs line interfaces. These line interfaces control the state of a line (i.e. open or closed) and implement a hybrid function. The hybrid function separates the signal on the line, which is the sum of the incoming and outgoing signals, into two individual signals. These individual receive and transmit signals can then be controlled independently.

To enable this hardware to work as required, a PABX system needs software. The software has to control the peripheral devices to obtain specific information from the system. The information can consist of the state of the lines (i.e open or closed), information on established connections, call progress information (i.e. busy, ring or dial tone on the line) and so on. This information must be analysed and as a result the software will take action by driving other devices.

© Copyright 1993 Sierra Semiconductor

2.1 GENERAL DESCRIPTION

The system diagram on page 3-1 shows the partitioning and interaction between the operating system, application software and hardware. The kernel is formed by an 8032 microprocessor, while the PABX consists of the SC11390 chip and auxiliary PABX hardware such as the line interfaces.

2.2 THE SC11390 INTEGRATED PRIVATE AUTOMATIC BRANCH EXCHANGE

2.2.1 General Description

The SC11390 is a five volt CMOS system integration of a PABX for use with up to five external lines and as many as twelve extensions. The chip comprises all transmission, reception, call progress circuitry for mixing voice and control signals used in multi feature PABXs. The key elements of the SC11390 are two high ohmic analog switch matrices of 20x23 ports and 4x4 ports respectively, which can be programmed by software control. Also included in the chip are two DTMF transmitters for dialling and two DTMF receivers for recognition of tones. An on-chip ring tone generator produces sine or square wave signals that can be amplified to produce the ringing signals to the extensions. Programmable call progress facilities are built in for flexible, multi country tone detection allowing one unit to be developed for PTT approval with only minor software changes required between countries. The digital interface of the chip is compatible with the address, data and control busses of Intel compatible microprocessors.

2.2.2 SC11390 Features

- Fully buffered 20x23 matrix.
- Two DTMF transmitters and receivers.
- Two programmable call progress monitors.
- Three programmable tone ringer frequencies.
- Digital gain controls.
- Facilities for conference calls, automatic fax detection.
- Auxiliary interfaces for connection to answerphone, voice mail, and music on hold
- 68 pin PLCC or 80 pin PQFP package.
- Low power 5 volt CMOS technology.

2.3 THE EVALUATION BOARD

To demonstrate the capabilities of the SC11390 chip to customers an evaluation board has been designed. The heart of the board is formed by the SC11390 chip and an 8032 microprocessor. The board has connections for two external and four internal lines. Every extension and trunk has an interface to adapt the analog signals to appropriate levels and provide signalling functions. A ringer amplifier is included on the board to increase the power of the ring signal generated by the SC11390. The 82C55 PIO (Peripheral Input Output) controls the mechanical devices for seizing the lines and detects the state of the lines. An RS232 interface is included which provides an interface between a PC and the evaluation board.

A full description of the evaluation board hardware is provided in Sierra's SC11390/391 hardware design manual.

2.4 SOFTWARE OVERVIEW

The software for the evaluation board consists of two parts: an operating system and application software. The operating system also consists of two parts. Part one is executing the commands given by the application software. These functions can be divided into four hardware groups:

- SC11390
- PIO
- RS232
- EEPROM

The second part of the operating system is supplying information to the application every 1ms. In addition to supplying information and executing commands, the operating system has other tasks as well.

The application software determines the features, functions and performance of the PABX system. To create a modular system it consists of seven parts:

- main; for start up activities for both operating system and application program
- applicat; the main program of the application software
- action; subprogram with functions for actions on states
- reaction; subprogram with functions for reactions on states
- function; subprogram with functions used for convenient programming
- interrup; interrupt service routine for time critical actions of the application program.
- at_comm; subprogram to handle commands from a PC link.

Three interrupts are used in the evaluation board software system. Every 1 ms *rt_int()* is activated to update the operating system variables. Another interrupt, every 10 ms, takes care of the watchdog-circuitry and controls the time critical matters of the application software. A third interrupt controls the reading

© Copyright 1993 Sierra Semiconductor

and writing of the RS232 port.

The operating system and the real time interrupt are written in assembly and are assembled using the Keil C-51 compiler and linked to the application program (written in C). Because of this, parameters of the operating system functions are passed via fixed CPU registers.

In C, a function declaration has the following syntax:

parameter1 <function_name> (parameter2, parameter3);

Parameters are passed via the following registers:

Parameter	Byte	Word
2	R7	R6,R7
3	R5	R4,R5

The return value, parameter1, of a function is passed via:

Return Value	Register
Bit	Carry Flag
Byte	R7
Word	R6,R7(with MSB in R6 and LSB in R7)

2.5 SPECIFICATION

The following paragraphs list the features included in firmware revision 5.0 of the SC11390 evaluation board.

2.5.1 Intercom function

It is possible to dial another extension by dialling the other extension number (11, 12, 13 or 14).

2.5.2 External dialling

It is possible to dial externally from an extension by dialling '0' for an outside line. The user will first hear a dial tone generated by the evaluation board and after the '0' is pressed the outside dial tone. All dialled digits will be transmitted to the outside line. If a command, like establishing a conference call, has to be given to the PABX then a hookflash must be given first.

2.5.3 Incoming calls

The origin of an incoming call to any extension will be obvious by the type of ringing signal heard at the extension. Calls from outside lines will have one long ring followed by a pause and internal calls will have two short rings followed by a pause.

2.5.4 Busy / call waiting

If an extension is busy while there is a ring signal on one of the trunks, the internal extension will hear a 'call waiting' beep. A hookflash will connect the extension to the 'call waiting'.

2.5.5 Unattended call pick-up

If an unattended phone is ringing, another user can pick it up from their own extension by lifting the handset and pressing hookflash-"7". Incoming calls from trunk lines will have priority over calls between extensions.

2.5.6 Call hold / transfer

It is possible to hold a call from an external or internal line and to make another call either internal or external. The hold function is initiated by a hook flash. To transfer the call to the other line just put down the handset.

2.5.7 Enquiry / Toggle

While a call is held it is possible to toggle between the two calls. The switch between calls will be initiated by another hook flash.

2.5.8 Conferencing

If a call is holding and the user has made another call it is possible to connect all three parties together in a conference call. This will be initiated by putting the last called party on hold (hookflash) followed by pressing '8'. The extension which built up the conference call, will be master. An error tone is heard when the conference line is already in use.

2.5.9 Trunk disconnection from conference

An outside line on the conference line or an outside line on hold can be disconnected by pressing '9'. The PABX system determines which trunk has a busy or dial tone and disconnects that trunk. If no busy or dial tone is detected all trunks are disconnected, which were connected or on hold by the current device. An internal busy tone is set for three seconds on the extension to confirm the disconnection.

2.5.10 Trunk configuration

The transmit modes of the two trunks can be set manually and independently. To enable the trunk and set the signalling mode to pulse dial 31 or 32, where 1 or 2 is the number of the trunk. To enable the trunk and set the signalling mode to DTMF dial 41 or 42 respectively.

2.5.11 Automatic fax detection

The system can automatically detect the CNG tone of incoming faxes on one or both trunks and automatically route the call to a designated fax extension. Default the fax select mode is off.Enabling the fax detection and trunk selection is performed by pressing '58x' where 'x' is '0' to reset the fax detection, '1' for trunk 1, '2' for trunk 2, '3' for both trunks.

2.5.12 Setting the fax extension

To change the default fax extension dial 59 followed by the extension number. Example: to set the fax to extension 1 dial '5911'.

2.5.13 Secretary filter

All incoming calls can be made to go to one extension for use by a receptionist or secretary. The mode can be enabled by pressing '61' and disabled with '60'. The selection of the designated extension is done by

pressing '69xx' where 'xx' is the extension number as for fax extension selection.

Notes:

- No action is possible for an extension on hold. Going onhook ends the hold status.
- Only one extension, the master, has control over the conference call. If this extension goes on hook then the other devices can continue the conversation but no action is possible. Going on hook ends a parties conference participation.
- Two outside lines are allowed in a conference call but are disconnected if the extension goes on hook.
- Extensions can be pulse- or DTMF dialling types. The transmit mode of the trunk can be set independently of the mode of the extensions.
- If a non-existent code or number is dialled, the busy tone will be heard. Hookflash ends the busy situation.
- Configuration commands, like assigning the trunk transmit mode, are confirmed by a dial tone
- A hookflash is determined as going on hook for less than 750ms.

An overview of the functions is given in Table 2-1.

Number	Action
0	The extension is connected to an outside line.
11 to 14	For making internal connections.
30	Disable pulse transmit to outside lines.
31 or 32	To assign pulse transmit mode to the outside line. 31 for outside line 1 32 for outside line 2.
40	Disable DTMF transmission to outside lines.
41 or 42	To assign DTMF transmit mode to the outside lines.
580	Release the automatic fax redirect.
581	Set automatic fax detection on outside line 1.
582	Set automatic fax detection on outside line 2.
583	Set automatic fax detection on both outside lines.
5911 to 5914	Set default fax-extension.
60	Release secretary filter.
61	Set secretary filter.
6911 to 6914	Set default secretary extension.
8	Conference call.
9	Disconnect outside line on hold or in conference.
Hookflash 7	Unattended extension call pick-up
Hookflash	Set call on hold or toggle between calls if call already on hold
Replace handset	Transfer call on hold to another extension

Table 2-1.

3.1 SOFTWARE DEVELOPMENT STRATEGY

The main object for the program was to get an application which did not rely on any time critical parameter or action. This was necessary to accomplish a system which can be very complex, and therefore very time consuming, and to feature the capability of adding extra features. In order to accomplish this, the switching conditions for the line matrix need to be constant while connecting the devices. This means that the parameters may not alter during one program cycle. Also, all the time critical actions like DTMF tone reading and the settings of audio tones, have to be centralized and may not be affected by the main program flow.

To cut down the program into a logical and compact flow, the program has to distinguish states for each device. The state of a device can depend on the state of the off-hook switch, the previous state of this switch, whether there is an internal, external or no call at all, etc.

Another point in the strategy was to let the operation be performed by an extension who wants this action. Since many actions are needed for some states, and these actions can affect different devices, a strategy like this will lead to a clear procedure for where decisions are taken. In other words: actions on other devices resulting from an earlier action are initiated by the extension who wanted the original operation.

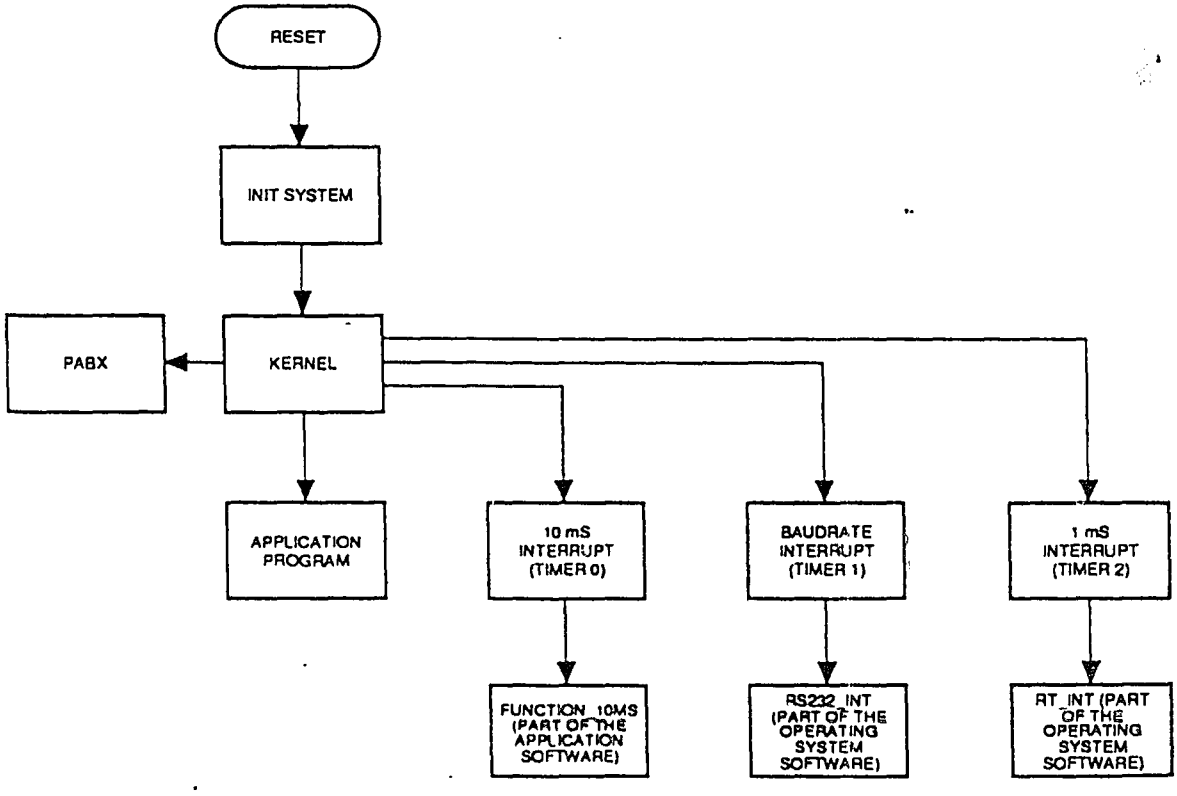
Example: If extension 1 tries to establish a connection with extension 2 then function connect_int() is called after the state of extension 1 is determined. The function sets the ring tone for extension 1 and the ring signal on extension 2. Function connect_int() will still be called by extension 1 every new program cycle and tests for extension 2 to go off hook. If so, then the extensions will be connected. Therefore, extension 2 goes off hook, but connection is made by the function which was called after determination of extension 1.

The flow charts should be viewed in conjunction with the source code. Program listings are not included as Sierra has a strategy of continuous development to improve the code and therefore the source code may have changed since the time of publication of this manual.

3.2 UNIT MAIN.C

In function main() of unit main.c all operating system variables are declared and initialized. In the function init_appl_var() the global application variables are initialized. Init_trunk_transmit_mode() can determine if the outside line is connected to a DTMF or a pulse telephone exchange. The function application() is nested in an infinite loop at the end of main().

SYSTEM DIAGRAM



© Copyright 1993 Sierra Semiconductor

State		Action
Extensions		
1	ONHOOK WAS ONHOOK	break
11	redial activated	handle_redial
2	ONHOOK WAS OFFHOOK	function *reset ()*
10	fax selected and detected	connect_fax
	OFFHOOK WAS ONHOOK	
3	no_ring signal	function *connect2pabx ()*
4	ring signal external	function *connect_trunk ()*
1	ring signal internal	break
6	OFFHOOK WAS OFFHOOK connection consultation/conference call ready for input busy tone ring tone internal	function *check_number_flash ()*
Trunks		
7	RING SIGNAL	function *ext_ring_set*
8	NO RING SIGNAL	function *ext_ring_reset*
1	BUSY TONE	break

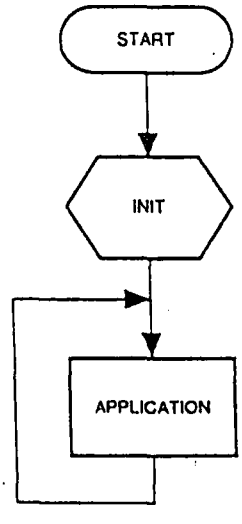
Table 3-2.

3.3 UNIT APPLICAT.C

3.3.1 Application()

At the beginning of application() all significant variables from the operating system, read rt_int(), are copied to variables for the application software. This is done to let the application software work with constant values for updating the states of all devices (i.e. extensions and trunks) and the total time for one complete application cycle can be longer than the time between two, 1ms, interrupts. With the constant parameters the state for each device is determined and action is taken depending on the state. Next, the operating system variables are copied once again and the application software can work out the new data.

VOID MAIN (VOID)



3.3.2 Copy_interrupt_variables()

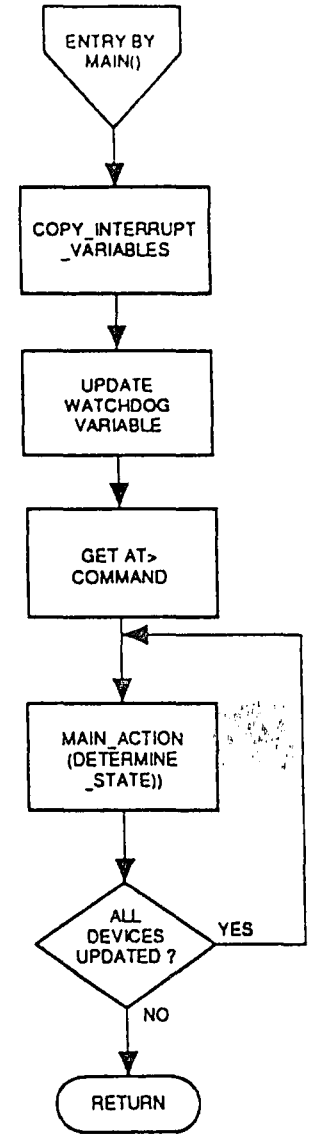
Time-critical variables are copied from the operating system variables to application variables. Also the filtered signals busy and dial tone are copied. *Appl_trunk[].status* is only used for filter purposes, *appl_trunk[].busy*, *appl_trunk[].dial*, *appl_trunk[].ring*, and redial are constant during one complete program cycle.

3.3.3 Determine_state()

One of the most powerful features of the program is the structure of first determining the state of a device and the next action of that state. This is powerful because this makes it easy to add features to the PABX just by defining a new state. An overview of the states is given in Table 3-2.

The state of a device depends on several factors. The device number is most important. device numbers 0-3 represent extensions 1-4.

