



ปีการศึกษา 2532

เรื่อง

ชุดสาย ELECTRONIC

ระบบ

STORE PROGRAM CONTROL

(SPC)

โดย

นายสุทิน งามสุข

นางมารินทร์ อนิวรรตน์

นายวันชัย นิลกำแหง

อาจารย์ที่ปรึกษา

อ.ประสิทธิ์ วัชรพิบูลย์

ปริญญาโทปีการศึกษา 2532

ภาควิชาเทคโนโลยีอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง


ชุดสาย ELECTRONIC ระบบ STORE PROGRAM CONTROL (SPC)

ผู้จัดทำ

1. นายสุทิน งามสุข
2. นางมารินทร์ อนิวรรณ
3. นายวันชัย นิลกำแหง


.....

อาจารย์ที่ปรึกษา


.....

อาจารย์ที่ปรึกษา

.....

อาจารย์ที่ปรึกษา

PROJECT

เรื่อง

ชุดสายโทรทัศน์ ELECTRONIC ระบบ Stored Program Control (SPC)

อ.ที่ปรึกษา

อาจารย์ ประดิษฐ์ วัชรนิบูลย์

ผู้ร่วมงาน

- นายสุทิน งามสุข 303637
- นางมารินทร์ อนิวรรณ 303619
- นายวันชัย นิลกำแหง 303623

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำนำ

ในปัจจุบันความเจริญก้าวหน้าทางด้าน Electronic ได้ก้าวไปอย่างรวดเร็วโดยเฉพาะ Computer จะเห็นว่างานในแทบทุกสาขาต้องมี Computer ไว้ใช้งานด้านเครื่องจักรต่าง ๆ จะมี Computer เป็นตัวควบคุมการทำงาน

ในเครื่องชุดสาย Electronic ในปัจจุบันก็ใช้ Computer เป็นตัวควบคุมการทำงานทั้งหมด เรียกว่าชุดสายระบบ Store Program Control (SPC) และชุดสาย SPC นี้ถูกนำเข้ามาใช้แทนชุดสาย Crossbar Switch ซึ่งกำลังจะหมดไปและจะถูกเลิกใช้ไปในที่สุด เนื่องจากทางบริษัทผู้ผลิตได้หยุดผลิตชุดสายชนิดนี้ เพราะอุปกรณ์เป็นอุปกรณ์ขนาดใหญ่ ใช้วัตถุดิบมากเป็นการไม่ประหยัด และชุดสาย Crossbar Switch ยังทำงานได้ช้ากว่าชุดสาย Electronic มาก

สำหรับ Project นี้ จะแสดงให้เห็นถึงตัวอุปกรณ์ต่าง ๆ และหน้าที่ของอุปกรณ์นั้น ๆ ว่ามีอะไรบ้าง และแต่ละอุปกรณ์นั้น ๆ ทำหน้าที่อะไร รวมทั้ง Flow chart, วงจร และการทำงานติดต่อกันระหว่างส่วน Control เอง และส่วน Control กับส่วน Switching part เพื่อให้เข้าใจการทำงานในขั้นตอนต่าง ๆ ของชุดสายโทรศัพท์เป็นอย่างดี

คณะผู้จัดทำ

จุดมุ่งหมาย

ในการทำ Project เกี่ยวกับตัวอย่างชুমสาย Stored Program Control (SPC) เพราะเห็นว่าในปัจจุบันนี้ทั่วโลกกำลังพัฒนาด้านชুমสาย SPC กันอย่างรีบเร่ง และชুমสายทั้งหมดในโลกนี้จะถูกเปลี่ยนไปใช้ เป็นระบบ SPC ทั้งหมด

ดังนั้นผู้ที่จะไปทำงานทางด้านชুমสายโทรศัพท์ จึงต้องศึกษาให้เข้าใจถึงระบบ โดยเฉพาะอย่างยิ่งการนำ Computer มาควบคุมการทำงานของชুমสาย SPC

แต่ชুমสายขนาดใหญ่ที่นำมาใช้ ทางบริษัทผู้ผลิตและจำหน่าย มีแต่คู่มือการใช้และควบคุมชুমสายเท่านั้น คู่มือหรือตำราเกี่ยวกับการ Control หรือติดต่อซึ่งกันและกันในระหว่างอุปกรณ์ในชুমสาย หรือทฤษฎีเกี่ยวกับด้านชুমสายนั้น ไม่มีแสดงไว้ จึงเป็นการที่จะ เข้าใจได้ดียากมาก

คณะผู้จัดทำจึงได้นำชুমสายโทรศัพท์ SPC ขนาด 4 เลขหมาย มาเป็นตัวอย่าง เพื่อแสดงให้เห็นถึงอุปกรณ์สำคัญ ๆ, หลักการสำคัญ ๆ ของชুমสาย ตลอดจนการทำงานร่วมกับช่องอุปกรณ์ต่าง ๆ ในระบบ SPC เพื่อให้ผู้สนใจได้ศึกษา เข้าใจเรื่องชুমสาย SPC

คณะผู้จัดทำหวังว่า ตัวอย่างชুমสาย SPC นี้คงเป็นเครื่องจูงใจให้ผู้สนใจได้ติดตามศึกษาค้นคว้า ในชুমสายขนาดใหญ่ต่อไป อันอาจก่อให้เกิดประโยชน์ต่อสังคมและประเทศชาติต่อไปด้วย

คณะผู้จัดทำ

บทคัดย่อ

เรื่อง ชุมสายโทรศัพท์ Electronic ระบบ Stored Program Control (SPC)

ชุมสายโทรศัพท์ตัวอย่างนี้ย่อมาจากชุมสายโทรศัพท์จริง

ส่วนประกอบของชุมสาย ประกอบด้วยภาค Switching part คือส่วนที่ใช้บรรจุเบอร์ผู้เข้ามีส่วนต่าง ๆ คือ Speech path ใช้ต่อผู้เข้าเข้าหากัน, LI สำหรับรับสภาวะการยกหูหรือวางหูของผู้เข้า KR ส่ง Dial tone และรับเลขหมายจากการกด Key set ของเครื่องโทรศัพท์, ABJ ต่อเชื่อมผู้เข้าให้พูดกันได้ ส่ง Ringing current ส่ง Ring Back tone, BS ส่ง Busy tone

ภาค control Part Unit (CPU) ประกอบด้วยส่วน control การทำงานในชุมสายทั้งหมด, ส่วน Program stored เก็บ Program ทั้งหมด ส่วน Data stored เก็บ Data Structure ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

1. ข้อดีของ เครื่องชুমสาย SPC.....	1
2. ข้อเสียของ เครื่องชুমสาย SPC.....	1
3. หลักการทั่วไปของชুমสาย SPC 4 เลขหมาย.....	2
4. ข้อกำหนดที่ต้องการ.....	3
5. การพิจารณาหน้าที่การทำงาน.....	4
6. ภาค Switching Part.....	11
7. รูปแบบของ CPU.....	21
8. Data Structure.....	25
9. Block Diagram.....	31
10. ตัวอย่าง Flow Chart และหลักการทำงาน.....	32
11. Flow Chart สมบูรณ์.....	40
12. Program.....	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องชุมสายโทรศัพท์ Stored Program Control SPC

1. ข้อดีของเครื่องชุมสาย SPC

- 1.1 มีขนาดเล็ก
- 1.2 ประหยัดเงินสร้างวงจรควบคุมที่มีขนาดเล็ก
- 1.3 ประหยัดเนื้อที่ติดตั้งชุมสาย
- 1.4 ทำงานได้เร็วมาก บางขั้นตอนใช้เวลาเป็น μs
- 1.5 ง่ายต่อการเปลี่ยนแปลง เพิ่มเติม หน้าที่การทำงานของชุมสาย
- 1.6 สามารถบริการพิเศษ (Facilities) ต่าง ๆ ได้มากขึ้น
- 1.7 มีระบบควบคุมและวิเคราะห์เหตุเสียได้ดีขึ้น
- 1.8 เมื่อมีอุปกรณ์เสียสามารถเปลี่ยนใหม่ได้อย่างรวดเร็ว
- 1.9 ง่ายต่อการเปลี่ยนแปลงและพัฒนาระบบ Numbering และระบบการคิดเงิน
- 1.10 ขณะเครื่องชุมสายทำงานไม่เกิดเสียงดัง

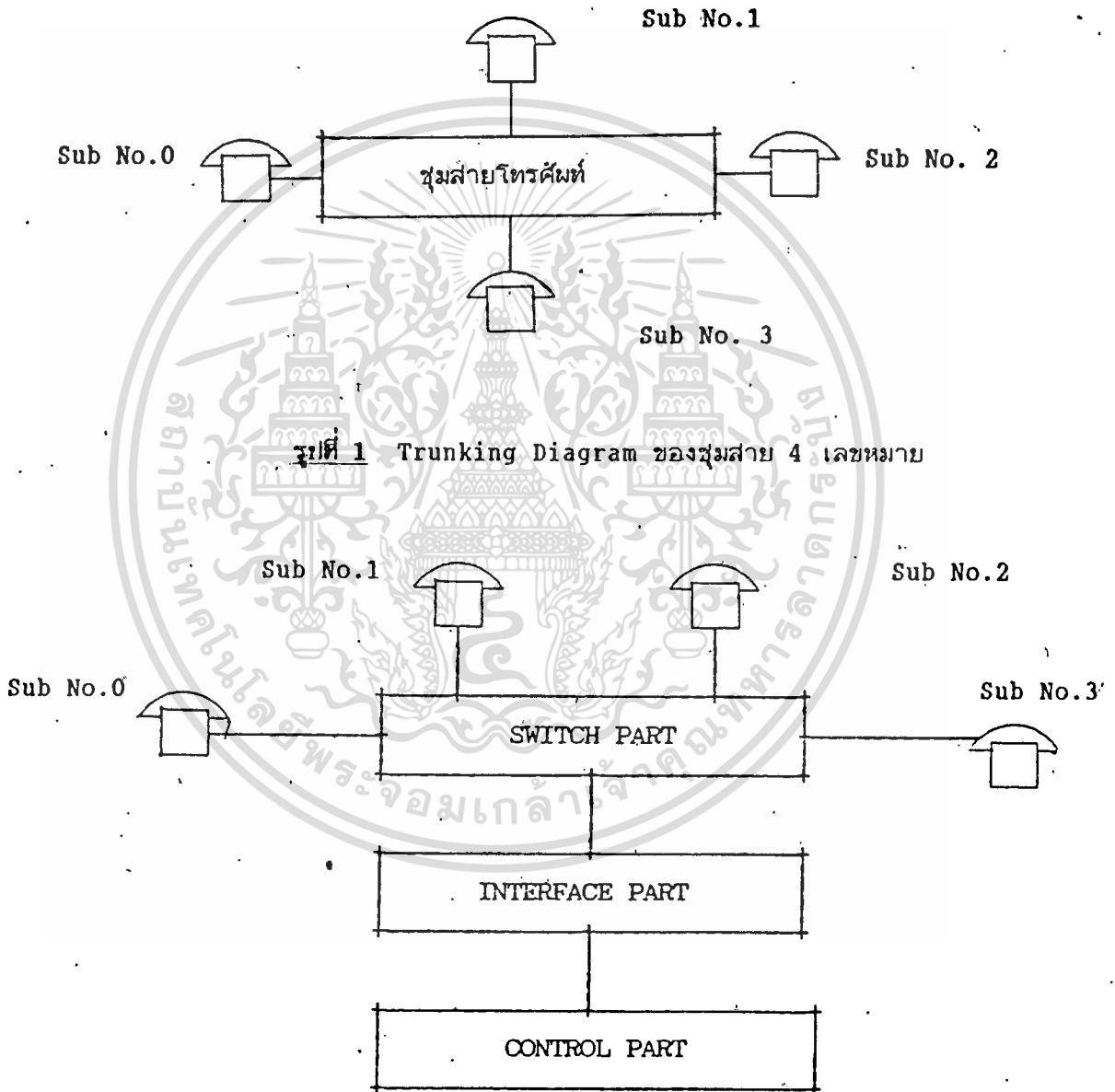
2. ข้อเสียของเครื่องชุมสาย SPC

- 2.1 ต้องจ่ายค่าไฟฟ้ามากกว่า เนื่องจากอุปกรณ์ของชุมสาย SPC ทำงานตลอดเวลา ส่วน Crossbar Switch ขณะไม่มีผู้เข้าเรียก อุปกรณ์ส่วนมากจะอยู่ในสภาพไม่ทำงาน
- 2.2 อุณหภูมิและความชื้นภายในห้องชุมสาย ต้องคงที่ตลอดเวลา ดังนั้นระบบ Air Condition ต้องสามารถควบคุมอุณหภูมิและความชื้นตลอดเวลา

สรุปแล้วจะเห็นว่าระบบ SPC ดีกว่าระบบ Crossbar Switch มาก

3. หลักการทั่วไปของชุมสาย SPC 4 เลขหมาย

3.1 Trunking Diagram



รูปที่ 2 Trunking Diagram แสดงส่วนต่าง ๆ ของชุมสาย SPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ข้อกำหนดที่ต้องการ

เมื่อใช้เป็นเครื่องสาธิตหลักการเบื้องต้นของชุมสาย Electronic ระบบ Store Program Control (SPC) จึงขอแนะนำตัวอย่างเครื่องชุมสายขนาด 4 เลขหมาย ซึ่งจัดทำขึ้นเพื่อสำหรับการติดต่อกันภายในชุมสายเดียวกัน (Internal Call) เท่านั้น สำหรับตัวเครื่องโทรศัพท์ (Telephone Set) นั้น ใช้แบบกดปุ่ม (Push Button)

กฎเกณฑ์การทำงานของชุมสายนี้ มีดังนี้ :-

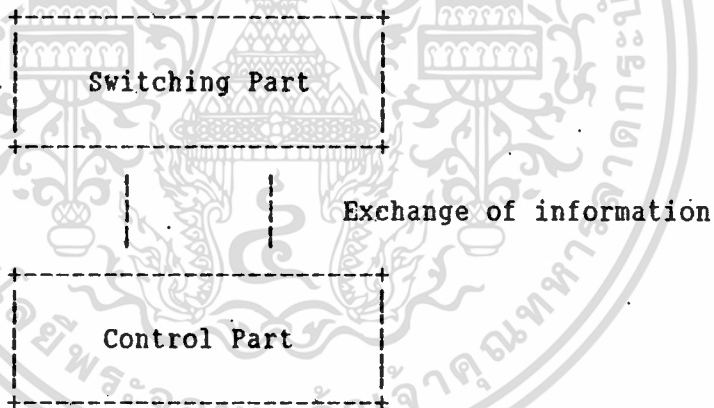
- ผู้เข้าที่มีความประสงค์จะทำการเรียก (Sub A) เมื่อกดปุ่มโทรศัพท์ของเขาขึ้นแล้ว เครื่องชุมสายจะต้องส่งสัญญาณ Dialling tone ให้ผู้เข้า เพื่อบอกให้รู้ว่าชุมสายพร้อมทำงานให้แล้ว
- เมื่อผู้เข้า Sub A ได้ยินสัญญาณ Dialling tone แล้ว สามารถกด Key set ของเครื่องโทรศัพท์ส่ง เลขหมายของผู้เข้าที่เขาต้องการติดต่อ (Sub B) ไปยังเครื่องชุมสายได้
- เมื่อเครื่องชุมสายได้รับเลขหมายของผู้เข้าที่ถูกเรียก (Sub B) แล้ว จะตรวจดูสถานะของ Sub B ว่าว่างหรือไม่ ถ้าว่างจะตรวจดู Trunk (ในที่นี้ใช้ ABj) ว่าอยู่ในสถานะว่างหรือไม่ ถ้าว่างจะทำการต่อ Sub A เข้า ABj และ Sub B เข้า ABj แล้วจะส่ง Ringing Signal (กระแสไฟ) ไปยังกระดิ่งของ Sub B และ Ringing control tone ไปยัง Sub A
- เมื่อผู้เข้า Sub B ยกหูตอบรับ Ringing Signal และ Ringing control tone จะถูกตัดออก Sub B กับ Sub A จะถูกต่อเข้าถึงกัน (Speech connection) ให้สามารถคุยกันได้ และ Meter ของ Sub A จะถูก Step ขึ้น 1 ครั้ง เมื่อบันทึกการเรียกของ Sub A
- เมื่อผู้เข้าฝ่ายใดฝ่ายหนึ่งวางโทรศัพท์ลง Speech path จะถูกตัดออกไป และจะมีสัญญาณ Busy tone ส่งไปยังอีกฝ่ายหนึ่งทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ต่าง ๆ ในเครื่องชุมสายขนาดเล็กนี้จะมีอย่างละ 1 ตัว เพื่อความจำเป็นใช้ทดลองเท่านั้น ดังนั้นเมื่อผู้เช่าใดผู้เช่าหนึ่งกำลังติดต่อกับ ตัวส่ง Dial tone หรือกำลังส่ง เลขหมายให้ตัวเก็บ เลขหมาย (KR) ขณะนั้นถ้าผู้เช่ารายอื่นยกหูเพื่อจะเอา Dial tone ก็จะถูกต่อเข้ากับตัวส่ง Busy tone (BS) ผู้เช่ารายนั้นจะได้รับ Busy tone แสดงให้รู้ว่าเครื่องชุมสายยังไม่พร้อมจะทำงานได้

ความจุ (Capacity) สามารถต่อผู้เช่าให้สามารถติดต่อกันได้ครั้งละ 1 คู่ (One speech connection at a time) เนื่องจากมี trunk (ABj) เพียงตัวเดียวเท่านั้น ดังนั้นหากผู้เช่า รายใดทำการเรียกขานผู้เช่าสองรายกำลังพูดกันอยู่ หรือกำลังรับ Ringing signal หรือ Ringing control tone อยู่ ผู้เช่ารายนั้นจะถูกต่อเข้า BS และจะได้รับ Busy tone

5. การพิจารณาหน้าที่การทำงาน



รูปที่ 3 แสดงโครงสร้างพื้นฐานของชุมสายโทรศัพท์ที่ใช้กันทั่ว ๆ ไป โครงสร้าง ทั่วไปจะประกอบด้วย 2 ส่วน คือ ส่วน Switching Part และ Control Part

ส่วน Switching Part

ประกอบไปด้วยสายต่อและ contacts ต่าง ๆ สำหรับใช้ต่อทางด้าน Signals และ Speech Part นอกจากนี้ยังมีวงจรซึ่งทำหน้าที่เหมือนโทรศัพท์ทั่ว ๆ ไป เช่น Tone generators ซึ่งใช้สำหรับส่ง Busy tone, Dial tone, Ringing control tone และ Generator สำหรับส่ง Ringing Signal วงจรต่าง ๆ สำหรับ Hook Signal และ Key set signal การค้า และวงจรสำหรับแปลงสัญญาณเหล่านี้ให้เป็นรูปแบบที่เหมาะสม เพื่อส่งไปให้ Control Part

ส่วน Control Part

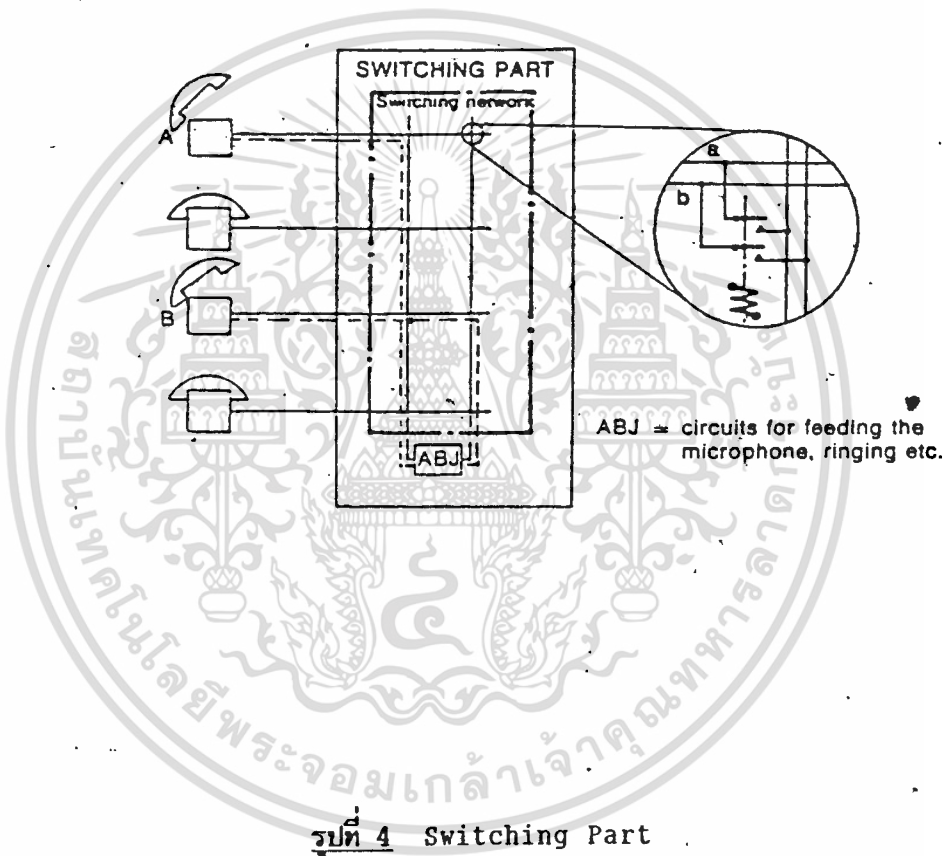
ประกอบด้วยวงจรและโปรแกรมต่าง ๆ ซึ่งควบคุมหน้าทำงานสำคัญ ๆ ในชุมสาย เช่น วงจรตรวจสอบสถานะใน Switching Part เช่น การวางหู หรือยกหูของผู้เข้าตลอดจนวงจรสำหรับควบคุมการทำงานของวงจรใน Switching Part ให้ทำงานตามโปรแกรมที่กำหนดให้ เครื่องชุมสายต้องปฏิบัติในสถานการณ์ต่าง ๆ กันด้วย

