การ์ดถอดรหัสเพจเจอร์บนไมโครคอมพิวเตอร์

Pager Decoder Card on Microcomputer

โดย

นาย	ชัยศักดิ์ รุ่งคณาวุฒิ	36012046 2M
นายอดุลย์	สันติรักษ์พงษ์	36012082 2M
นายอุดมพร	สุนันทชัยกุล	36012087 2M

ปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ การ์ดถอดรหัสเพจเจอร์บนไมโครคอมพิวเตอร์

Pager Decoder Card on Microcomputer

ชื่อนักศึกษา นายเกรียงศักดิ์ รุ่งคณาวุฒิ 36012046 2M

นายอดุลย์ สันติรักษ์พงษ์ 36012082 2M

นายอุดมพร สุนันทชัยกุล 36012087 2M

อาจารย์ที่ปรึกษา อ.กฤดากร กล่อมการ

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2537

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

การ์ดถอดรหัสเพจเจอร์บนไมโครคอมพิวเตอร์

โดย นายเกรียงศักดิ์ รุ่งคณาวุฒิ 36012046 2M

นายอดุลย์ สันติรักษ์พงษ์ 36012082 2M

นายอุดมพร สุนันทชัยกุล 36012087 2M

อาจารย์ที่ปรึกษา อ.กฤดากร กล่อมการ

ปีการศึกษา 2537

บทคัดย่อ

โครงงานชุดนี้เป็นการ์ดถอดรหัสเพจเจอร์บนไมโครคอมพิวเตอร์ ซึ่งสามารถใช้ตรวจสอบการทำงานของระบบการส่งเพจเจอร์ โครงงานนี้สามารถตรวจจับหมายเลขเครื่องรับเพจเจอร์ที่ต้องการรับโดยจะมีสัญญาณเสียงเตือนเมื่อมีข้อมูลของหมายเลขเครื่องที่ต้องการ โดยไมโครคอมพิวเตอร์จะได้รับสัญญาณอินเตอร์รัพท์จากการ์ดถอดรหัสเพจเจอร์ และยังทำหน้าที่เก็บข้อมูลเพจเจอร์ที่รับได้ชั่วขณะหนึ่งสำหรับแสดงผลออกทางจอภาพ หากต้องการทราบข่าวสารก็เพียงแต่ป้อนหมายเลขเครื่องเพจเจอร์ให้กับโปรแกรมที่สร้างขึ้นทางแป้นพิมพ์เท่านั้น การทำงานของทั้งระบบจะอาศัยทั้งชุดวงจร และโปรแกรมคอมพิวเตอร์ที่เขียนขึ้นด้วยภาษาซี ความสามารถของโครงงานนี้ สามารถถอดรหัสสัญญาณเพจเจอร์ในโหมดตัวเลข และโหมดตัวอักษร-ตัวเลขของมาตราฐาน POCSAG ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pager Decoder Card on Microcomputer

BY MR.KRIANGSAK RUNGKANAVUT 36012046 2M

MR.ADUL SANTIRUKPONG 36012082 2M

MR.UDOMPORN SUNUNTACHAKUL 36012087 2M

ADVISER MR.KITDAKORN KROMKRAN

YEASR 1995

ABSTRACT

This project is presentation of pager decoder card on personal computer. It can test paging transmisters system. This project is used to find plugcode of hardware decodes address plugcode. Thise is alarm noise when the data is sent from pager that we want to know. Microcomputer reserive intercept signal from card of computer. It oftain the data of pager, to receive in short time for presenting result on monitor. If we want to know the message. We can call the program and input the plugcode of pager. This systems cooperate function together. This program is writen by C lanqauge. The program can decode pager signal in numeric mode and alpha-numeric modé by POCSAG standard format.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการปริญญานิพนธ์สำเร็จลุล่วงลงได้โดยได้รับความช่วยเหลือจากหลาย ๆ ฝ่ายด้วยกัน และที่สำคัญที่สุดพระคุณของบิดา มารดาผู้ให้ทุกสิ่งทุกอย่าง ตลอดจนคำปรึกษาและแนะนำของท่านอาจารย์ กฤดากร กล่อมการ และเพื่อที่ให้ความช่วยเหลือทุกท่าน อีกทั้งยังได้รับความช่วยเหลือทานด้วยสถานที่และเครื่องไม้เครื่องมือในการปฏิบัติงาน จากคณะวิศวกรรมศาสตร์ ภาควิชาเทคนิคอุตสาหกรรม ทางคณะผู้จัดทำจึงของขอบพระคุณ ขอบคุณ ณ. โอกาสนี้ด้วย

คณะผู้จัดทำ

เกรียงศักดิ์ รุ่งคณาวุฒิ

อดุลย์ สันติรักพงษ์

อุดมนร สุนันทชัยกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ.....1

โครงสร้างของระบบ.....2

การขยายพื้นที่.....3

บทที่ 2 POCSAG.....6

รูปแบบของสัญญาณเพจเจอร์.....6

สัญญาณพัลส์ส่วนหน้า.....6

โครงสร้างมาตราฐาน POCSAG.....7

โครสร้างของรหัสคำ.....8

รหัสคำการซิงโครไนซ์.....10

รหัสคำหมายเลขเรียกขาน.....11

รหัสคำข่าวสารเพจเจอร์รุ่นตัวเลข.....14

รหัสคำข่าวสารเพจเจอร์รุ่นตัวอักษร-ตัวเลข.....17

รหัสคำเทียม.....24

บทที่ 3 ภาครับสัญญาณเพจเจอร์.....25

หลักการทั่วไปของภาครับสัญญาณเพจเจอร์.....26

วงจรภาครับสัญญาณ.....29

วงจรลิมิตเตอร์.....37

วงจรภาคกู้สัญญาณนาฬิกา.....38

วงจรภาคตรวจจับเฟรมซิง.....40

SIPO.....42

วงจรภาคกำเนิดสัญญาณอ่าน/เขียนแรม และวงจรนับแอดเดรส.....42

วงจรภาคแรมอีมูเลเตอร์.....45

วงจรภาคตรวจจับหมายเลขเรียกขาน.....49

วงจรภาคสร้างสัญญาณอินเตอร์รัพท์.....49

คอมพิวเตอร์.....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 โปรแกรมแสดงผลข้อมูลเพจเจอร์.....59

รายละเอียดของเฮ็ดเดอร์ไฟล์ Pager.h.....60

รายละเอียดของเฮ็ดเดอร์ไฟล์ N_pager.h.....62

รายละเอียดของโปรแกรม Demons.c.....67

รายละเอียดของโปรแกรม Pager.c.....70

รายละเอียดของโปรแกรม Binary.c.....75

รายละเอียดของโปรแกรม Demons.c.....88

รายละเอียดของโปรแกรม Plugcode.c.....99

บทที่ 5 สรุปผลการทดลอง.....100

ผลการทดลองวงจรลิมิตเตอร์.....100

ผลการทดลองวงจรภาคกู้สัญญาณนาฬิกา.....102

ผลการทดลองวงจรภาคตรวจจับเฟรมซิง.....103

ผลการทดลองวงจรภาคกำเนิดสัญญาณล่าน/เยียนแรม และวงจรนับแอดเดรส.....103

ผลการทดลองแสดงข้อมูลเพจเจอร์รุ่นตัวเลข.....107

ผลการทดลองแสดงข่าวสารเพจเจอร์รุ่นตัวเลข.....107

ผลการทดลองแสดงข้อมูลเพจเจอร์รุ่นตัวอักษร-ตัวเลข.....108

ผลการทดลองแสดงข่าวสารเพจเจอร์รุ่นตัวอักษร-ตัวเลข.....109

ภาคผนวก

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ทฤษฎีทั่วไปของวิทยุติดตามตัว (Pager)

ในระบบการติดต่อสื่อสารจากอดีตถึงปัจจุบันได้มีการพัฒนาอุปกรณ์ที่ใช้ในการสื่อสารมากมายหลายชนิดตั้งแต่อุปกรณ์สื่อสารแบบมีสายจนกระทั่งถึงอุปกรณ์แบบไร้สาย และจากความเจริญก้าวหน้าทางเทคโนโลยีก็ได้มีการพัฒนาอุปกรณ์เหล่านี้เพื่ออำนวยความสะดวกแก่ผู้ใช้มากขึ้น ไม่ว่าจะเป็นการเพิ่มรัศมีในการติดต่อสื่อสาร การลดขนาดเพื่อให้เล็กกะทัดรัด