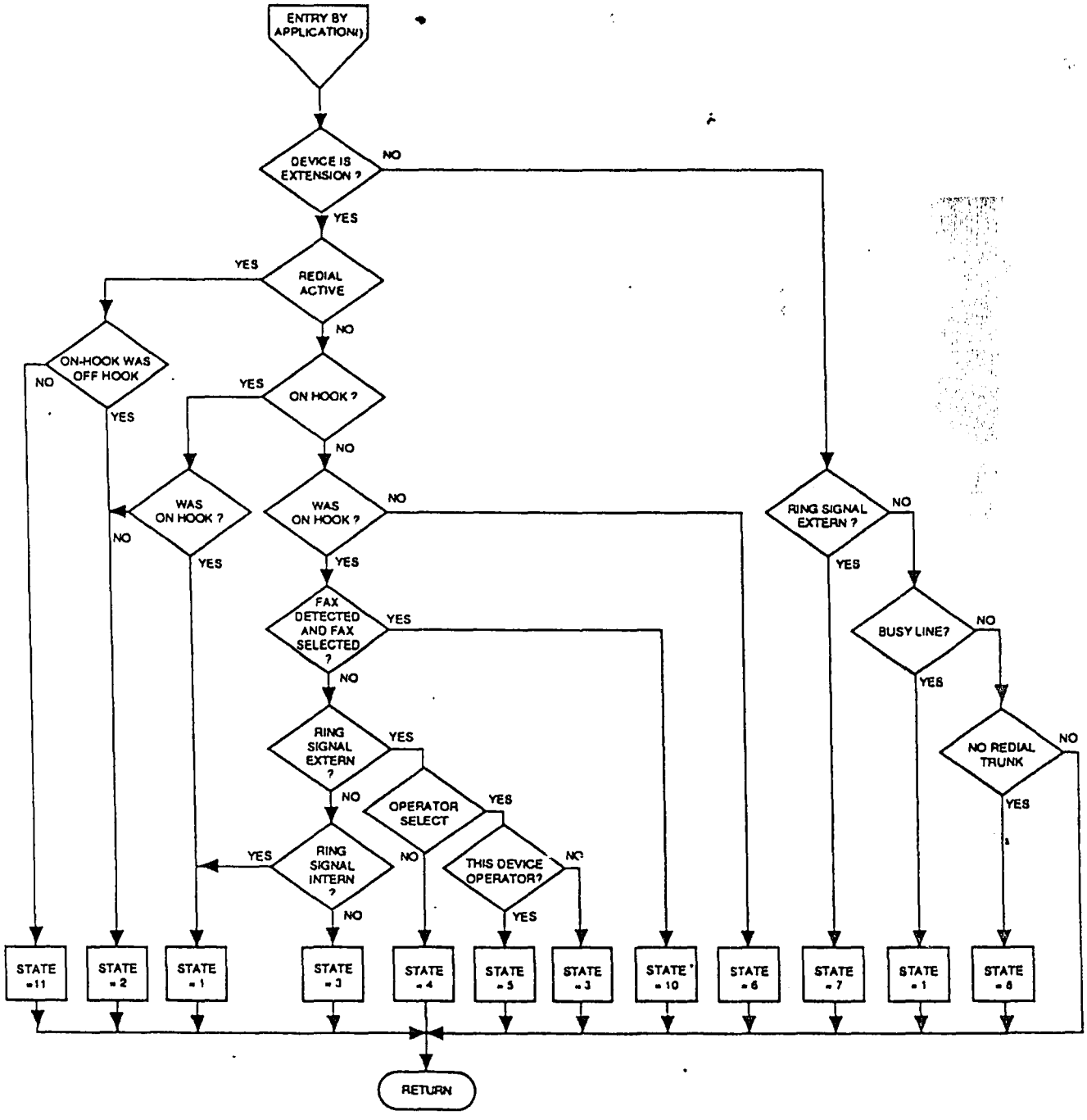
VOID APPLICATION ()



device numbers 4 and 5 represent trunk 0 and trunk 1 respectively.

Extensions and trunks have different states. States 1-6 and 10,11 are extension states. States 1, 7, and8 are states of the trunks. For an extension, on hook/off hook, the old situation of on hook/off hook and internal or external ring signal are the factors which determine the state. The factors external ring signal and busy tone determine the state of the trunk. The function determine_state() returns a variable which contains a value corresponding with the current state for the selected device.

BYTE DETERMINE_STATE ()



3.3.4 Main_action()

In main_action() a function call is made corresponding to the value of the variable state. For example if state is six, the function check_-number_flash() is called.

3.3.5 Init_appl_var()

Used to initialize the application variables with default values and to set default connections (i.e. DTMF transmitters on the trunks and the tone generator to the line matrix)

3.3.6. Init_trunk_transmit_mode()

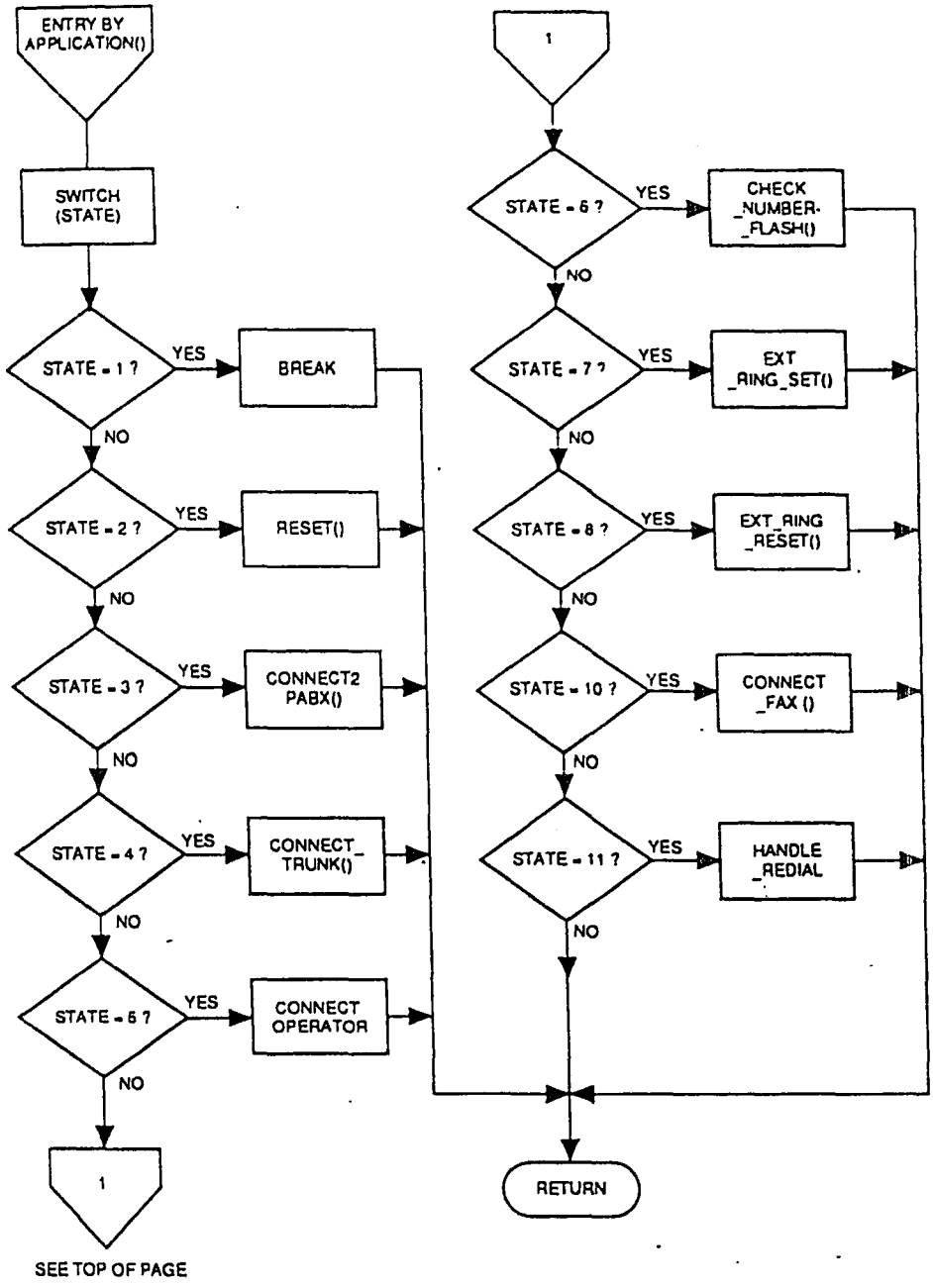
As a compilable option the PABX can configure the trunk lines itself. If a dial tone is present on the trunk, a DTMF '0' will be transmitted. The telephone exchange responds by resetting the dial tone, the transmit mode is set to DTMF, else the transmit mode is pulse. So if there is no outside line connected, the trunk is disabled

3.4 Unit ACTION.C

3.4.1 Connect2PABX()

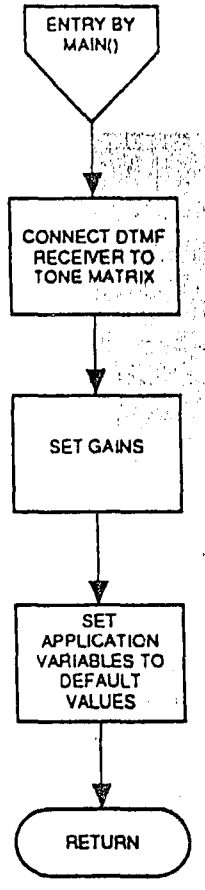
When an extension goes from on hook to off hook and there are no ring signals (internal or external) *Connect2PABX()* is called. The only

VOID MAIN_ACTION (BYTE STATE)

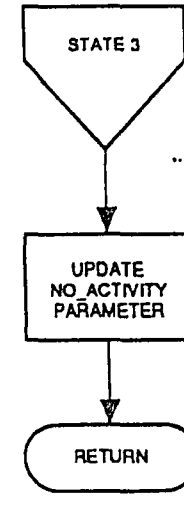


SEE TOP OF PAGE

VOID INIT_APPL_VAR (VOID)



VOID CONNECT2PABX ()



thing it supports is initialization of the variable no_activity[device].

3.4.2 Check_number_flash()

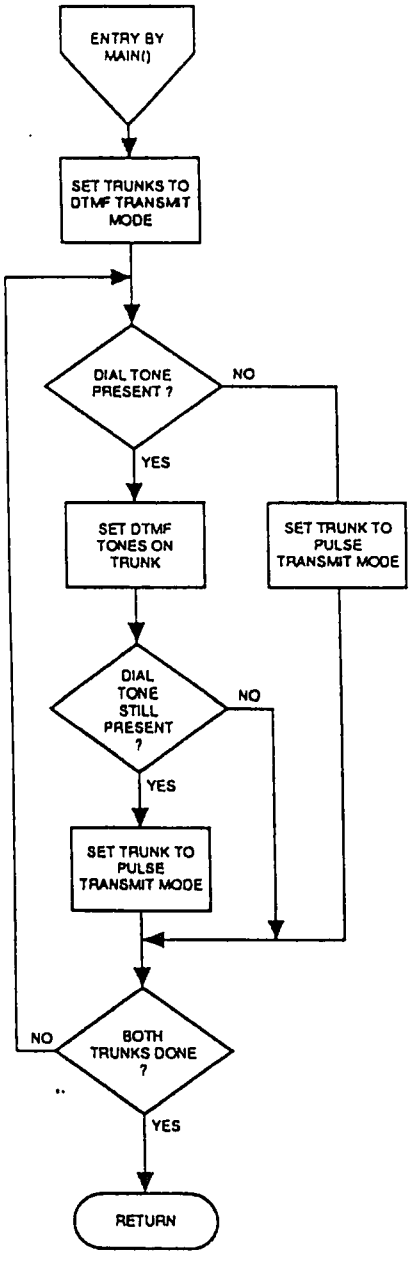
When an extension is off hook, all the operations for this extension will be initiated from *check_number_flash()*. This function checks for hookflashes and number input. Hookflashes can always be given and cause the extension to be connected to the PABX[1] or to toggle between two lines if there is consultation.

The first step for function *check_number_flash()* is to check if the extension is on hold. If positive, no action is taken and the program is returned to its caller. The second step is to check if the extension is trying to establish an internal connection. If so, then function *connect_int()* is started to connect the devices. Next, hookflash is checked. If there was no hookflash, then only the possibility of dialling a number or doing nothing (i.e. while a connection is already established) is left.

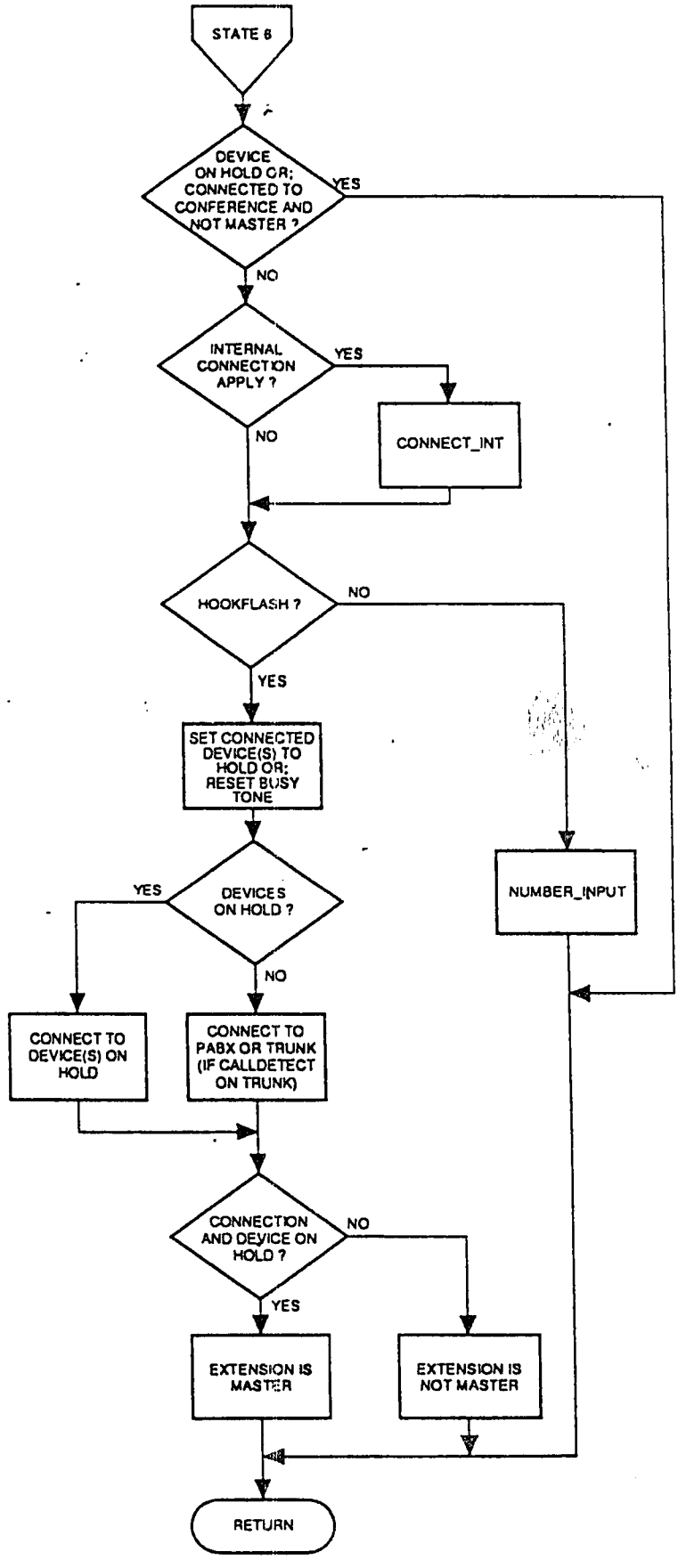
Giving a hookflash causes the program to set the device(s) that the extension was connected to, on hold. If there was no device connected then the audio signals for

[1] a connection to the PABX means that a dial tone is set and the extension is ready for dialling.

VOID INIT_TRUNK_TRANSMIT (VOID)



VOID CHECK_NUMBER_FLASH ()



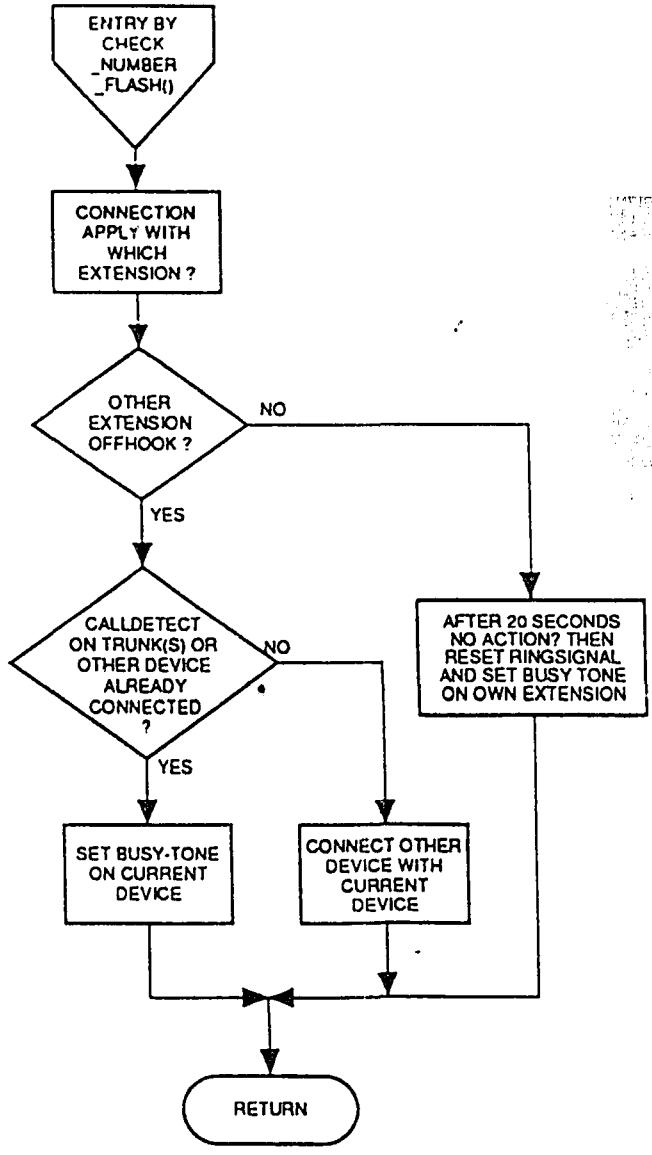
this extension will be reset. If there was a device on hold for this extension then the device on hold will be connected to the extension. If no devices were on hold then a hook-flash means that the extension has to be connected to the PABX[1] or, if an outside line was ringing, to the trunk.