นั่นคือ Control Part เป็นตัวกำหนดว่าจะต้องทำอะไรและเมื่อใด เมื่อมีการเปลี่ยนแปลงสถานะทาง Switching Part เช่น ผู้เข้ายกหูโทรศัพท์ (มีการเปลี่ยนแปลงจากโทรศัพท์ที่วางอยู่บนเครื่องถูกยกขึ้น) Control Part รับรู้สถานะแล้วจะต้องดำเนินการอย่างไร เช่น ถ้าผู้เข้าต้องการเรียกผู้อื่น ก็จะดำเนินการต่อ Dial ให้ หรือถ้ายกหูเมื่อรับสายผู้อื่น ก็จะต่อ Speech path ให้ ฯลฯ

เทคนิคต่าง ๆ ที่จะนำมาใช้ในส่วนตัวต่าง ๆ ต่อไปจะเหมือนกับที่ใช้ในชุมสาย ระบบ SPC ในปัจจุบัน กล่าวคือ ใช้ Electro-mechanics และ Electronics สำหรับ Circuit logic ใน Switching Part และใช้ Electronic data processing พร้อมด้วย Memory storage ของ Program และข้อมูล (Data) ใน Control Part

เทคนิคที่ใช้ในส่วนทั้งสองดังกล่าว มีความแตกต่างกัน เนื่องจากปัจจัยหลายอย่าง เช่น Electro mechanical circuit โดยปกติต้องใช้แรงดันไฟฟ้าสูงกว่าที่ใช้ใน Electronic circuit และความเร็วในการทำงานของวงจรก็ต่างกันมาก การเปลี่ยนแปลงของภาวะใน Electro-mechanical circuit ต้องการเวลาดคิดเป็น MS แต่ใน Electronic control Part การเปลี่ยนแปลงดังกล่าวเกิดขึ้นเพียงช่วงเวลาคิดเป็น μ S ซึ่งเร็วกว่ามาก ความแตกต่างในทางเทคนิคดังกล่าวของ Switching Part และ Control Part ทำให้มีความจำเป็นต้องมีส่วนที่เรียกว่า "Interface Part" เข้ามาเพื่อให้ Switching Part และ Control Part สามารถทำงานสัมพันธ์กันได้ดังที่ต้องการ

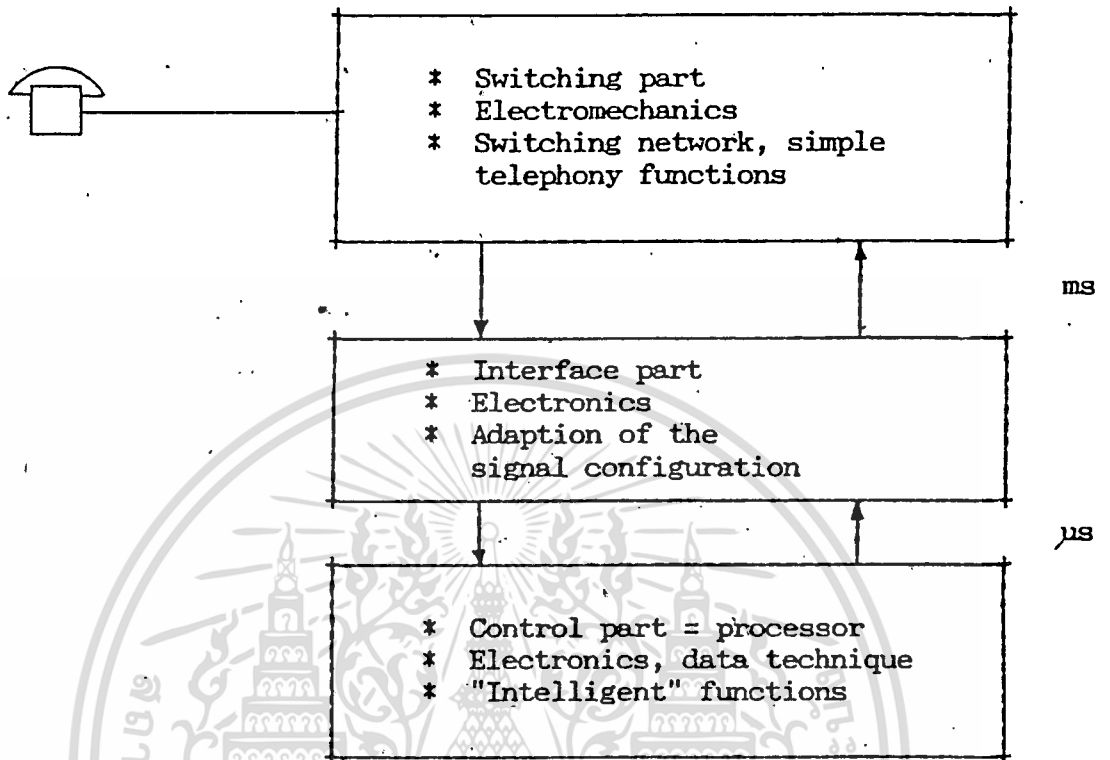
Interface Part ประกอบด้วยวงจร Electronic ที่ทำงานได้เร็วในระดับ μs และยังมี Memory element ซึ่งสามารถเก็บสภาวะการเปลี่ยนแปลงใน Switching Part และสามารถรับคำสั่งจาก Control Part ไปควบคุมให้ทาง Switching Part ปฏิบัติงานได้



รูปที่ 4 Switching Part

จากรูปที่ 4 จะเห็นว่าใน Switching part ประกอบด้วย Relay ต่าง ๆ ซึ่งทำงานโดยถูกควบคุมจาก Processor Processor ก็คือ Computer ที่ควบคุมขบวนการ (Process) ต่าง ๆ เมื่อมีสัญญาณจากภายนอกป้อนเข้ามาและดำเนินการที่จำเป็นต้องการของสัญญาณนั้น ๆ นั่นคือ Processor ควบคุมขบวนการด้าน Switching Part สำหรับการต่อโทรศัพท์ (Telephone connection)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5 ส่วนต่าง ๆ ของชุมสายระบบ SPC

รูปที่ 5 เป็นรูปแสดงให้เห็นถึงส่วนประกอบของชุมสาย SPC และการทำงานสัมพันธ์กัน โดยมี Interface Part เป็นตัวเชื่อมโยง

Processor ต้องรู้ว่าเมื่อไหร่จะเกิดขึ้นใน Switching Part ตลอดเวลา. เช่น มีผู้เข้าขอกหรือว่างหูโทรศัพท์หรือเปล่า ฯลฯ นอกจากนั้น Processor ยังต้องสามารถไปยังค้ำให้ Relay ต่าง ๆ ทาง Switching Part ทำงานได้ด้วย เพื่อจะได้ต่อผู้เข้าหนึ่ง เข้ากับอีกผู้เข้าหนึ่ง เพื่อให้สามารถพูดโทรศัพท์กันได้

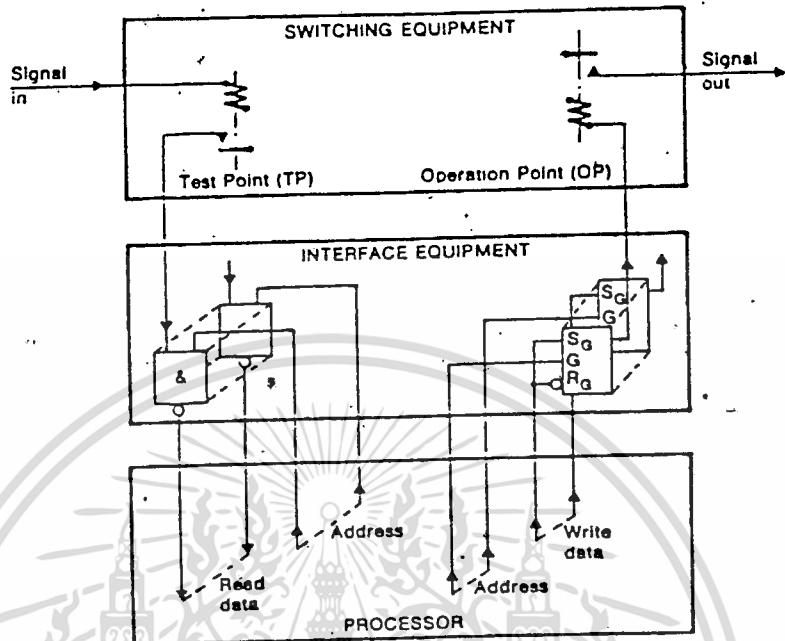
ดังนั้น จึงเห็นว่ามี การแลกเปลี่ยนข่าวสารกันระหว่าง Switching Part และ Control Part โดยผ่าน Interface Part อยู่ตลอดเวลา ดังรูปที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการที่ Processor สามารถตรวจผ่านทาง Interface ว่ามีอะไรเกิดขึ้นใน Switching Part โดยใช้ Address signal เพื่อไปเปิด Gate ให้อ่านสถานะของ Relay ต่าง ๆ ใน Switching Part โดยเมื่อมีการยกหู หรือวางหูใน Switching Part สถานะการวางหูหรือยกหูใน Switching Part นั้น จะมาปรากฏใน Interface และ Processor จะไปอ่านสถานะเอามาจาก Interface การตรวจสอบนี้เรียกว่า "การอ่าน" (Reading) และ ข้อมูลที่อ่านได้นี้เรียกว่า Read data จะเห็นว่าเมื่อเกิดการเปลี่ยนแปลงสถานะทาง Switching Part (ยกหูหรือวางหู) ทาง Switching Part มิได้ส่งการเปลี่ยนแปลงไปยัง Control Part (Processor) โดยตรง แต่สถานะนั้นจะไปปรากฏอยู่ใน Interface ช่วงนี้ทาง Processor ยังไม่รู้ว่าเกิดอะไรขึ้นบ้างทาง Switching Part จนกว่าจะถึงเวลาที่กำหนดให้ Processor ไปอ่านสถานะใน Interface. (สำหรับขั้วสายตัวอย่างนี้กำหนดให้ทุก ๆ 10 MS ให้ Processor ไปอ่านข้อมูลใน Interface 1 ครั้ง)

การอ่านนี้จะไปอ่านสถานะ Contact ของ Relay ซึ่งส่งผลมาปรากฏใน Interface จุดที่ Processor อ่านข้อมูลมานี้เรียกว่า Test point

ใน Interface Part ต้องมี Memory element ไว้ด้วย เมื่อรับคำสั่งมาจาก Processor ไปบังคับให้ Relays ต่าง ๆ ใน Switching Part ทำงาน จุดที่ถูกบังคับให้ทำงานเรียกว่า Operater point Memory element นี้อาจเป็นเพียง Bistable flip flop ก็ได้ เพื่อจะให้ Relay ตัวใดทำงาน Processor ก็จะต้องออกคำสั่งโดยใช้ Address บ่งชี้ Flip Flop ที่ควบคุม Relay นั้น ๆ ให้ทำงาน Flip Flop จะถูก Set หรือ Reset ด้วย "ข้อมูลที่เขียน" (Write data) ในสถานะเป็น 1 หรือ 0 หลังจากรับคำสั่งมาจาก Processor แล้ว Flip Flop จะเก็บคำสั่งนั้นไว้ และจะไปบังคับให้ Relay ทำงานอีกต่อหนึ่ง ทำให้ Control Part และ Switching part สามารถทำงานสัมพันธ์กันได้อย่างถูกต้อง โดยมี Interface เป็นตัวกลางของการติดต่อ ดังรูปที่ 6



รูปที่ 6 การตรวจสอบและการทำงานให้แก่ Switching Part ด้วยโมรเซสเซอร์

รูปที่ 6 เป็นการขยายให้ดูว่าภายใน Switching Part, Interface Part และ Control Part (Processor มีอะไรรอบๆและสามารถติดต่อกันได้อย่างไร)

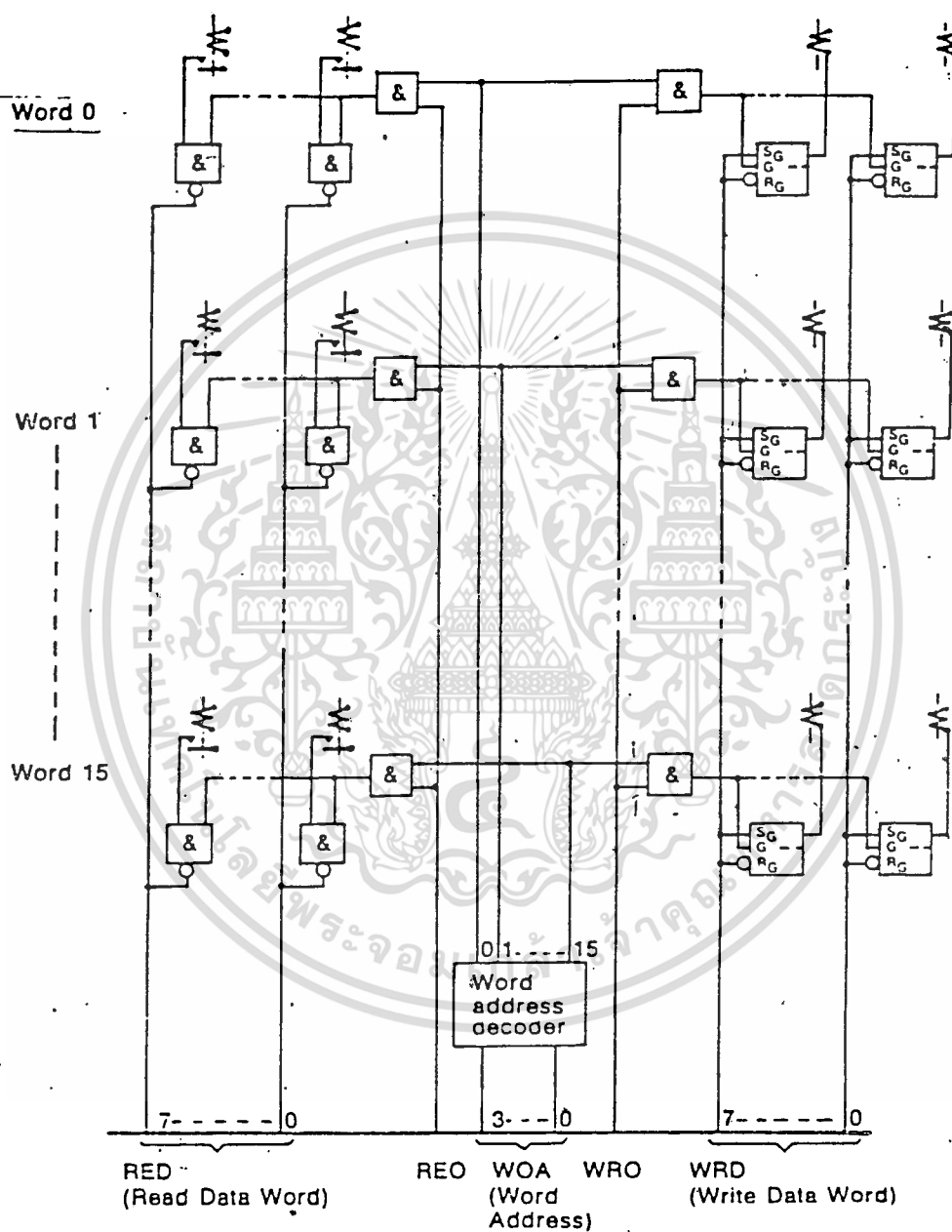
การอ่านหรือเขียนข้อมูลมักกระทำพร้อม ๆ กัน คราวละหลาย ๆ Bit positions Bit positions จำนวนหนึ่ง ๆ ประกอบกันเข้าเป็น "Word" Word หนึ่ง ๆ ซึ่งประกอบด้วย Test points จำนวนหนึ่ง เราเรียกว่า Test word และ Word หนึ่ง ๆ ซึ่งประกอบด้วย Operation points จำนวนหนึ่ง เราเรียกว่า "Operation word"

สัญญาณที่ต้องการสำหรับการอ่าน ประกอบด้วย Address และ Read order (REO) เมื่อป้อนข้อมูลที่อ่านและ Address ที่ต้องการอ่านออกมา สำหรับการเขียน ประกอบด้วย Address และ Write order (WRO) เมื่อป้อน การเขียนและ Address ที่ต้องเขียนเข้าไป (ตามตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ซุ่มสายนี้ใช้ 8 Bits/Add)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารที่ 026907



รูปที่ 7. ตัวอย่างของ 16 Words Word ละ 8 Bits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7 แสดงตัวอย่างของ 16 Words แต่ละ Word มี 8 Bits ในชุดสาย SPC ที่ยกมาเป็นตัวอย่าง เราอาจใช้ Word 0 สำหรับ Test points แสดงภาวะของสายผู้เช่า (ใช้ 1 Bit ต่อหนึ่งผู้เช่า) และสำหรับการทำงานของ Switching Part สำหรับผู้เช่าที่เป็นฝ่ายเรียก (1 Bit ต่อหนึ่งผู้เช่า) Word ที่ Address 1 อาจบรรจุ Test points สำหรับการรับจาก Key set และ Operation points สำหรับ รับจาก Key set และ Operation points Switching Part ของ Sub B ซึ่งจะได้กล่าวต่อไป

สำหรับรูปที่ 7 นี้ เป็นวงจรตัวอย่างแสดงการ Read หรือ Write ในทาง Switching Part

ตัวอย่างการอ่าน Processor ต้องการอ่าน Test Word 5

วิธีปฏิบัติ Processor ต้องส่ง Word Address 5 และ Read Order เข้าไป Processor ก็จะสามารถอ่านข้อมูลใน Word ที่ 5 ออกมาได้ ทั้ง 8 Bits ของ Word ที่ 5 ออกมา นั่นคือถ้าข้อมูลใน Word ที่ 5 มีอะไรบรรจุอยู่ ข้อมูลนั้นจะถูกถ่ายมายัง Processor

ตัวอย่างการเขียน Processor ต้องการเขียนข้อมูลเข้าไปที่ Test points (เพื่อบังคับให้ Relay operated) สมมติ ต้องการเขียนใน Address Word 0

วิธีปฏิบัติ Processor ต้องส่ง Address 0 และ Write Order เข้าไป ข้อมูลจาก Processor จะถูกเขียนเข้าไปใน Address 0 แล้วแต่ข้อมูลที่ต้องการจะเขียน ซึ่งส่งไปจาก Processor

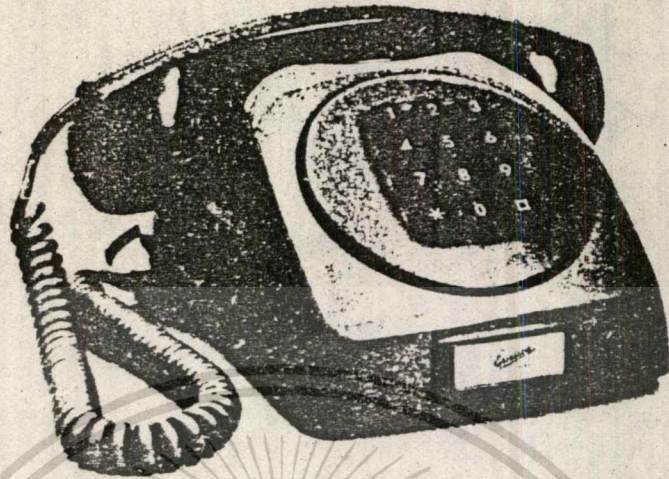
6. ภาค Switching Part

ใน Switching Part แบ่งเป็นส่วนต่าง ๆ ได้ดังนี้ :-

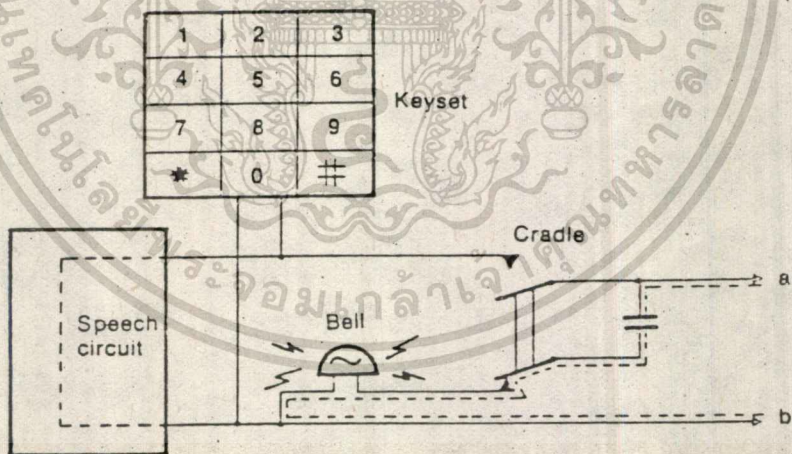
6.1 เครื่องโทรศัณท์ (Telephone Set) โทรศัณท์ที่ใช้ในเครื่องตัวอย่าง ใช้เครื่องแบบ Push Button.