วิทยุติดตามตัวหรือเพจเจอร์ (pager) เป็นเครื่องมือสื่อสารชนิดเคลื่อนที่แบบหนึ่งที่นิยมใช้สำหรับผู้ที่มีภาระหน้าที่ไม่ค่อยประจำที่ แต่จำเป็นต้องมีการติดต่อได้ตลอดเวลาเช่นเดียวกับวิทยุโทรศัพท์แบบรวงผึ้ง (cellular) แต่มีลักษณะการสื่อสารในทิศทางเดียวจากผู้ส่งไปยังผู้รับ

การติดต่อระหว่างผู้ส่งกับผู้รับอาจเป็นการแจ้งให้ผู้รับติดต่อกลับไปยังผู้ส่ง หรืออาจเป็นการส่งข่าวสารที่ต้องการไปยังผู้รับโดยตรงก็ได้ เพจเจอร์ที่มีใช้แบ่งออกได้เป็น 5 แบบ คือ

1. Voice pager จะทำการส่งข่าวสารเป็นเสียงพูดไปยังผู้รับ
2. Digital display pager จะทำการส่งข่าวสารเป็นตัวเลข ซึ่งตัวเลขนี้จะแทนข่าวสารที่มีความหมายเป็นที่เข้าใจกันระหว่างผู้ส่งกับผู้รับ
3. Alpha-Numeric pager แบบนี้สามารถที่จะส่งข่าวสารเป็นได้ทั้งตัวเลข และตัวอักษรซึ่งเป็นการช่วยให้ส่งข่าวสารได้ละเอียดขึ้น แต่จำเป็นต้องใช้ระบบที่พิเศษออกไป ซึ่งเป็นแบบที่นิยมใช้ในปัจจุบันภายในประเทศ
4. Tone-alert pager เป็นการส่งสัญญาณเสียงเพื่อเป็นการบอกให้ผู้รับติดต่อไปยังศูนย์เพื่อรับข่าวสารอีกทีหนึ่ง
5. Dual address pager จะทำการส่งสัญญาณเสียงเหมือนกัน tone-alert pager แต่จะให้สัญญาณเสียงได้ 2 ลักษณะเพื่อให้ผู้รับทราบว่าจะติดต่อไปยังที่ใดเวลาที่ใช้ในการส่งข่าวสารด้วยเพจเจอร์ในแต่ละแบบจะแตกต่างกันคือ เพจเจอร์แบบใช้เสียงพูดจะใช้เวลาตั้งแต่ 10 วินาทีลงมา ส่วนเพจเจอร์แบบที่ไม่ใช้เสียงพูดจะใช้เวลาในการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งเป็นมิลลิวินาที

ข้อดีของเพจเจอร์เมื่อเปรียบเทียบกับระบบการสื่อสารเคลื่อนที่ชนิด 2 ทิศทางคือ

1. สามารถพกติดตัวได้สะดวก เนื่องจากขนาดเล็กและมีน้ำหนักเบาและไม่ต้องมีแหล่งจ่ายไฟฟ้าจากภายนอก
2. ค่าใช้จ่ายในการเช่าใช้ถูกกว่ามาก
3. มีจำนวนผู้เช่าใช้ต่อหนึ่งความถี่ที่มากกว่า
4. เครื่องรับสามารถใช้ร่วมกันได้โดยส่วนใหญ่จากหลาย ๆ ยี่ห้อ
5. ที่ตัวเพจเจอร์แทบไม่ต้องมีการบำรุงรักษา
6. ไม่ต้องมีการติดตั้งใด ๆ ในส่วนของผู้ใช้
7. สามารถบันทึกข่าวสารไว้ได้ในตัวเครื่อง

โครงสร้างของระบบเพจเจอร์

เพจเจอร์ใช้การสื่อสารด้วยความถี่วิทยุเช่นเดียวกันกับระบบวิทยุโทรศัพท์ดังนั้น โครงสร้างของระบบจึงไม่แตกต่างกันนักดังแสดงในรูปที่ 1.1 ซึ่งประกอบด้วยส่วนต่าง ๆ ดังนี้

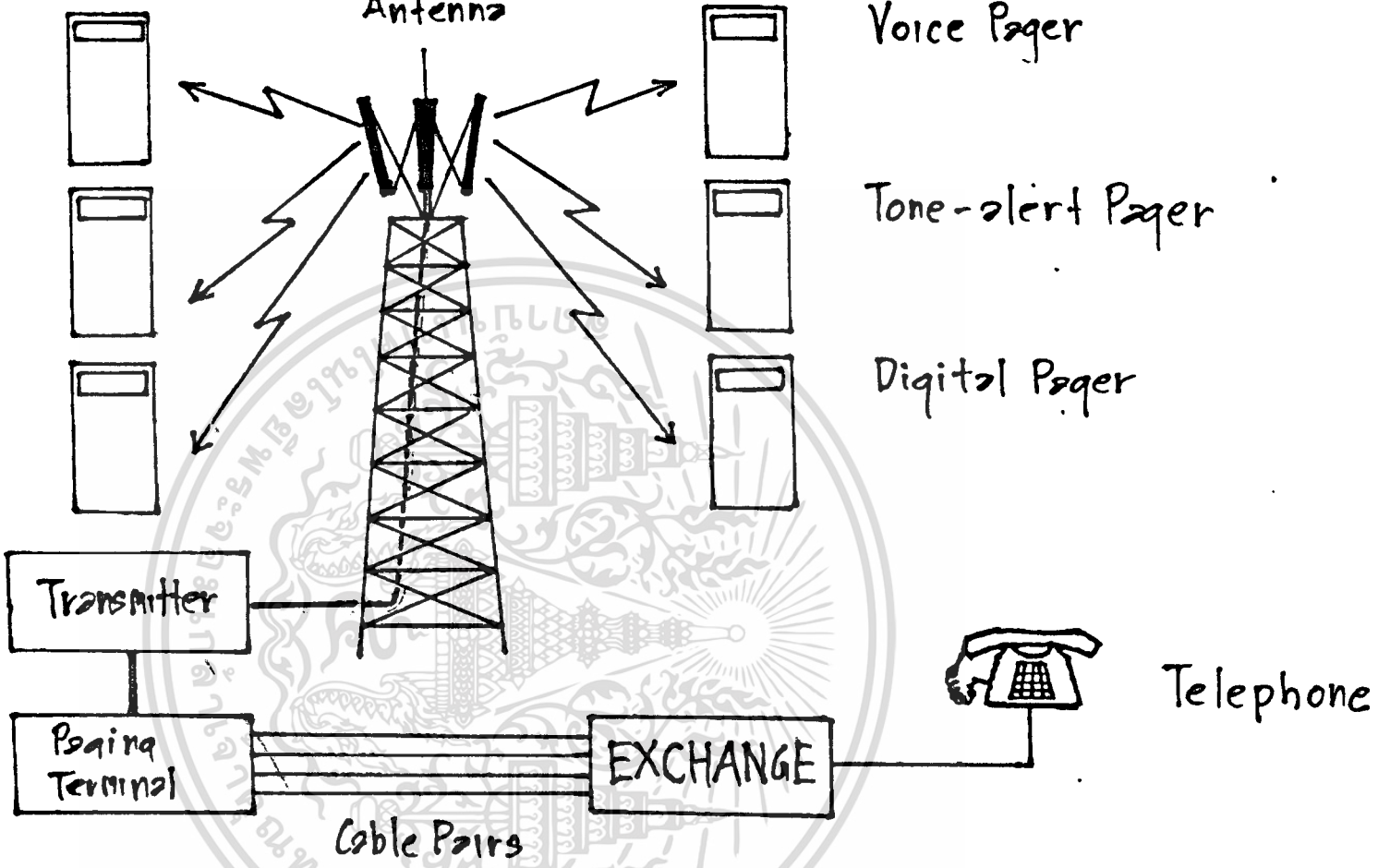
1. ระบบโทรศัพท์ เป็นส่วนที่ผู้ส่งสามารถส่งข่าวสารผ่านทางระบบโทรศัพท์ไปยังศูนย์เพื่อส่งต่อไปยังผู้รับที่ต้องการ
2. ศูนย์กลางของระบบ (Páging Terminal) เป็นศูนย์กลางในการติดต่อสื่อสารโดยมีคอมพิวเตอร์เป็นผู้จัดการทุกอย่างเกี่ยวกับการให้บริการการรับส่งข่าวสารจากผู้ส่ง ไปยังผู้รับ ซึ่งถือ ว่าเป็นส่วนที่สำคัญของระบบ
3. เครื่องส่ง ทำหน้าที่ส่งข่าวสารในรูปของคลื่นวิทยุผ่านสายอากาศไปยังเครื่องรับ ซึ่งเครื่องส่งอาจตั้งอยู่ที่ศูนย์หรือที่อื่นก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