3.4.3 Connect_Int()

As mentioned above, *connect_int()* connects two internal devices (i.e. extensions). Since the strategy in the application software is to execute all operations by the extension that wants to command a function, the establishing of the internal connection is executed by the extension which is calling.

Therefore the program needs to know which extension is being called and then this extension is checked for going off hook. If this extension does not go off hook for a certain period, default 20 seconds, the call will be reset. If the extension goes off hook, then an extra test is needed to check if there is an external call or if the extension is already connected. If so, the busy tone of the current extension is set, because an external call has a higher priority then an internal call. If the test is negative, then the ring signal for the caller and the ring tone on the current extension will be reset and the extensions are connected via the matrix.

VOID CONNECT_INT ()



3.4.4 Number_Input()

When an extension is connected to the PABX or when it is involved in a multiple call (i.e. one connected line and one line on hold) then it is possible to input digits. Function *number_input()* filters this input to the desired operations.

If the current extension is calling another extension, input is not accepted except for the redial command. In case no number input is given and there is no connection, the program performs a time test. This is done to disconnect the DTMF receiver in case an extension is off hook and no key is pressed. If a connection already exists, then this test is overruled.

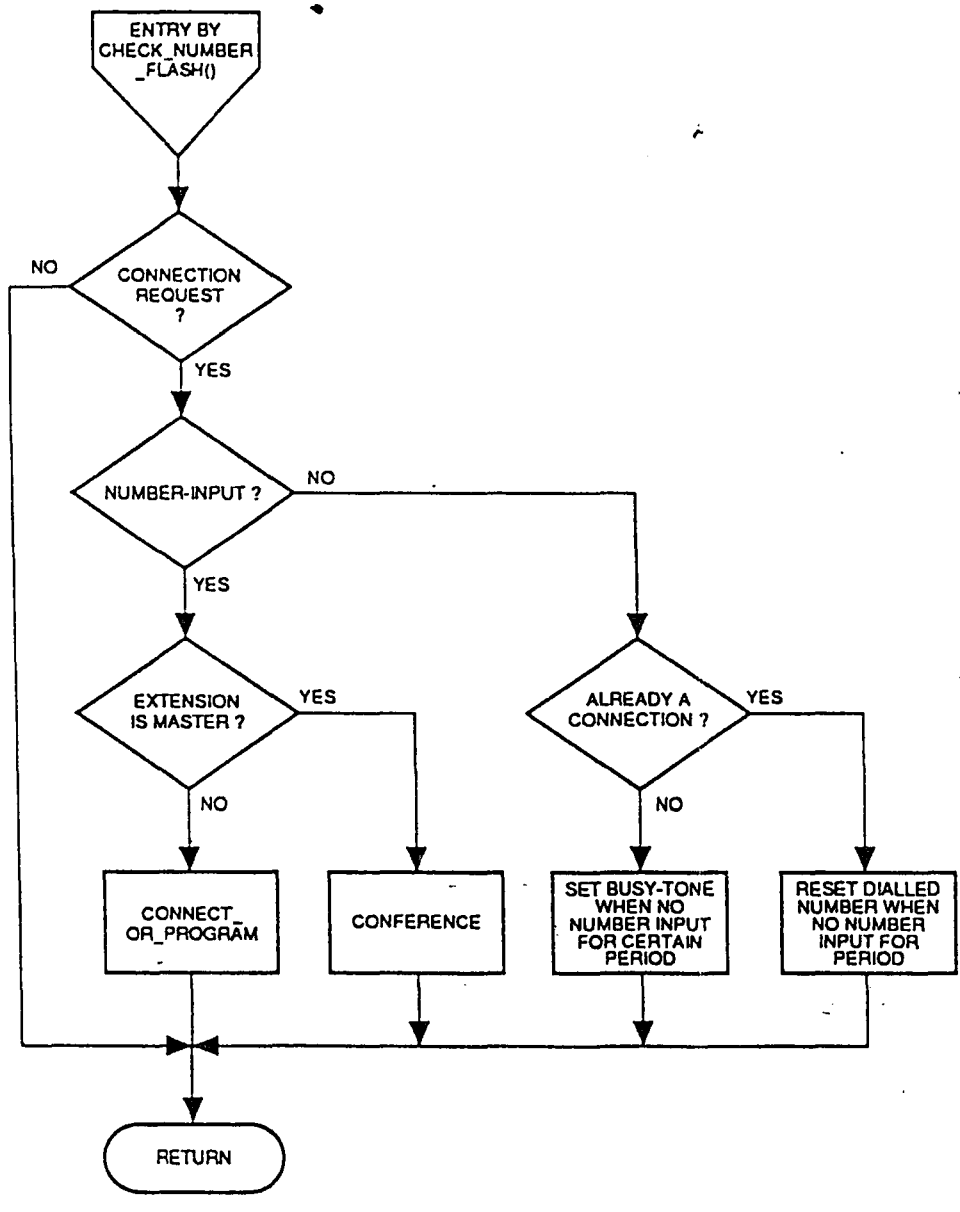
In case of number input, there will be two options. First, there could already be a connection and that implies that a conference call is required. Secondly, no connection is in progress so dialling is expected or the PABX is to be programmed. The function *connect_program()* is called.

3.4.5 Connect_or_program()

Connect_or_program is called as soon as digits are entered and no connection was established. This means that digits were entered while connected to the PABX.

The function checks the dialled digits for valid commands and calls functions to execute these commands. A valid command can be an extension number or a code for programming the PABX. Depending on the number of features this function will expand.

[1] a connection to the PABX means that a dial tone is set and the extension is ready for dialling.



The valid numbers are stored in rxd_buf[device]. Every extension has its own buffer. By means of rxd_get[device] and rxd_put [device], the buffer is read or written. Rxd_buf[device][0] is the start address of the receive buffer of an extension. Rxd_get is the write offset value of this buffer. The read offset value is rxd_put. So with a pointer it is simple to write or read the number receive buffer (rxd_-buf[extension] [maximal numbers]).

3.4.6 Conference()

The function *conference()* is entered as soon as an extension has a line connected. At the moment that an extension has another line on hold as well, it will be possible to connect the three devices together in conference.

The program checks the dial buffer for the conference command and the device for having a device on hold. If so then three or more (max= 17) devices will be connected to the conference line, if available.

3.4.7 Connect_conf()

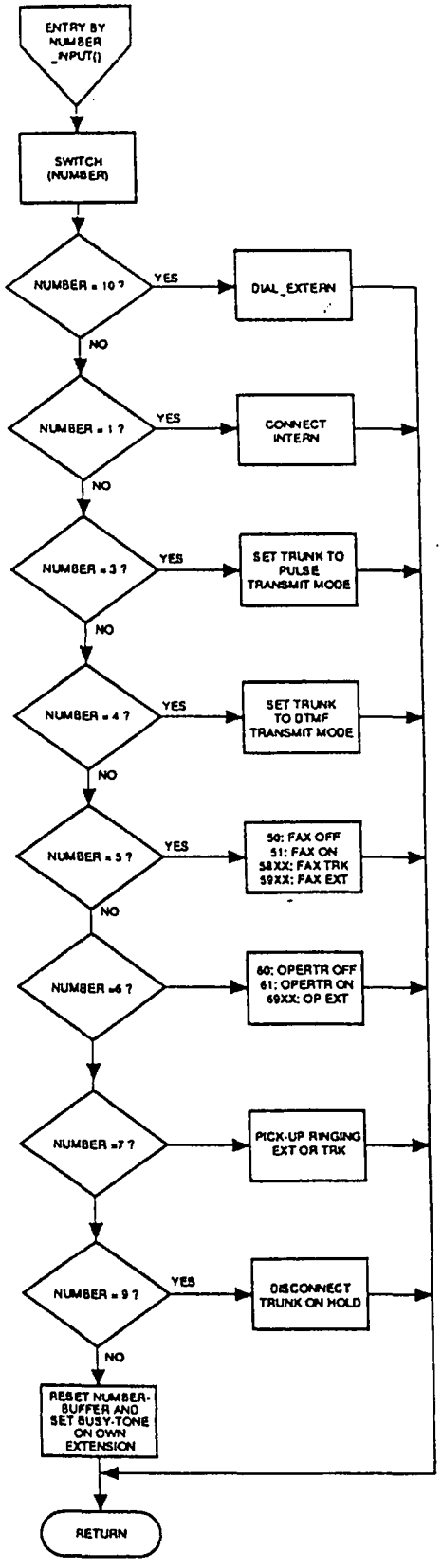
Connect_conf() is called by th function *conference()*. to establish conference call. Depending on th variable connection, a device c hold is connected to the conferenc line, or a new conference is built.

3.5 UNIT REACTION.C

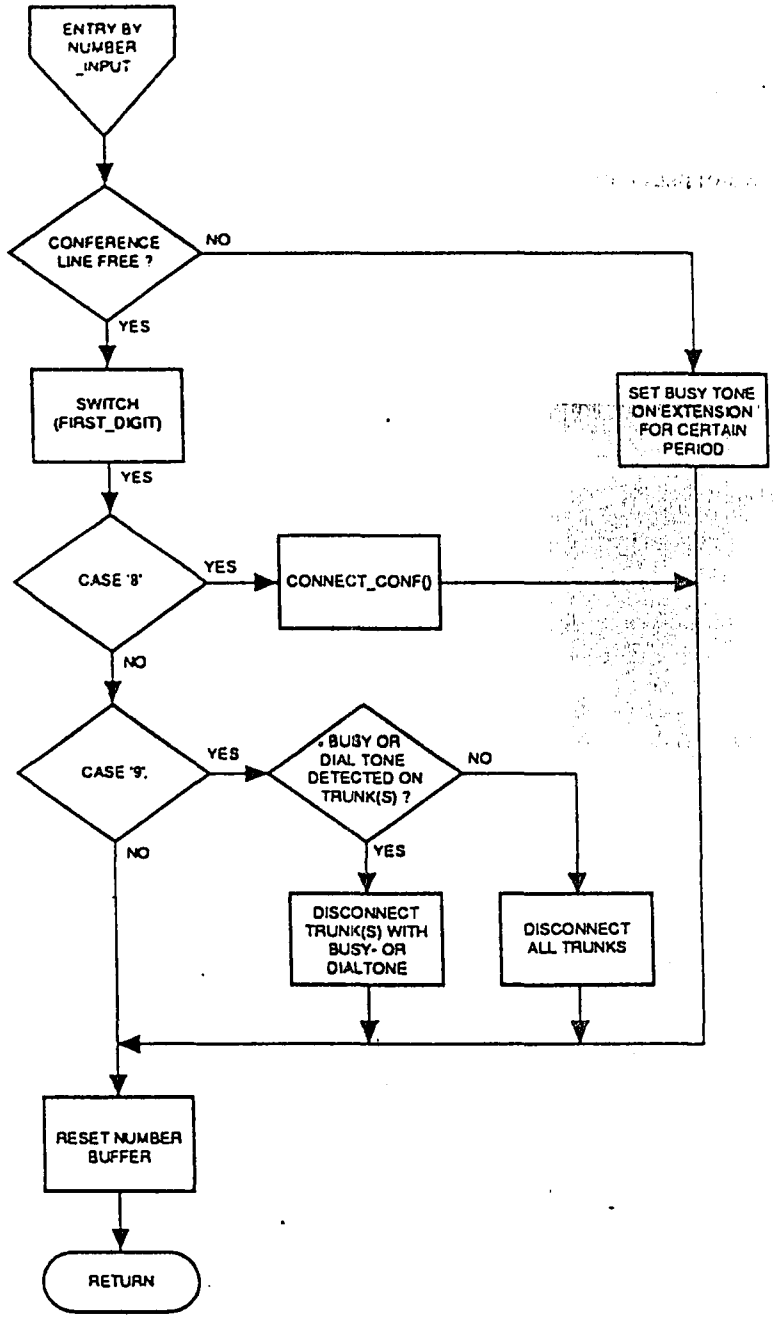
3.5.1 Reset()

The function *reset()* is called whe an extension is on hook and was hook. The extension has to be res and variables, tones and conne

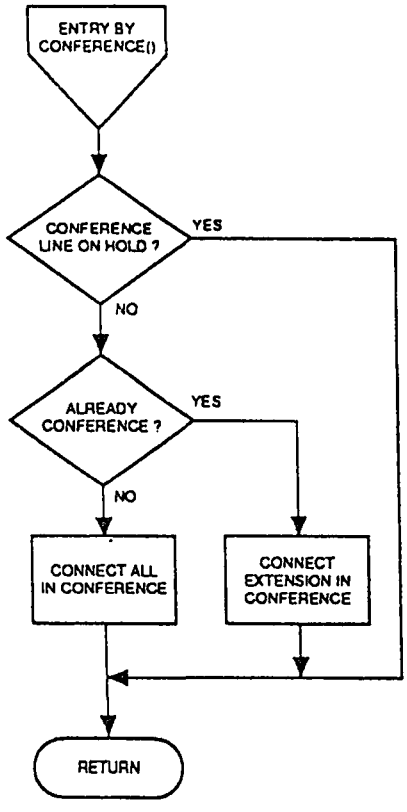
VOID CONNECT_OR_PROGRAM ()



VOID CONFERENCE ()



VOID CONNECT_CONF (VOID)



tions have to be updated (set or reset). This function also supports the updating of variables, tones and connections of the devices which it was connected to (or which it had on hold). For example when extension 0 is connected to extension 1 and extension 0 goes on hook, the *reset()* of extension 0 supports the busy tone on extension 1.

The first criterion which is checked is consultation. If there is no consultation then this means there is a single connection, remember conference call is a single connection as well, otherwise the extension has a device, or conference line, on hold. If there is no consultation, a *switch case* is activated, which switches on the variable connection[device]. Depending on which device, or conference line, the current device is connected to, action is taken. If there was consultation, the second criterion is connection. No connection is another switch case which switches on the variable hold[device]. Depending on this variable, action is taken. When the current device has a device on hold as well a device connected, there are three possibilities left. First, it is connected with the conference-line and has a device on hold. Second, it has the conference line on hold and is connected with a device. Third, it has one single connection and one device on hold. Again depending on these three possibilities action is taken. Lastly, all variables for the current device will be reset. Current device is the device which is on hook and was off hook.

3.5.2 Connect_trunk()

At the moment that an outside line rings and an extension reacts to this by going off hook or by giving a hook flash, *connect_trunk()* will be activated.

Depending on which trunk is calling, the current extension will be connected with the calling trunk. *Connect_trunk()* will reset the ring signals and warning tones on all extensions if only one trunk is calling. If both trunks are calling, only the ring signal of the current device will be reset.

3.5.3 Ext_ring_set()

The task of this function is to set an external ring signal on the operator extension, if selected. If not the external ring signal is set on all extensions onhook. Extensions which are off hook will receive a warning beep. A selected fax extension will not be involved. If a call from an outside line is not answered and the ring signal goes off the line, the ring signal on the extensions has to be reset.

3.5.4 Ext_ring_reset()

The function *ext_ring_reset()* tests if another trunk is calling. If so, then no ring signals need to be reset. If no other trunk is ringing and the ring signals have not been reset already, the signals are reset.

3.5.5 Connect_fax()

If a fax call is detected on either programmed trunk (fax_trunk) and the fax(=extension=fax_select) goes off hook, connect_fax() is called. This function establishes the connection between the trunk and the internal fax, it updates the involved variables and the ring signal is cleared.

3.6 UNIT INTERRUP.C

3.6.1 Function_10ms()

Function_10ms() is the interrupt service routine of timer 0. Every 10mS *function_10ms()* is called. The purpose of this function is to centralize all time critical application software actions. *Function_10ms()* uses the following functions: *filter_signals()*, *switch_DTMFr()* and *set_audio_tones()*.