ผู้เช่าแต่ละเบอร์จะต้องมีเครื่องโทรศัณท์พร้อม Key Set ดังแสดงในรูปที่ 8

ซึ่งเราสามารถแยกส่วนได้ 4 ส่วน คือ วงจรพูด (Speech Circuit), Cradle Switch, ขนด้านการค้า กระดิ่งและ Key Set พร้อมด้วย Tone Generator ดังรูปที่ 9 ถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 Telephone Set with a key set



รูปที่ 9

วงจรชุดประกอบด้วย ปากพูด หูฟัง และ Speech Transformer ส่วนประกอบ
ทั้งสองนี้ รวมอยู่ใน Handset (ที่เราถือขณะพูด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Cradle Switch จะทำงานเมื่อ Handset ถูกยกขึ้นหรือวางลง ใช้ Cradle Switch ไว้สำหรับบอกสถานะของการเรียก การตอบรับ การเลิกพูด และขณะวาง Handset ไว้บนเครื่อง จะเป็นตัวต่อวงจรกระดิ่งไว้คอยรับสัญญาณเรียก ขณะยก Handset ขึ้น จะไปต่อด้าน Key set และ Speech Circuit เมื่อใช้ส่งเลขหมายและพูดติดต่อกัน

กระดิ่งเป็นแบบ A.C.Bell. ต่ออยู่กับ A,B Wires ของสายผู้เช่า ผ่าน Capacitor และ Cradle Switch

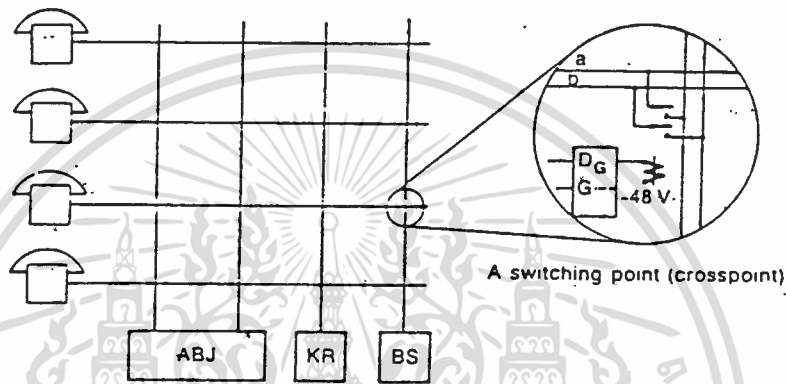
Key set มีไว้สำหรับส่งเลขหมาย (Digit) 0 - 9 และสัญญาณพิเศษ 2 อย่าง คือ ดาว (*) และรูปเหลี่ยม (#) สัญญาณทั้งสองนี้มีไว้ใช้บริการพิเศษเท่านั้น เมื่อกด Key อันใดอันหนึ่งจะมี Tone frequency 2 freq. ส่งออกไปดังในตารางรูปที่ 10

Hz->	1209	1336	1477
697	1	2	3
770	4	5	6
852	7	8	9
941	*	0	#

รูปที่ 10

ในชุด Key set จะมี Oscillator อยู่ 7 ชุด แต่ละชุดจะผลิต freq. ตามตาราง freq. เหล่านี้จะถูกส่งออกไป 2 freq. บวกกัน เมื่อกด Key set ตัวใดตัวหนึ่ง เช่น ถ้ากด 1 จะมี freq. 697 + 1209 ส่งออกไป ถ้ากด # จะมี freq. 941 + 1477 ส่งออกไป ฯลฯ

6.2 Switching Part



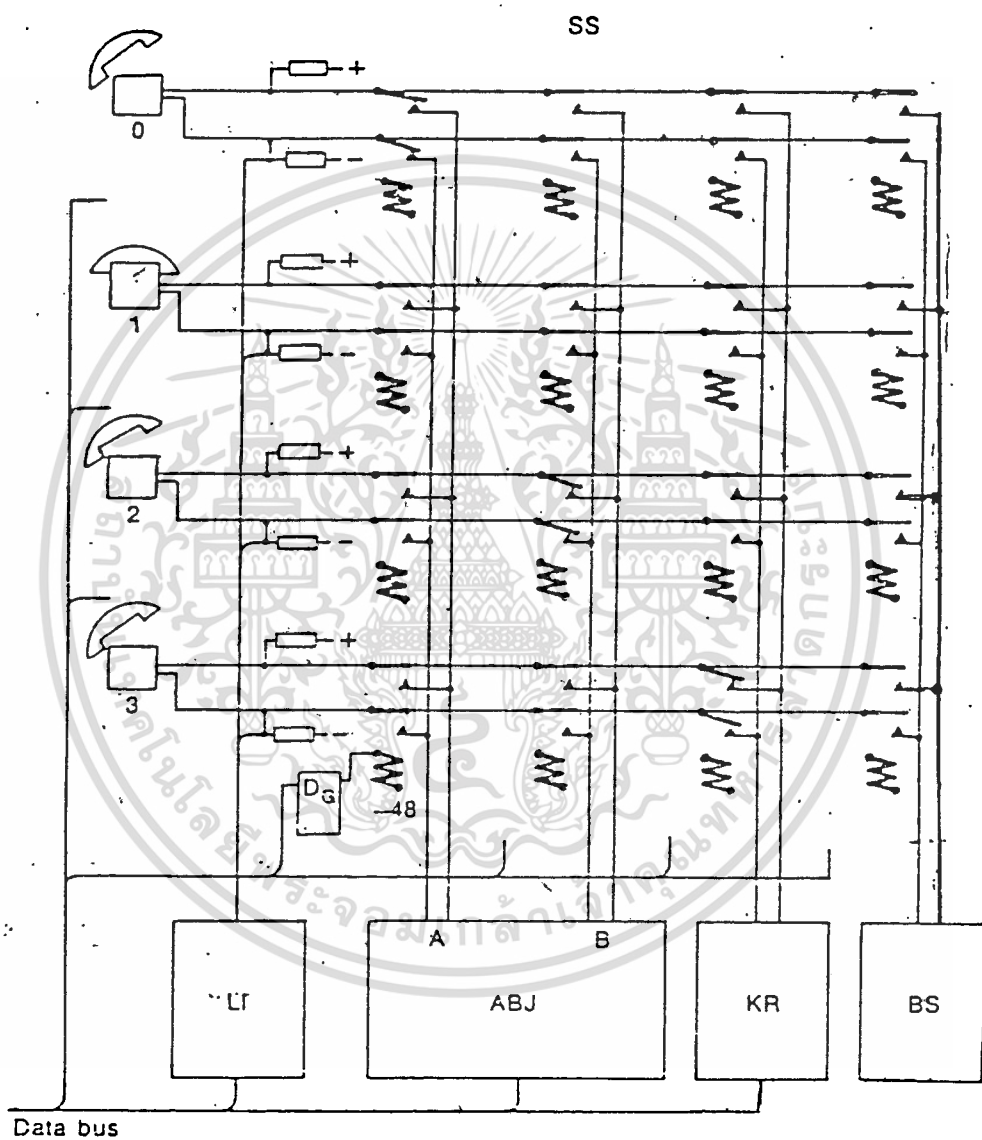
รูปที่ 11

ใน Switching Part จากรูปที่ 11 สำหรับเครื่องตัวอย่างนี้ มีผู้เช่า 4 เลขหมาย ผู้เช่าแต่ละเลขหมาย จะมี Relay ประจำเลขหมาย 4 ตัว (ดูจากรูปวงกลม) Relay ทุก ๆ ตัว จะทำหน้าที่ต่อ Crosspoint ไปยังอุปกรณ์ต่าง ๆ คือ ABj (AB junction) เพื่อให้ผู้เช่า พูดกันได้ โดยฝ่ายเรียกจะเชื่อมต่อทางด้าน Aj ผู้ถูกเรียกจะถูกต่อเข้าทาง Bj นอกจากนี้ ABj ยังทำหน้าที่ส่ง Ringing tone ไปให้ผู้ถูกเรียก Ringing control tone (Ring back tone) ไปยังฝ่ายเรียก ต่อ KR เพื่อรับ Dial tone และส่งเลขหมายที่ต้องการเรียกไปยัง KR (จาก Key set ไปเก็บใน KR) ต่อ BS ในกรณีการเรียกครั้งนั้น ชุมสายไม่สามารถดำเนินการต่อให้ได้ นอกจากนี้ อุปกรณ์ LI (Line Interface) อุปกรณ์ตัวนี้ไม่มี Relay ควบคุม เพราะอุปกรณ์ LI ถูกต่อตรงกับสาย A ของผู้เช่าแต่ละเลขหมายมีไว้สำหรับสภาวะการวางหูหรือยกหูของผู้เช่า (เปลี่ยนแปลงสภาวะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของอุปกรณ์ทั้งหมดทางภาค Switch Part ทั้งหมดนี้ได้รับการควบคุมจาก Processor ผ่านทาง Data bus

ตัวอย่างการต่อเข้าอุปกรณ์ดังกล่าวแสดงโดย รูปที่ 12



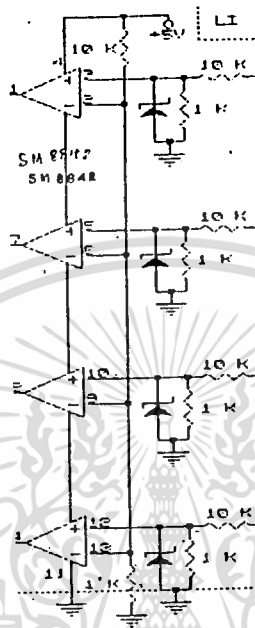
- LI = Line interface
- ABJ = AB-junctor
- KR = Key set receiver
- BS = Busy tone sender
- SS = Subscriber Stage

รูปที่ 12 เป็นรูปแสดงทางด้าน Switching Part ของชุมสาย Electronic ระบบ SPC 4 เลขหมาย จะเห็นว่าทางด้าน Switching Part นี้ ประกอบด้วย Subscriber Stage ซึ่งมีเลขหมายทั้งหมด 4 เลขหมาย (มีโทรศัพท์ 4 เครื่อง) แต่ละเลขหมายจะมีสาย A ซึ่งต่อเข้า Resister 1 ตัว ต่อมาเข้า LI (Line Interface) สำหรับแสดงสภาวะการยกหูหรือวางหูของผู้เข้า (อุปกรณ์ LI, ABj, KR และ BS จะอธิบายเฉพาะต่อไป) ที่ ABj มี Relay เลขหมายละ 2 ตัว ตัวซ้ายมือสุด สำหรับ สาย A และ B ของผู้เข้าฝ่ายเรียกเข้ากับ Aj ของ ABj และตัวถัดมาจะเป็น ตัวต่อผู้เข้าฝ่ายถูกเรียกเข้ากับทางด้าน Bj ของ ABj Relay ตัวถัดมาอีก จะเป็นตัวต่อผู้เข้าเข้ากับ KR และตัวขวาสุดจะต่อผู้เข้าเข้ากับ BS

ในรูปจะเห็นว่า หมายเลข 0 เป็นฝ่ายเรียกเพราะต่อเข้าทางด้าน Aj ของ ABj หมายเลข 2 เป็นฝ่ายถูกเรียกเพราะต่อเข้าทางด้าน Bj ของ ABj หมายเลข 0 กำลังถูกต่อเข้ากับ ABj ซึ่งขณะนี้เบอร์ 0 อาจกำลังได้รับสัญญาณ Ringing control tone เบอร์ 2 อาจกำลังได้รับ Ringing signal (สัญญาณกระดิ่ง) หรือทั้ง 2 อาจกำลังพูดกันก็ได้ ซึ่งจะขยายให้เห็นช่วงต่อไป หมายเลข 3 อาจกำลังได้รับ Dial tone หรือกำลังส่งเลขหมายให้ KR เพราะกำลังถูกต่อเข้า KR อยู่

หน้าที่และส่วนประกอบต่าง ๆ

อุปกรณ์ LI (Line Interface)



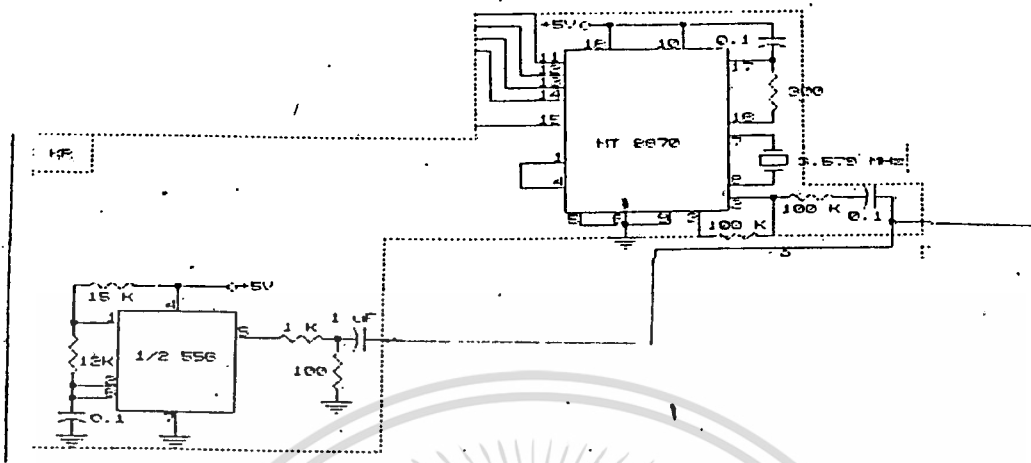
รูปที่ 13. Line Interface Device

ประกอบด้วย Op-Amp (LM324) มีหน้าที่สำหรับรับสภาวะการวางหูหรือยกหูโทรศัพท์ เพื่อบอกให้ CPU (Processor) ทราบ

การทำงาน โดยปกติที่สาย A และสาย B ของเครื่องโทรศัพท์จะมี Resistor 600 โอห์มต่อกันและไปต่อกันกับ 48 VDC (ดูรูป 12 ประกอบ) ขณะที่ผู้เช่าไม่ได้ยกหูโทรศัพท์ขึ้น จะมีบวก 48 VDC ส่งเข้าไปยัง Op-Amp เมื่อ Op-Amp ได้รับแล้วจะนำไป compare ใน Op-Amp จะได้ logic 0 ออกไปเข้ายัง Interface part ถ้าผู้เช่ายกหูโทรศัพท์ขึ้น ขั้วบวกและขั้วลบจากสาย A และ B จะครบวงจรกันโดยผ่านทางเครื่องโทรศัพท์ จะทำให้ Voltage ที่ส่งไปยัง LI ตกลงไป (ไม่ถึง 48V.) เมื่อเข้าไปใน Op-Amp ทำการ compare แล้ว จะได้เป็น Logic 1 ไปเข้ายัง Interface part ต่อไป ในทุก ๆ 10 MS CPU (Processor) จะไปอ่านค่า (สภาวะ) ใน Interface part แล้วนำค่าที่อ่านได้ไปเปรียบเทียบกับค่าเดิมที่อ่านไปครั้งล่าสุดที่เก็บไว้ใน Substage CPU ก็จะทำให้รู้ว่ามีคนวางหูหรือยกหูโทรศัพท์หรือไม่

ไม่ว่ากรณีใดๆทั้งสิ้น อีจันท์หาบิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ KR (Key Set Receiver)



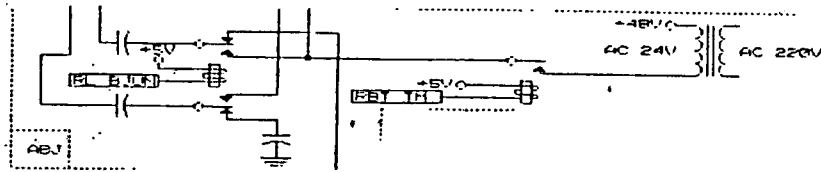
รูปที่ 14. Key Set Receiver Device

ประกอบด้วย :-

- 1/2 556 ผลิต freq. เพื่อเป็น Dial tone สำหรับส่งให้ผู้เช่า
- MT 8870 ทำหน้าที่เป็น filter freq. จากการกดเรียกจาก Key Set ของเครื่องโทรศัณท์ เมื่อไว้สำหรับรับตัวเลข 10 ตัว และเครื่องหมาย * # จากเครื่องรับโทรศัณท์หน้าทีของ KR
- ส่ง Dial tone ให้ผู้เรียก
- รับเลขหมายและบริการมีเดย์จากเครื่องรับโทรศัณท์ส่งให้ Processor การทำงาน

เมื่อ CPU อ่านค่าใน LI รู้ว่ามีผู้ยกหูโทรศัณท์และวิเคราะห์แล้วรู้ว่าเบอร์ไหนต้องการต่อผู้อื่น CPU จะตรวจจุดสภาวะของ KR Device ว่าว่างหรือไม่ ถ้า KR ว่าง CPU จะส่งสัญญาณไปทางสาย RL KR- จะเป็น KR- เท่าไรนั้นขึ้นอยู่กับว่าเบอร์ไหนยกหูโทรศัณท์ เมื่อต้องการต่อผู้อื่น CPU ก็ จะส่งสัญญาณไปยัง RL- KR- นั้น เพื่อให้ Relay สำหรับต่อ KR ของเบอร์นั้น operate เพื่อต่อเบอร์นั้นเข้ากับ KR Device ใน KR Device มี 1/2 556 ผลิต freq ออกมาอยู่ตลอดเวลา ดังนั้นเมื่อผู้เช่าถูกต่อเข้า KR Device ก็จะได้รับ Dial tone ทันที เมื่อผู้เช่าได้รับ Dial tone ก็จะกด Key set ตามเลขหมายที่ต้องการก็จะมี freq. 2 freq. (ครูป 10 ประกอบ) ส่งเข้ามา KR-Device ที่ 8870 8870 จะทำหน้าที่ filter เอา freq. นั้นส่งเข้า CPU ที่ 8155 เพื่อให้ CPU รู้เลขหมายของเบอร์ที่ต้องการเรียกเพื่อเอาไปดำเนินการต่อไป

อุปกรณ์ ABJ (A B Junctor)



รูปที่ 15. ABj Device

บางที่เรียกว่า Cord Line Relay Set ประกอบด้วย:-

- Relay change over contacts สำหรับต่อ Ring back tone ให้ฝ่ายเรียก และต่อให้ผู้เช่าทั้งสองฝ่ายพูดกันได้
- Relay สำหรับส่ง Ringing current ให้ฝ่ายถูกเรียก
- 24 V. AC สำหรับไปสั่นกระดิ่ง (หรือ Buzzer) ของฝ่ายถูกเรียก

หน้าที่ของ ABj

- ส่ง Ringing signal ไปยังฝ่ายถูกเรียก
- ส่ง Ringing control tone ไปยังฝ่ายเรียก
- ตัด Ringing control tone และ Ringing signal ออกเมื่อฝ่ายถูกเรียก ยกหูรับ
- ต่อ Speech part ให้ทั้งสองฝ่ายพูดกันได้

การทำงาน

เมื่อ CPU ตรวจสอบภาวะของฝ่ายถูกเรียกรู้ว่าว่างแล้ว CPU จะดำเนินการต่าง ๆ ดังนี้:-

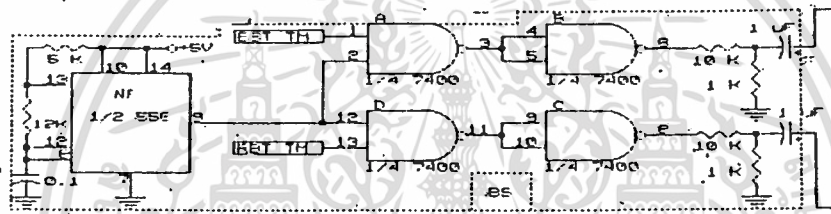
- ต่อผู้เรียกเข้ากับ Aj ของ AB junctor โดยส่งสัญญาณไปทาง RL-Aj- จะเป็น Aj- เท่าไรขึ้นอยู่กับผู้เช่าเบอร์ไหนเป็นฝ่ายเรียก
- ต่อผู้ถูกเรียกเข้ากับ Bj ของ AB junctor โดยส่งสัญญาณไปทาง RL- Bj- จะเป็น Bj- เท่าไรขึ้นอยู่กับเบอร์ไหนเป็นผู้ถูกเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่งสัญญาณไปทาง RBT TM สัญญาณนี้แยกออกไป 2 ทาง ทางที่หนึ่ง เข้ามาที่ ABj เมื่อบังคับให้ Relay ตัวขวามือทำงาน เมื่อต่อวงจรส่งไฟ 24 VAC ไปสั่นกระดิ่งหรือ Buzzer ของฝ่ายถูกเรียก ทางที่สอง เข้าสู่ BS เพื่อเป็น Time control ทำให้สัญญาณ Ring back tone จาก OSC (1/2 556) ส่งผ่าน Aj ของ ABj ไปยังผู้เข้าสายเรียก
- เมื่อผู้เข้าสายถูกเรียกยกหูรับ CPU จะส่งสัญญาณมาจาก RL BJUN บังคับให้ Relay ตัวซ้ายมือทำงาน เพื่อต่อ Speech part ให้ทั้งสองพูดกันได้

ดำเนินการตัด RBT TM เพื่อตัด Ringing current และ Ring back tone

อุปกรณ์ BS (Busy Tone Sender)



รูปที่ 16. Busy Tone Sender Device

ประกอบด้วย :-

- 1/2 556 เป็น OSC ผลิต freq. สำหรับส่งเป็น Busy tone
- 7400 2 ชุด ชุดบนสำหรับ control การส่ง Busy tone ให้มีการตัดต่อเป็นช่วง ๆ ชุดล่าง control การส่ง Ring back tone ให้ผู้เข้าสายเรียกให้มีการตัดต่อเป็นช่วง ๆ

การทำงาน

ปกติ 1/2 556 จะผลิต freq. ถ้า CPU ดำเนินการต่อผู้เข้าเบอร์ใดก็ตามไม่สำเร็จ อาจจะเป็นผู้เข้าสายกหูอยู่ตลอดเวลาเพื่อเรียกเบอร์อื่นแต่ขณะนั้น KR ไม่ว่าง หรือเมื่อต้องการต่อ ABj แต่ ABj หรือต้องการต่อเบอร์ใดแล้วเบอร์นั้นไม่ว่างก็ตาม CPU จะส่งสัญญาณออกมาทาง BST TM เมื่อเปิด gate ให้สัญญาณ Busy tone จาก 1/2 556 ส่งไปยังผู้เข้าเบอร์ที่ต้องการ ต่อในครั้งนั้น ได้รับ Busy tone เมื่อแจ้งให้ทราบว่า การต่อครั้งนั้นไม่สามารถดำเนินการให้สำเร็จตามประสงค์ได้ หรือในกรณีที่กำลังพูดกันอยู่แล้วฝ่ายใดฝ่ายหนึ่งวางหูก่อน CPU ต้องดำเนินการต่อ BS ให้ฝ่ายตรงข้ามได้รับ Busy tone ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. รูปแบบของ CPU