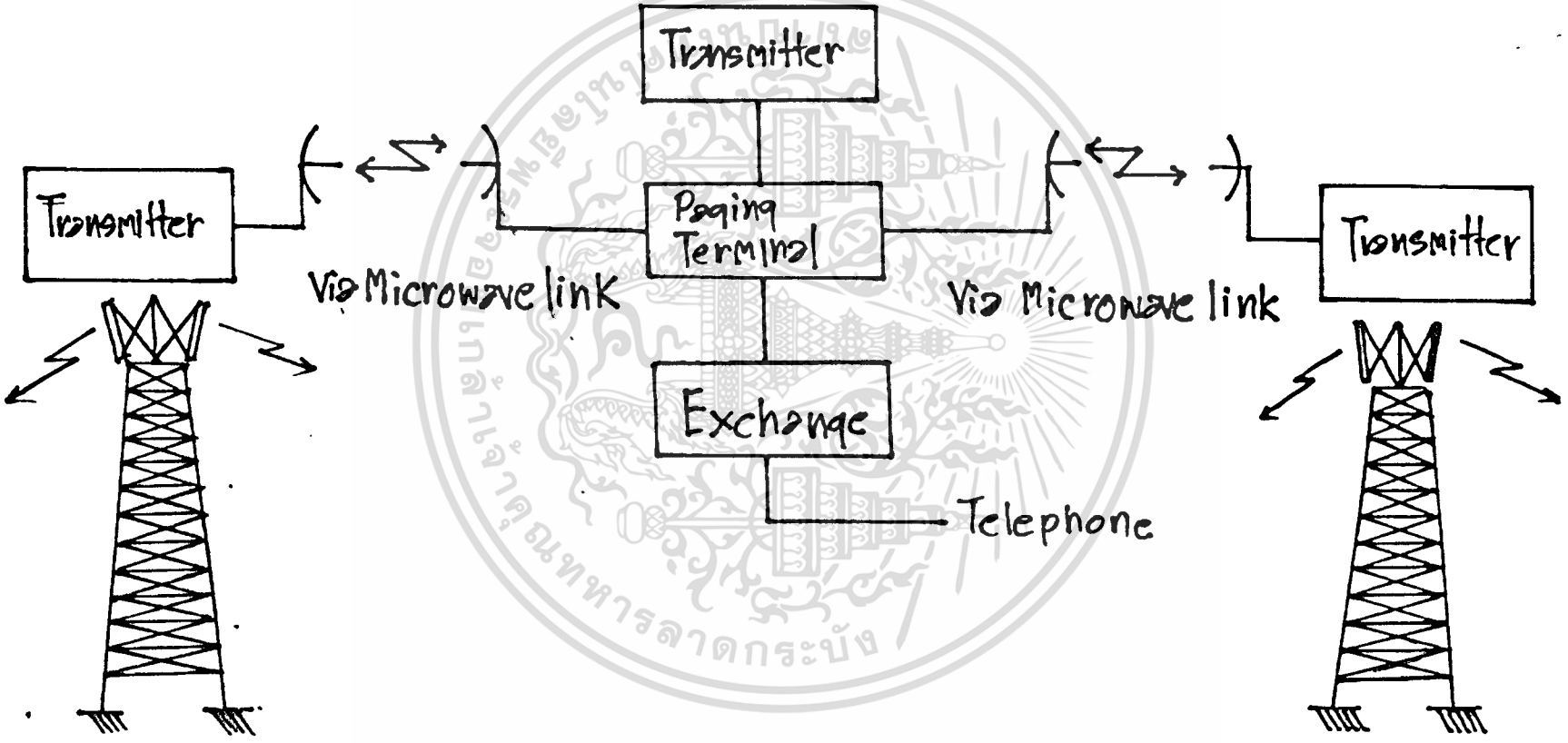
การขยายพื้นที่ของระบบ

เพื่อให้พื้นที่ในการให้บริการแก่ผู้ใช้กว้างออกไป จำเป็นต้องเพิ่มสถานีส่งกระจายออกไปตามจุดต่างๆให้ครอบคลุมอาณาบริเวณที่ต้องการ โดยความถี่ที่ส่งออกไปในเครื่องส่งแต่ละเครื่องยังคงเป็นความถี่เดิม ส่วนข่าวสารจะมีการติดต่อถึงกันด้วยคลื่นวิทยุอีกความถี่หนึ่งหรือใช้ระบบไมโครเวฟปัญหาในเรื่องการรบกวนของเครื่องส่งที่ส่งความถี่เดียวกัน ในพื้นที่ที่เหลื่อมกันนั้นใช้วิธีปรับเฟสของคลื่นที่ส่งออกอากาศให้มีเฟสตรงกัน เมื่อคลื่นมาถึงพื้นที่ดังกล่าวลักษณะของการขยายระบบเป็นดังรูปที่ 1.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 โครงสร้างของวิทยุติดตามตัว



รูปที่ 1.2 ลักษณะการขยายพื้นที่ของระบบ

บทที่ 2

POCSAG

(Post Office Code Standardization Advisory Group)

รูปแบบของสัญญาณเพจเจอร์

สัญญาณข่าวสารที่จะส่งไปยังผู้รับนั้นเป็นสัญญาณรหัสข้อมูลเลขฐานสองเรียงต่อเนื่องกัน ซึ่งรหัสข้อมูลที่ใช้เป็นรหัสของ POCSAG (Post Office Code Standardization Advisory Group) ซึ่งเป็นรูปแบบและรหัสมาตรฐานสำหรับวิทยุติดตามตัวที่ถูกกำหนดโดย CCIR รูปแบบของสัญญาณข่าวสารที่ใช้ส่งประกอบไปด้วยสัญญาณพัลส์ส่วนหน้า (preamble) จำนวน 576 บิท และตามด้วยรหัสคำตั้งแต่หนึ่งชุดขึ้นไป (Batch Structure) ซึ่งแต่ละชุดจะเริ่มต้นด้วยรหัสคำการ ซิงโครไนซ์ (Synchronization Codeword) ตามด้วยรหัสคำเทียม (Idle Codeword) รหัสคำหมายเลขเรียกขาน (Address Codeword) และ รหัสคำข่าวสาร (Message Codeword) ถ้าข้อมูลที่ส่งไม่เต็ม 1 ชุด (Batch Structure) จะส่งรหัสคำเทียมต่อท้ายมาจนเต็ม 1 ชุด รูปแบบของสัญญาณแสดงได้ดังรูปที่ 2.1

สัญญาณพัลส์ส่วนหน้า (Preamble)

ประกอบด้วยรูปแบบที่มีลักษณะของบิทตรงข้ามกัน คือ "1010....1010" จะถูกส่งเป็นจำนวนอย่างน้อย 576 บิท (มีค่าเท่ากับ 1 batch บวก 1 codeword) สัญญาณพัลส์ส่วนหน้านี้ถูกใช้เป็นตัวช่วยในการตรวจหาจุดเริ่มต้นของการส่งเพจเจอร์

โครงสร้างของแบทช์ข้อมูล (Batch Structure)

รหัสคำจะถูกส่งลงในแบทช์ ซึ่งในแต่ละแบทช์จะเริ่มด้วยรหัสคำการซิงโครไนซ์ และตามด้วยเฟรมของรหัสคำอีก 8 เฟรม โดยที่เฟรมแรกเป็นเฟรม 0 และเฟรมสุดท้ายเป็นเฟรมที่ 7