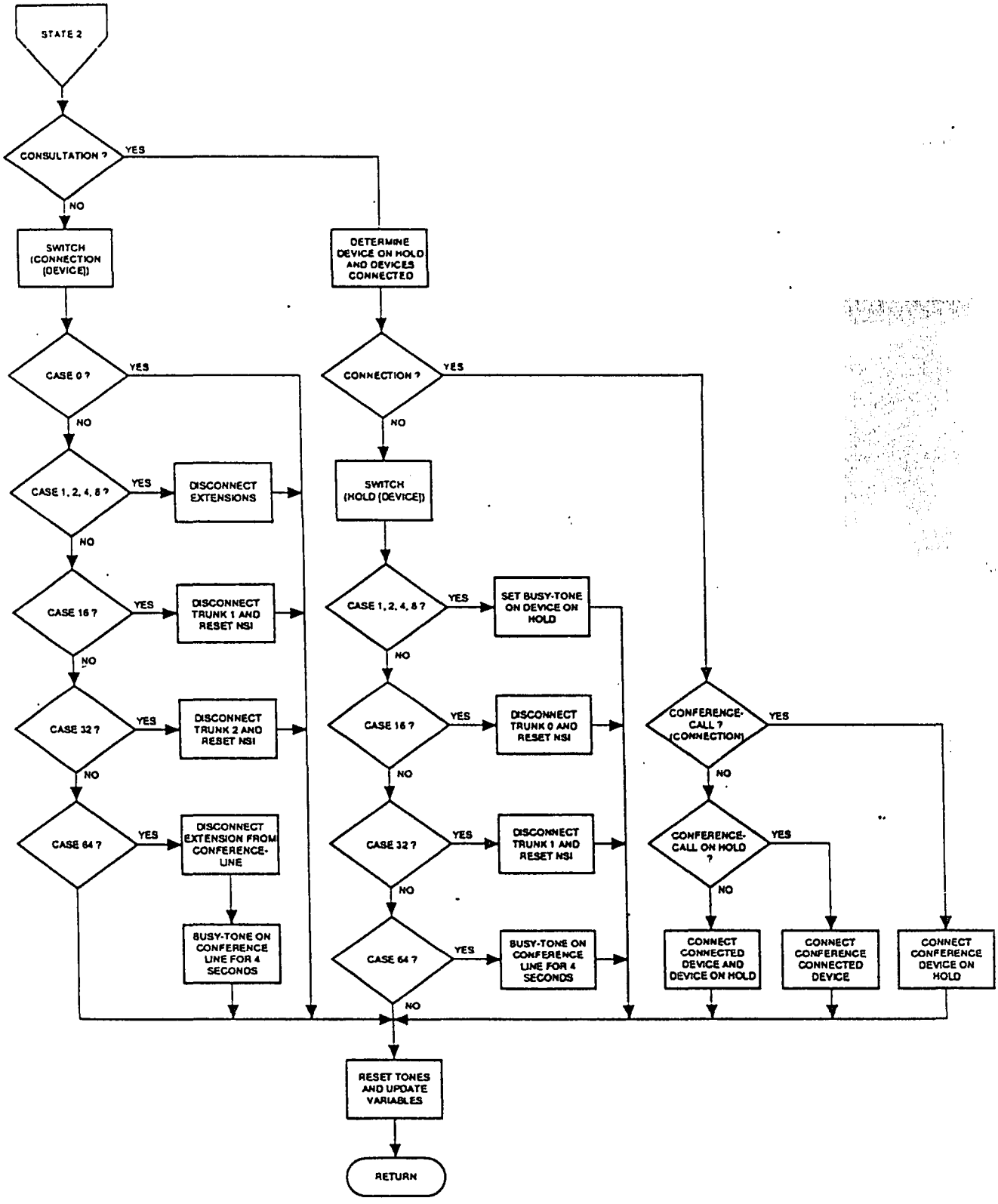
First the watchdog circuit is refreshed. Secondly the connected DTMF receivers are checked for new information and if found, the information is stored in corresponding buffers.

The third step will transmit the correct pulses or DTMF tones to one of the trunks if neccesary. When extensions and trunks are in tone mode they are connected through the matrix. The tones are transmitted directly from extension to trunk.

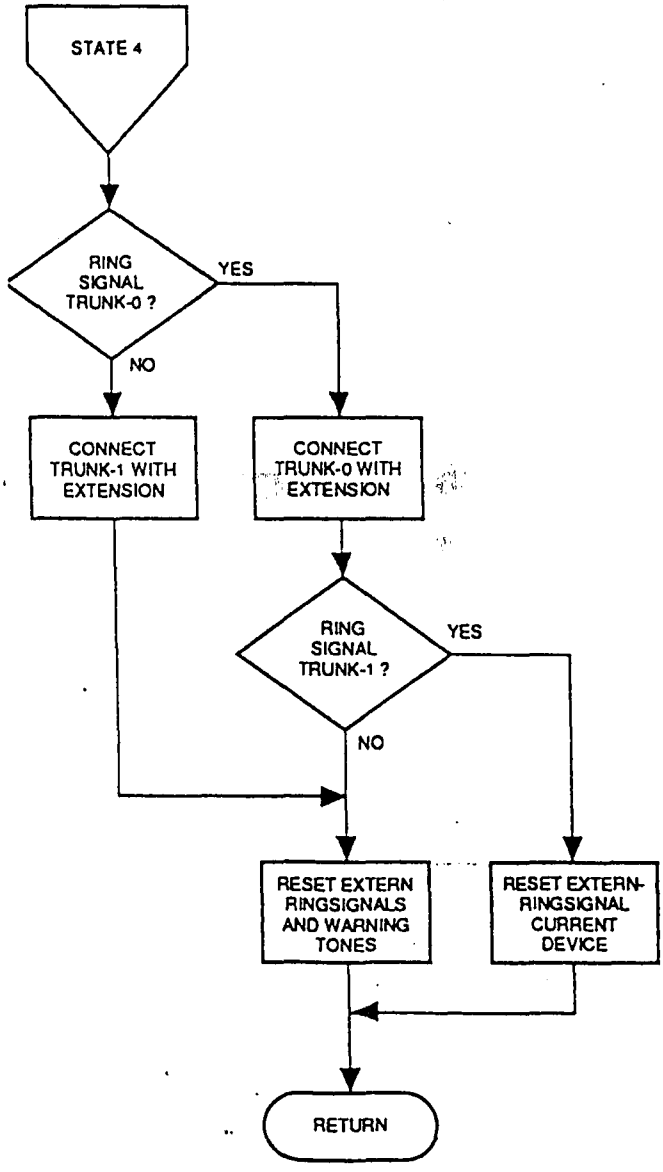
For all other mode (pulse/tone) combinations only the trunk input is connected to the extension output (seen from the matrix). By doing this we are able to 'translate' pulses to tones and the other way around. Every digit transmitted by DTMF over the trunk is detected by a DTMF receiver and stored. A busy tone is set to the extension if digi entering takes too long.

Finally, three functions are called These functions do not require a time resolution of 10ms. The function *set_audio_tones()* is called every 100ms and *switch_DTMFr()* *filter_signals()* is called every 20ms In order to keep *function_10ms()* o more or less constant length, these functions are called in shifted phase.

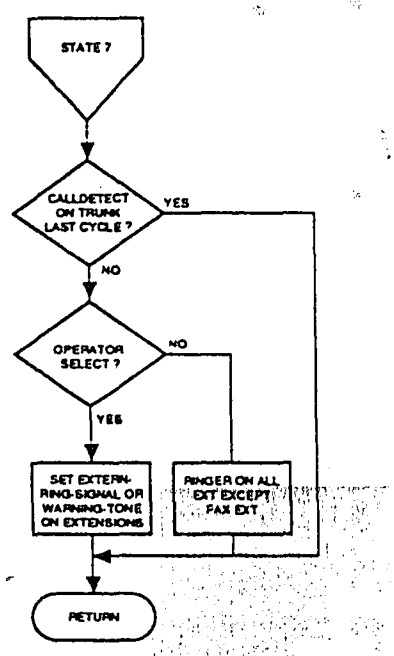
VOID RESET ()


STATE 2
CONSULTATION ?
YES
NO
SWITCH (CONNECTION (DEVICE))
DETERMINE DEVICE ON HOLD AND DEVICES CONNECTED
CASE 0 ?
YES
NO
CASE 1, 2, 4, 8 ?
YES
DISCONNECT EXTENSIONS
NO
CASE 16 ?
YES
DISCONNECT TRUNK 1 AND RESET NSI
NO
CASE 32 ?
YES
DISCONNECT TRUNK 2 AND RESET NSI
NO
CASE 64 ?
YES
DISCONNECT EXTENSION FROM CONFERENCE-LINE
BUSY-TONE ON CONFERENCE LINE FOR 4 SECONDS
CONNECTION ?
YES
NO
SWITCH (HOLD (DEVICE))
CASE 1, 2, 4, 8 ?
YES
SET BUSY-TONE ON DEVICE ON HOLD
NO
CASE 16 ?
YES
DISCONNECT TRUNK 0 AND RESET NSI
NO
CASE 32 ?
YES
DISCONNECT TRUNK 1 AND RESET NSI
NO
CASE 64 ?
YES
BUSY-TONE ON CONFERENCE LINE FOR 4 SECONDS
CONFERENCE-CALL ? (CONNECTION)
YES
NO
CONFERENCE-CALL ON HOLD ?
YES
NO
CONNECT CONNECTED DEVICE AND DEVICE ON HOLD
CONNECT CONFERENCE CONNECTED DEVICE
CONNECT CONFERENCE DEVICE ON HOLD
NO
RESET TONES AND UPDATE VARIABLES
RETURN

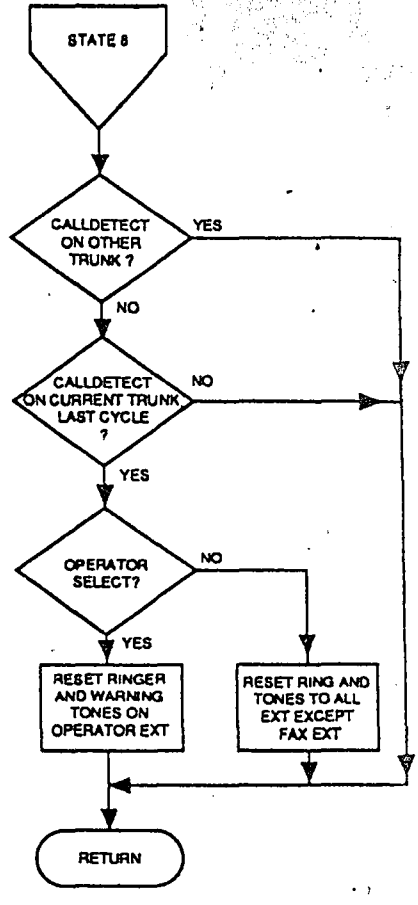
VOID CONNECT_TRUNK ()



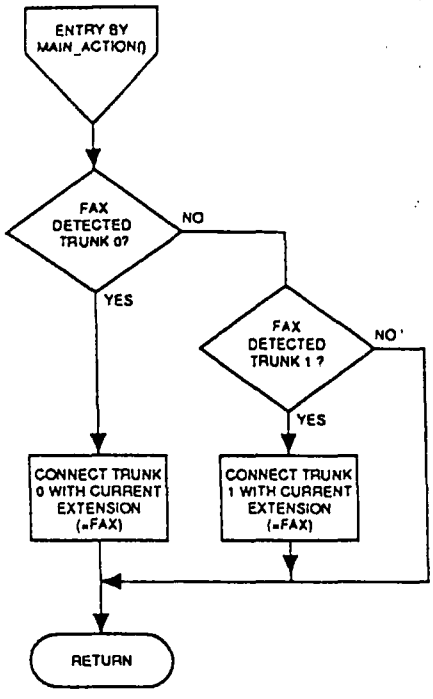
VOID EXT_RING_SET ()



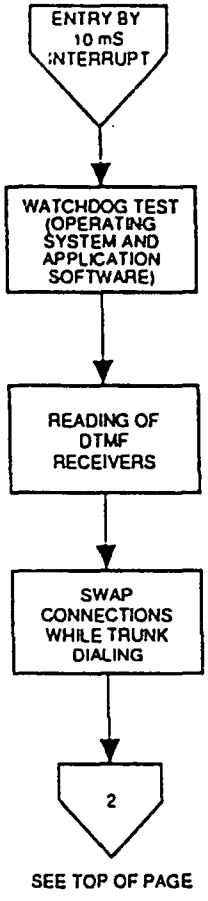
VOID EXT_RING_RESET ()

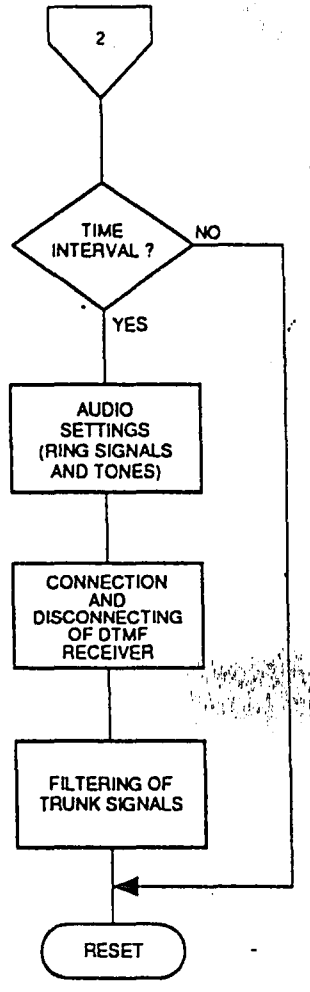


VOID CONNECT_FAX ()



VOID FUNCTION_10MS





3.6.2 Filter_signals()

Time related signals like external ring signal, dial tone and busy tone are filtered in this function. The application software needs constant values for ring signal, dial tone and busy tone instead of values that alter in the rhythm of a ring signal or busy tone.

3.6.3 Fax_detect()

If a fax is installed (fax_select=1-4), this function will be called by filter_signals(). Fax_detect() consists of 6 phases (phase 0 to phase 5). Phase 0 detects whether there is a call detect or not. If there is a call detect, phase is incremented. The preparations for the fax detect are made in phases 0 and 2. Phases 3 and 4 are the fax_detect phases. Phase 3 is the sample phase. Phase 4 determines if the calling party was a fax or a normal telephone, depending on the information gained in phase 3. Depending on the fact, fax or not, action is taken. Phase 5 provides the ring signal for 30 seconds on the device(s), unless a busy tone is detected or a device goes offhook.

3.6.4 Switch_DTMFr()

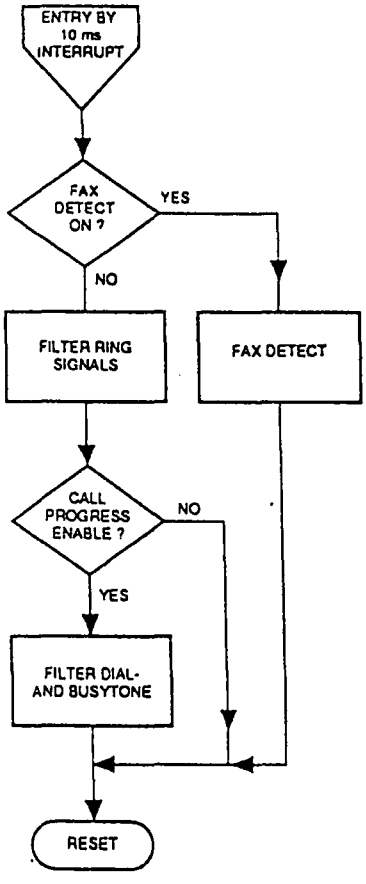
The conditions of the extensions are tested to see if the extension needs a DTMF receiver. If so, and a DTMF receiver is available, the DTMF receiver is connected to the extension. If there is no DTMF available, then the busy tone parameter is set for the extension. If the extension does not need a DTMF receiver but a DTMF receiver is connected, then this receiver is disconnected. The conditions when a DTMF receiver is required by an extension are:

1. Another extension is not trying to call the extension
2. Extension is not on hold by another extension
3. Extension is off hook
4. Extension not connected
   or: extension connection request
   or: extension on hold
   or: extension is master
   or: extension is connected to a trunk and not ready with trunk dialling.