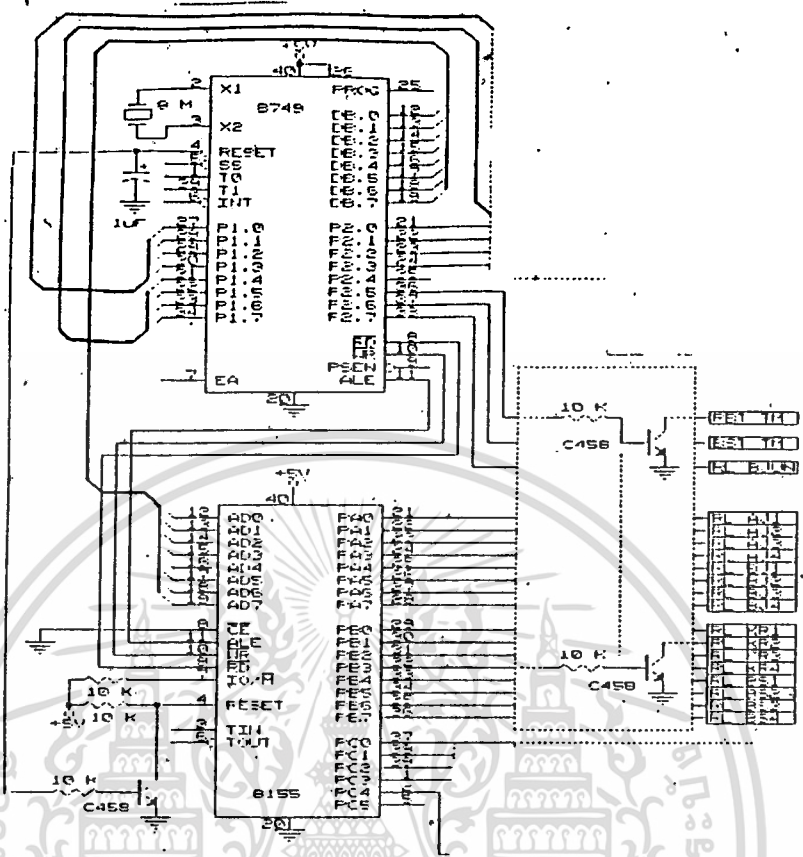
ชมสายตัวอย่างที่ใช้ IC 8749 แต่ 8749 จำนวน Part ไม่พอใช้ จึงใช้ 8155
เข้ามาช่วยเพื่อเพิ่มจำนวน Part ให้มากขึ้น โดยใช้

8749

- P1 สำหรับควบคุมการ Step meter ของผู้เช่า
- P2 - ใช้รับสภาวะการรบกวนหรือวางหูของเครื่องโทรศัพท์
 - Control การต่อ Ring back tone, Ringing tone
 - " " Busy tone
 - " " Speech part ให้ผู้เช่าพูดกันและตัด Ringing tone
Ring back tone

8155

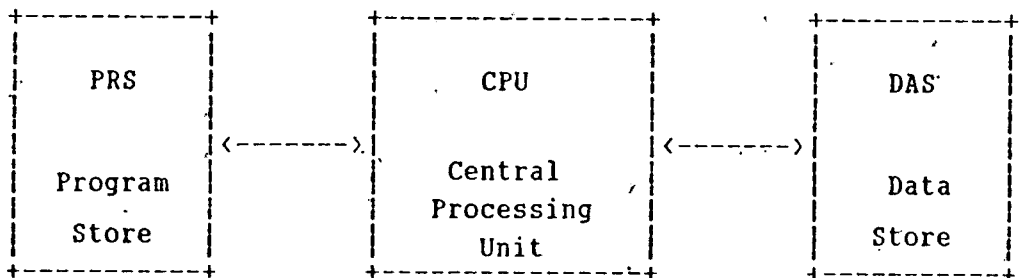
- PA0 - PA3 สำหรับ control การ operate Aj1 - Aj4 เพื่อต่อผู้เช่า
ฝ่ายเรียกเข้ากับ Aj ของ ABj
- PA4 - PA7 สำหรับ control การ operate Bj1 - Bj4 เพื่อต่อผู้เช่า
ฝ่ายถูกเรียกเข้ากับ Bj ของ ABj
- PB0 - PB3 สำหรับ control การ operate KR1 - KR4 เพื่อต่อผู้เช่า
เข้ากับ KR Device เพื่อส่ง Dial tone ให้ผู้เช่า
- PB4 - PB7 สำหรับ control การ operate BS1 - BS4 เพื่อต่อผู้เช่า
เข้ากับ BS Device เพื่อส่ง Busy tone ให้ผู้เช่า
- PC0 - PC3 สำหรับรับเลขหมายจาก KR Device จากการกด Key set ของผู้เช่า



รูปที่ 17. CPU Part

Processor ซึ่งใช้ควบคุมการต่อทั้งหมด จะต้องประกอบไปด้วยส่วนสำคัญ 3 ส่วน คือ

- Control Processor Unit (CPU)
- Program Store (PRS)
- Data Store (DAS)



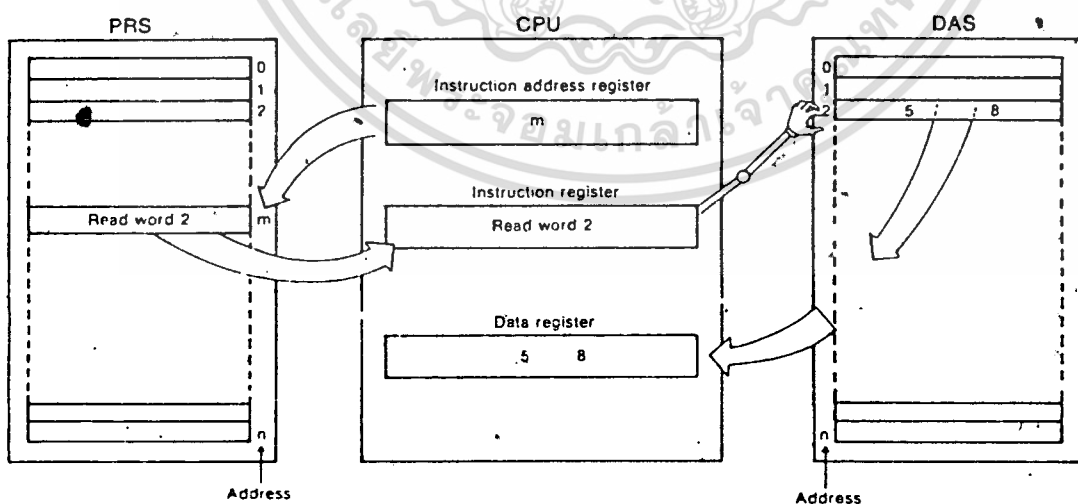
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งรูปที่ 18. ส่วนประกอบสำคัญของ Processor

CPU เป็น Active unit ซึ่งดำเนินการต่าง ๆ เช่น ส่งถ่ายข้อมูล (Transferring data) เปลี่ยนแปลงข้อมูล (Changing data) และทำการตัดสินใจ ในการทำงานทั้งหมด PRS และ DAS เป็น Passive parts ของ Processor ที่ใช้ในการเก็บข้อมูล

- Program Store (PRS) บรรจุ Program ต่าง ๆ ที่บอก CPU ให้ทราบว่าต้องทำอะไร
- Data Store (DAS) บรรจุข้อมูลที่ Processor จะใช้ดำเนินการ ว่าทำอย่างไร

ความจริงเราสามารถใส่ Program และข้อมูลต่าง ๆ ไว้ในที่เก็บ (Store) เดียวกันได้ แต่ในการปฏิบัติมักจะใช้ที่เก็บแยกกัน อันจะทำให้ CPU สามารถอ่านคำสั่ง (Instruction) ออกจาก Program store ในเวลาเดียวกันกับที่มันอ่านหรือเขียนข้อมูลใน Data store ได้ ซึ่งเป็นผลให้ Processor ทำงานได้เร็วขึ้น

CPU จำต้องทำงานร่วมกับ Switching part แต่เราจะนำมากล่าวในตอนต่อไป ในตอนนี้เราจะพูดถึงการทำงานของ Processor ภายในตัวมันเอง

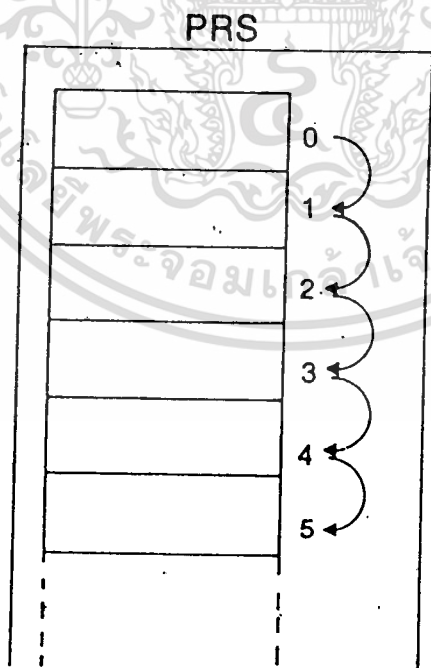


รูปที่ 19. การทำงานร่วมกันระหว่างส่วนทั้งสามของโมรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 19 แสดงตัวอย่างการทำงานร่วมกันระหว่างส่วนทั้งสามของ Processor ใน CPU มี Register ซึ่งเรียกว่า "Instruction-Address Register" สิ่งที่ยังอยู่ใน Register นี้ก็คือ Address ของ Word (คำสั่ง) ใน Program store เมื่อส่ง Address นี้ไปยัง PRS CPU จะได้รับคำสั่งที่ Address ดังกล่าวตอบมา ในตัวอย่าง Address ๓ เราพบคำสั่ง "Read word 2" คำสั่งนี้เวลาสั่งจริง ๆ ไม่ได้เป็นถ้อยคำ แต่เป็น Binary code คำสั่งที่อ่านมาได้จะนำมาเก็บไว้ใน Instruction Register CPU จะเอาคำสั่งมาแปล แล้วจัดการ control การอ่านข้อมูล (Word) ใน Data store จาก Address ที่ 2 ของ Data store ก็จะได้ข้อมูลที่อยู่ที่ Address ที่ 2 จาก Data store ซึ่งเป็นงานมาเก็บใน Data register เป็นการเสร็จสิ้นคำสั่งว่า Read word 2 ส่วนข้อมูลจะให้ทำอะไรนั้น CPU จะมีคำสั่งจัดการต่อไป

โดยทั่ว ๆ ไปนั้น CPU จะอ่านข้อมูลจาก PRS ที่ Address ที่เท่าไรก่อนนั้น ขึ้นอยู่กับ การที่จะทำงานในช่วงไหนและงานนั้น ๆ เริ่มต้นที่ Address ที่เท่าไร CPU จะต้องเริ่มที่ Address นั้น และเมื่อทำงานไหน Address นั้น เสร็จเรียบร้อยแล้วก็จะไปทำงานในคำสั่งถัด ๆ ไป ดังรูปที่ 20

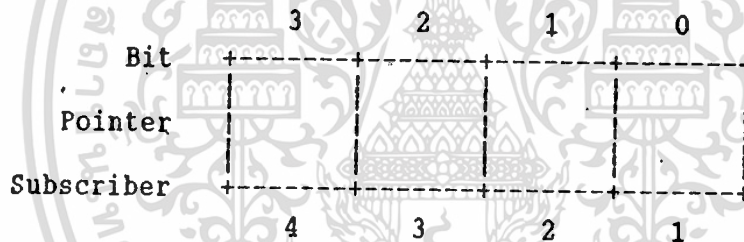


ยกเว้นจะไปเจอคำสั่งว่า Jump ก็ Jump ไปยัง Address ที่กำหนด เช่น CPU ได้รับคำสั่งให้ทำงานชิ้นหนึ่ง งานชิ้นเริ่มที่ Address 2 ใน PRS ดังนั้น CPU ก็จะเริ่มปฏิบัติงานที่ Address 2 เมื่อเสร็จแล้วก็จะไปทำงานที่ Address 3.....ไปเรื่อย ๆ ยกเว้นถ้าไปเจอคำสั่ง Jump ก็ Jump ไป Address นั้น ต่อไป

Data Structure

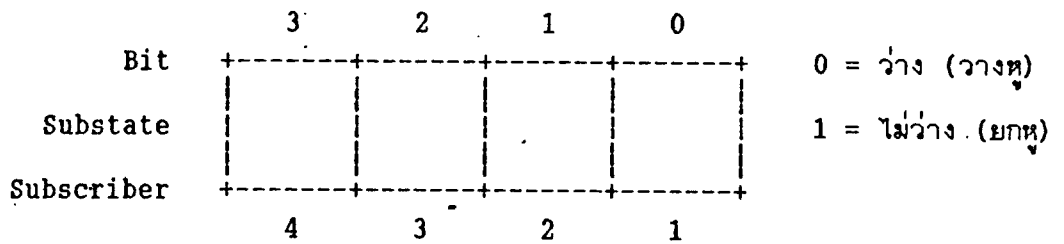
ต่อไปนี้เป็น Data struction สำหรับซุ่มสายตัวอย่าง โดยใช้ bit 0 ถึง 3 แทนผู้เช่าเลขหมาย 0 ถึง 3 ดังนั้นจะใช้ bit 0 - 3 สำหรับ Data word อื่น ๆ เท่าที่จำเป็น ดังนี้:-

Word Pointer สำหรับบ่งชี้ว่าขณะนี้จะต้องจัดการทำงานให้เลขหมายใด ๆ



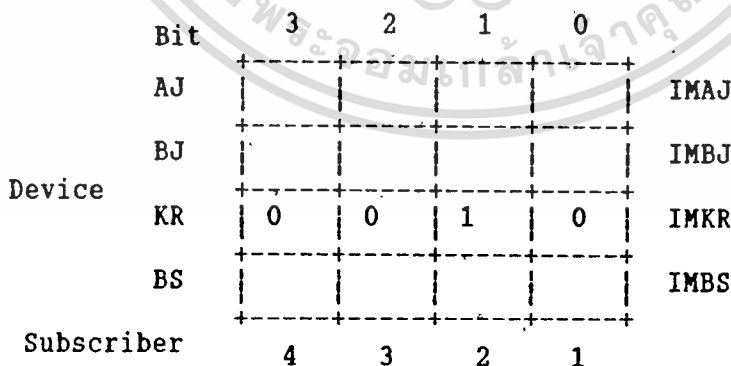
รูปที่ 21. Pointer

Word Substate เป็นตัวจดจำสถานะ Cradle state ของผู้เช่าก่อนหน้านี้ นั่นก็คือการจำสถานะการยกหูหรือวางหูของผู้เช่าก่อนหน้านี้ช่วงปัจจุบันนี้ สถานะนั้นคือ วางหู = 0, ยกหู = 1 สถานะครั้งสุดท้ายก่อนหน้านี้อาจจะถูกเก็บข้อมูลใน Substate



รูปที่ 22. Substate

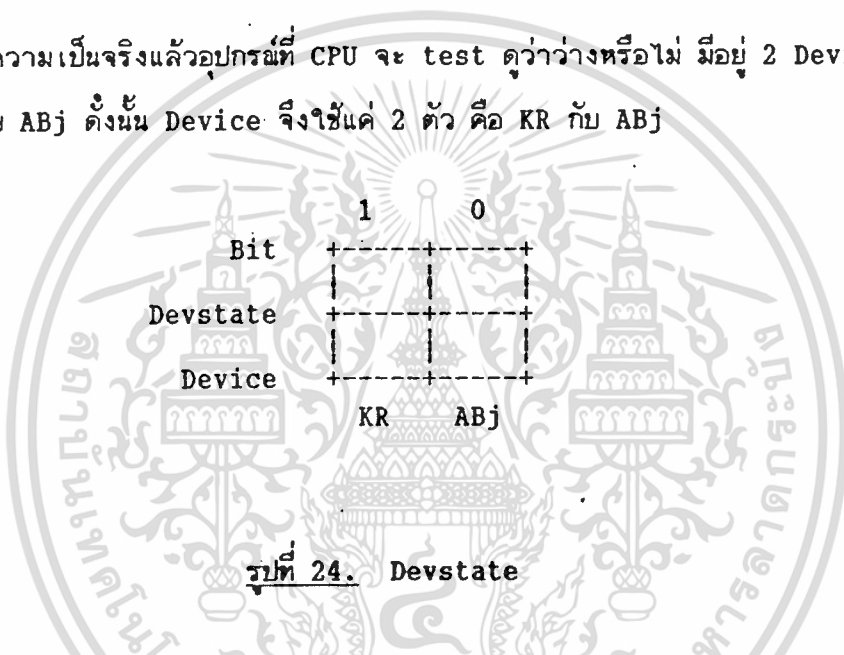
Devices ในแต่ละ Switching point ใน Switching state ต้องการหนึ่ง Bit เพื่อ Mark ว่า Switching point นั้นว่าง (=0) หรือไม่ว่าง (=1) เพื่อให้ CPU ทราบสถานะของ Switching state นั้น ๆ ดังนั้น 16 Switching points จึงต้องการ 16 Bits เพื่อให้ CPU ทราบสถานะทาง Switching part ว่าอุปกรณ์นั้น ๆ ถูกใช้งานอยู่หรือเปล่า จึงแบ่ง Bit ต่าง ๆ ออกเป็น 4 Word Word ละ 4 Bit โดยให้ตำแหน่งของ Bit ตรงกับเลขหมายผู้เช่าและเบอร์ ดังรูปที่ 21 เป็นรูปที่แสดงถึงการที่อุปกรณ์โทรศัพท์ (Telephone Device) อันใดว่าง (=0) หรือไม่ว่าง (=1) เราใช้ 1 Bit ต่อ หนึ่งอุปกรณ์ สำหรับ BS นั้น ไม่ต้องใช้ Bit ใด ๆ แสดงเพราะผู้เช่าทุก ๆ เบอร์สามารถต่อชานกันเข้า BS ได้พร้อม ๆ กัน ถึงแม้ BS จะไม่ว่างก็ตาม



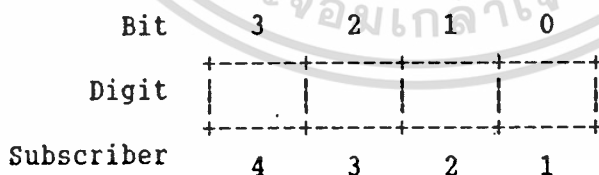
รูปที่ 23. Word สำหรับ Telephone Devices

เวลา CPU ดำเนินการต่ออุปกรณ์ใดไปแล้วจะ Mark อุปกรณ์นั้นให้มีสถานะเท่ากับไม่ว่าง เช่นถ้า CPU ดำเนินการต่อ KR 1ให้กับเบอร์ใดไป เช่นในตัวอย่างมี 1 อยู่ที่ Bit 1 แสดงว่า CPU ต่อ KR เข้ากับผู้เช่าเบอร์ 1 แล้ว ถ้าทราบใดที่เบอร์ 1 ยังไม่เลิกใช้ KR อยู่ ถ้ามีผู้เช่าเบอร์อื่นต้องการใช้ KR อีก (ผู้เช่ายกหูต้องการเรียกเบอร์อื่น) เมื่อ CPU test ดู KR CPU จะรู้ทันทีว่า KR ไม่ว่าง (ถูกเบอร์ 1 ใช้อยู่)

ในความเป็นจริงแล้วอุปกรณ์ที่ CPU จะ test ดูว่าว่างหรือไม่ มีอยู่ 2 Device เท่านั้นคือ KR กับ ABj ดังนั้น Device จึงใช้แค่ 2 ตัว คือ KR กับ ABj



Digit เป็นตัวเก็บเลขหมาย KR ที่รับมาจากการด Key set ของผู้เช่า



รูปที่ 25. Digit

เมื่อรอให้ Processor มาอ่านไปต่อไป

Anyjob : Processor จะต้องทราบว่าผู้เข้ามีการเปลี่ยนแปลงสภาวะหรือไม่ เช่น เปลี่ยนแปลงจากว่าง เป็นไม่ว่าง คือ จากว่างอยู่เป็นยกหูเพื่อทำการเรียกเบอร์อื่น (หรือยกหูเพื่อตอบรับ) หรือจากไม่ว่างเป็นว่าง คือ จากการยกหูอยู่เป็นวางหู (Clearing) การเปลี่ยนแปลงนี้ Processor จะทราบได้โดยการเอาข้อมูลทีอ่านมาได้จาก LI ในปัจจุบัน เปรียบเทียบกับข้อมูลเดิม (ครั้งสุดท้ายก่อนหน้านี้) ที่เก็บอยู่ใน Substate การเปรียบเทียบทำดังนี้:-



รูปที่ 26. วิธีการ เปรียบเทียบข้อมูล

วิธีการ เปรียบเทียบกันนั้น Processor จะเอาข้อมูลทั้ง 2 มาทำการ Exclusive

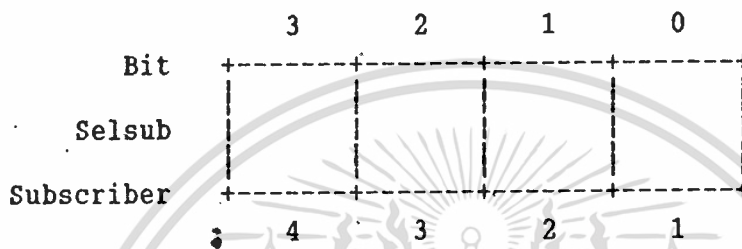
OR กัน เช่น

ผลจากการอ่านจาก LI	0 1 0 1
ข้อมูลเดิมทีเก็บอยู่ใน Substate	<u>1 1 0 1</u>
การเปลี่ยนแปลง	<u>1 0 0 1</u>

ผลการ เปรียบเทียบแสดงว่ามีการเปลี่ยนแปลงที่หมายเลข 1 จากการยกหูอยู่เป็นวางหู เบอร์ 4 จากการวางหูอยู่เป็นยกหู แสดงว่า Processor ต้องทำงานให้แก่เบอร์ 2 เบอร์นี้ ตาม ประสงค์ต่อไป

ผลที่ Processor เอาข้อมูลทั้ง 2 นี้มาเปรียบเทียบกัน ได้ผลออกมาอย่างไร จะถูกนำไปเก็บใน Anyjob เพื่อรอให้ Processor นำไปดำเนินการต่อไป

Selsub (Selected Subscriber)



รูปที่ 27. Selsub

การเปลี่ยนแปลงจากว่าง เป็นไม่ว่าง สำหรับผู้เช่าอาจหมายความว่าผู้เช่าโทรศัพท์ยกหูโทรศัพท์ขึ้นเพื่อทำการเรียก (Calling Signal) หรือผู้เช่ายกหูเพื่อตอบรับสัญญาณเรียก (Ringing หรือ B Answer Signal) ก็ได้ เพื่อให้ Processor สามารถจำแนกได้ว่าเป็นแบบไหน เราจึงใช้ Word Selsub (Selected Subscriber) เพื่อเป็นตัวบ่งชี้ว่าถ้าผู้เช่าเบอร์ใดได้ถูกต่อเข้ากับ Bj ของ ABJ และได้รับ Ringing signal ผู้เช่าเบอร์นั้นจะถูก Mark ไว้ใน Selsub เพื่อแสดงให้ Processor ทราบว่าถ้าเขายกหูขึ้น แสดงว่ายกหูเพื่อตอบรับ

Data word ต่าง ๆ ที่กล่าวมาทั้งหมดนี้ ถูกเก็บอยู่ใน Data store ที่ Address ต่าง ๆ (0 - 9) ดังรูปที่ 28.