รหัสคำที่บรรจุอยู่ในแบทช์แบ่งออกได้เป็น 4 ชนิด ดังนี้

1. รหัสคำการซิงโครไนซ์ (Synchronization Codeword)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

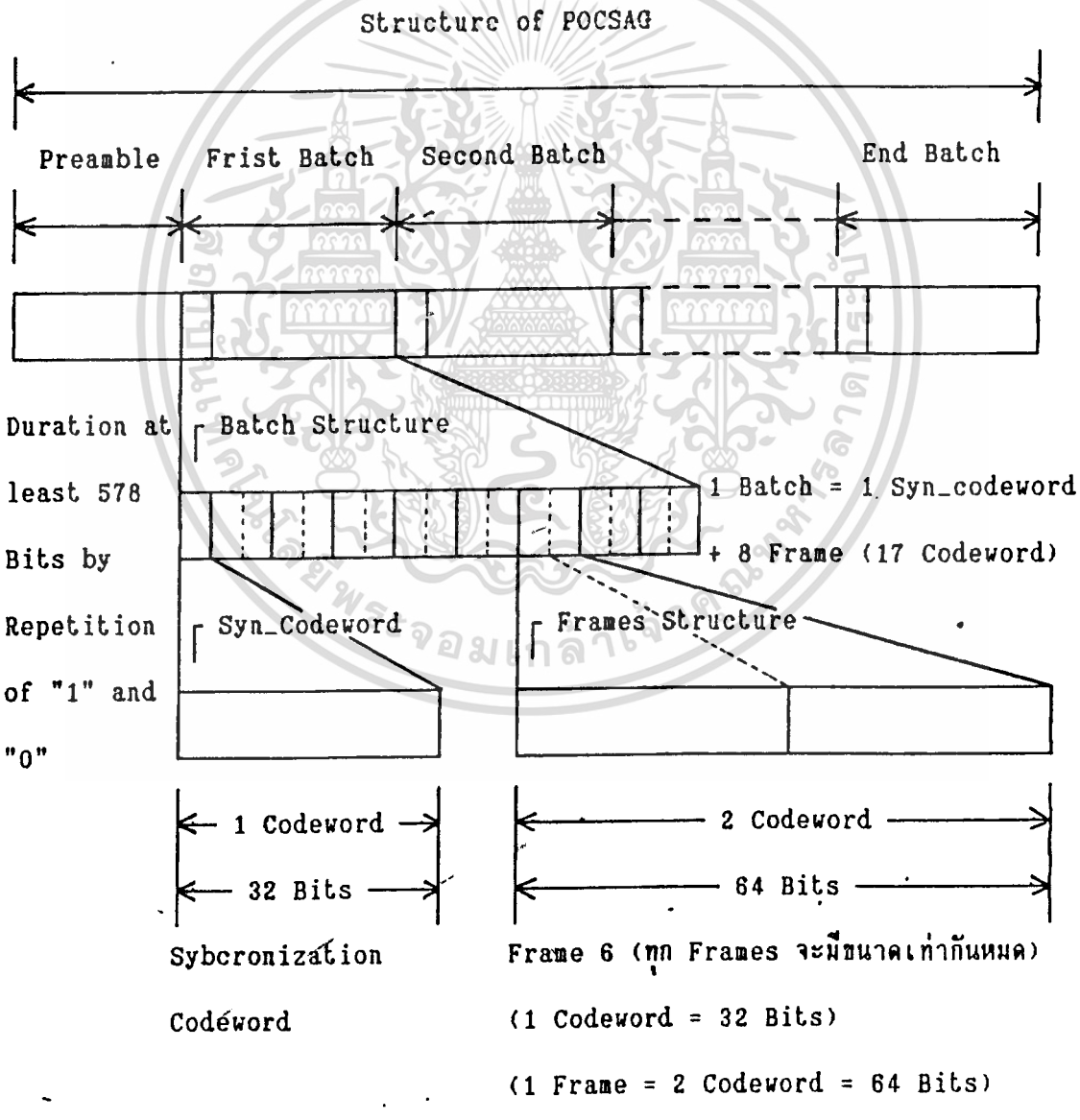
2. รหัสคำของหมายเลขเรียกขาน (Address Codeword)

3. รหัสคำของข่าวสาร (Message Codeword)

4. รหัสคำเทียม (Idle Codeword)

รหัสคำทั้ง 4 ชนิดนี้แต่ละชนิดประกอบด้วยเลขฐานสองจำนวน 32 บิท และแต่ละชนิดมีรูปแบบซึ่งแสดงได้ดังตารางที่ 2.2

โครงสร้างรูปแบบของมาตราฐาน POCSAG

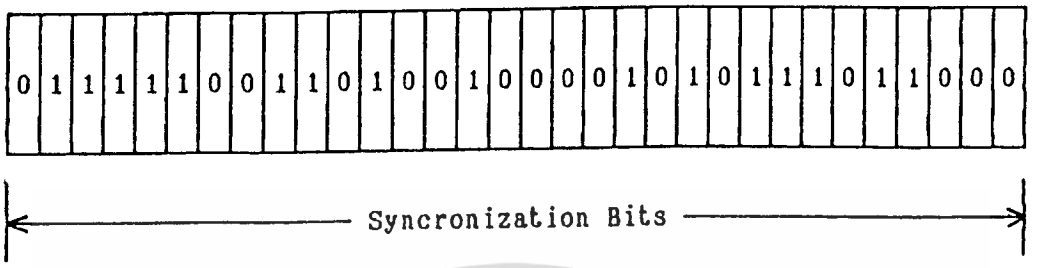


รูปที่ 2.1 รูปแบบของสัญญาณเพจเจอร์มาตราฐาน POCSAG

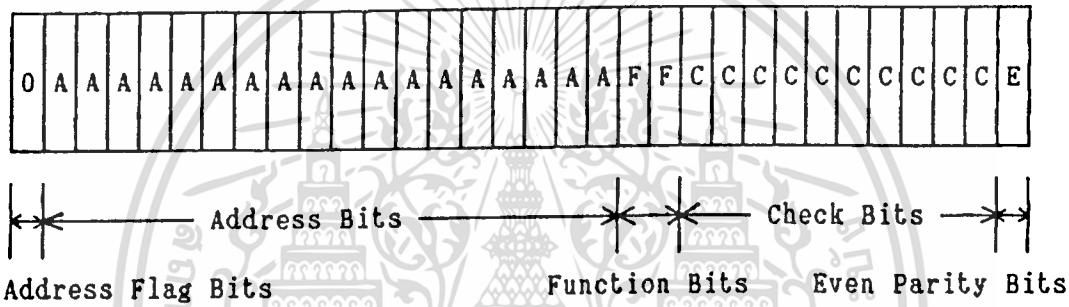
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของรูปแบบรหัสคำทั้ง 4 แบบ