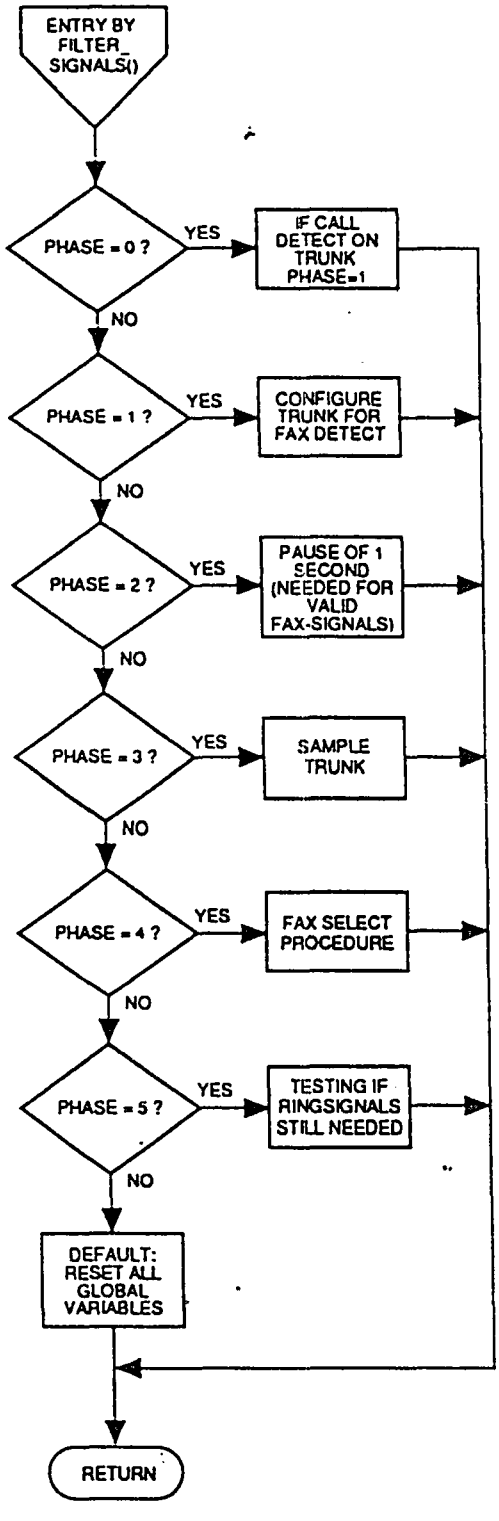
3.6.5 Set_audio_tones()

Since the application software sets only parameters for the tones there is a need for a tone connecting function. This is done in *function_10ms()*. First, actions for tones with a specific duration are taken. Next, the setting of dial tones and external ring signals are done. The external ring signal is set on the

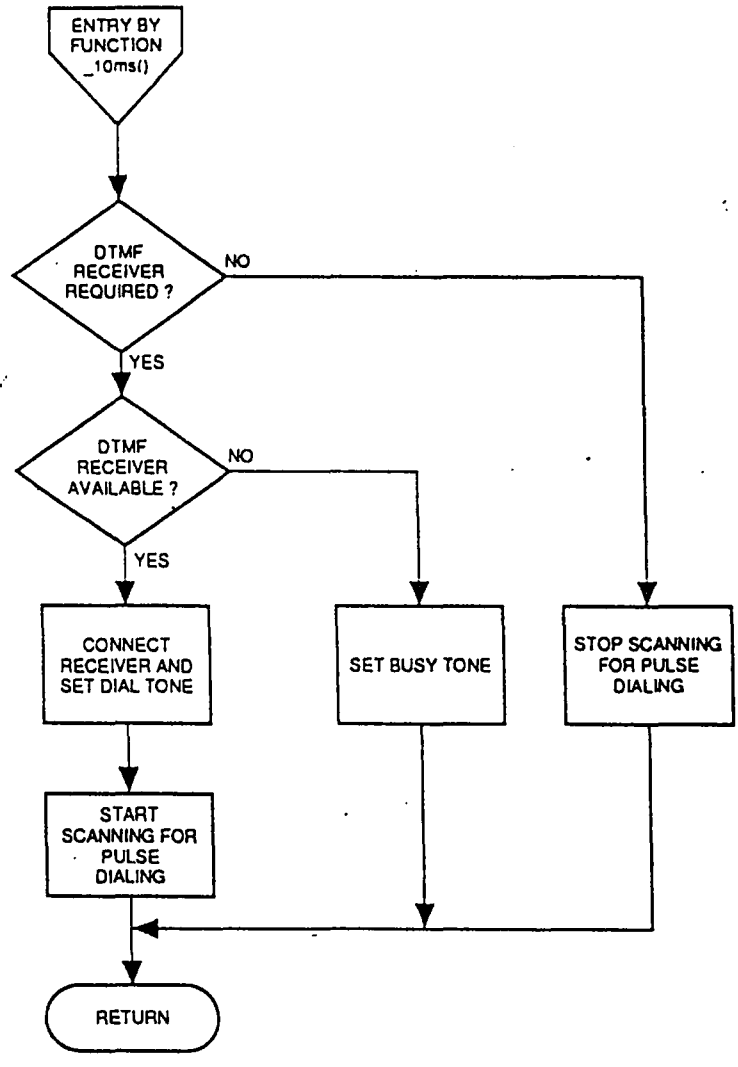
VOID FILTER_SIGNALS (VOID)



VOID FAX_DETECT ()



VOID SWITCH_DTMFr (VOID)



extensions with the same rhythm as the incoming ring signal. The last step is to determine the phase in the tone cycle (t0-t14) and if needed, tones i.e. ring signals are set or reset.

3.6.6 Switch_tone()

Audio tones are set in the 10ms-interrupt. To set and reset tones rapidly, function *switch_tone()* is used. The function swaps the tone generator from connect to disconnect or vice versa.

3.6.7 Operator mode

If the operator mode is set, all incoming calls are routed to the designated extension. If the operator is dealing with a call already and another call comes in on another trunk, the operator will hear a warning bleep to signify a call is waiting.

3.7 UNIT FUNCTION.C

Besides the functions which are described by means of flowcharts, there are some short functions, used for convenient programming, which have not been explained yet. These functions are merged in FUNCTION.C.

3.7.1 General

Column_device(): used for the conversion of the column value of a device to the device number itself. The column value is the value for a certain device on the column of the switch matrix, see SC11390 datasheet memory map. The device number is a number in the array of 0-3 for extensions and 4-5 for trunks.

Device_column(): for conversion from device numbers to column values

Device_row(): converts a device number into a row number of the switch matrix

log2(): calculates the second logarithm of a variable

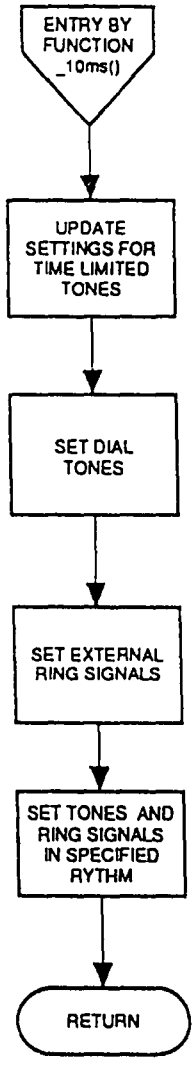
power2(): calculates the second power of a value. Standard C has a function for calculating the power of a variable but the application software needed the return-value to be limited to a single byte.

3.7.2 Reset_extern()

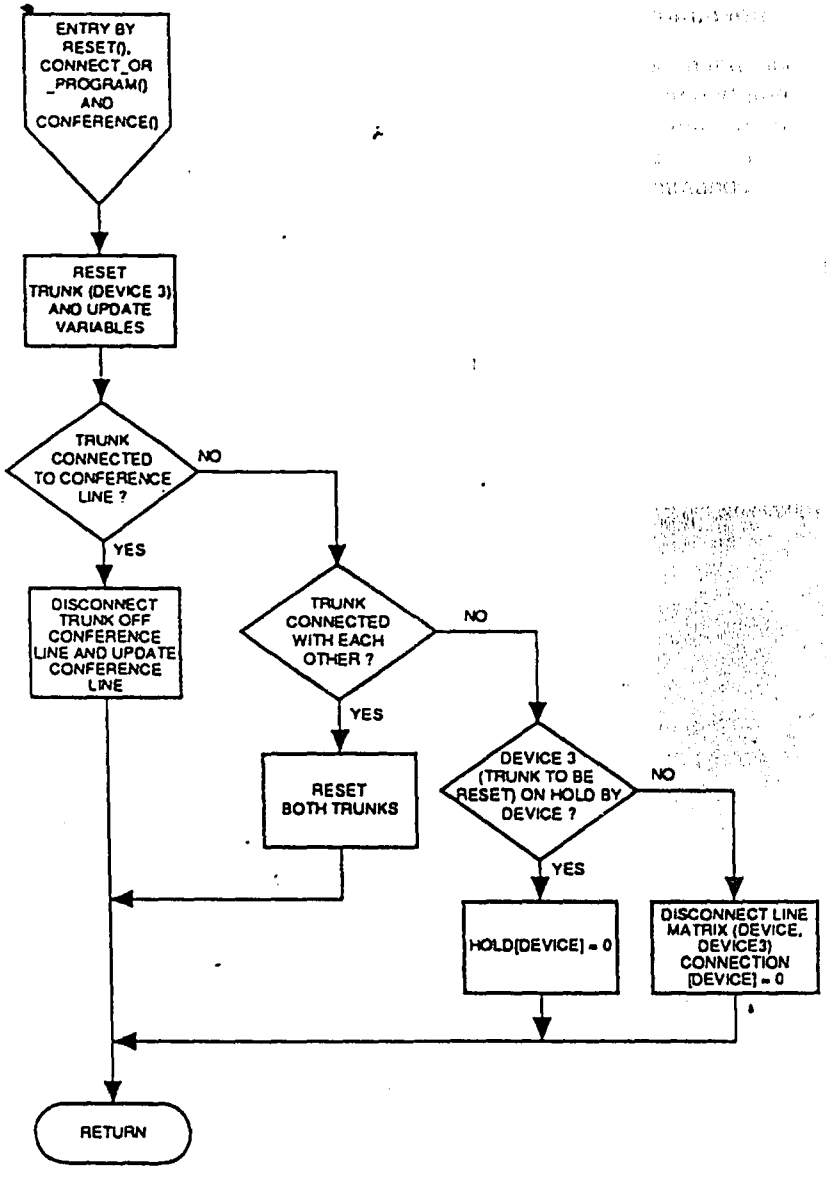
When an outside line goes on hook while it was connected to an extension or conference call or it was on hold, and this extension dials a '9' then function *reset_extern()* is called.

If the outside line is on hold by an extension, then the outside line will be reset (disconnected) and the extension which had the line on hold will hear a busy tone. If the outside line is connected to the conference line, then the conference line will hear the busy tone from the outside line for a short period and then the outside line will be disconnected. However, if the outside line was connected to this conference line but an extension had the conference line on hold, then there are two options. The first is that only one device is left on the conference line, therefore this device will be disconnected from the conference line and put on hold for the extension. The second option is that more than one device is left on the conference line. The beep will be set on the conference line after the trunk is disconnected and the conference connections remain.

VOID SET_AUDIO_TONES (WORD AUDIO_CLOCK)



VOID RESET_EXTERN (DEVICE 3)



Nothing is done if the outside line was simply connected to one extension, the outside line will be reset as soon as the extension goes onhook.

3.8 UNIT REDIAL.C

3.8.1 Handle_redial

The redial software consists of two parts, one for internal redial the other for external redial.

The internal redial will start dialling after the redial command is entered. For external redial, a 0 is transmitted towards the trunk. When the result of this action is a dial tone the actual dialling is started. The dialling uses connect_or_program to output the digits.

After dialling the number, internal redial waits until the called extension goes off hook. If so an internal ring signal is generated for the calling extension. The external redial will use the call progress detection circuitry to detect the ring back signal from the trunk. When ringing is detected from the called party we wait for the phone to be picked up, indicated by silence on the line. When silence is detected an external ring signal is set on the calling extension. The redial action is completed if the extension goes off hook.

All redial states contain timeouts, once a timeout expires the redial action is stopped and completely repeated after 30 seconds.

If during the redial session the redial extension goes off hook the redial session will end.

3.9 UNIT AT_COMM.C

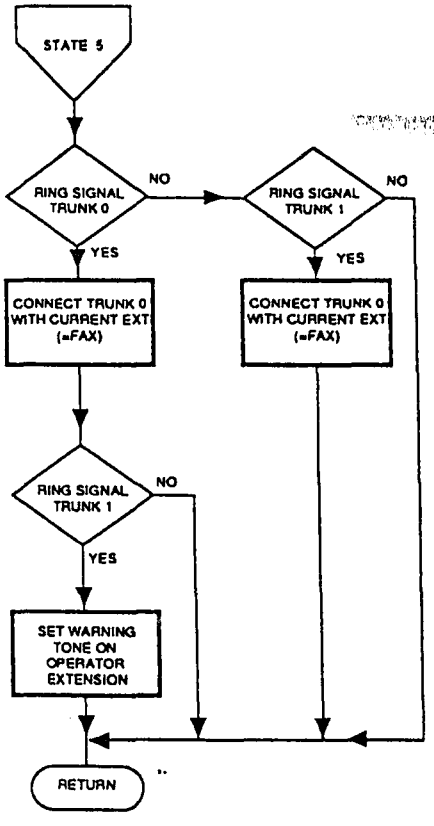
3.9.1 AT_handle_command

After the AT_get command function receives a valid command from the serial port, AT_handle_command will process the command. There are two kinds of commands.

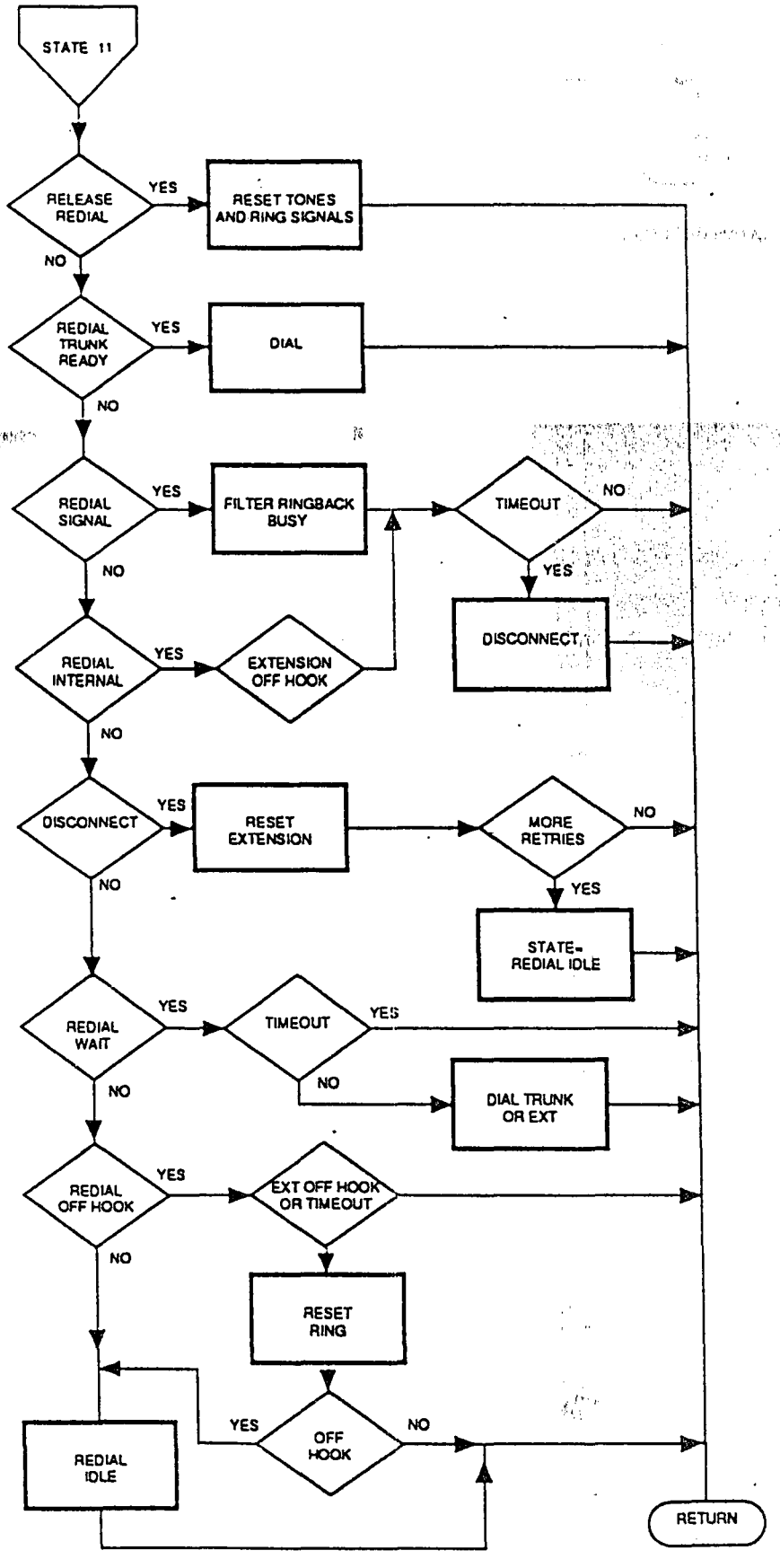
The first command group only sets or changes system parameters to meet their objectives. Second group commands are handled by inserting digits into the Rx buffer which are processed by calling connect_or_- program as at this point an extension is simulated.

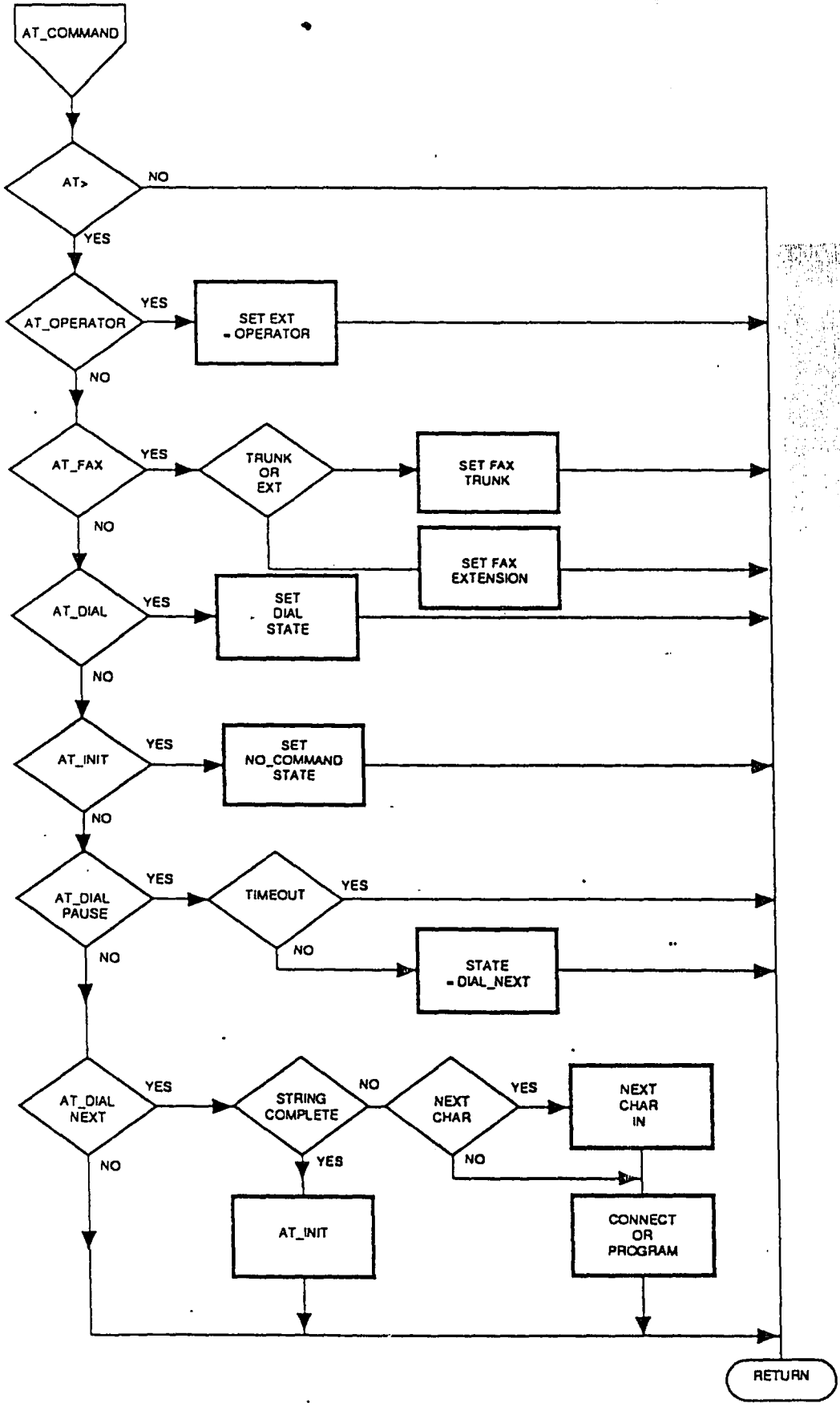
For a detailed command description see section 5 of this document.