Data structure in DAS

Function	Data name	Bit				Word address	
		3	2	1	0		
0 = idle 1 = busy	Image of Oper. Word 0					0	
	Image of Oper. Word 1					1	
	Image of Oper. Word 2					2	
	Image of Oper. Word 3					3	
Subscriber state	POINTER					4	
	SUBSTATE					5	
	ANYJOB					6	
	DIGIT					7	
	SELSUB					8	
	Selected B-sub. Device state	DEVSTATE					9
					KR	ABJ	

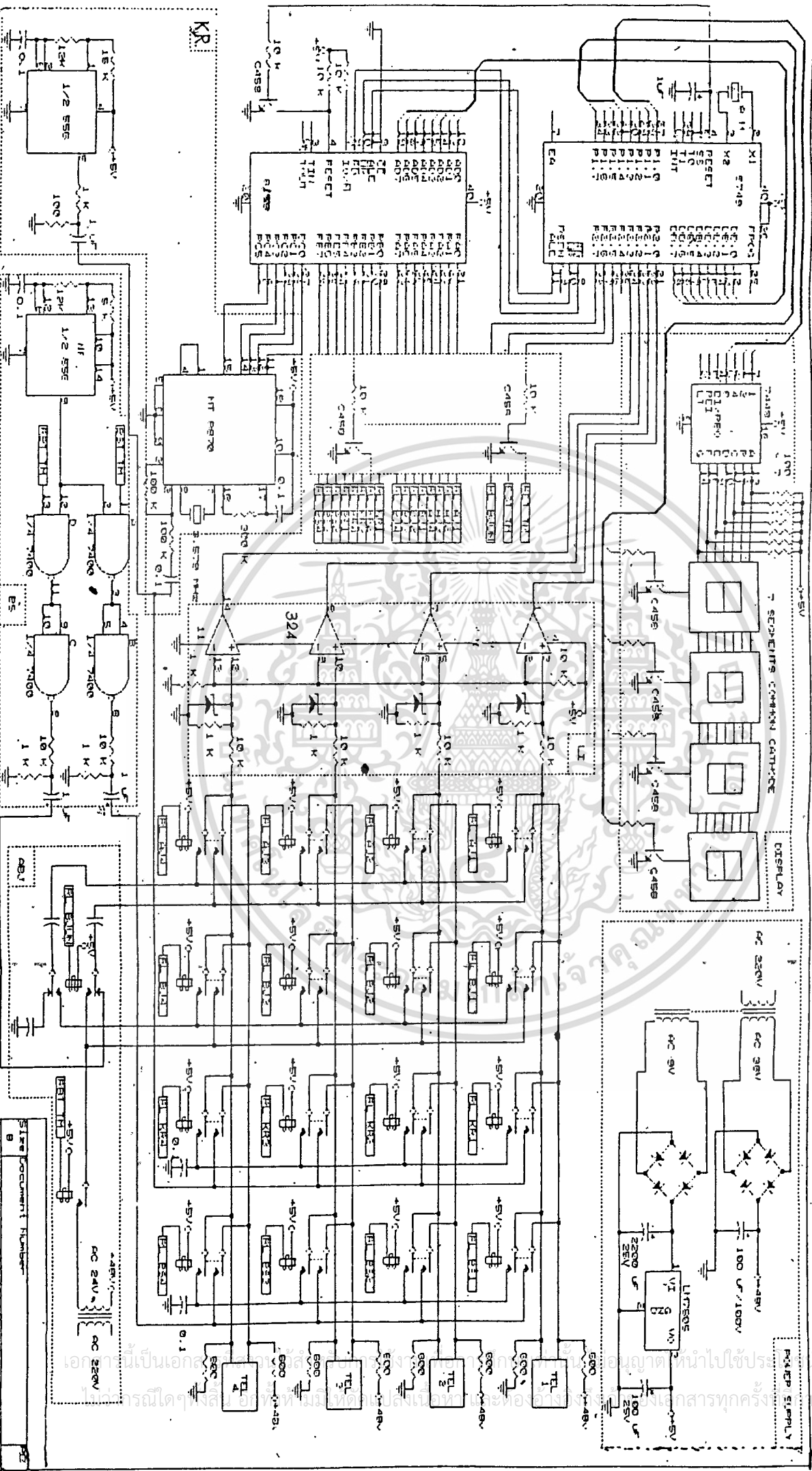
รูปที่ 28. Data Structure DAS

ขอให้เราทบทวนหน้าที่ของ Data words อีกครั้งหนึ่ง :-

- Word 0 ถึง 3 ใน Data store ตรงกับ 4 - Operation words สำหรับควบคุมให้ Switching relays ทำงาน Word 0 ตรงกับคอลัมน์ของ Relay 4 ตัวที่ต่อด้าน A ของ ABj เข้ากับผู้เช่า Bit 0 ถึง 3 ใน Word 0 ตรงกับ Relay ที่ต่อกับผู้เช่า เลขหมาย 0 ถึง 3 Word นี้ต้องจัดหาภาพ (Image) ของภาวะสำหรับ Relay เหล่านี้ ดังนั้นเราจึงเรียก Word ดังกล่าวว่าเป็น IMAGE ของ Operation word Word 1 เป็นภาพของด้าน B ของ ABj Word 2 เป็นภาพของการต่อ KR และ Word 3 จัดหาภาพของ Relay สำหรับการต่อของ BS
- Word 4, POINTER ใช้เพื่อบ่งชี้ผู้เช่าใดที่ต้องจัดการ ณ เวลานั้น
- Word 5, SUBSTATE ใช้สำหรับกับภาวะที่มียุก่อนของผู้เช่า
- Word 6, ANYJOB ใช้เก็บผลลัพธ์ที่ได้จากการเปรียบเทียบระหว่าง Test answer จาก LI และ Word 5
- Word 7, DIGIT ใช้สำหรับเก็บเลขหมายที่ได้รับจาก KR
- Word 8, SELSUB บอกให้เราทราบว่าผู้เช่าใดถูกต้องเข้ากับผู้เช่า B ในฐานะเป็นผู้เช่า B ก่อนที่ผู้เช่าจะยกหูตอบรับ เขาได้ถูกหมายให้ว่างใน SUBSTATE เมื่อเขาตอบรับมา เราต้องไม่แปลความว่าเป็นการเรียก แต่ต้องแปลว่า B ตอบรับ
- Word 9, DEVSTATE หมายถึงรู้ว่า ABJ หรือ KR ว่างหรือไม่ว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าใน Words 0 - 8 Bit ต่าง ๆ ตรงกับผู้เช่าทั้งสี่เลขหมายไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. Block Diagram
 7-SEGMENTS COMMON CATHODE DISPLAY
 5-VUU STORED PROGRAM CONTROL (SPC)



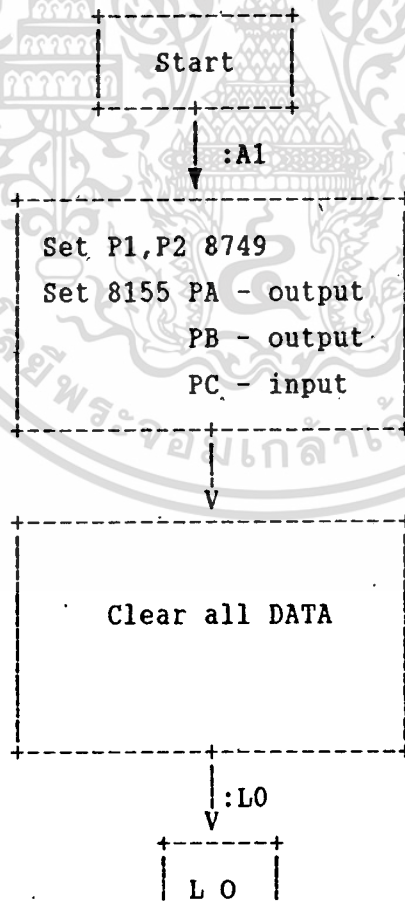
Block Diagram ของขุมสาย Electronic ระบบ SPC (ตัวอย่าง) ดังรูปที่ 29

การจะดูการทำงานของขุมสายโทรทัศน์ทำให้เข้าใจนั้นจะต้องดู Trunking diagram ประกอบกับ Flow chart และจะยิ่งเข้าใจยิ่งขึ้นถ้าท่านสามารถเข้าใจ Program ต่าง ๆ ของขุมสายด้วย

ต่อไปนี้จะยกตัวอย่างการทำงานของขุมสายโดยใช้ Trunking diagram ประกอบกับ Flow chart และจะแนะนำเพียงให้รู้ว่า ณ จุดนั้น ๆ ของ Flow chart ใช้ Program อะไรพอเป็นตัวอย่างให้เข้าใจ

ใน Trunking diagram จะเห็นว่าภาค CPU ใช้ 8749 แต่ 8749 ตัวเดียวมีจำนวน Port ไม่พอใช้ จึงเอาก 8155 มาช่วยอีกตัวหนึ่ง ดังเคยกล่าวมาแล้วใน ภาค CPU เมื่อประกอบเป็นขุมสายดัง Trunking diagram เสร็จเรียบร้อยแล้ว ก่อนจะเอาขุมสายนี้มาใช้งานให้ได้ประโยชน์สมบูรณ์ จะต้องดำเนินการคือ (ดู Flow chart)

10. ตัวอย่าง Flow Chart และหลักการทำงาน



รูปที่ 30. A ตัวอย่าง Flow Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า - ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Set ค่า P1, P2 ของ 8749 เนื่องจากในขณะที่ทำการประกอบนั้นอาจจะมีค่าหนึ่งค่าใดปรากฏอยู่ใน 8749 เพื่อให้ปฏิบัติงานจริงอาจจะมีการผิดพลาดเกิดขึ้นได้ จึงต้อง Set ค่า P1, P2 ให้มีค่า = 0 (ศูนย์) คืออยู่ในสภาวะว่าง

Set PA, PB, PC ของ 8155 กำหนดให้ PA, PB เป็น Output ไป Control การทำงานของ Relay ใน Switching part

PC เป็น Input สำหรับรับเลขหมายจาก KR เราต้อง Set ค่าของ PA, PB, PC ให้ = 0 (ศูนย์) เพื่อให้ทั้งหมดอยู่ในสภาวะว่าง

Clear all Data คือการ clear program ต่าง ๆ ใน Data store ให้มีค่าเป็น = 0 ทั้งหมด

การ Set ค่า P1, P2 ใน Program ซึ่งอยู่ด้านหลังนั้นจะเริ่มตั้งแต่ลำดับที่ 41 - 43 (Initial)

Set 8155 เริ่มลำดับที่ 44 - 51

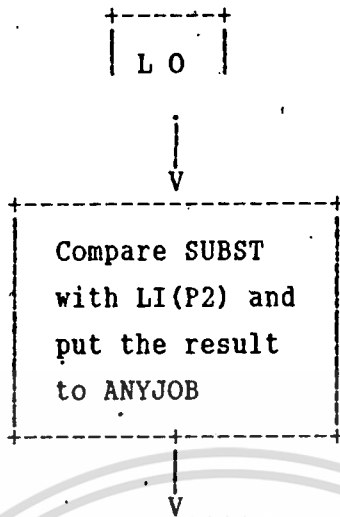
Clear data store เริ่มลำดับที่ 52 - 96

เมื่อ Set 8749 และ 8155 เสร็จเรียบร้อยแล้ว ชุมสายโทรศัพท์ตัวอย่างนี้ก็จะพร้อมปฏิบัติงานได้สมบูรณ์ต่อไป

ต่อไปนี้เป็นตัวอย่างการทำงานของชุมสาย จะเริ่มที่ L0 (ดู Flow chart ประกอบ) จาก Flow chart จะเห็นว่าเขียนไว้ว่า Compare Subst with LI (P2) and put the result to, Anyjob ให้เอาค่าที่บันทึกไว้ใน Substate มาเปรียบเทียบกับค่าที่อ่านได้จาก LI ได้ผลลัพธ์ออกมา เอาผลลัพธ์นั้นไปเก็บไว้ใน Anyjob จากตาราง Program จะเห็นว่าใช้ Program ลำดับที่ 97 หรือ Add ที่ 005E OA ของ CPU ถึงลำดับที่ 102 Add 0066A1

เป็นอันเสร็จคำสั่งตาม Flow chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