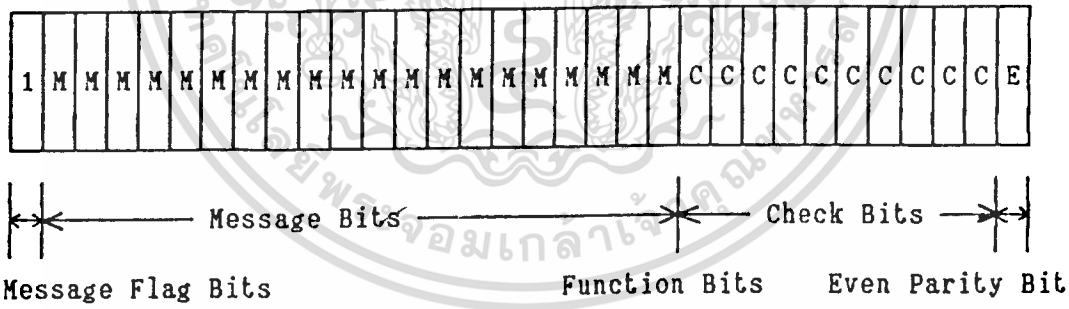
Syncronization Codeword



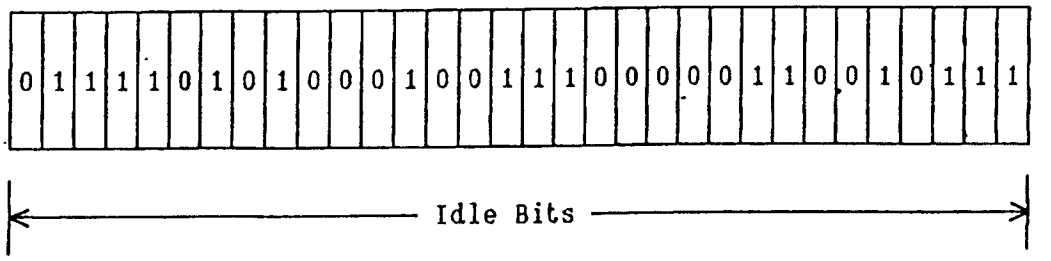
Address Codeword



Message Codeword



Idle Codeword



รูปที่ 2.2 รูปแบบของโครงสร้างรหัสคำต่างๆของสัญญาณเพจเจอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT NO.	SYNCHRONIZATION CODEWORD	ADDRESS CODEWORD	MESSAGE CODEWORD	IDLE CODEWORD
01	0	Adress Flag = 0	Message Flag=1	0
02	1	Adress Bit	Message Bit	1
03	1	Adress Bit	Message Bit	1
04	1	Adress Bit	Message Bit	1
05	1	Adress Bit	Message Bit	1
06	1	Adress Bit	Message Bit	0
07	0	Adress Bit	Message Bit	1
08	0	Adress Bit	Message Bit	0
09	1	Adress Bit	Message Bit	1
10	1	Adress Bit	Message Bit	0
11	0	Adress Bit	Message Bit	0
12	1	Adress Bit	Message Bit	0
13	0	Adress Bit	Messgae Bit	1
14	0	Adress Bit	Message Bit	0
15	1	Adress Bit	Message Bit	0
16	0	Adress Bit	Message Bit	1
17	0	Adress Bit	Message Bit	1
18	0	Adress Bit	Message Bit	1
19	0	Adress Bit	Message Bit	0
20	1	Function Bit	Message Bit	0
21	0	Function Bit	Message Bit	0
22	1	Check Bit	Check Bit	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

034836

BIT NO.	SYNCHRONIZATION CODEWORD	ADDRESS CODEWORD	MESSAGE CODEWORD	IDLE CODEWORD
23	0	Check Bit	Check Bit	0
24	1	Check Bit	Check Bit	1
25	1	Check Bit	Check Bit	1
26	1	Check Bit	Check Bit	0
27	0	Check Bit	Check Bit	0
28	1	Check Bit	Check Bit	1
29	1	Check Bit	Check Bit	0
30	0	Check Bit	Check Bit	1
31	0	Check Bit	Check Bit	1
32	0	Even Parity	Even Parity	1

ตารางที่ 2.1 รูปแบบของรหัสคำ

รหัสคำการซิงโครไนซ์ (Synchronization Codeword)

เป็นรหัสคำที่มีรูปแบบแฉพาะเป็นเอกลักษณ์และไม่มีการซ้ำจากรหัสคำอื่น ๆ รหัสคำชนิดนี้จะอยู่เป็นอันดับแรกของแต่ละแบทช์จึงทำหน้าที่กำหนดจุดเริ่มต้นของแบทช์ด้วย โดยมีรูปแบบของบิทดังตารางที่ 2.2

n.	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
BIT	0	1	1	1	1	1	0	0	1	1	0	1	0	0	1	0	0	0	0	1	0	1	0	1	1	1	0	1	1	0	0	0

ตารางที่ 2.2 รูปแบบของบิทของรหัสคำการซิงโครไนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสคำของหมายเลขเรียกขาน (Address Codeword)

รหัสคำชนิดนี้จะประกอบไปด้วยเลขฐานสองจำนวน 32 บิท ซึ่งบิทแรกจะเป็น "0" เสมอ รูปแบบของรหัสนี้ถูกแสดงไว้ในตารางที่ 2.1 บิทที่ 2 ถึงบิทที่ 19 เป็นรหัสหมายเลขเฉพาะที่กำหนดให้กับเพจเจอร์แต่ละเครื่อง ซึ่งถูกแปลงจากเลขฐานสิบจำนวน 7 หลักมาเป็นเลขฐานสองจำนวน 21 บิท ซึ่งเลขฐานสองที่ถูกแปลงมานั้น จะใช้เพียง 18 บิทที่มีความสำคัญมาก (MSB) ส่งไปกับรหัสของหมายเลขเรียกขาน ส่วนอีก 3 บิทที่มีความสำคัญน้อย (LSB) จะไม่ถูกส่งไปด้วย แต่จะถูกใช้เป็นตัวกำหนดหมายเลขของเฟรมที่จะใช้ในการส่งรหัสของหมายเลขเรียกขานออกไปว่าเป็นเฟรมที่เท่าใดของแบทช์ ตัวอย่างเช่น ต้องการส่งข่าวสารให้กับเครื่องที่มีหมายเลขเรียกขาน 0210570 ก่อนอื่นจะต้องแปลงเลขฐานสิบดังกล่าวเป็นเลขฐานสองดังตารางที่ 2.3 ซึ่งมีทั้งหมด 21 บิทดังนี้ "000110011011010001010" แต่ในการส่งจริงจะใช้เพียง 18 บิทที่มีความสำคัญมากเท่านั้น ส่วน "010" ที่เหลือจะใช้เป็นตัวกำหนดหมายเลขของเฟรมที่จะส่งรหัสหมายเลขเรียกขานชุดนี้ออกไป จากตัวอย่างนี้ $010_2 = 2_{10}$ ดังนั้นรหัสคำนี้จะอยู่ในเฟรมที่ 2 ของแบทช์ ดังนั้นเครื่องรับเพจเจอร์ที่มีหมายเลขเรียกขาน 021570 ก็จะมองหารหัสคำหมายเลขเรียกขานเฉพาะเฟรมที่ 2 ของแบทช์เท่านั้น วิธีการส่งลักษณะนี้จะช่วยลดจำนวนบิทของหมายเลขเรียกขานให้น้อยลงไปได้ถึง 3 บิท และถ้ามีบิทใดใน 3 บิทนั้นเกิด การผิดพลาดไปในการรับส่งข้อมูลก็จะไม่มีผลกระทบใด ๆ ต่อรหัสหมายเลขเรียกขานเลย จึงช่วยเพิ่มประสิทธิภาพของเครื่องรับให้ดีขึ้นด้วย

n	2^n	Binary Code	Decimal	Binary Code	Decimal
0	1	0	-	1	1
1	2	1	2	1	+2
2	4	0	-	0	-
3	8	1	+8	0	-
4	16	0	-	1	+16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