VOID CONNECT_OPERATOR ()



REDIAL OPERATION




AT_COMMAND
AT>
NO
YES
AT_OPERATOR
YES
SET EXT = OPERATOR
NO
AT_FAX
YES
TRUNK OR EXT
SET FAX TRUNK
NO
SET FAX EXTENSION
AT_DIAL
YES
SET DIAL STATE
NO
AT_INIT
YES
SET NO_COMMAND STATE
NO
AT_DIAL PAUSE
YES
TIMEOUT
YES
NO
STATE = DIAL_NEXT
NO
AT_DIAL NEXT
YES
STRING COMPLETE
NO
NEXT CHAR
YES
NEXT CHAR IN
NO
YES
NO
AT_INIT
CONNECT OR PROGRAM
RETURN

4.1 UNIT SC11390.A51

This code unit SC11390 can be divided into the following parts:

1.1.Matrix programming
1.2.DTMF receive and transmit
1.3.Gain setting
1.4.Read gain
1.5.Others: ring generator and tone filter programming.

For every part, the functions are described in corresponding paragraphs.

4.1.1 Matrix

The matrix of the SC11390 consists of 20 rows and 23 columns and is the main part of the SC11390. The matrix is subdivided into a line matrix and a tone matrix. For both matrices, the software contains sub-modules or functions to connect or disconnect the matrix rows and columns. A function to connect or disconnect a conference with the line matrix is available as well. The external tone generator can be connected with all the extensions by the functions *connect_tone_generator* and *disconnect_tone_generator.*

At any time, it is possible to read the status of the line matrix or tone matrix with the functions *read_line_matrix_status* or *read_tone_matrix_status.* In this matrix section, functions for the registers 0 to 38 of the SC11390 can be found.

void connect_line_matrix (byte row, byte column);

Input: row (0..18),
column (0..21)

Output: none

Affected: dptr,acc,r7,r6,r5,r4

Description: A connection is made from an extension to another extension or from an extension to a trunk. The rows can vary from 0 to 18 and the columns can vary from 0 to 21. As the SC11390 uses a four wire architecture where receive and transmit are on separate lines, two switch connections have to be made. In the matrix, columns are inputs and rows are outputs so row transmit is connected with column receive. If the row equals 15, 16, 17 or 18 the line_matrix is only connected in one direction (row to column) as the selected row is for one of the tone detectors.

Example:
connect_line_matrix(0,19)

A connection is made between extension 1 and extension 2. This means row 0 (extension transmit 1) is connected to column 19 (extension receive2) and row 1 (extension transmit 2) is connected to column 20 (extension receive1).

void disconnect_line_matrix (byte row, byte column);

Input: row(0..18),
column(0..21)

Output: none

Affected: dptr,acc,r7,r5,r4

Description: The description is identical to connect_line_matrix. Both transmit and receive paths are disconnected. First, the row (output) is connected to column 0 (analog ground). After that the column value (input) is used to point to the row to which it is connected (transmit) and the row is connected to analog ground.

If the same example is used, disconnect_line_matrix (0,19), row 0 (extension transmit 1) is connected to analog ground. Column 19 (extension receive 2) is translated to row 1 (extension transmit 2). Thus row 2 is connected to column analog ground. If row equals R17 or R18 (DTMFR1 or DTMFR2) then only these rows R17 or R18 are connected to analog ground as the DTMF receivers only use a unidirectional path.

void connect_tone_matrix (byte row, byte column);

Input: row(19..22),
column(0..3)

Output: none

Affected: dptr,acc,r7,r5

Description: Connect the selected row with the selected column in one direction. If row equals 19 or 20 and column equals 3 then the DTMF receivers are available on the line matrix. To establish a connection from an extension or a trunk to a DTMF receiver use *connect_line_matrix.*

void disconnect_tone_matrix (byte row);

Input: row(19..22)

Output: none

Affected: dptr,acc,r7,r5

Description: Connect the row in use with the analog ground column (C0).

void connect_conference (byte row);

Input: row(0..18)

Output: none

Affected: dptr,pc,acc,r7,r6,r5,r4

Description: The row is connected with the conference column. Then the row is translated to a column and this new column is connected with the right conference row. This means a connection with register 32, 33 or 34.

Example:
Connect_conference(0)

Row 0 (extension transmit1) is connected with the conference column (column 22). After translation of the row into a column (the new column becomes extension receive1), this column is connected with the conference row 34.

© Copyright 1993 Sierra Semiconductor

void disconnect_conference (byte row);

Input: row(0..18)

Output: none

Description: The specified row is disconnected from the conference. If there are still lines connected in a conference, the conference remains, else the conference row R19 is connected to analog ground.

void connect_tone_generator (byte row);

Input: row (0..19)

Output: none

Affected: dptr,acc,r7,r6,r5,r4

Description: The row is connected with the tone generator. Depending on the row, a connection is made to the right row of the tone generator 36, 37 or 38. Row 0 to row 18 correspond with the rows of the line matrix. If row equals 19 (conference column), the tone generator can be connected with the conference column.

void disconnect_tone_generator (byte row);

Input: row (0..19)

Output: none

Affected: dptr,acc,r7,r6,r5,r4

Description: Disconnect the selected row from the tone generator row 36, 37 or 38. A disconnection means a connection with the analog ground column C0.

byte read_line_matrix_status (byte row);

Input: row(0..18)

Output: column(0..21)

Description: Read status from the line matrix. The function returns the column connected to the selected row. A return value of 0 means that the row is connected with analog ground and therefore not used.

byte read_tone_matrix_status (byte);

Input: row(19..22)

Output: column(0..3)

Description: Read status from the tone matrix. This means, from the requested row, the routine responds with which column is connected to this row. If the column equals 0, this means the row is connected with analog ground and is not used at this time.

4.1.2 DTMF Transmit and Receive

This section can be used to transmit or receive DTMF tones. Both DTMF transmitters translate a given digit into DTMF tones. The digit must be written into the DTMF transmit register 68 or 69. This is possible with the functions *set_DTMFt1* and *set_DTMFt2*. To stop DTMF transmit, the functions *clr_DTMFt1* and *clr_DTMFt2* can be used.

Both DTMF receivers translate a DTMF tone into a digit. The result of this translation is placed into the DTMF receive registers 64 or 65. Valid digits are placed in these registers after the STD signal has gone high.

byte DTMFr1_status (void);

Input: none

Output: byte; d3.d0 = digit

Digit	d3.d0	Digit	d3.d0
1	0001	9	1001
2	0010	0	1010
3	0011	*	1011
4	0100	#	1100
5	0101	A	1101
6	0110	B	1110
7	0111	C	1111
8	1000	D	0000

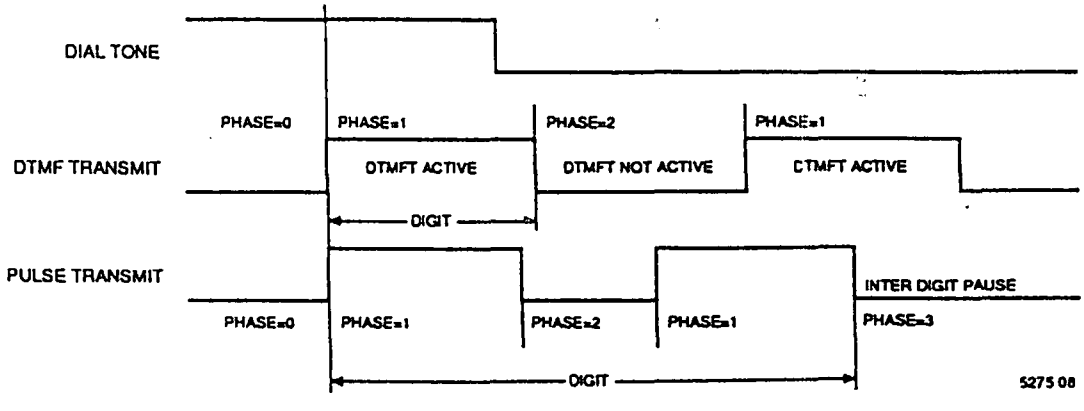
d4 = EST; Early steering: bit is set when the DTMF receiver has recognized the presence of a valid tone pair.

d5 = STD; Delayed steering: If EST has been high long enough, STD is set.

Affected: dptr,acc,r7

Description: After the STD flag goes high, DTMF receiver1 will latch its digit into register 64. With this function register 64 is read and the digit is returned with its EST and STD flag.

TIME DIAGRAM DIGIT TRANSMITTING



byte DTMFr2_status (void);

Input: none

Output: byte: d3..d0 = digit (see DTMFr1_status)

d4 = EST; Early steering: bit is set when the DTMF receiver has recognized the presence of a valid tone pair.

d5 = STD; Delayed steering: If EST has been high long enough, STD is set.

Affected: dptr,acc,r7

Description: With this function register 65 is read and the digit is returned with its EST and STD flag (see DTMFr1_status).

void set_DTMFt1 (byte digit);

Input: byte:d3..d0

Digit	d3..d0	Digit	d3..d0
1	0001	9	1001
2	0010	0	1010
3	0011	*	1011
4	0100	#	1100
5	0101	A	1101
6	0110	B	1110
7	0111	C	1111
8	1000	D	0000

d4 = RTONE
d5 = CTONE

Output: none

Affected: dptr,acc,r7,r5

Description: Set DTMF transmitter1 on with the selected digit. The digit is written into DTMF transmit register 68. The DTMF transmitter converts this digit into the corresponding frequencies with RTONE and CTONE high.

void set_DTMFt2 (byte digit);

Input: byte:d3..d0 = digit (see set_DTMFt2)

d4 = RTONE
d5 = CTONE

Output: none

Affected: dptr,acc,r7,r5

Description: Set DTMF transmitter2 on with the selected digit. The digit is written into DTMF transmit register 69. The DTMF transmitter converts this digit into the corresponding frequencies with RTONE and CTONE high.

void clr_DTMFt1 (void);

Input: none

Output: none

Affected: dptr,acc,r7,r5

Description: DTMF transmitter1 is turned off by means of RTONE=0 and CTONE=0.

void clr_DTMFt2 (void);

Input: none

Output: none

Affected: dptr,acc,r7,r5

Description: DTMF transmitter2 is turned off by means of RTONE=0 and CTONE=0.

byte envelope_status (void);

Input: none

Output: byte: d0 = tone
d1 = VOICE
d2 = EST DTMFR1
d3 = EST DTMFR2
d4 = STD DTMFR1
d5 = STD DTMFR2
d6 = call progress1
d7 = call progress2

Affected: dptr,acc,r7

Description: Envelope register 70 is read in this function. This register reflects the status of the tone detector, voice detector, both DTMF receivers and both call progress detectors. The bits call progress1 and call progress2 differ from the call progress pins of the SC11390. If a valid tone is detected (a valid tone means a frequency between 300 and 550 Hz), the bits call progress1 and call progress2 are set.

4.1.3 Gain

Functions for the PGA registers 80 to 84 and 86 to 93 can be found in this part. With these functions, all PGAs present at trunk transmit, trunk receive, extension transmit, tone detector, voice detector, DTMF receive1 and 2 and call progress1 and 2 can be programmed separately. On start up of the evaluation board, all PGAs are default programmed with gain 0. Excepted are the PGAs at call progress. Default setting is 15.

void trunk1_transmit (byte gain);

Input: gain(0..15)

Setting	Gain (dB)
0	0
1	-0.3
2	-0.6
3	-0.9
4	-1.2
5	-1.5
6	-1.8
7	-2.1
8	-2.4
9	-2.7
10	-3.0
11	-3.3
12	-3.6
13	-3.9
14	-4.2
15	-4.5

Output: none

Affected: dptr,acc,r7,r5

Description: The gain of the PGA for trunk1 transmit is written into register 80. With settings 0 to 15, every gain between 0 and -4.5 dB can be chosen in steps of 0.3 dB. Default is -4.5 dB.

void trunk2_transmit (byte gain);

Input: gain(0..15)
(see trunk1_transmit)

Output: none

Affected: dptr,acc,r7,r5

Description: The gain of the PGA for trunk2 transmit is written in register 81. With settings 0 to 15, every gain between 0 and -4.5 dB can be chosen in steps of 0.3 dB. Default is -4.5 dB.

void trunk1_receive (byte gain);

Input: gain(0..15)

Setting	Gain (dB)
0	2
1	2.5
2	3
3	3.5
4	4
5	4.5
6	5
7	5.5
8	6
9	6.5
10	7
11	7.5
12	8
13	8.5
14	9
15	9.5

Output: none

Affected: dptr,acc,r7,r5

Description: The gain of the PGA for trunk1 receive is written into register 82. With settings 0 to 15, every gain between 2 and 9.5 dB can be chosen. Default is 2 dB.

void trunk2_receive (byte gain);

Input: gain(0..15)
(see trunk1_receive)

Output: none

Affected: dptr,acc,r7,r5

Description: The gain of the PGA for trunk2 receive is written into register 83. With settings 0 to 15, every gain between 2 and 9.5 dB can be chosen. Default is 2 dB.

void gain_extension_transmit (byte gain);

Input: gain(0..7)

Selection	Gain (dB)
0	-5
1	-5.5
2	-6
3	-6.5
4	-7
5	-7.5
6	-8
7	-9

Output: none

Affected: dptr,acc,r7,r5

Description: The gain of the PGA for extension transmit is written into register 84. When setting is 0 to 7, every gain between -5 and -9 dB can be chosen. Default is -5 dB.

void detection_level_tone_detector (byte gain);

Input: gain(0..15)

Selection	Min. Detect level (dBm)
0	-22
1	-24
2	-26
3	-28
4	-30
5	-32
6	-34
7	-36
8	-38
9	-40
10	-42
11	-44
12	-46
13	-48
14	-50
15	-52

Output: none

Affected: dptr,acc,r7,r5

Description: The setting of the PGA at the tone detector is written into register 86. Selections between 0 and 15 set the minimum accept level. Default is -22 dBm.

void detection_level_voice_detector (byte gain);

Input: gain(0..15) (see detection_level_tone_detector)

Output: none

Affected: dptr,acc,r7,r5

Description: The setting of the PGA for the voice detector is written into register 87. Selections between 0 and 15 set the minimum accept level. Default is -22 dBm.

void detection_level_DTMFr1 (byte gain);

Input: gain(0..7)

Selection	Min. Detect Level (dBm)
0	-30
1	-33
2	-36
3	-39
4	-42
5	-45
6	-48
7	-51

Output: none

Affected: dptr,acc,r7,r5

Description: The setting of the PGA for DTMF receiver1 is written into register 90. Selections between 0 and7 set the minimum accept level. Default is -30 dBm.

void detection_level_DTMFr2 (byte gain);

Input: gain(0..7) (see detection_level_DTMFr1)

Output: none

Affected: dptr,acc,r7,r5

Description: The setting of the PGA for DTMF receiver2 is written into register 91. Selections between 0 and7 set the minimum accept level. Default is -30 dBm.

void detection_level_call_progress1 (byte gain);

Input: gain(0..15)dBm

Selection	Min. Detect Level (dBm)
0	-18
1	-20
2	-22
3	-24
4	-26
5	-28
6	-30
7	-32
8	-34
9	-36
10	-38
11	-40
12	-40
13	-42
14	-44
15	-46

Output: none

Affected: dptr,acc,r7,r5

Description: The setting of the PGA for call progress detector 1 is written into register 92. Selections between 0 and 15 set the minimum accept level. Default is -48 dBm.

void detection_level_call_progress2 (byte gain);

Input: gain(0..15) (see detection_level_call_progress1)

Output: none

Affected: dptr,acc,r7,r5

Description: The setting of the PGA for call progress detector 2 is written into register 93. Selections between 0 and 15 set the minimum accept level. Default is -48 dBm.