97  005E  0A      LO      IN A,P2          ; ANYJOB = SUBST (EXC) LI(P2)
98  005F  53 0F      ANL A,#00FH
99  0061  B8 25      MOV R0,#SUBST
100 0063  B9 26      MOV R1,#ANYJOB
101 0065  D0          XRL A,@R0
102 0066  A1          MOV @R1,A
  
```

รูปที่ 30. B

หลักการทางาน (ดู Trunking Diagram ประกอบ)

ขุมสายนี้เรากำหนดได้ทุก 10 MS CPU จะต้องไปอ่านข้อมูลใน LI เพื่อตรวจดูว่ามีเบอร์ยกหูหรือมีเบอร์ใดวางหูบ้าง แล้วไปอ่านใน KR ว่ามีคดเลขหมาย (Key set) บ้าง เพื่อจะเอางานนั้น ๆ มาปฏิบัติการให้เป็นไปตามมุ่งหมายของผู้เรียก

๗ ตำแหน่ง LO ใน Flow chart เป็นจุดกำหนดให้ CPU ไปอ่านข้อมูลมาจาก LI แล้วนำไปเปรียบเทียบกับข้อมูลที่เก็บไว้ใน Substate เราจะเห็นว่าตอนที่เรามา start เครื่อง เราได้ทำการ clear ข้อมูลต่าง ๆ ใน Data หมดแล้ว เพราะฉะนั้นขณะนี้ Substate จะถูก clear มีค่าเป็น 0

Substate = 0 0 0 0

สมมติ CPU ไปอ่านค่าใน LI ได้ = 0 0 0 1

นำมา Exclusive OR กันได้ = 0 0 0 1

นำเอาค่า 0 0 0 1 คือผลลัพธ์ที่ได้ไปเก็บไว้ใน Anyjob เป็นอันเสร็จงานใน Flow chart ตัวอย่างนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปเป็นการ Set pointer = 0001 เพื่อให้ CPU ไปตรวจดูว่าใน Anyjob เลขหมาย No.1 มีงานที่ CPU จะต้องปฏิบัติการให้หรือไม่ (Is the indicate bit in Anyjob = 1 ?) Program การทำงานนี้อยู่ลำดับที่ 105 - 109 ถ้าไม่ (No) จะไปเริ่มที่ลำดับที่ 147 ถ้าใช่คือหมายเลข 1 มีงานให้ทำ Program จะถูก run ไปเรื่อย ๆ ตอนแรกนี้ จะสมมติว่าหมายเลข 1 ไม่มีงานให้ทำ ตาม Flow chart จะไปเริ่มที่ L2 (ต่อไปนี้คิดว่าท่าน คงทราบแล้วว่าการทำงานแต่ละจุดต้องใช้ Program ลำดับที่เท่าไรถึงลำดับที่เท่าไร ดังนั้น ต่อไปจะไม่อ้าง Program ให้ดูอีก ให้ดู Program สมบูรณ์เอง ซึ่งมีอยู่ทางท้ายเล่มนี้แล้ว ส่วน Flow chart ก็เช่นกัน มี Flow chart สมบูรณ์อยู่แล้ว จะไม่ทำ Flow chart แต่ละจุด ให้ดูให้ไปดู Flow chart สมบูรณ์เอาเอง).

ข จุด L2 มีข้อความว่า Is bit 3 in Pointer = 1 ? Bit 3 ใน Point = 1 หรือไม่ คือ CPU จะตรวจดู Bit 3 ใน Pointer ว่า = 1 หรือยัง ถ้า = 1 แสดงว่า CPU ตรวจดูเลขหมายทุกเลขหมายแล้ว มีงานให้ทำหรือไม่ แล้ว CPU จะไปอ่านดูใน KR ว่ามีการรับ เลขหมายมาจากเครื่องโทรศัพท์ไหม ถ้า Bit 3 ใน Pointer ไม่เท่ากับ 1 แสดงว่า CPU ยังตรวจเลขหมายยังไม่ครบ CPU จะทำการคูณค่า Pointer ด้วย 2 (Pointer x 2) เพื่อเลื่อนค่า Pointer ไป 1 ตำแหน่ง เพื่อให้ CPU ไปตรวจดูเลขหมายถัดไป แล้ว CPU จะไปดูค่าใน Anyjob ใหม่ว่าเลขหมายที่ตรงกับค่า Point มีงานให้ทำหรือไม่

ในตัวอย่าง เราจะเห็นว่าผลลัพธ์ที่เราได้จากการ กั้นระหว่างข้อมูลเดิมจาก Substate กับข้อมูลที่อ่านมาใหม่จาก LI ได้ = 0001 แสดงว่าเดิมนั้น ตอนที่ clear ชุมสาย ยังไม่มีเบอร์โดยที่โทรศัพท์เลย แต่หลังจาก clear เสร็จ มีผู้เข้าเบอร์ 1 (ตรงกับ Bit 0) ยกโทรศัพท์ขึ้นเพราะผลลัพธ์ 0001 ค่านี้ถูกเก็บอยู่ใน Anyjob เมื่อ CPU ไปตรวจดูค่าใน Anyjob เจอค่า 0001 CPU จะรู้ทันทีว่า ๑ เบอร์ 1 มีงานให้ทำ คือ CPU รู้ว่าเบอร์ 1 ยกหู แต่ยังไม่รู้ว่า ยกหูเพื่อตอบรับเบอร์อื่นหรือยกหูเพื่อเอา Dial tone (ต้องการต่อผู้อื่น) ดังนั้น CPU จะต้องไปดู ค่าใน Subst ว่าเบอร์ 1 ที่เก็บอยู่ใน Subst เป็น 1 อยู่หรือไม่ (Is the indicated bit in SUBST = 1 ?) ถ้าไม่ใช้แสดงว่าแต่ก่อนนี้ไม่ได้ยกหูอยู่ แต่ปัจจุบันได้ยกหูขึ้นมา ถ้าไม่ต้องการเรียก ผู้อื่นก็ต้องยกหูตอบรับผู้อื่น เมื่อมีการยกหูขึ้นมาเช่นนี้ CPU จะต้องรีบ mark เบอร์ที่ยกหูนี้ให้มีค่าเท่ากับ 1 (Mark the subscriber busy) เพื่อกันไม่ให้เบอร์อื่นเรียกเบอร์ที่ยกหูนี้ แล้ว CPU จะ

ตรวจดูการยกหูขึ้นเป็นการยกหูในฐานะ B Subs คือตอบรับใช่ไหม (Is the subscriber chosen as B Subs) ถ้าใช่จะไปทาง LB ถ้าไม่ใช่ (No = call) ก็แสดงว่าผู้เขายกหูเพื่อเรียกผู้อื่น (ต้องการ Dial tone) CPU จะไปตรวจดูสภาพ KR ว่าว่างหรือไม่ (Is KR idle) ถ้า KR ไม่ว่าง (No) CPU จะ control เบอร์นี้ต่อเข้ากับ BS เพื่อส่ง Busy tone ให้ผู้ที่ยกหู ถ้า KR ว่าง (Yes = LI2) CPU จะ mark KR ให้มีค่าเท่ากับไม่ว่างเพื่อเป็นการจอง KR ไว้ แล้ว control การต่อผู้เข้าเข้ากับ KR เมื่อต่อได้แล้วต้องไป mark ที่ KR Device state ให้มีค่าเป็น 1 เพื่อแสดงสถานะให้รู้ว่า KR ไม่ว่าง เป็นอันเสร็จต่อผู้เข้าเข้ากับ KR ให้ KR ส่ง Dial tone ให้ผู้เข้าต่อไป

แต่ถ้าผู้ที่ยกหูขึ้นเป็นการยกหูเพื่อตอบเรียกจากผู้อื่น (Yes = B Answer = L13) CPU จะไป clear Selsub แล้วไปดำเนินการ step meter ของ Sub A (ไปดำเนินการงาน ณ Z2)

ในกรณีใน SUBST = 1 คือผู้เขายกหูโทรคืนที่อยู่แต่ปัจจุบันนี้ว่างหูลง (Yes) CPU จะไปดำเนินการงาน ณ L3 (Clearing) คือ Mark the subscriber free คือต้องทำให้เบอร์ที่ว่างหูลงอยู่ในสถานะว่าง เพื่อให้เบอร์อื่น ๆ สามารถเรียกเบอร์ที่ว่างหูลงได้ เสร็จแล้ว CPU จะมาตรวจดูว่าเบอร์ที่ว่างหูลงนี้เขาอยู่ในสถานะผู้เรียกหรือผู้ถูกเรียก (Does he use Aj) ถ้า Yes ใช่เขาใช้ Aj ก็แสดงว่าเขาเป็นผู้เรียกผู้อื่น CPU จะทำงานไปทาง L7 โดย จะตรวจดูว่า Sub B ยกหูหรือยัง ถ้ายัง (No) จะทำงานไปทาง L11 คือ clear Selsub แล้ว clear RBT, และ RAT เสร็จแล้วไปทำงานที่ L10 แต่ถ้า Sub B ยกหูรับแล้ว (Yes) จะดำเนินการต่อ BS ให้ Sub B ได้รับ Busy tone แล้วไปดำเนินการงานต่อที่ L10 คือ Mark path to Aj free Mark path to Bj free Mark ABj idle แล้วตัดผู้เข้าทั้งคู่ออกจากกัน นั่นคือทำให้ ABj กลับไปอยู่ในสถานะว่างเพื่อให้ผู้เขารายอื่น ๆ ได้มีโอกาสใช้ ABj ต่อไป เสร็จแล้ว CPU จะกลับไป L2 เพื่อตรวจดูสถานะเบอร์อื่นต่อไป

ในกรณี No (เขาไม่ได้ใช้ Aj อยู่) CPU จะตรวจดูว่าเขาต่อ Bj อยู่ใช่ไหม ถ้าใช่ (Yes) จะดำเนินการงานไปทาง L8 โดย Mark path A Sub to BS แล้วต่อ Sub A กับ BS (Connect A Sub to BS) เพื่อส่ง Busy tone ให้ Sub A แล้วไปดำเนินการงานที่ L10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณี No (เขาไม่ได้ใช้ B_j) CPU จะตรวจดูว่าเขาใช้ KR อยู่หรือไม่
 (Does he use KR?) ถ้าใช่ (Yes) CPU จะ Mark KR free Mark KR idle
 Disconnect the Sub from KR นั่นคือดำเนินการตัด KR ออกจากผู้เช่าเพื่อให้ KR
 กลับไปสู่สภาวะว่างอยู่ แล้ว CPU จะไปตรวจดูเบอร์อื่นต่อไป

ในกรณี No (เขาไม่ได้ใช้ KR) Mark path BS to BS free Disconnect
 the Sub from BS ตัดผู้เช่าออกจาก BS แล้วไปตรวจ เบอร์อื่นต่อไป

การอ่านค่าใน KR

ในทุก ๆ 10 MS เมื่อ CPU อ่านค่าใน LI เสร็จแล้ว CPU จะมาอ่านค่าใน KR
 เมื่อตรวจดูว่ามีผู้เช่าเบอร์ไหนบ้างถูกเรียก (Is there any digit?) ในกรณี Yes คือมี
 การกด Key set ส่งเลขหมายมา KR CPU จะนำเลขหมายที่อ่านได้ไปเก็บใน data store
 ที่ Word digit แล้ว CPU จะไปอ่านค่าที่ Word digit มาใหม่ มาตรวจดูว่าใน Word digit
 มีค่า = 0 หรือไม่ (Digit = 0?)

ในกรณี No แสดงว่ามีการส่งเลขหมายเข้ามา คือมีการ เรียกเบอร์ใดเบอร์หนึ่ง
 CPU จะเอา Sub A ใน Pointer แล้วดำเนินการทำให้ KR ว่างคือตัด Sub A ออกจาก KR
 เพื่อให้ผู้เช่ารายอื่น ๆ มีโอกาสใช้ KR ต่อไป

ต่อไป CPU จะตรวจดู Sub B ว่าว่างไหม (Is the B-Sub free?) ถ้าว่าง
 CPU จะไปตรวจดู AB_j ว่าว่างหรือไม่ (Is AB_j idle?) ถ้า Sub B หรือ AB_j ไม่ว่าง
 จะไปที่ L15 ไปดำเนินการต่อ BS เข้า Sub A เพื่อให้ผู้เช่าได้รับ Busy tone แล้วจึง
 clear Digit ในกรณี Sub B และ AB_j ว่าง จะไปที่ L16 คือ ดำเนินการต่อ Sub A
 เข้าทาง A_j ของ AB_j และต่อ Sub B เข้า B_j ของ AB_j แล้วบันทึก Sub B Number ไว้ใน
 Selsub แล้ว clear Digit เสร็จแล้วดำเนินการ start Ringing เพื่อส่ง Ringing tone
 ให้ Sub B และ Ring back tone ให้ Sub A

SUBROUTINEConvert Digit

เมื่อ CPU ไปอ่านค่ามาจาก KR แล้ว CPU จะตรวจดูว่ามีการกดเลขหมายเข้ามาหรือไม่ (มีการเรียกหรือเปล่า) โดย CPU จะตรวจดูว่า เลขหมาย = 1 หรือเปล่า ถ้าใช้ก็ดำเนินการต่อเลข 1 ถ้าไม่ใช่จะดูว่าเลข 2 ใช่ไหม ถ้าใช่ก็ดำเนินการต่อ ไปอย่างนี้เรื่อย ๆ จนกระทั่งถึงดูว่าเลขหมายเป็นเลข 4 หรือไม่ ถ้าใช่ก็ดำเนินการต่อเลข 4 ถ้าไม่ใช่แสดงว่าไม่มีการเรียกเข้ามา

B. Answer (Add Meter)

เป็นช่วงที่ Sub B ยกหูรับ CPU จะดำเนินการ Ringing tone และ Ring back tone ออก แล้วต่อ Sub B เข้า Bj ของ ABj เสร็จแล้ว CPU จะไปดู Sub A ว่าเป็นเลขหมายอะไร เพราะว่าเมื่อ Sub B ยกหูรับจะได้ Step meter ของเบอร์นั้น ตาม Flow chart CPU จะตรวจดูว่า Sub A เป็นเลข 1 ใช่ไหม ถ้าใช่ก็ดำเนินการ Step meter ของหมายเลข 1 ถ้าไม่ใช่จะไปดูว่าเป็นเลข 2 ใช่ไหม ถ้าใช่ก็ Step meter ของเลขหมาย 2 เป็นอย่างนี้ไปเรื่อย ๆ จนถึงเลข 4

B. No Answer (Clear RBT, RGT)

เมื่อส่ง Ringing ถึง Sub B แล้ว แต่ Sub B ไม่ยกหูรับ ต้องทำการตัด Ringing tone และ Ring back tone ออก คือ CPU จะไปดำเนินการให้ IMRBT ซึ่งเป็นตัว control การต่อ Ringing และ Ring back tone มีค่า = 0 เพื่อตัด RBT ออก

Start Ringing

หลังจากตรวจสอบสถานะทาง Sub B ว่าว่างแล้ว ก็จะเริ่ม start การส่งสัญญาณ Ring ไปยัง Sub B และ Ring back tone ไปยัง Sub A โดยจะไป start function การทำงานของการ check timing ของ RBT

Check Timing RBT

ตรวจวงจร Ring back tone ว่าว่างหรือไม่ การ set timing ของสัญญาณ Ring back tone เป็นอัตราส่วน การ ON : OFF เช่น จาก 2MS off 4MS ส่งผลไป start วงจรทำงาน ให้ส่งสัญญาณ RBT การทำงานจะเป็น loop ไปเรื่อย ๆ จนกว่า Sub B จะยกหุ้บ ก็หยุดการทำงาน

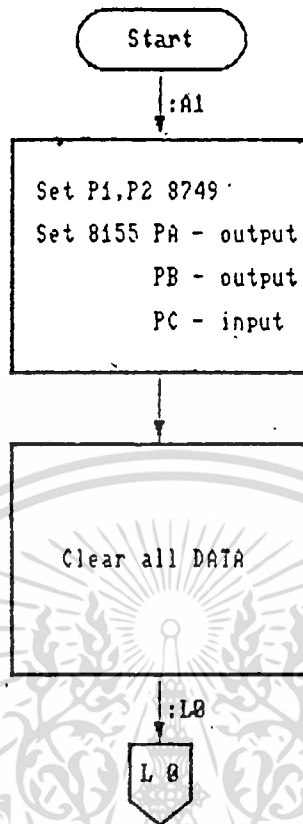
Check Timer BS

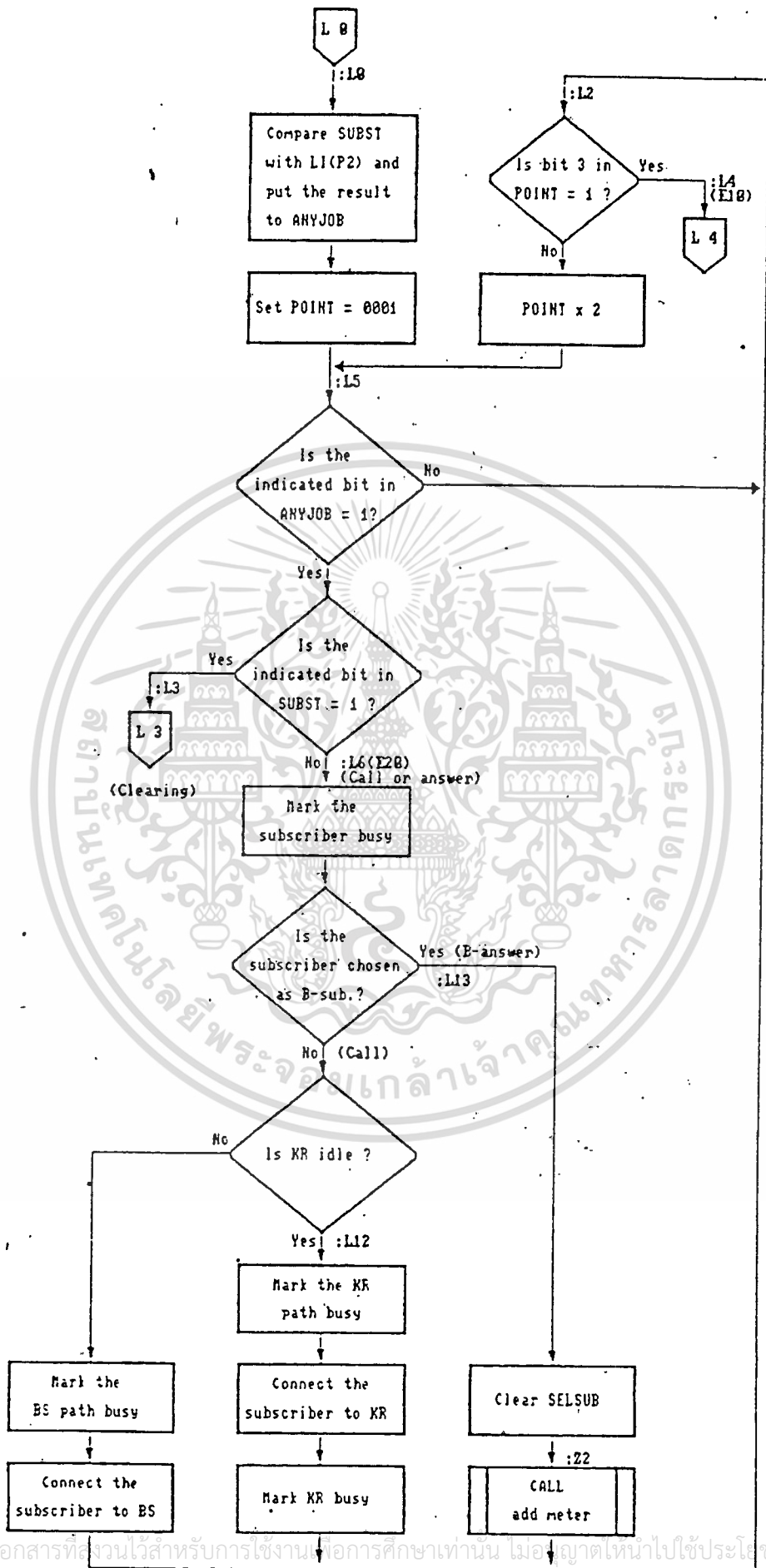
ในกรณีตรวจสอบทาง B-Sub แล้ว ปรากฏว่าไม่ว่าง ตรวจวงจร Busy tone ว่าว่างหรือไม่ ถ้าว่างเตรียม Set BST เพื่อให้ตัววงจรส่งสัญญาณ Busy tone ไปยัง A-Sub ต่อไป

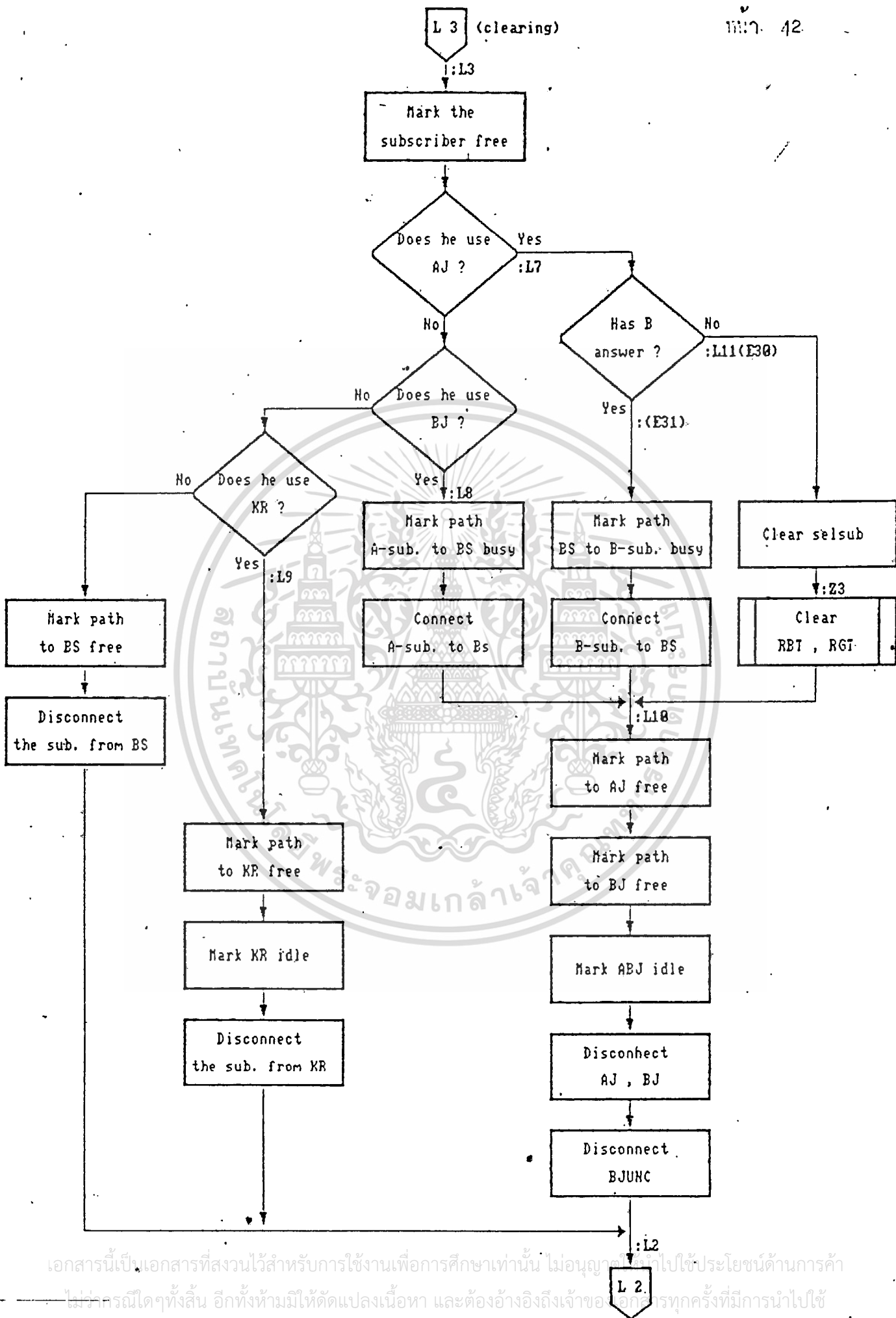
Display Meter

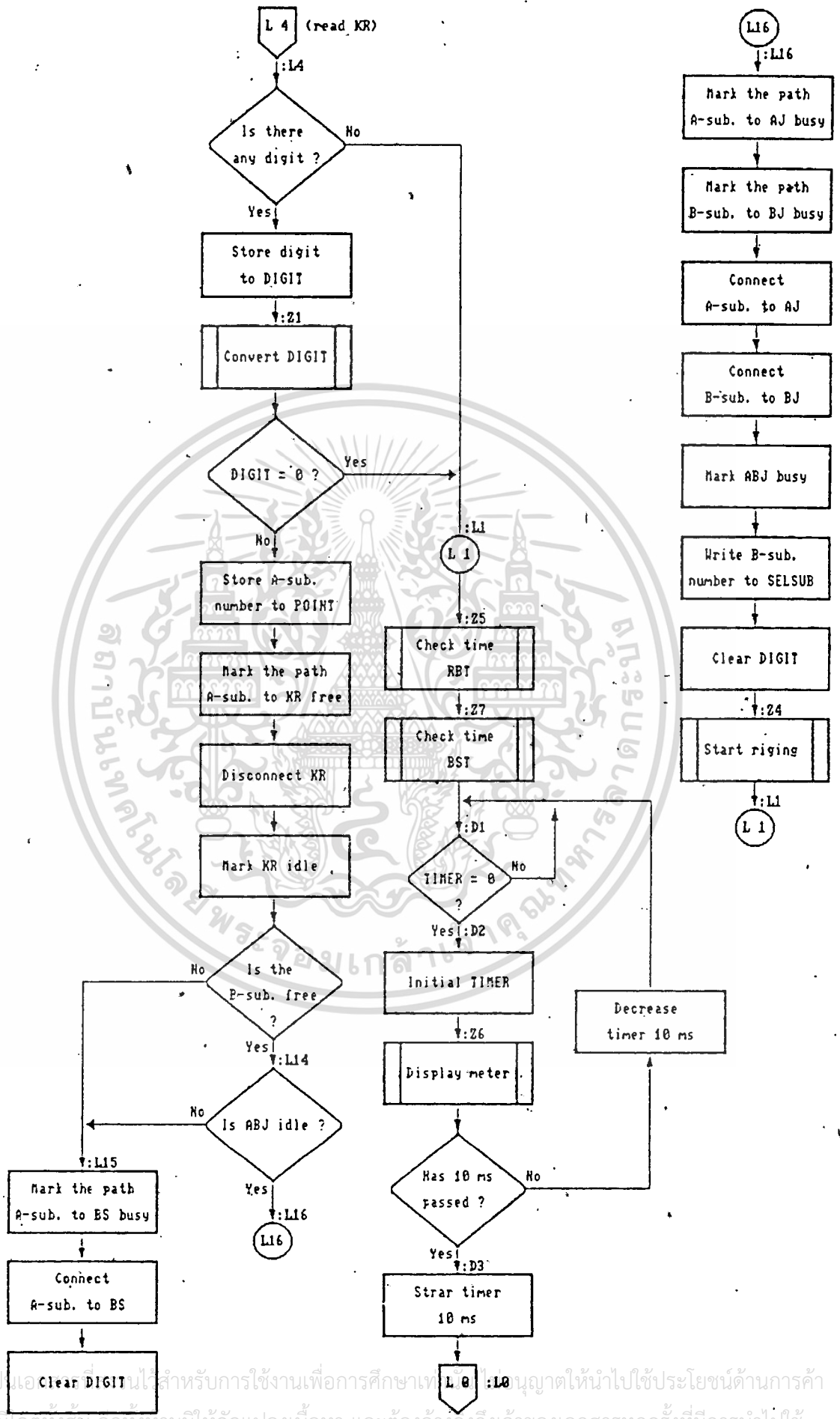
CPU จะ Test ดูว่า Column เดิมเป็น 4 ไร่ไหม ถ้าไร่ต้องเริ่มไปดำเนินการใน Column 1 ถ้าไม่ไร่ จะดูว่าของเดิมเป็น Column 1 ไร่ไหม ถ้าไร่ ไปดำเนินการใน Column 2 ถ้าไม่ไร่ จะดูว่าเป็น Column 2 ไร่ไหม จะเป็นอย่างไรไปเรื่อย ๆ การทำเช่นนี้ เมื่อเอาค่าของมันไปแสดงผลออกทางมิเตอร์ เพื่อให้เราทราบว่าเบอร์ไหนใช้โทรศัพท์กี่ครั้งแล้วเมื่อจะได้เป็นข้อมูลที่จะไปเก็บค่าใช้โทรศัพท์ต่อไป

11. Flow Chart

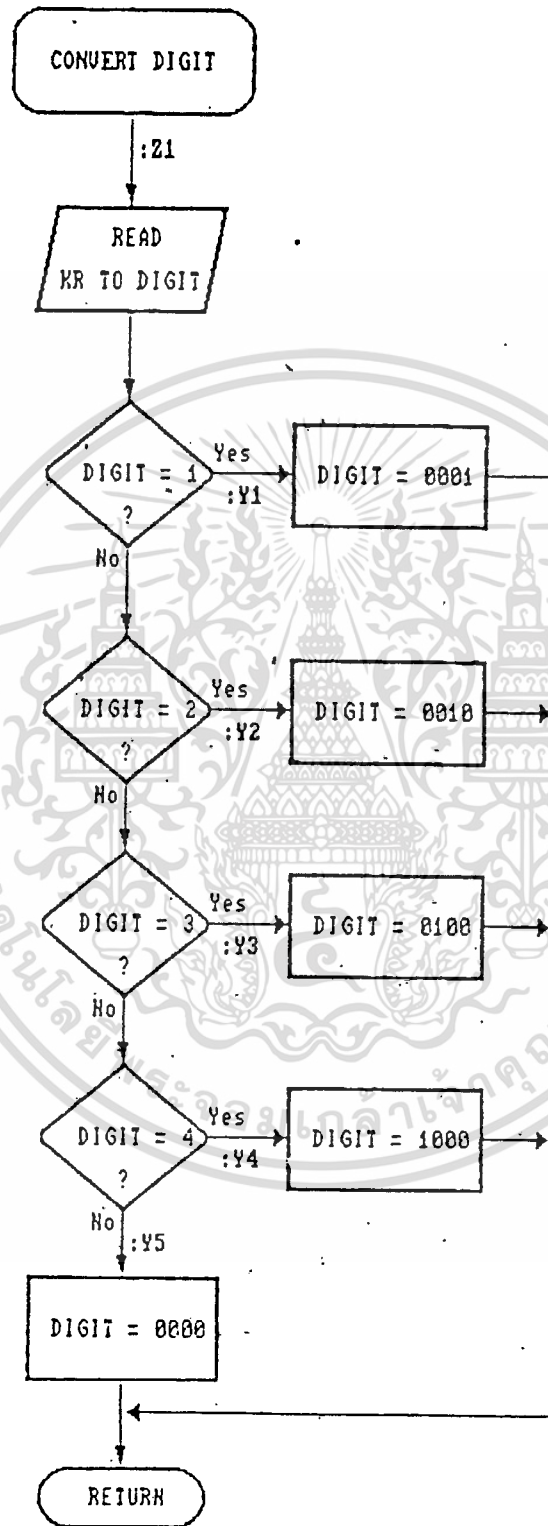




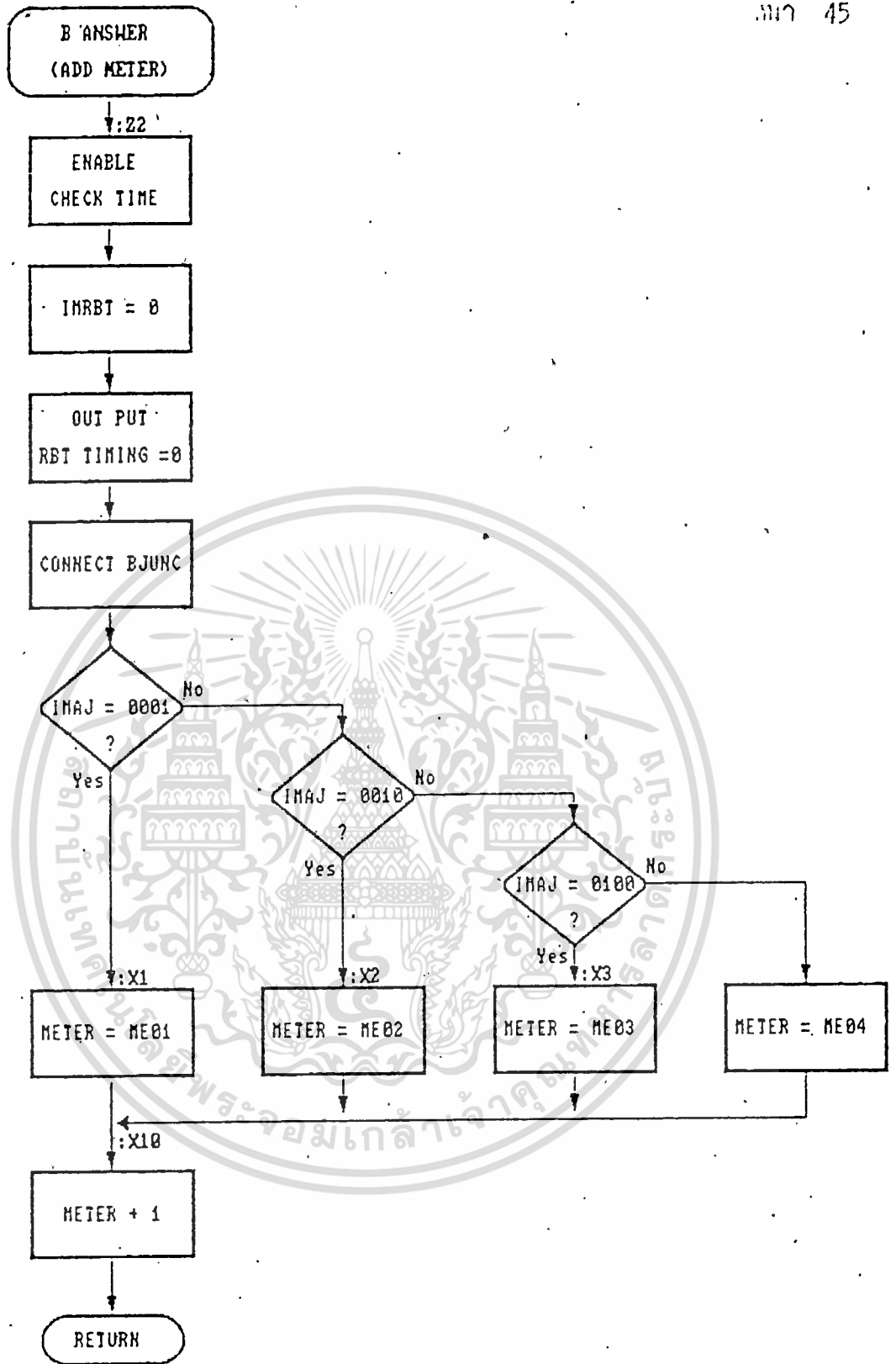


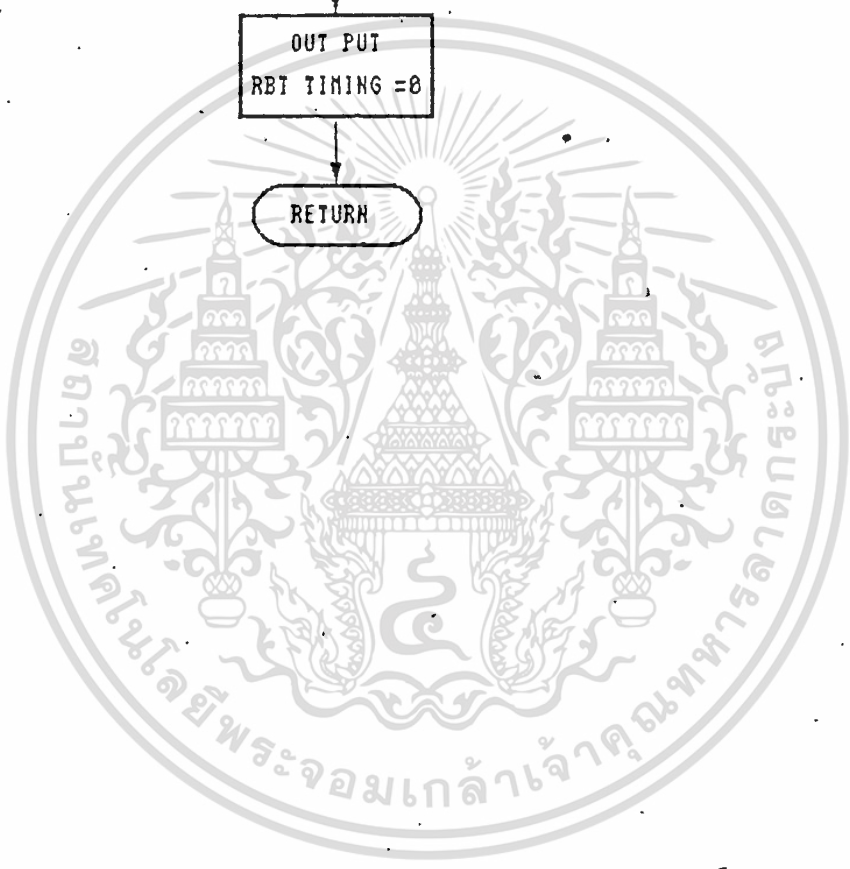
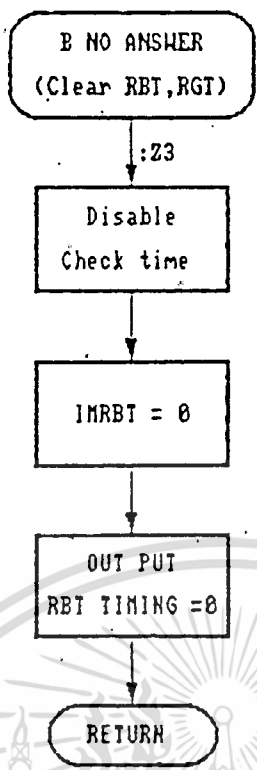


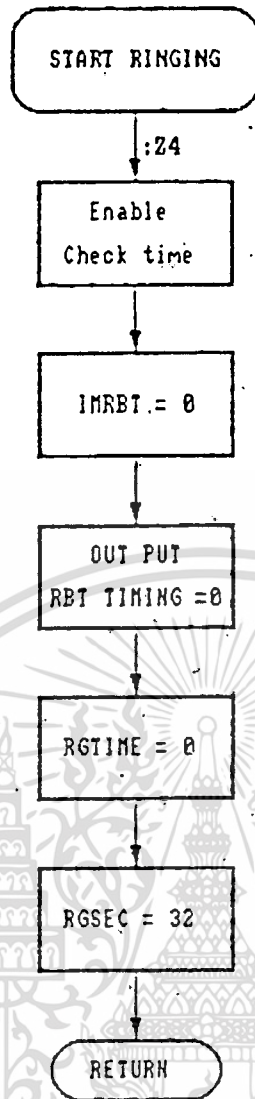
SUBROUTINE

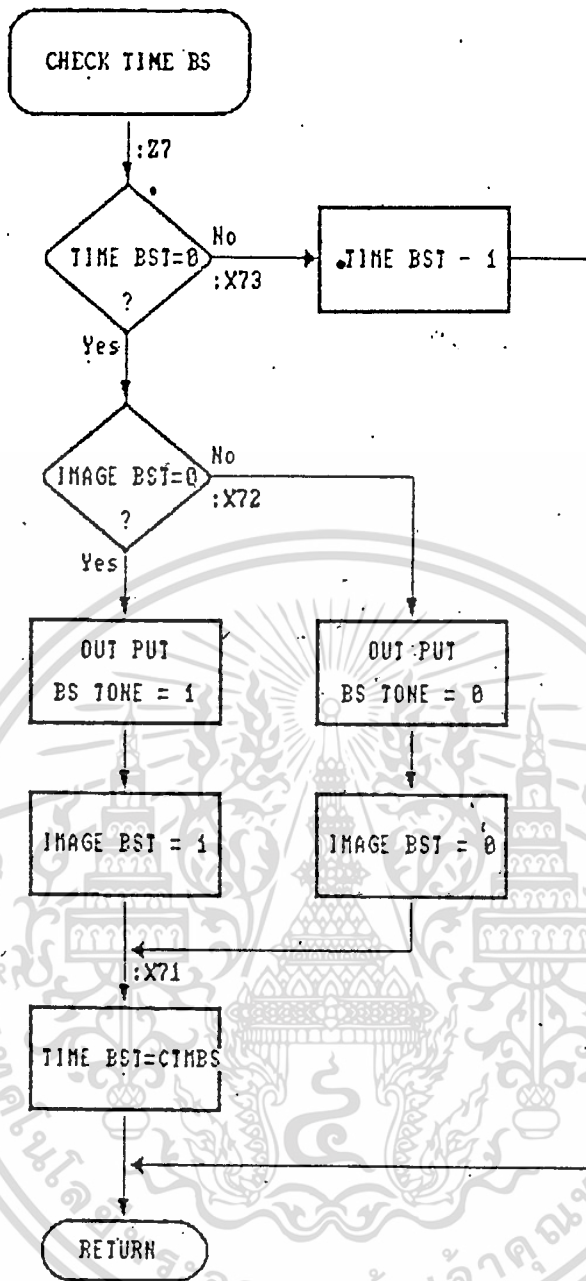


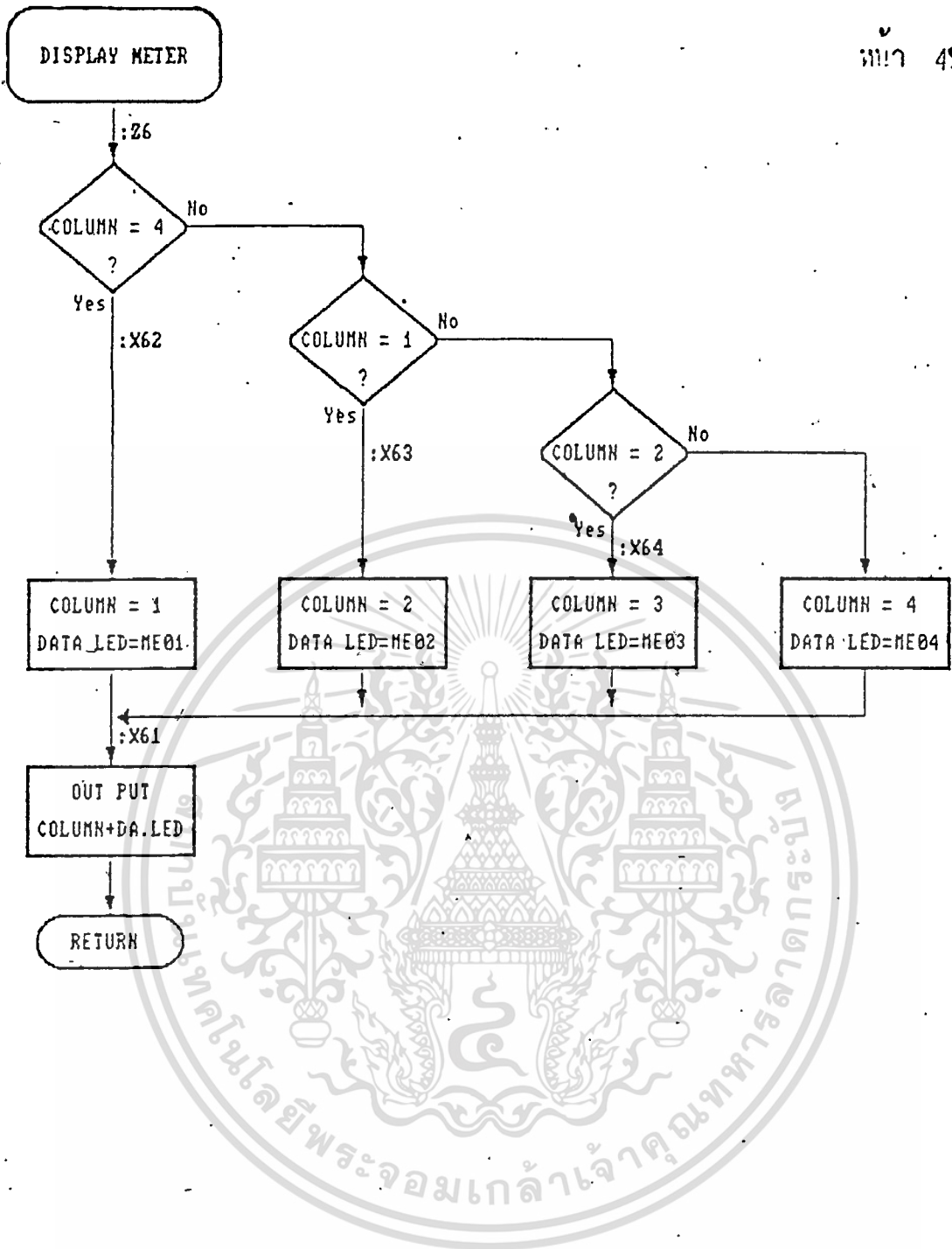
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีที่คัดลอกและเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



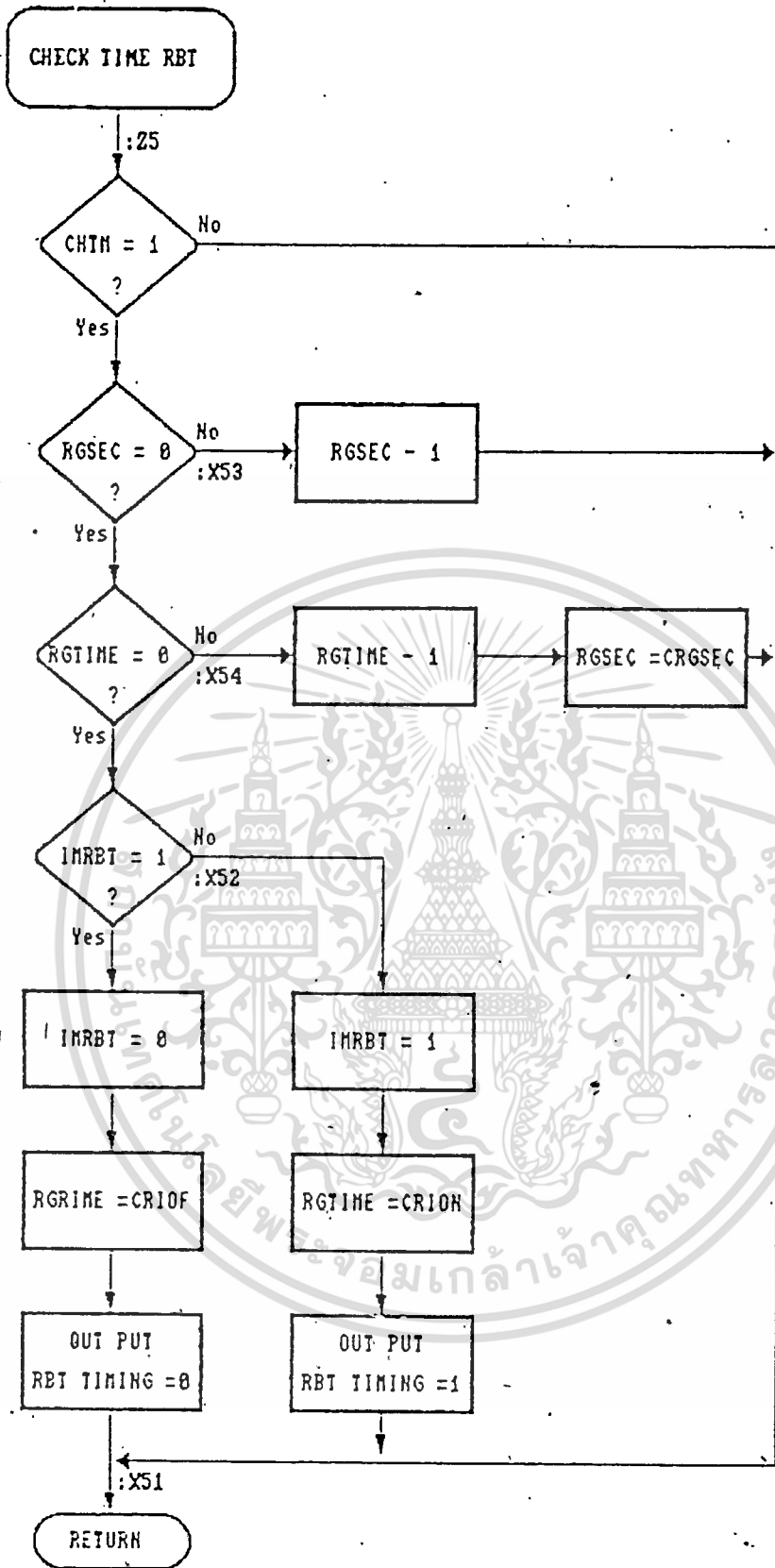








เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ทำแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. โปรแกรมขยาย SPC 4 เลขนท้าย

1	0001	CRION	EQU 001H
2	0003	CRIOF	EQU 003H
3	0031	CTMBS	EQU 031H
4	00AC	CTSYS	EQU 0ACH
5	0001	CTSPC	EQU 001H
6	0063	CRGSEC	EQU 063H
7	0020	IMAJ	EQU 020H
8	0021	IMBJ	EQU 021H
9	0022	IMKR	EQU 022H
10	0023	IMBS	EQU 023H
11	0024	POINT	EQU 024H
12	0025	SUBST	EQU 025H
13	0026	ANYJOB	EQU 026H
14	0027	DIGIT	EQU 027H
15	0028	SELSUB	EQU 028H
16	0029	DEVST	EQU 029H
17	002A	IMP1	EQU 02AH
18	002B	IMP2	EQU 02BH
19	002C	IMPA	EQU 02CH
20	002D	IMPB	EQU 02DH
21	002E	ME01	EQU 02EH
22	002F	ME02	EQU 02FH
23	0030	ME03	EQU 030H
24	0031	ME04	EQU 031H
25	0032	CHTM	EQU 032H
26	0033	IMRBT	EQU 033H
27	0034	RGTIME	EQU 034H
28	0035	RGSEC	EQU 035H
29	0036	DACOL	EQU 036H
30	0037	DALED	EQU 037H
31	0038	TMSPC	EQU 038H