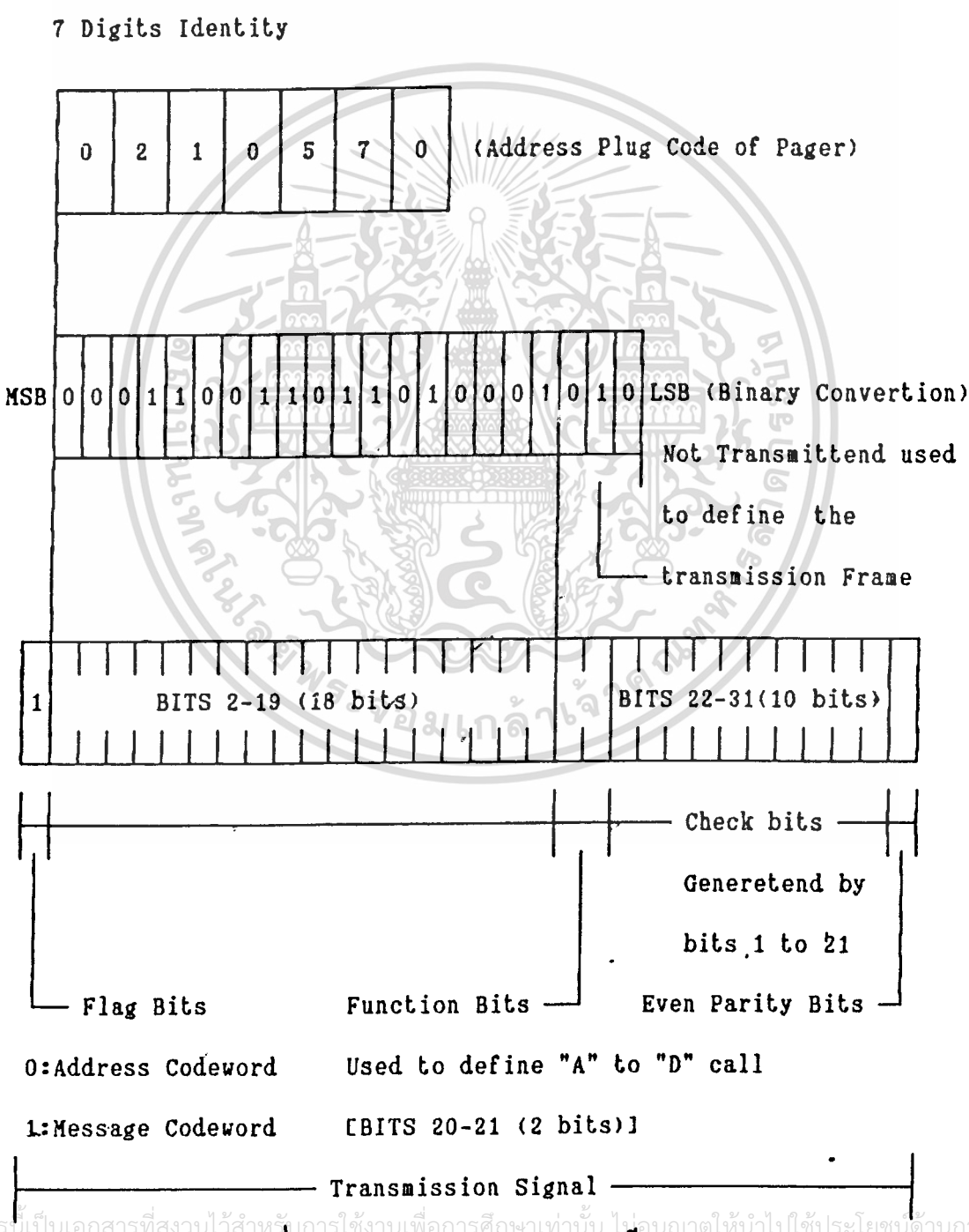
n	2^n	Binary Code	Decimal	Binary Code	Decimal
5	32	0	-	1	+32
6	64	0	-	0	-
7	128	1	+128	0	-
8	256	0	-	1	+256
9	512	1	+512	1	+512
10	1024	1	+1024	0	-
11	2048	0	-	0	-
12	4096	1	+4096	0	-
13	8192	1	+8192	0	-
14	16384	0	-	1	+16384
15	32768	0	-	0	-
16	65536	1	+65536	0	-
17	131072	1	+131072	0	-
18	262144	0	-	0	-
19	524288	0	-	0	-
20	1048576	0	-	0	-
		RESULLTING	---------		---------
		CODE	= 210570		= 17203

ตารางที่ 2.3 ตัวอย่างการแปลงเลขฐานสิบเป็นเลขฐานสองของหมายเลขประจำเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนบิทที่ 20 และ 21 จะแสดงเป็นฟังก์ชั่นบิทมี 4 รูปแบบที่ใช้ในทางปฏิบัติคือ 00, 01, 10, 11 โดยรหัสในส่วนนี้จะทำให้เกิดเป็นเสียง alert tone 1 Beep, 2 Beep, 3 Beep, และ 4 Beep ตามลำดับ ส่วนบิทที่ 22 ถึง 31 จะเป็น Check Bits สำหรับบิทที่ 32 จะถูกบวกเข้าไปเพื่อทำให้เป็นพาริตีคู่ (Even parity bit) รูปแบบของรหัสหมายเลขเรียกขานนี้แสดงได้ดังรูปที่ 2.2

รูปแบบของการเปลี่ยนหมายเลขเรียกขาน 7 ตัว



รูปที่ 2.3 รูปแบบของรหัสหมายเลขเรียกขาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่างการส่งข่าวสารให้แก่เครื่องหมายเลข 0210570 เมื่อรวมกับ function bits, Check bits และ event parity bits แล้วจะได้รหัสคำของหมายเลขเรียกขานดังนี้

n.	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
BIT	0	0	0	0	1	1	0	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	0	1	0	0	0	0	1	1	1	0

รหัสคำข่าวสาร (Message Codeword)

1. รหัสคำของข่าวสารเพจเจอร์รุ่นตัวเลข (Numeric)

รูปแบบของรหัสชนิดนี้ถูกแสดงไว้ในตารางที่ 2.1 รหัสคำของข่าวสารจะประกอบด้วยเวทฐานสองจำนวน 32 บิท โดยที่บิทแรกจะเป็น "1" เสมอ และรหัสคำนี้จะตามหลังรหัสของหมายเลขเรียกขาน รหัสคำของข่าวสารจะสิ้นสุดลงเมื่อมีการส่งรหัสคำของหมายเลขเรียกขานตัวต่อไป หรือมีการส่งรหัสคำเทียม หรือเมื่อการส่งสัญญาณได้มีการสิ้นสุดลง แต่ถ้าข่าวสารที่จะส่งยาวเลยแบทช์นี้นไป ซึ่งก็หมายความว่าจะถูกแทรกด้วยรหัสคำการซิงค์ก่อน แล้วข่าวสารส่วนที่เหลือจึงจะถูกส่งในแบทช์ถัดไปจนหมด โดยที่ไม่ต้องเริ่มด้วยรหัสคำหมายเลขเรียกขานอีกครั้ง ข่าวสารในวิทยุติดตามตัวรุ่นตัวเลขจะมีทั้งข่าวสารที่เป็นเลขฐานสิบ, เครื่องหมาย hyphen (-), เครื่องหมาย [,], ช่องว่าง (Space) และสัญลักษณ์ urgency "U" ซึ่งข้อมูลที่แสดงออกที่หน้าจอของวิทยุติดตามตัวก็จะใช้รูปแบบนี้ รูปแบบของข่าวสารนี้จะใช้เลขฐานสองจำนวน 4 บิทแทน 1 อักขระ ชุดของอักขระสำหรับข่าวสารถูกแสดงในตารางที่ 2.4 การส่งบิทข่าวสารของอักขระแต่ละตัวจะส่งเป็นลำดับโดยเริ่มจากบิทที่ 1 ซึ่งเป็นบิทที่มีความสำคัญน้อยที่สุดก่อน แล้วตามด้วยบิทที่ 2,3,4 ตามลำดับ อักขระตัวต่อๆ ไปก็จะถูกส่งตามกันไปจนครบ 5 ตัว หรือ 20 บิทต่อ 1 รหัสคำข่าวสาร โดยทที่ 20 บิทเหล่านี้จะเรียงจากบิทที่ 2 ถึงบิทที่ 21 ของรหัสคำข่าวสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