4.1.4 Read Gain Status

Functions in this section are used to read the programmed gain of the PGAs.

byte trunk1_transmit_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the gain of the PGA for trunk transmit 1 in register 80. The value of 0-15 corresponds with the same gain as shown in function trunk1_transmit.

byte trunk2_transmit_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the gain of the PGA for trunk transmit 2 in register 81. The value of 0 to 15 corresponds with the same gain as shown in function trunk2_transmit.

byte trunk1_receive_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the gain of the PGA for trunk receive 1 in register 82. The value of 0 to 15 corresponds with the same gain as shown in function trunk1_receive.

byte trunk2_receive_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the gain of the PGA for trunk receive 2 in register 83. The value of 0 to 15 corresponds with the same gain as shown in function trunk2_receive.

byte gain_extension_transmit_status (void);

Input: none

Output: byte(0..7)

Affected: dptr,acc,r7

Description: Read gain of the PGA for extension transmit in register 84. The value of 0 to 7 corresponds with the same gain as shown in extension_transmit.

byte detection_level_tone_detector_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the setting of the PGA for the tone detector in register 86. The value corresponds with the setting given in detection_level_tone_detector.

byte detection_level_voice_detector_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the setting of the PGA for the voice detector at register 87. The value corresponds with the setting given in detection_level_voice_detector.

byte detection_level_DTMFr1_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the setting of the PGA for DTMFr1 in register 90. The value corresponds with the setting given in detection_level_DTMFr1.

byte detection_level_DTMFr2_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

*Description:*Read the setting of the PGA for DTMFr2 in register 91. The value corresponds with the gain given in detection_level_DTMFr2.

byte detection_level_call_progress1_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the setting of the PGA for call progress detector1 in register 92. The value corresponds with the setting given in detection_level_call_progress1.

byte detection_level_call_progress2_status (void);

Input: none

Output: byte(0..15)

Affected: dptr,acc,r7

Description: Read the setting of the PGA for call progress detector2 in register 93. The value corresponds with the setting given in detection_level_call_progress2.

4.1.5 Others

In this section, functions for the tone filter (register 85) and the ring generator (register 94) can be found. The tone filter can be programmed with the function freq_tone_detector according to table 5 of the SC11390 datasheet. With the function freq_tone_status, the programmed tone can be read. The ring generator can be programmed for ringer frequencies of 16.6Hz, 25Hz or 50Hz. The default frequency used on the evaluation board is 25 Hz. This chapter also contains a function set_defaults_11390 to set all PGAs to a default value on startup of the evaluation board.

void freq_tone_detector (byte tone);

Input: tone(6..250)
see table 5 datasheet SC11390

Output: none

Affected: dptr,acc,r7,r5

Description: Sets the divide factor used for the programmable band pass filter in register 85.

byte freq_tone_status (void);

Input: none

Output: tone(6..250)

Affected: dptr,acc,r7

Description: Read divide factor of programmable band pass filter of register 85. The divide factor of this filter corresponds with the divide factor given in freq_tone_detector.

void set_ring_generator (byte freq);

Input: freq. (0..7)

0	ring generator off
1	16.6 Hz sine wave
2	25 Hz sine wave
3	50 Hz sine wave
4	ring generator off
5	16.6 Hz square wave
6	25 Hz square wave
7	50 Hz square wave

Output: none

Affected: dptr,acc,r7,r5

Description: The ring generator is programmed with the selected frequency. Every value between 0 and 7 can be chosen. Default is a 25 Hz ring signal.

Note: With this function it is also possible to turn the ring generator off by setting the frequency to 0 or 4. However, the ring generator can also be turned off with the function clr_ring_generator.

void clr_ring_generator (void);

Input: none

Output: none

Affected: dptr,acc,r7,r5

Description: The ring generator is turned off. This is done by programming the ring generator with frequency 0.

void set_defaults_11390 (void);

Input: none

Output: none

Affected: dptr,acc,r7,r6,r5

Description: On start up the SC11390 will be configured with standard values.

The following levels are set:

Trunk1 transmit	PGA-tt1:	0 dB
Trunk2 transmit	PGA-tt2:	0 dB
Trunk1 receive	PGA-tr1:	2 dB
Trunk2 receive	PGA-tr2:	2 dB
Extension transmit	PGA-ext.:	-5 dB
Tone detection	PGA-tone	-22 dBm
Voice detection	PGA-voice	-22 dBm
DTMFr1	PGA-DTMFr1	-30 dBm
DTMFr2	PGA-DTMFr2	-30 dBm
Call progress1	PGA-cp1	-48 dBm
Call progress2	PGA-cp2	-48 dBm

The ring generator is turned off.

4.2 Unit PIO.A51

The PIO used on the evaluation board is the 82C55. On startup of the evaluation board, this PIO is initialised with the function init_pio. With this routine, the external tone generator on the evaluation board is initialised on startup. Also in this routine, functions are found to set a ring signal on an extension and to clear this ring signal and to connect or disconnect a trunk from the evaluation board, the function set_nsi and clr_nsi are provided.

void init_pio (void);

Input: none

Output: none

Affected: dptr,acc

Description: The PIO ports are initialised with port A and port C (high byte) input, port B and port C (low byte) output. The ports are configured as follows:

Port A:
pa.0 = offhook extension1
pa.1 = offhook extension2
pa.2 = offhook extension3
pa.3 = offhook extension4
pa.4 = call detection trunk1
pa.5 = call detection trunk2
pa.6 = not used
pa.7 = not used.

Port B:
pb.0 = ring extension1
pb.1 = ring extension2
pb.2 = ring extension3
pb.3 = ring extension4
pb.4 = not used
pb.5 = nsi trunk1
pb.6 = not used
pb.7 = nsi trunk2

Port C:
low byte output:
pc.0 = SC11314 din
pc.1 = SC11314 wrclock
pc.2 = SC11314 wr
pc.3 = not used

high byte Input:
pa.4 = tone detect
pa.5 = call progress detection 1
pa.6 = call progress detection 2
pa.7 = not used

void Init_11314 (void);
(Optional programmable tone generator SC11314)

Input: none

Output: none

Affected: dptr,acc,r7,r6,r4

Description: Initializes the frequency, amplitude and gain of the SC11314, a programmable sine/squarewave generator. Default frequency is set to 425 Hz. The SC11314 is not used on evaluation board version 2.0.

void set_ring (byte extension);

Input: extension

0 = extension1
1 = extension2
2 = extension3
3 = extension4

Output: none

Affected: dptr,acc,b,r7,r6,r1,r0

Description: Ring signal is set on the selected extension. The ring generator is also turned on.

Attention: first check for offhook status; if the handset is onhook this function can be used.

void clr_ring (byte extension);

Input: extension

0 = extension1
1 = extension2
2 = extension3
3 = extension4

Output: none

Affected: dptr,acc,b,r7,r1

Description: This function clears the ring signal at the selected extension. If a ring signal exists on another extension than the selected extension, that ring signal stays and the ring generator will also stay on.

void set_nsi (byte trunk);

Input: trunk

0 = trunk1
1 = trunk2

Output: none

Affected: dptr,acc,r7,r6

Description: This function can be used to close the relay used as the hook switch (nsi = Nummer Schalter für Impulse) at the selected trunk. This function can also be used to pulse dialling (pulse_break or interdigit pause) to a trunk.

void clr_nsi (byte trunk);

Input: trunk

0 = trunk1
1 = trunk2

Output: none

Affected: dptr,acc,r7,r6

Description: This function can be used to open the nsi relay contact of the selected trunk. This function can also be used to transmit pulses to a trunk (pulse_make).

4.3 UNIT EEPROM.A51

This unit is written for an XL93C66 serial EEPROM. In this routine functions can be found to read or to write the EEPROM. The first 10 addresses of the EEPROM are used by the real time interrupt.

void ee_wds (void);

Input: none

Output: none

Affected: psw,acc,b

Description: Lock EEPROM write.

void ee_wen (void);

Input: none

Output: none

Affected: psw,acc,b

Description: Unlock EEPROM write. Before the EEPROM can be written, an ee_wen() must be given.

void ee_eral (void);

Input: none

Output: none

Affected: psw,acc,b

Description: Erase whole EEPROM.

void ee_wral (word pattern);

Input: word

Output: none

Affected: psw,acc,b,r5,r4

Description: Fill EEPROM with pattern.

word ee_read (byte address);

Input: address(0..255)

Output: word

Affected: psw,acc,b,r7,r6

Description: Read EEPROM at

selected address and return data standing at that address.

bit ee_write (byte address, word data);

Input: address(0..255)
word(0..65556)

Output: bit

Affected: psw,acc,b,r7,r5,r4,carry

Description: Write word at selected address. If bit is set, then write cycle was not completed.

void ee_erase (byte address);

Input: address(0..255)

Output: none

Affected: psw,acc,b,r7

Description: Erase word at selected address.

4.4 UNIT RS232.A51

For various purposes, an RS232 port is included on the evaluation board. On startup of the board, the RS232 port is configured with the function ser_init. This unit uses timer 1 of the controller 80C32. In this routine, functions can be found to read or write the serial port.

void ser_init (void);

Input: none

Output: none

Affected: none

Description: Configuration of RS232 with baudrate 9600 and no parity used. The dataformat is 8 data bits and 1 stop bit.

byte ser_rd (void);

Input: none

Output: byte

Affected: dptr,acc,r7

Description: Get serial data from buffer.

void ser_wr (byte txdata);

Input: byte

Output: none

Affected: dptr,acc,r7

Description: Put serial data in buffer.

void ser_wr_text (word address);

Input: word

Output: none

Affected: dptr,acc,r7,r6

Description: Put string terminated by null.

void ser_wr_hex (byte txdata);

Input: byte

Output: none

Affected: dptr,acc,r7,r0

Description: Convert byte into 2 bytes ASCII Hex.

void ser_wr_hexword (word txdata);

Input: byte

Output: none

Affected: dptr,acc,r7,r0

Description: Convert word into 2 bytes ASCII Hex.

void ser_wr_bcd (byte txdata);

Input: byte

Output: none

Affected: dptr,acc,r7,r0

Description: Convert BCD into 2 bytes ASCII.

byte ser_wait_key (void);

Input: none

Output: byte

Affected: dptr,acc,r7

Description: Wait until data is received.

void ser_flush (void);

Input: none

Output: none

Affected: none

Description: Flush serial buffers and reset flags.

4.5 UNIT RTI.A51

To realize a connection between the operating system and the application software, a real time interrupt is used. This interrupt is generated every 1 millisecond and uses timer 2 of the controller 80C32. The features of this real time interrupt are:

1. General: read status tone detector;
2. Extensions: read offhook, pulse receive;
3. Trunks: read call detection and envelope status; transmit a telephone number.

The offhook status of every extension is reported by the interrupt routine. With this status the pulses, appearing at an offhook line, are counted. The result of this count is written into a buffer. Every extension has an rxd_buffer with a length of 22 bytes. If a pulse telephone is connected to the evaluation board, the telephone number is written into this buffer. If a DTMF telephone is connected to the evaluation board, the application can detect the telephone number with the functions DTMFr1_status or DTMFr2_status from unit SC11390.

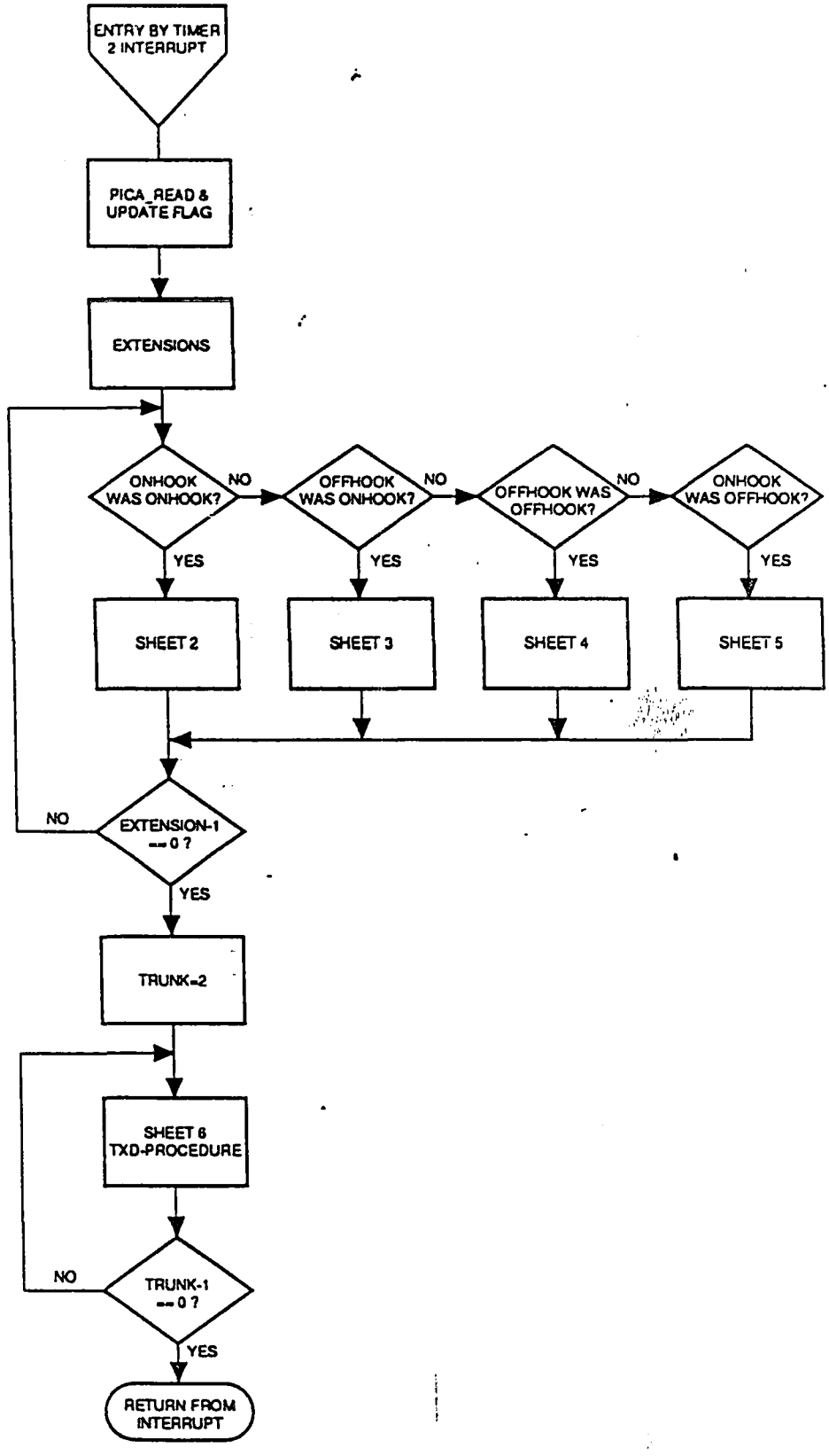
Call detection is reported for every trunk. With the envelope detector of the SC11390, the call progress detect bits are read. Every trunk has a txd_buffer with a length of 22 bytes available. The application software can fill these buffers with digits which will be sent to the trunk by the real time interrupt. The application has to enable this process.

The minimum execution time needed for the real time interrupt is about

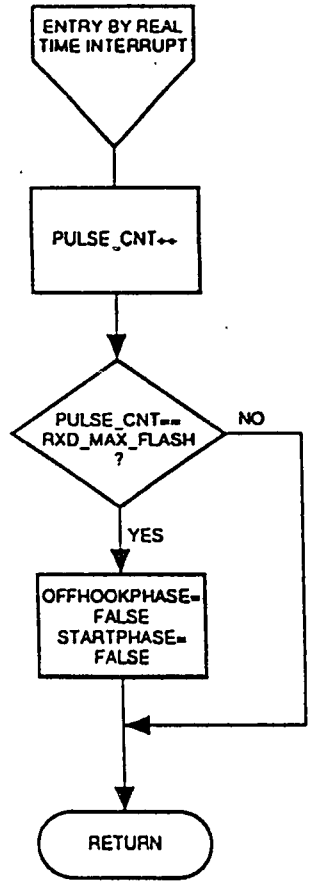
240 µseconds. If transmission of digits to a trunk is enabled, execution time goes up to 280 µseconds.

The memory map of these variables can be found in Appendix A. The variables used by the interrupt are described in Appendix C.