```

32 0039 TMSYS EQU 0039H
33 003A IMTBS EQU 003AH
34 003B TMBS EQU 003BH

35 ORG 00
36 0000 04 09 JMP A1
37 ORG 03
38 0003 04 09 JMP A1
39 ORG 07
40 0007 04 09 JMP A1
41 0009 27 A1 CLR A ; INITIAL
42 000A 39 OUTL P1,A
43 000B 3A OUTL P2,A
44 000C 23 03 MOV A,#003H ; SET 8155
45 000E B8 00 MOV R0,#000H
46 0010 90 MOVX @R0,A
47 0011 27 CLR A
48 0012 B8 01 MOV R0,#001H ; PORT A
49 0014 90 MOVX @R0,A
50 0015 B8 02 MOV R0,#002H ; PORT B
51 0017 90 MOVX @R0,A
52 0018 B8 20 MOV R0,#1MAJ ; CLEAR ALL DAS
53 001A A0 MOV @R0,A
54 001B B8 21 MOV R0,#1MBJ
55 001D A0 MOV @R0,A
56 001E B8 22 MOV R0,#1MKR
57 0020 A0 MOV @R0,A

```

```

58 0021 B8 23      MOV R0,#IMBS
59 0023 A0         MOV @R0,A

60 0024 B8 24      MOV R0,#POINT
61 0026 A0         MOV @R0,A

62 0027 B8 25      MOV R0,#SUBST
63 0029 A0         MOV @R0,A

64 002A B8 26      MOV R0,#ANYJOB
65 002C A0         MOV @R0,A

66 002D B8 27      MOV R0,#DIGIT
67 002F A0         MOV @R0,A

68 0030 B8 29      MOV R0,#SELSUB
69 0032 A0         MOV @R0,A

70 0033 B8 29      MOV R0,#DEVST
71 0035 A0         MOV @R0,A

72 0036 B8 2E      MOV R0,#ME01
73 0038 A0         MOV @R0,A

74 0039 B8 2F      MOV R0,#ME02
75 003B A0         MOV @R0,A

76 003C B8 30      MOV R0,#ME03
77 003E A0         MOV @R0,A

78 003F B8 31      MOV R0,#ME04
79 0041 A0         MOV @R0,A

80 0042 B8 2C      MOV R0,#IMPA
81 0044 A0         MOV @R0,A

82 0045 B8 2D      MOV R0,#IMPB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

83  0047  A0          MOV @R0,A
84  0048  B8 32        MOV R0,#CHTM
85  004A  A0          MOV @R0,A
86  004B  B8 3A        MOV R0,#IMTBS
87  004D  A0          MOV @R0,A
88  004E  B8 3B        MOV R0,#TMBS
89  0050  B0 31        MOV @R0,#CTMBS
90  0052  B8 38        MOV R0,#TMSPC
91  0054  B0 01        MOV @R0,#CTSPC
92  0056  B8 36        MOV R0,#DACOL
93  0058  B0 80        MOV @R0,#080H
94  005A  2C AC        MOV A,#CTSYS ; TIMER
95  005C  62          MOV T,A
96  005D  55          STRT T
97  005E  0A L0        IN A,P2 ; ANYJOB = SUBST (EXC) LI(P2)
98  005F  53 0F        ANL A,#00FH
99  0061  B8 25        MOV R0,#SUBST
100 0063  B9 26        MOV R1,#ANYJOB
101 0065  D0          XRL A,@R0
102 0066  A1          MOV @R1,A
103 0067  B8 24        MOV R0,#POINT ; POINT = 1
104 0069  B0 01        MOV @R0,#001H
105 006B  B8 24 L5     MOV R0,#POINT ; Indicated bit in ANYJOB = 1 ?
106 006D  B9 26        MOV R1,#ANYJOB
107 006F  F0          MOV A,@R0
108 0070  51          ANL A,@R1
109 0071  C6 A8        JZ L2 ; No (=0) go to L2

```