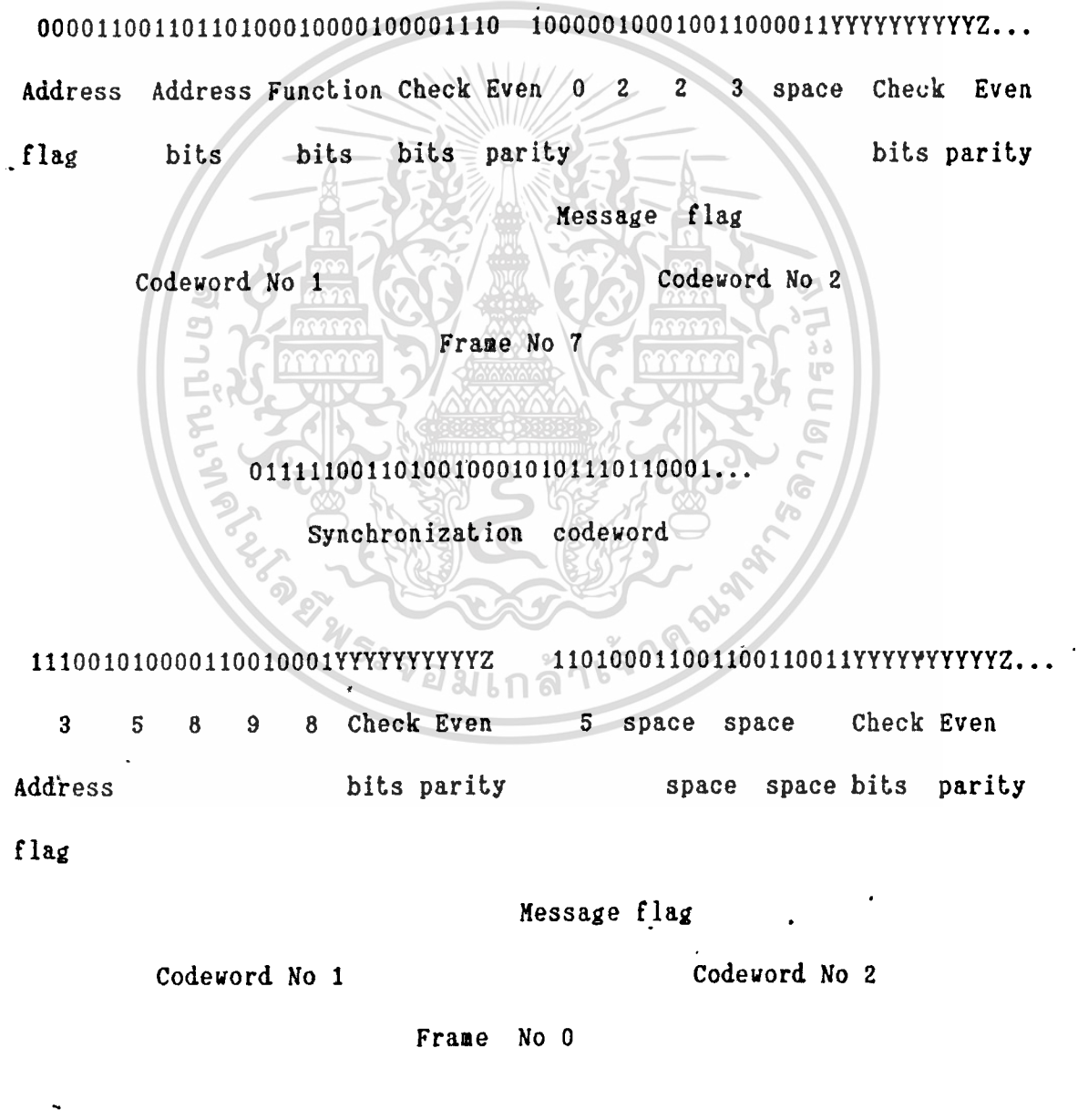
4-bits Combination Bit No. 4321	Displayed Character
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	=
1011	U(Urgency indicator)
1100	Space
1101	hyphen
1110	]
1111	[

ตารางที่ 2.4 ชุดอักขระของข่าวสารเพจเจอร์รุ่นตัวเลข(Numeric)

ตัวอย่างเช่น ต้องการส่งข่าวสารเป็นหมายเลขโทรศัพท์ "0223 358985" ให้กับเครื่องรับหมายเลข 021570 ซึ่งเป็นหมายเลขเดียวกับตัวอย่างที่แล้ว จะสังเกตุข่าวสารดังกล่าวต้องใช้รหัสคำมากกว่าหนึ่งรหัสคำสำหรับการส่ง เนื่องจากในหนึ่งรหัสคำส่งตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้เพียง 5 ตัว เท่านั้น และสมมติว่าการส่งหมายเลขเรียกขานและข่าวสารอยู่ในเฟรมสุดท้าย (เฟรมที่ 7) ของแบทช์พอดี ดังนั้นเมื่อส่งข้อความ "0223" ในรหัสคำที่สองของเฟรมที่ 7 แล้ว จึงต้องขั้นด้วยรหัสคำการซิงค์ของแบทช์ถัดไปก่อนแล้วจึงตามด้วยเฟรมแรก(เฟรมที่ 0) ของแบทช์ถัดไปนั้น ซึ่งบรรจุด้วยข่าวสารส่วนถัดมาคือ "35898" สำหรับรหัสคำแรก และ "5" สำหรับรหัสคำสุดท้ายของเฟรมเดียวกันเพื่อให้กระจ่างชัดมากขึ้นอาจจะพิจารณารูปแบบการส่งในรูปที่ 2.4 ประกอบ



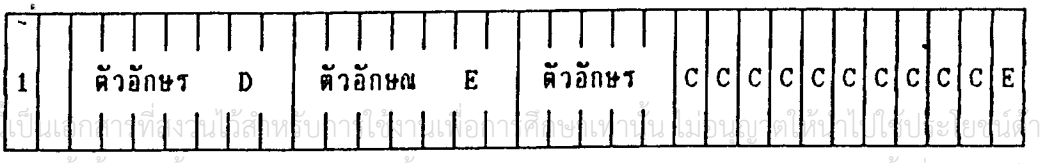
รูปที่ 2.4 ตัวอย่างของรูปแบบการส่งข่าวสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

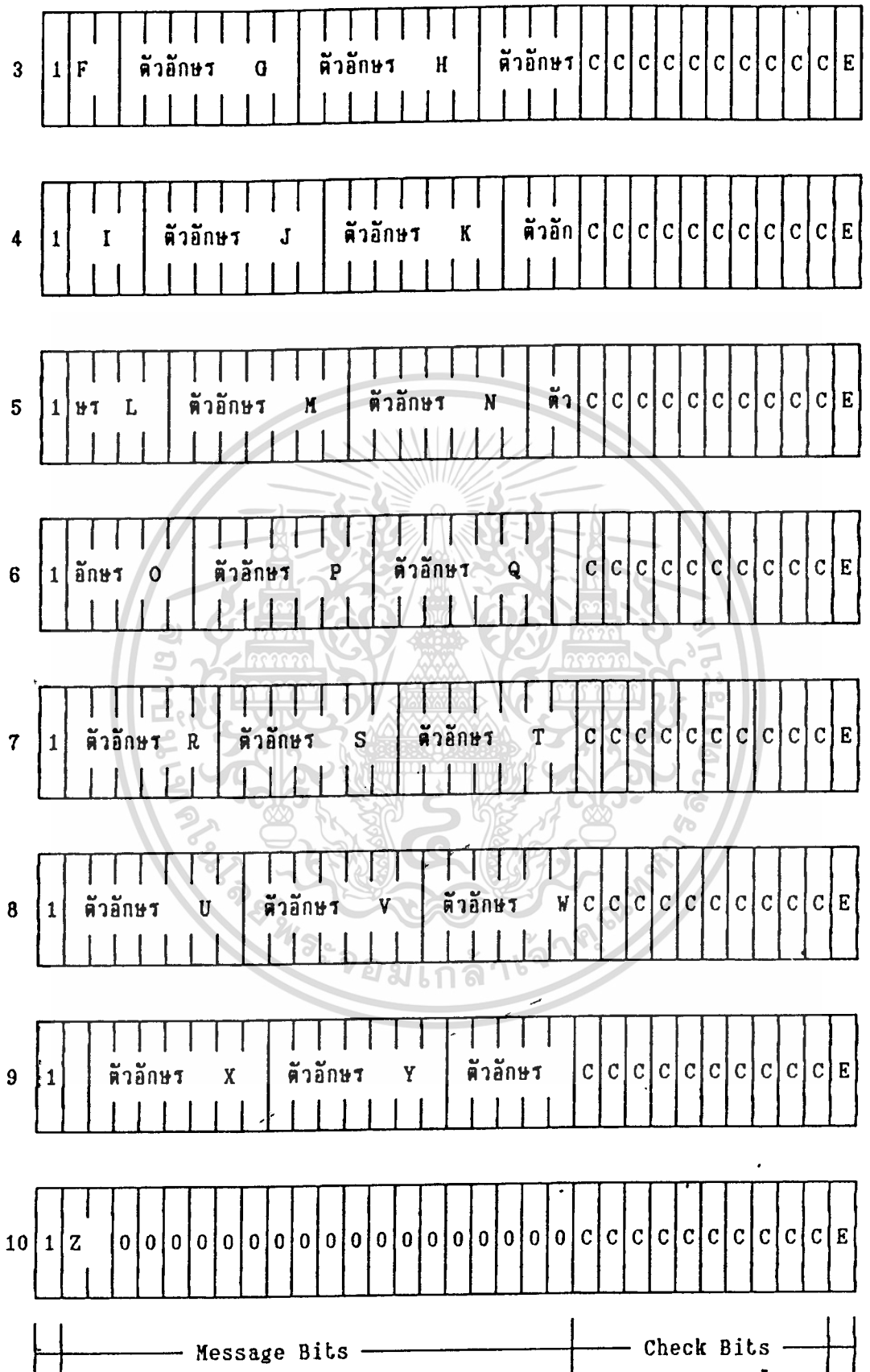
2. รหัสคำของข่าวสารเพจเจอร์รุ่นตัวอักษร-ตัวเลข (Alpha-Numeric)