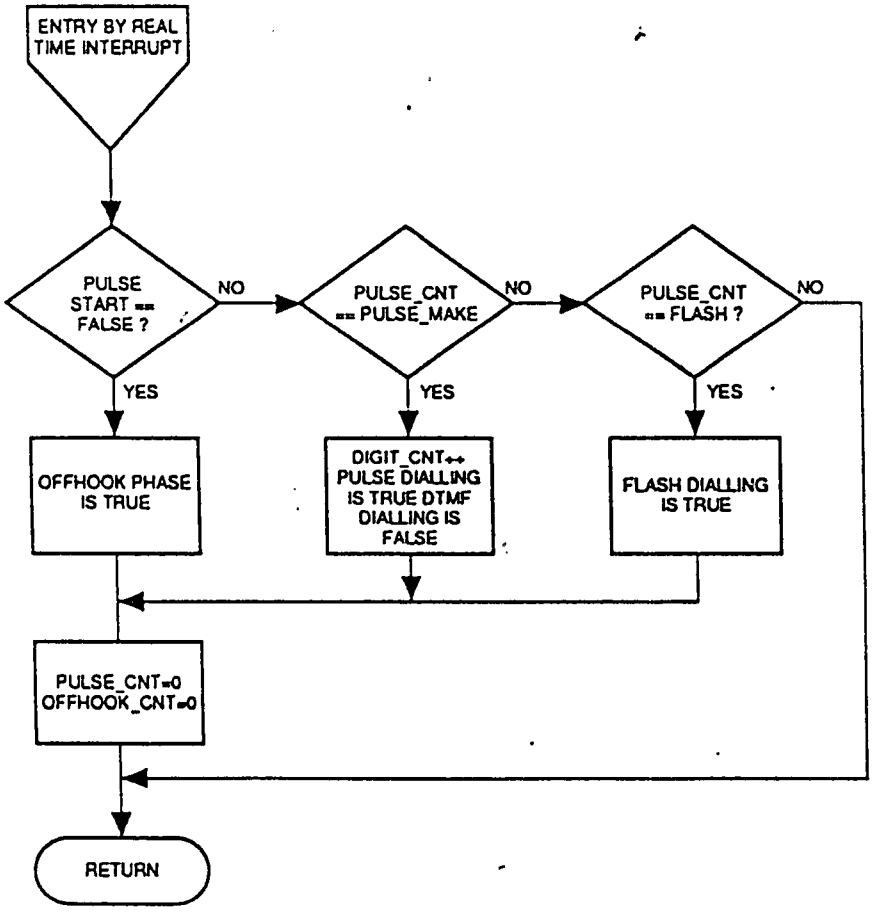
REAL-TIME INTERRUPT (1ms-interrupt)



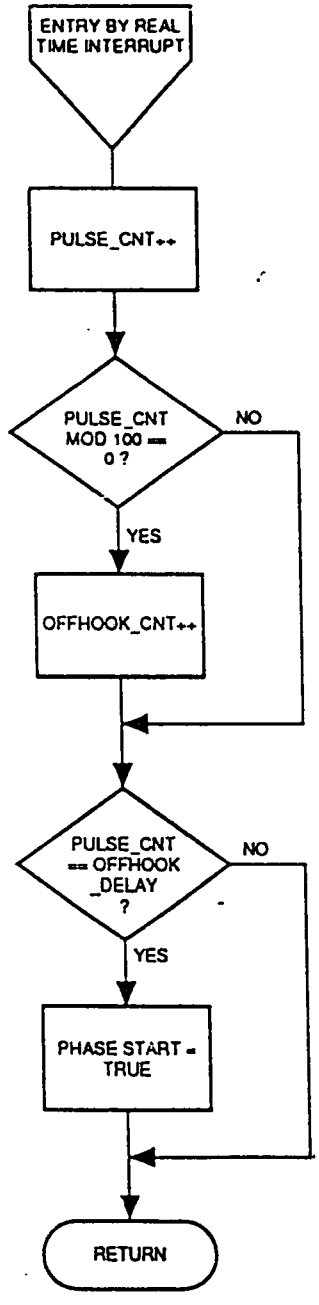
ONHOOK WAS ONHOOK (SHEET 2)



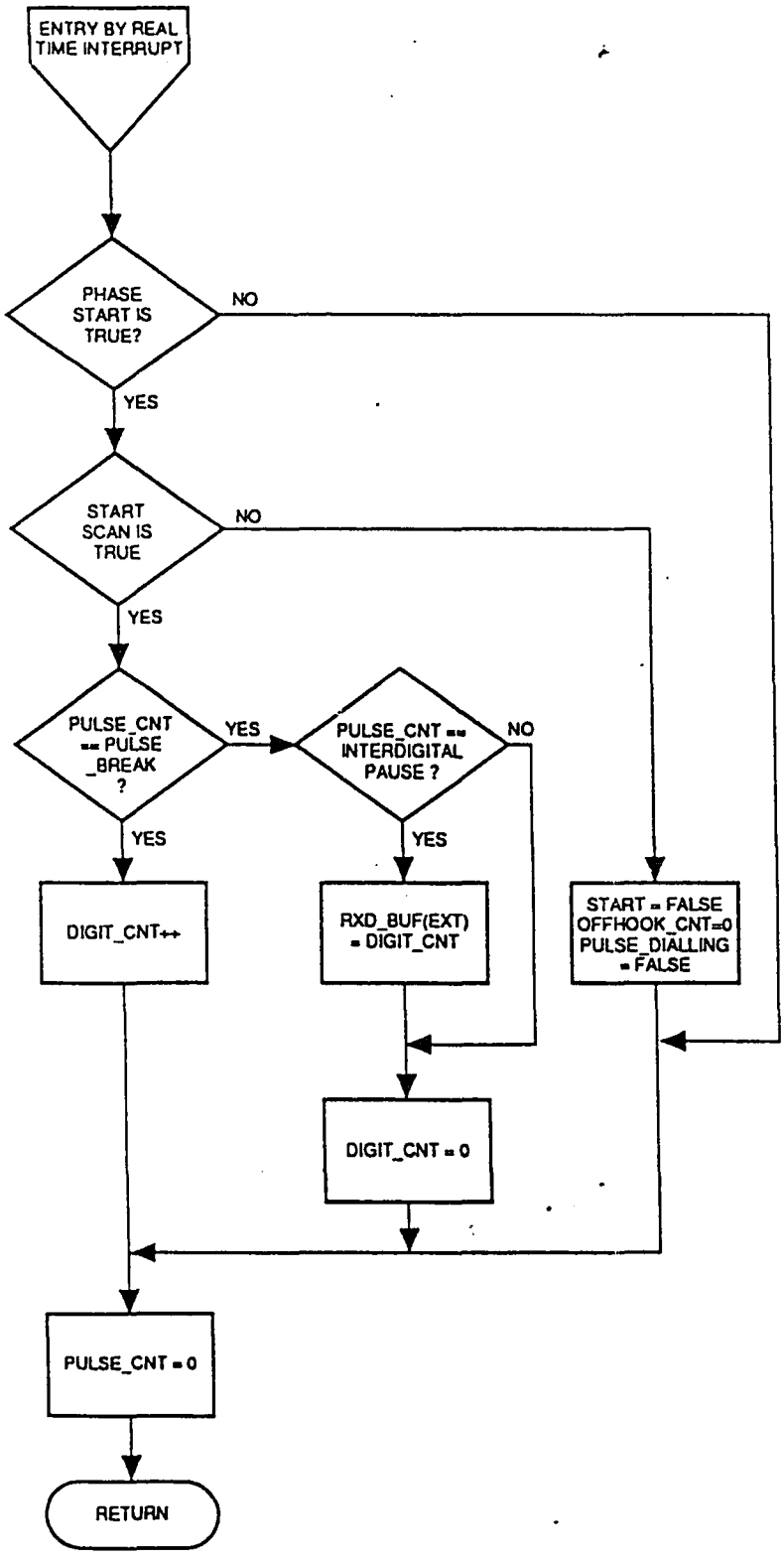
OFFHOOK WAS ONHOOK (SHEET 3)



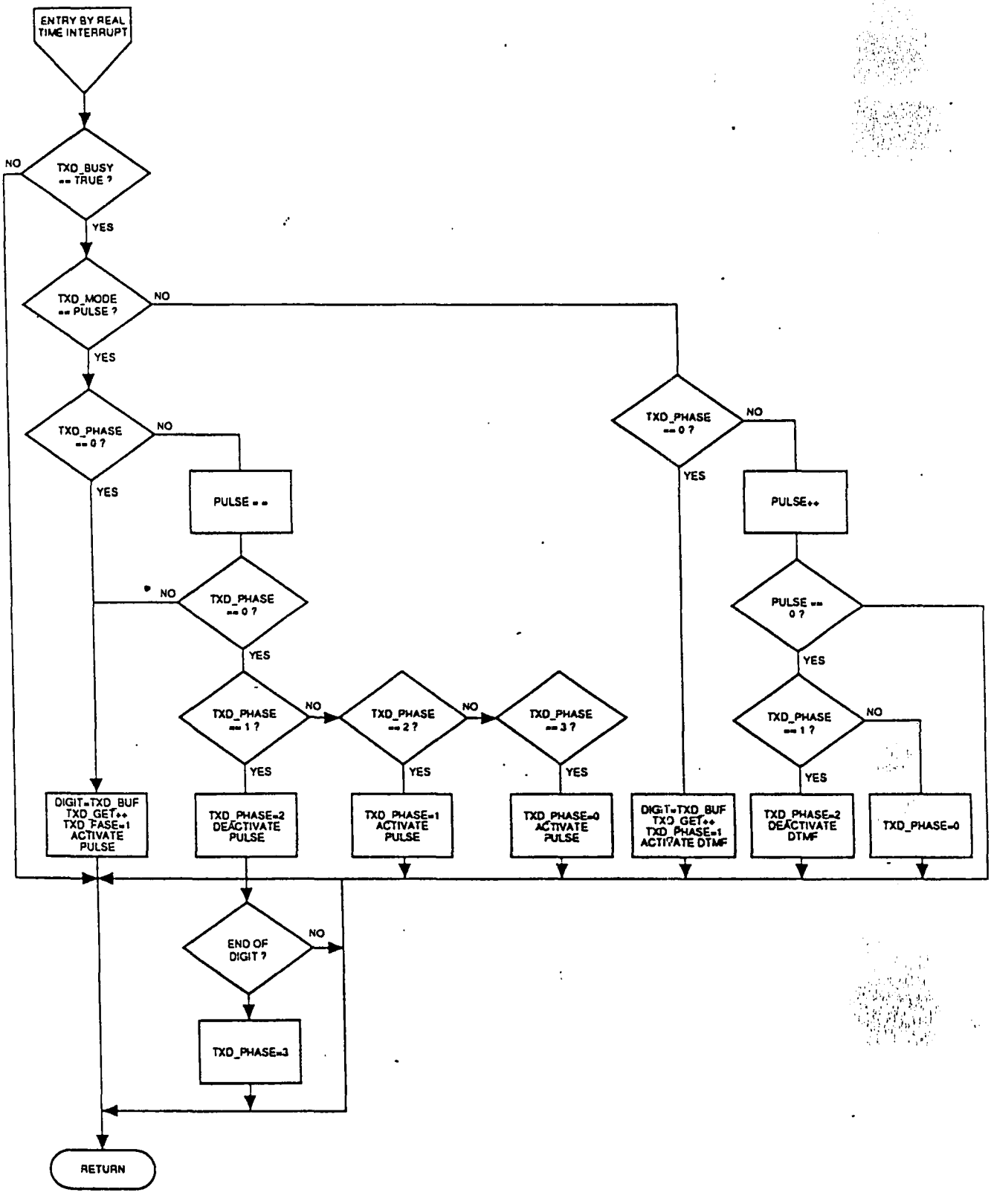
OFFHOOK WAS OFFHOOK (SHEET 4)



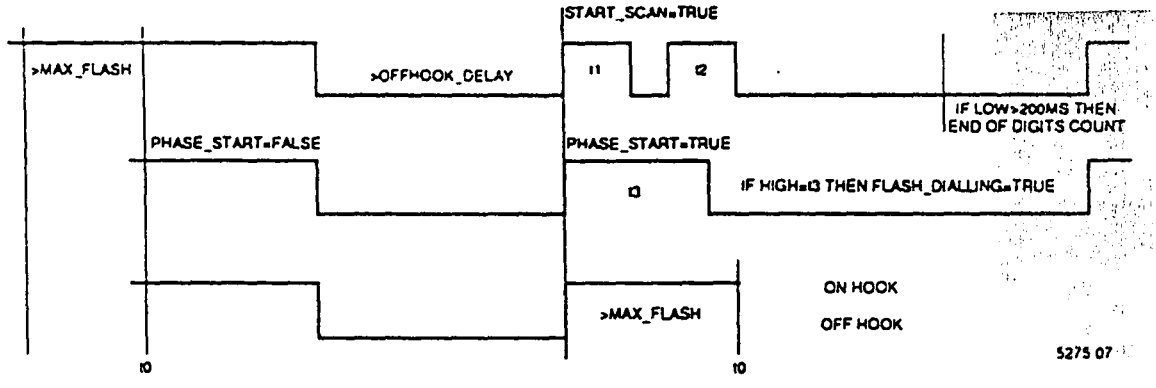
ONHOOK WAS OFFHOOK (SHEET 5)



TXD PROCEDURE (SHEET 6)



TIMING DIAGRAM REAL TIME INTERRUPT



MIN_PULSE_BREAK < t2 < MAX_PULSE_BREAK
MIN_FLASH < t3 < MAX_FLASH
MIN_PULSE_MAKE < t1 < MAX_PULSE_MAKE

5.1 SERIAL INTERFACE

The RS232 port accepts and transmits data only at 9600bps with 8 data bits, no parity and one stop bit. The following explains the format of the commands available.

5.2 DEBUGGING INFORMATION

For diagnostic purposes and debugging, the application program can send information on states and actions of the evaluation board to the RS232 port. This information can be used for test purposes as well as to check the program proceeding.

Two types of information can be selected to be sent to the RS232 port. Some information can be sent constantly and other information can be sent after certain actions. The information sent constantly can be used to form a status line at the top of the screen. This status line gives the following information:

STATE: The state for the extensions and trunks is printed. The first two digits represent the state of extension 1 and the last two digits the state of trunk 2. State is the number determined in the function determine_state().

CONN: Gives for all extensions and trunks its connections.

HOLD: Gives the devices on hold to that extension.

MASTER: Shows the extension assigned as master.

All the information on devices is defined by the byte:

(D7..D0) = (n.u, conference line, tr2, tr1, ext4, ext3, ext2, ext1).

Therefore extension 1 is represented by 0x01 and the conference line by 0x40. Note: information is printed in the hexadecimal format by two digits.

The occasional information is sent after certain actions. This information can be used to check on the program flow. Action information is printed below the status line.

5.3. "AT>" OPERATION COMMANDS

The SC11390 evaluation board has an RS232 interface through which all of the features that can be operated from the telephone keypad can also be effected from a PC. Commands are issued in the 'AT' style like a modem only a delimiter '>' is used between the 'AT' and the command letter.

5.3.1 AT> command format:

Each command has the same format which identifies the extension requesting the action, the command and the data for the command

AT>{Address}{Command}{Data}<CR><LF>
Where:

Address= {E/T}{xx}

(E = extension, T = trunk)
xx = 01,02,11,12,13,14
trunk/extension number
xx=00=ALL trunks, extensions
Command: see table 5-1
Data: see table 5-1

Command	Data	Action
D = Dial	T	Tone dialling
	P	Pulse dialling
	!	Hookflash
	0..9, *,#	Digits
	,	2 Sec pause
	@	5 Sec pause
F = Fax	1	Set fax detection on if address is trunk or set default fax location if address is extension.
	0	Reset fax
O = Operator	1	Set operator to address
	0	Reset operator to address
S = Set Register	{{reg. address}{reg. data}}	
	00,.., FF	Register address: 2 digits HEX ASCII format
	00,.., FF	Register data: 2 digits HEX ASCII format
R = Read Register(s)	{reg. address}	
	00,.., FF	2 Digits HEX ASCII format
	?	? Returns status of all registers

Table 5-1 AT> Commands

© Copyright 1993 Sierra Semiconductor

In the normal way, each AT> command has a set of responses. The response format is:

AT>{address}{status}{master}{connection}{connect_req}{hold}OK<CR><LF>

ADDRESS: as above

STATUS: format HEX ASCII

for extensions
00 Onhook
01 Offhook
for trunks
00 No Ring
01 Ring

MASTER: format HEX ASCII

01 Extension 11
02 Extension 12
04 Extension 13
08 Extension 14

CONNECTION: format HEX ASCII

01 Extension 11
02 Extension 12
04 Extension 13
08 Extension 14
10 Trunk 01
20 Trunk 02
40 ConferenceLine

CON_REQUEST: format HEX ASCII

01 Extension 11
02 Extension 12
04 Extension 13
08 Extension 14
10 Trunk 01
20 Trunk 02

HOLD: See connection

Register read AT>00R?
Response

AT>{data R0}.....{data R15}OK<CR><LF>
AT>{data R16}....{data R31}OK<CR><LF>
AT>{data R32}....{data R47}OK<CR><LF>
AT>{data R48}....{data R63}OK<CR><LF>
AT>{data R64}....{data R79}OK<CR><LF>
AT>{data R80}....{data R95}OK<CR><LF>
AT>{data R96}....{data R111}OK<CR><LF>
AT>{data R112}...{data R127}OK<CR><LF>
OK<CR><LF>

If request data = {Reg. n}

AT>{data Rn}OK<CR><LF>

Data Rn =2 digits HEX ASCII 00,...,FF
NOTE: If register does not exist or cannot be read, invalid data is returned.

5.4 COMMAND SEQUENCE EXAMPLES.

Return status of all extensions and trunks.

AT>E00?<CR><LF>

Response.

AT>E110100000000OK<CR><LF>

AT>E120100000000OK<CR><LF>

AT>E130100000000OK<CR><LF>

AT>E140100000000OK<CR><LF>

AT>E010000000000OK<CR><LF>

AT>E020000000000OK<CR><LF>

Return status of extension 12

AT>E12?<CR><LF>

Response.

AT>E1201000000OK<CR><LF>

Fax select on extension 11

AT>E11F1<CR><LF>

Fax detection added on trunk 1

AT>T01F1<CR><LF>

Fax detection added on trunk 2

AT>T02F1<CR><LF>

Now fax detection is on both trunks

Operator on extension 12

AT>E12O1<CR><LF>

Extension 14 dials extension 13

AT>E14D13<CR><LF>

Connect extension 11 to DTMFT1

AT>E11S0B03<CR><LF>

Return data of extension11 transmit register

AT>E11R01<CR><LF>

Response

AT>02OK<CR><LF>

เอกสารอ้างอิง

1. นาวาตรีธวัชชัย เลื่อนฉวี , " เทคโนโลยีโทรศัพท์ " , พิมพ์ครั้งที่ 3 สำนักพิมพ์บันเทิงการพิมพ์ กรุงเทพมหานคร , 2533

2. ดร.วิวัฒน์ กิตรานนท์ , " พื้นฐานการสื่อสาร์ " , สำนักพิมพ์สุภา , กรุงเทพมหานคร

3. SIERRA SEMICONDUCTOR , " SC11390 PABX Evaluation board and Software Manual. " Revision 5.0 , June 1993

4. BERNARD SKLAR , " DIGITAL COMMUNICATIONS " , Prentice-Hall International Editions Los Angeles , 1988

5. GARY M. MILLER , " MODERN ELECTRONIC COMMUNICATION " , Third Edition , Prentice-Hall International Editions , New Jersey , 1988

6. MIKELL P. GROOVER , " AUTOMATION , PRODUCTION SYSTEMS AND COMPUTER INTEGRATED MANUFACTURING " , Prentice-Hall International Edit ons , New Jersey , 1987