```

110 0073 B9 25      MOV R1,#SUBST    ; Indicated bit in SUBST = 1 ?
111 0075 F0        MOV A,@R0
112 0076 51        ANL A,@R1
113 0077 C6 7B     JZ E20           ; No (=0) go to L6

114 0079 04 7D     JMP L3

115 007B 24 14     E20            JMP L6

116 007D B8 24 L3  MOV R0,#POINT    ; Mark the subscriber free
117 007F B9 25     MOV R1,#SUBST    ; (SUBST bit = 0)
118 0081 F0        MOV A,@R0
119 0082 37        CPL A
120 0083 51        ANL A,@R1
121 0084 A1        MOV @R1,A

122 0085 B9 20     MOV R1,#IMAJ     ; Does he use AJ (sub A) ?
123 0087 F0        MOV A,@R0
124 0088 D1        XRL A,@R1
125 0089 C6 F0     JZ L7           ; Yes (=0) go to L7

126 008B B9 21     MOV R1,#IMBJ     ; Does he use BJ (sub B) ?
127 008D F0        MOV A,@R0
128 008E D1        XRL A,@R1
129 008F C6 C8     JZ L8           ; Yes (=0) go to L8

130 0091 B9 22     MOV R1,#IMKR     ; Does he use KR ?
131 0093 F0        MOV A,@R0
132 0094 D1        XRL A,@R1
133 0095 C6 B3     JZ L9           ; Yes (=0) go to L9

134 0097 B9 23     MOV R1,#1MBS    ; Mark path to BS free
135 0099 F0        MOV A,@R0
136 009A 37        CPL A
137 009B 51        ANL A,@R1
138 009C A1        MOV @R1,A
139 009D AA        MOV R2,A

```

```

140 009E B9 2D      MOV R1,#IMPB      ; Disconnect the sub from BS
141 00A0 F1        MOV A,@R1
142 00A1 53 F0      ANL A,#0F0H
143 00A3 4A        ORL A,R2
144 00A4 A1         MOV @R1,A
145 00A5 B9 02      MOV R1,#002H
146 00A7 91        MOVX @R1,A

147 00A8 B8 24 L2    MOV R0,#POINT     ; Is bit 3 in POINT = 1 ?
148 00AA F0        MOV A,@R0
149 00AB 72 B1      JB3 E10           ; Yes (=1) go to L4

150 00AD E7         RL A              ; Multiply POINT by 2
151 00AE A0        MOV @R0,A
152 00AF 04 6B      JMP L5
153 00B1 24 55 E10   JMP L4

154 00B3 B9 22 L9    MOV R1,#IMKR      ; Mark path to KR free
155 00B5 B1 00      MOV @R1,#000H
156 00B7 B9 29      MOV R1,#DEVST     ; Mark KR idle
157 00B9 F1        MOV A,@R1
158 00BA 53 FD      ANL A,#0FDH
159 00BC A1         MOV @R1,A

160 00BD B9 2D      MOV R1,#IMPB      ; Disconnect the sub from KR
161 00BF F1        MOV A,@R1
162 00C0 53 0F      ANL A,#00FH
163 00C2 A1         MOV @R1,A
164 00C3 B9 02      MOV R1,#002H
165 00C5 91        MOVX @R1,A
166 00C6 04 AB      JMP L2

```

```

167 00C8 B9 20 L8    MOV R1,#IMAJ      ; Mark path A-sub to BS busy

```

```

168 00CA F1          MOV A,@R1
169 00CB B9 23      MOV R1,#IMBS
170 00CD 41          ORL A,@R1
171 00CE A1          MOV @R1,A
172 00CF AA          MOV R2,A

173 00D0 B9 2D      MOV R1,#IMPB ; Connect A-sub to BS
174 00D2 F1          MOV A,@R1
175 00D3 4A          ORL A,R2
176 00D4 A1          MOV @R1,A
177 00D5 B9 02      MOV R1,#002H
178 00D7 91          MOVX @R1,A

179 00D8 27 L10     CLR A ; Mark path AJ free
180 00D9 B9 20      MOV R1,#IMAJ
181 00DB A1          MOV @R1,A
182 00DC B9 21      MOV R1,#IMBJ ; Mark path BJ free
183 00DE A1          MOV @R1,A

184 00DF B9 29      MOV R1,#DEVST ; Mark ABJ idle
185 00E1 F1          MOV A,@R1
186 00E2 53 FE      ANL A,#0FEH
187 00E4 A1          MOV @R1,A

188 00E5 B9 2C      MOV R1,#IMPA ; Disconnect AJ and BJ
189 00E7 27          CLR A ; (PA)
190 00E8 A1          MOV @R1,A
191 00E9 B9 01      MOV R1,#001H
192 00EB 91          MOVX @R1,A

193 00EC 9A 7F      ANL P2,#07FH ; Disconnect BJUNC
194 00EE 04 A8      JMP L2 ; go to L2

195 00F0 B9 21 L7    MOV R1,#IMBJ ; Has B answered ?
196 00F2 F1          MOV A,@R1
197 00F3 B9 25      MOV R1,#SUBST
198 00F5 51          ANL A,@R1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

199 00F6 C6 FA      JZ E30      ; No (=0) go to L11
200 00F8 04 FC      JMP E31
201 00FA 24 0C      E30  JMP L11
202 00FC B9 21      E31  MOV R1,#IMBJ ; Mark path BS to B-sub busy
203 00FE F1          MOV A,@R1
204 00FF B9 23      MOV R1,#IMBS
205 0101 41          ORL A,@R1
206 0102 A1          MOV @R1,A

```

```

207 0103 B9 2D      MOV R1,#IMPB ; Connect B-sub to BS
208 0105 41          ORL A,@R1
209 0106 A1          MOV @R1,A
210 0107 B9 02      MOV R1,#002H
211 0109 91          MOVX @R1,A
212 010A 04 D8      JMP L10      ; go to L10
213 010C B9 28      L11  MOV R1,#SELSUB ; Clear SELSUB
214 010E B1 00      MOV @R1,#000H
215 0110 54 35      CALL Z3      ; CALL CLEAR RBT , RGT
216 0112 04 D8      JMP L10      ; go to L10

```

; #### Call or answer #####

```

217 0114 B8 24      L6  MOV R0,#POINT ; Mark the subscriber busy
218 0116 B9 25      MOV R1,#SUBST
219 0118 F0          MOV A,@R0
220 0119 41          ORL A,@R1
221 011A A1          MOV @R1,A
222 011B B9 28      MOV R1,#SELSUB ; Is the subscriber chosen as
; B-subscriber ?
223 011D F0          MOV A,@R0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

224 011E D1      XRL A,@R1
225 011F C6 4D   JZ L13          ; Yes (=0) B-answer go to L13

226 0121 B9 29   MOV R1,#DEVST  ; Is KR idle ?
227 0123 F1      MOV A,@R1
228 0124 37      CPL A
229 0125 32 35   JB1 L12        ; Yes (=0)

230 0127 B5 23   MOV R1,#IMBS   ; Mark the BS path busy
231 0129 F0      MOV A,@R0
232 012A 41      ORL A,@R1
233 012B A1      MOV @R1,A

234 012C B9 2D   MOV R1,#IMPB   ; Connect the subscriber to BS
235 012E 41      ORL A,@R1
236 012F A1      MOV @R1,A
237 0130 B9 02   MOV R1,#002H
238 0132 91      MOVX @R1,A

240 0133 04 A8   JMP L2          ; go to L2

241 0135 B9 22 L12 MOV R1,#IMKR   ; Mark the KR path busy
242 0137 F0      MOV A,@R0
243 0138 41      ORL A,@R1
244 0139 A1      MOV @R1,A

245 013A E7      RL A           ; Connect the subscriber to KR
246 013B E7      RL A
247 013C E7      RL A
248 013D E7      RL A
249 013E B9 2D   MOV R1,#IMPB
250 0140 41      ORL A,@R1
251 0141 A1      MOV @R1,A
252 0142 B9 02   MOV R1,#002H
253 0144 91      MOVX @R1,A

254 0145 B9 29   MOV R1,#DEVST  ; Mark KR busy
255 0147 F1      MOV A,@R1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

256 0148 43 02      ORL A,#002H
257 014A A1         MOV @R1,A

258 014B 04 A8     JMP L2             ; go to L2

259 014D B9 28 L13  MOV R1,#SELSUB   ; Clear SELSUB
260 014F B1 00     MOV @R1,#000H

261 0151 54 05     CALL Z2           ; ADD METER

262 0153 04 A8     JMP L2           ; go to L2

; ##### Read KR #####

263 0155 B8 03 L4   MOV R0,#003H     ; Is there any digit ?
264 0157 80        MOVX A,@R0
265 0158 AA        MOV R2,A
266 0159 37        CPL A
267 015A 92 C6     JB4 L1           ; No (=0) go to L1

268 015C FA        MOV A,R2         ; Store digit to DIGIT
269 015D 53 0F    ANL A,#00FH
270 015F AB        MOV R3,A

271 0160 34 E0     CALL Z1         ; *** CALL CONVERT

272 0162 FB        MOV A,R3

273 0163 C6 C6     JZ L1          ; *** If DIGIT = 0 , go to L1

274 0165 B9 27     MOV R1,#DIGIT
275 0167 A1        MOV @R1,A

276 0168 B8 24     MOV R0,#POINT   ; Store A-sub number to POINT
277 016A B9 22     MOV R1,#IMKR
278 016C F1        MOV A,@R1
279 016D A0        MOV @R0,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

280 016E B1 00      MOV @R1,#000H    ; Mark the path A-sub to KR free
281 0170 B9 2D      MOV R1,#IMPB     ; Disconnect KR
282 0172 F1         MOV A,@R1
283 0173 53 0F      ANL A,#00FH
284 0175 A1         MOV @R1,A
285 0176 B9 02      MOV R1,#002H
286 0178 91         MOVX @R1,A

287 0179 B9 29      MOV R1,#DEVST    ; Mark KR idle
288 017B F1         MOV A,@R1
289 017C 53 FD      ANL A,#0FDH
290 017E A1         MOV @R1,A

291 017F B9 27      MOV R1,#DIGIT    ; Is the B-sub free ?
292 0181 F1         MOV A,@R1
293 0182 B9 25      MOV R1,#SUBST
294 0184 51         ANL A,@R1
295 0185 C6 99      JZ L14           ; Yes (=0) go to L14

296 0187 B9 23 L15  MOV R1,#IMBS     ; Mark the path A-sub to BS busy
297 0189 F0         MOV A,@R0
298 018A 41         ORL A,@R1
299 018B A1         MOV @R1,A

300 018C B9 2D      MOV R1,#IMPB     ; Connect A-sub to BS
301 018E 41         ORL A,@R1
302 018F A1         MOV @R1,A
303 0190 B9 02      MOV R1,#002H
304 0192 91         MOVX @R1,A

305 0193 B9 27      MOV R1,#DIGIT    ; Clear DIGIT
306 0195 B1 00      MOV @R1,#000H

307 0197 24.C6     JMP L1           ; go to L1

308 0199 B9 29 L14  MOV R1,#DEVST    ; Is ABJ idle ?

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

309 019B F1          MOV A,@R1
310 019C 12 87      JB0 L15           ; No (=1) go to L15

311 019E B9 20    L16  MOV R1,#IMAJ      ; Mark the path A-sub to AJ busy
312 01A0 F0          MOV A,@R0
313 01A1 A1         MOV @R1,A
314 01A2 AB         MOV R3,A

315 01A3 B9 27      MOV R1,#DIGIT    ; Mark the path B-sub to BJ busy
316 01A5 F1          MOV A,@R1
317 01A6 B9 21      MOV R1,#IMBJ
318 01A8 A1         MOV @R1,A

319 01A9 E7          RL A             ; Connect A-sub to AJ
320 01AA E7          RL A             ; Connect B-sub to BJ
321 01AB E7          RL A
322 01AC E7          RL A
323 01AD 4B         ORL A,R3
324 01AE B9 2C      MOV R1,#IMPA
325 01B0 A1         MOV @R1,A
326 01B1 B9 01      MOV R1,#001H
327 01B3 91         MOVX @R1,A

328 01B4 B9 29      MOV R1,#DEVST    ; Mark ABJ busy
329 01B6 F1          MOV A,@R1
330 01B7 43 01      ORL A,#001H
331 01B9 A1         MOV @R1,A

332 01BA B8 27      MOV R0,#DIGIT    ; Write B-sub number to SELSUB
333 01BC F0          MOV A,@R0
334 01BD B9 28      MOV R1,#SELSUB
335 01BF A1         MOV @R1,A

336 01C0 B0 00      MOV @R0,#000H    ; Clear DIGIT

337 01C2 54 40      CALL Z4          ; Start ringing

338 01C4 24 C6      JMP L1           ; go to L1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

339 01C6 54 53 L1 CALL Z5 ; CHECK TIME RBT
340 01C8 54 AC CALL Z7 ; CHECK TIME BST
341 01CA 16 CE D1 JTF D2
342 01CC 24 CA JMP D1
343 01CE 23 AC D2 MOV A,#CTSYS
344 01D0 62 MOV T,A
345 01D1 54 86 CALL Z6 ; DISPLAY METER
346 01D3 B8 38 MOV R0,#TMSPC ; TMSPC = 0 ?
347 01D5 F0 MOV A,@R0
348 01D6 C6 DC JZ D3
349 01D8 07 DEC A ; Yes (TMSPC-1)
350 01D9 A0 MOV @R0,A
351 01DA 24 CA JMP D1 ; go to D1
352 01DC B0 01 D3 MOV @R0,#CTSPC ; TMSPC = CTSPC
353 01DE 04 5E JMP L0 ; go to L0

; #### SUBROUTINE ####

354 01E0 FB Z1 MOV A,R3
355 01E1 D3 01 XRL A,#001H
356 01E3 C6 F6 JZ Y1 ; DIGIT = 1 (go to Y1)

357 01E5 FB MOV A,R3
358 01E6 D3 02 XRL A,#002H
359 01E8 C6 F9 JZ Y2 ; DIGIT = 2 (go to Y2)

```

```

360 01EA FB      MOV A,R3
361 01EB D3 03   XRL A,#003H
362 01ED C6 FC   JZ Y3           ; DIGIT = 3 (go to Y3)

363 01EF FB      MOV A,R3
364 01F0 D3 04   XRL A,#004H
365 01F2 C6 FF   JZ Y4           ; DIGIT = 4 (go to Y4)

366 01F4 44 02   JMP Y5

```

```

367 01F6 BB 01   Y1   MOV R3,#001H   ; DIGIT 1 = SUB 1
368 01F8 83      RET
369 01F9 BB 02   Y2   MOV R3,#002H   ; DIGIT 2 = SUB 2
370 01FB 83      RET
371 01FC BB 04   Y3   MOV R3,#004H   ; DIGIT 4 = SUB 3
372 01FE 83      RET
373 01FF BB 08   Y4   MOV R3,#008H   ; DIGIT 8 = SUB 3
374 0201 83      RET
375 0202 BB 00   Y5   MOV R3,#000H   ; DIGIT LIMIT
376 0204 83      RET

```

; #### B ANSWER ####

```

377 0205 B8 32   Z2   MOV R0,#CHTM   ; CHTM = 0
378 0207 B0 00   MOV @R0,#000H

379 0209 B8 33   MOV R0,#IMRBT  ; IMRBT = 0
380 020B B0 00   MOV @R0,#000H

381 020D 9A BF   ANL P2,#0BFH   ; O/P = 0
382 020F 8A 80   ORL P2,#080H   ; Connect BJUNC

```

```

383 0211 B8 20      MOV R0,#1MAJ      ; ADD METER
384 0213 F0         MOV A,@R0

385 0214 12 29      JB0 X1            ; ME01
386 0216 32 2D      JB1 X2            ; ME02
387 0218 52 31      JB2 X3            ; ME03

388 021A 38 31      MOV R0,#ME04     ; ME04

389 021C F0         X10 MOV A,@R0     ; Meter + 1
390 021D 17         INC A
391 021E AA         MOV R2,A
392 021F D3 0A      XRL A,#00AH

393 0221 C6 26      JZ X5

394 0223 FA         MOV A,R2
395 0224 A0         MOV @R0,A
396 0225 83         RET

397 0226 B0 00      X5 MOV @R0,#000H
398 0228 83         RET

399 0229 B8 2E      X1 MOV R0,#ME01
400 022B 44 1C      JMP X10

401 022D B8 2F      X2 MOV R0,#ME02
402 022F 44 1C      JMP X10

403 0231 B8 30      X3 MOV R0,#ME03
404 0233 44 1C      JMP X10

```

; #### B NO ANSWER ####

```

405 0235 B8 32      Z3 MOV R0,#CHTM     ; CHTM = 0

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการทำงานเพื่อ MOV @R0, #000H อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรณำไปใช้

```

407  0239  B8 33      MOV R0,#IMRBT ; IMRBT = 0
408  023B  B0 00      MOV @R0,#000H

409  023D  9A BF      ANL P2,#0BFH ; O/P = 0

410  023F  83          RET

```

; #### START RINGING ####

```

411  0240  B8 32  Z4      MOV R0,#CHTM ; CHTM = 1
412  0242  B0 01      MOV @R0,#001H

413  0244  B8 33      MOV R0,#IMRBT ; IMRBT = 0
414  0246  B0 00      MOV @R0,#000H

415  0248  9A BF      ANL P2,#0BFH ; O/P = 0

416  024A  B8 34      MOV R0,#RGTIME ; RGTIME = 0
417  024C  B0 00      MOV @R0,#000H

418  024E  B8 35      MOV R0,#RGSEC ; RGSEC = 32
419  0250  B0 32      MOV @R0,#032H

420  0252  83          RET

```

; #### CHECK TIME RBT ####

```

421  0253  B8 32  Z5      MOV R0,#CHTM ; CHTM = 1 ?
422  0255  F0          MOV A,@R0
423  0256  C6 6F      JZ X51 ; No go to X51

424  0258  B8 35      MOV R0,#RGSEC ; RGSEC = 0 ?
425  025A  F0          MOV A,@R0
426  025B  96 7A      JNZ X53 ; No go to X53

427  025D  B8 34      MOV R0,#RGTIME ; RGTIME = 0 ?
428  025F  F0          MOV A,@R0

```

```

429 0260 96 7E      JNZ X54           ; No go to X54

430 0262 B8 33      MOV R0,#IMRBT    ; IMRBT = 1 ?
431 0264 F0         MOV A,@R0
432 0265 C6 70      JZ X52           ; No go to X52

433 0267 B0 00      MOV @R0,#000H   ; IMRBT = 0

434 0269 B8 34      MOV R0,#RGTIME  ; RGTIME = CRIOF
435 026B B0 03      MOV @R0,#CRIOF

436 026D 9A BF      ANL P2,#0BFH    ; O/P = 0

437 026F 83         X51  RET

438 0270 B0 01      X52  MOV @R0,#001H  ; IMRBT = 1
439 0272 B8 34      MOV R0,#RGTIME  ; RGTIME = CRION
440 0274 B0 01      MOV @R0,#CRION

441 0276 8A 40      ORL P2,#040H    ; O/P = 1

442 0278 44 6F      JMP X51

443 027A 07         X53  DEC A           ; RGSEC - 1
444 027B A0         MOV @R0,A

445 027C 44 6F      JMP X51

446 027E 07         X54  DEC A           ; RGTIME - 1
447 027F A0         MOV @R0,A

448 0280 B8 35      MOV R0,#RGSEC   ; RGSEC = CRGSEC
449 0282 B0 63      MOV @R0,#CRGSEC

450 0284 44 6F      JMP X51

```

```

451  0286  B8 36  Z6  MOV R0,#DACOL ; COL = 4 ?
452  0288  F0      MOV A,@R0
453  0289  F2 97      JB7 X62 ; COLUMN = 1
454  028B  92 9E      JB4 X63 ; COLUMN = 2
455  028D  B2 A5      JB5 X64 ; COLUMN = 3
456  028F  B0 80      MOV @R0,#080H ; COLUMN = 4
457  0291  B9 31      MOV R1,#ME04 ; DALED = ME04
458  0293  F1      MOV A,@R1
459  0294  40      X61  ORL A,@R0 ; OUT P1 = DACOL + DALED
460  0295  39      OUT P1,A
461  0296  83      RET
462  0297  B0 10  X62  MOV @R0,#010H ; COLUMN = 1
463  0299  B9 2E      MOV R1,#ME01 ; DALED = ME01
464  029B  F1      MOV A,@R1
465  029C  44 94      JMP X61
466  029E  B0 20  X63  MOV @R0,#020H ; COLUMN = 2
467  02A0  B9 2F      MOV R1,#ME02 ; DALED = ME02
468  02A2  F1      MOV A,@R1
469  02A3  44 94      JMP X61
470  02A5  B0 40  X64  MOV @R0,#040H ; COLUMN = 3
471  02A7  B9 30      MOV R1,#ME03 ; DALED = ME03
472  02A9  F1      MOV A,@R1

```

473 02AA 44 94

JMP X61

; #### CHECK TIME B.S. TONE ####

474 02AC B8 3B Z7 MOV R0,#TMBS ; TMBS = 0 ?

475 02AE F0 MOV A,@R0

476 02AF 96 C9 JNZ X73 ; No go to X70

477 02B1 B8 3A MOV R0,#1MTBS ; 1MTBS = 0 ?

478 02B3 F0 MOV A,@R0

479 02B4 96 C1 JNZ X72 ; No go to X71

480 02B6 8A 20 ORL P2,#020H ; OUT BST = 1

481 02B8 B8 3A MOV R0,#1MTBS ; 1MTBS = 1

482 02BA B0 01 MOV @R0,#001H

483 02BC B8 3B X71 MOV R0,#TMBS ; TMBS = CTMBS

484 02BE 20 21 MOV @R0,#CTMBS

485 02C0 83 RET

486 02C1 9A DF X72 ANL P2,#0DFH ; OUT BST = 0

487 02C3 B8 3A MOV R0,#1MTBS ; 1MTBS = 0

488 02C5 B0 00 MOV @R0,#000H

489 02C7 44 BC JMP X71

490 02C9 07 X73 DEC A ; TMBS - 1

491 02CA A0 MOV @R0,A

492 02CB 83 RET

493 02CC END