สำหรับรหัสคำข่าวสารชนิดนี้จะมีลักษณะโครงสร้างทางบิท และรูปแบบเหมือนกันกับรหัสคำข่าวสารของรุ่นตัวเลขทุกประการดังที่ได้กล่าวมาแล้วในหัวข้อที่ 1 ของรหัสคำข่าวสารแต่จะมีความแตกต่างกันตรงจำนวนบิท เนื่องจากเป็นรุ่นตัวอักษร-ตัวเลข จึงจำเป็นต้องใช้จำนวนบิทเพิ่มมากขึ้น ตามมาตรฐานของ POSSAG 1 อักษร หรือ 1 อักขระ จะใช้จำนวนบิท 7 บิท ด้วยกัน ซึ่งจะต่างกับจำนวนบิทของรุ่นตัวเลขที่มีเพียง 4 บิทเท่านั้น และรุ่นตัวอักษร-ตัวเลขนี้รหัสข่าวสารที่มีจะมีถึง 80 รูปแบบ โดยมี ตัวเลข 0-9 อักษรภาษาอังกฤษพิมพ์ใหญ่ และพิมพ์เล็ก รวมทั้งสัญลักษณ์พิเศษต่างๆ ลักษณะตำแหน่งของข่าวสารในบิทข่าวสารนี้จะมีการเลื่อนไปของข้อมูลตามความยาวของข่าวสารที่ส่งมา หมายความว่าในบิทข่าวสารของรหัสคำข่าวสารแรกนั้นจะบรรจุได้ 3 ตัวอักษร แต่อักษรตัวที่ 3 จะมีเพียง 6 บิท ส่วนบิทสุดท้ายของอักษรตัวที่ 3 จะไปอยู่ในบิทข่าวสารของรหัสคำข่าวสารถัดไป ถ้ามีการส่งเป็นข้อความยาวๆ จำนวนบิทของอักษรแต่ละตัวจะถูกเลื่อนออกไป เนื่องจากรูปแบบของบิทข่าวสารกำหนดมีความยาว 20 บิท หรือ 2 ตัวกับอีก 6 บิท นั่นเอง นั้นหมายความว่าจะมีอักษรบางตัวที่อยู่คร่อมระหว่างบิทข่าวสารของรหัสคำข่าวสาร 2 รหัสคำ ส่วนรหัสของแต่ละตัวอักษรแสดงไว้ในตารางที่ 2.5 เพื่อความกระจางจึงขอยกตัวอย่างการส่งข้อความในรุ่นตัวอักษร-ตัวเลข (Alpha-Numeric) ดังนี้

สมมติให้ข้อความยาว 26 ตัวอักษร คือ "ABCDEFGHIJKLMNOPQRSTUVWXYZ" หลังจากรหัสคำหมายเลขเรียกขานแล้วก็จะตามด้วยรหัสคำข่าวสารโดยให้รหัสคำแรกจะบรรจุตัวอักษร A และB กับอีก 6 บิท ของตัวอักษร C และบิทสุดท้ายของตัวอักษร C จะไปอยู่ในรหัสคำข่าวสารที่สองรวมทั้งตัวอักษร DและE กับอีก 5 บิทของตัวอักษร F โดยอีก 2 บิทจะไปอยู่ในรหัสคำข่าวสารที่สาม และจะมีการเลื่อนของข้อมูลไปเรื่อย ๆ ในที่นี้มีการส่งเกิน 20 ตัวอักษรใน เฟรมที่ 8 จะวนกลับมามีลักษณะการเลื่อนเหมือนเฟรมแรก ดังแสดงการเลื่อนของข้อมูลในรูปที่ 2.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงการเลื่อนของบิทข่าวในรหัสคำข่าวสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงรหัสของเทจเจอร์รุ่นตัวอักษร-ตัวเลข

7-bits Combination Bit No. 7654321	Displayed Character
0000110	0
1000110	1
0100110	2
1100110	3
0010110	4
1010110	5
0110110	6
1110110	7
0000110	8
1001110	9
0101110	:
1101110	;
0011110	<
1011110	=
0111110	>
1111110	?

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงรหัสของเพจเจอร์รุ่นตัวอักษร-ตัวเลข

7-bits Combination Bit No. 7654321	Displayed Character
0000001	@
1000001	A
0100001	B
1100001	C
0010001	D
1010001	E
0110001	F
1110001	G
0000001	H
1001001	I
0101001	J
1101001	K
0011001	L
1011001	M
0111001	N
1111001	O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงรหัสของเพจเจอร์รุ่นตัวอักษร-ตัวเลข

7-bits Combination Bit No. 7654321	Displayed Character
0000101	P
1000101	Q
0100101	R
1100101	S
0010101	T
1010101	U
0110101	V
1110101	W
0000101	X
1001101	Y
0101101	Z
1101101	[
0011101	
1011101	]
0111101	^
1111101	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงรหัสของเพจเจอร์รุ่นตัวอักษร-ตัวเลข

7-bits Combination Bit No. 7654321	Displayed Character
0000011	"
1000011	a
0100011	b
1100011	c
0010011	d
1010011	e
0110011	f
1110011	g
0000011	h
1001011	i
0101011	j
1101011	k
0011011	l
1011011	m
0111011	n
1111011	o

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงรหัสของเพจเจอร์รุ่นตัวอักษร-ตัวเลข

7-bits Combination Bit No. 7654321	Displayed Character
0000111	p
1000111	q
0100111	r
1100111	s
0010111	t
1010111	u
0110111	v
1110111	w
0000111	x
1001111	y
0101111	z
1101111	{
0011111	|
1011111	}
0111111	~
1111111	#

ตารางที่ 2.4 ชุดอักขระของข่าวสารเพจเจอร์รุ่นอักษร-ตัวเลข(Alpha-Numeric)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากลักษณะของการเลื่อนในตารางจะเห็นได้ว่าอักษรบางตัวจะไม่อยู่ในรหัสคำข่าวสารเดียวแต่จะคร่อมระหว่างรหัสคำข่าวสาร เช่น อักษร E,F,H เป็นต้น ข่าวสารที่ส่งไม่เต็มแบทต์จำนวนบิทที่เหลือจะถูกเติมให้เต็มด้วยรหัสเทียมดังจะกล่าวในหัวข้อต่อไป

รหัสคำเทียม (Idle Codeword)

ถ้าข่าวสารที่ส่งมาหมดลงก่อนการสิ้นสุดลงของเฟรม รหัสคำเทียมนี้จะถูกใส่แทนรหัสคำปกติ เพื่อให้เฟรมนั้นมีครบสองสองรหัสคำ (64 บิท) และในกรณีที่ข่าวสารที่จะส่งหมดลงก่อนที่จะสิ้นสุดแบทช์ รหัสคำเทียมก็จะได้รับการใส่ไว้เพื่อให้ครบแปดเฟรมด้วยเช่นกัน รูปแบบของรหัสคำเทียมแสดงไว้ในตารางที่ 2.6

n.	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
BIT	0	1	1	1	1	0	1	0	1	0	0	0	1	0	0	1	1	1	0	0	0	0	0	1	1	0	0	1	0	1	1	1

ตารางที่ 2.6 รูปแบบของรหัสคำเทียม

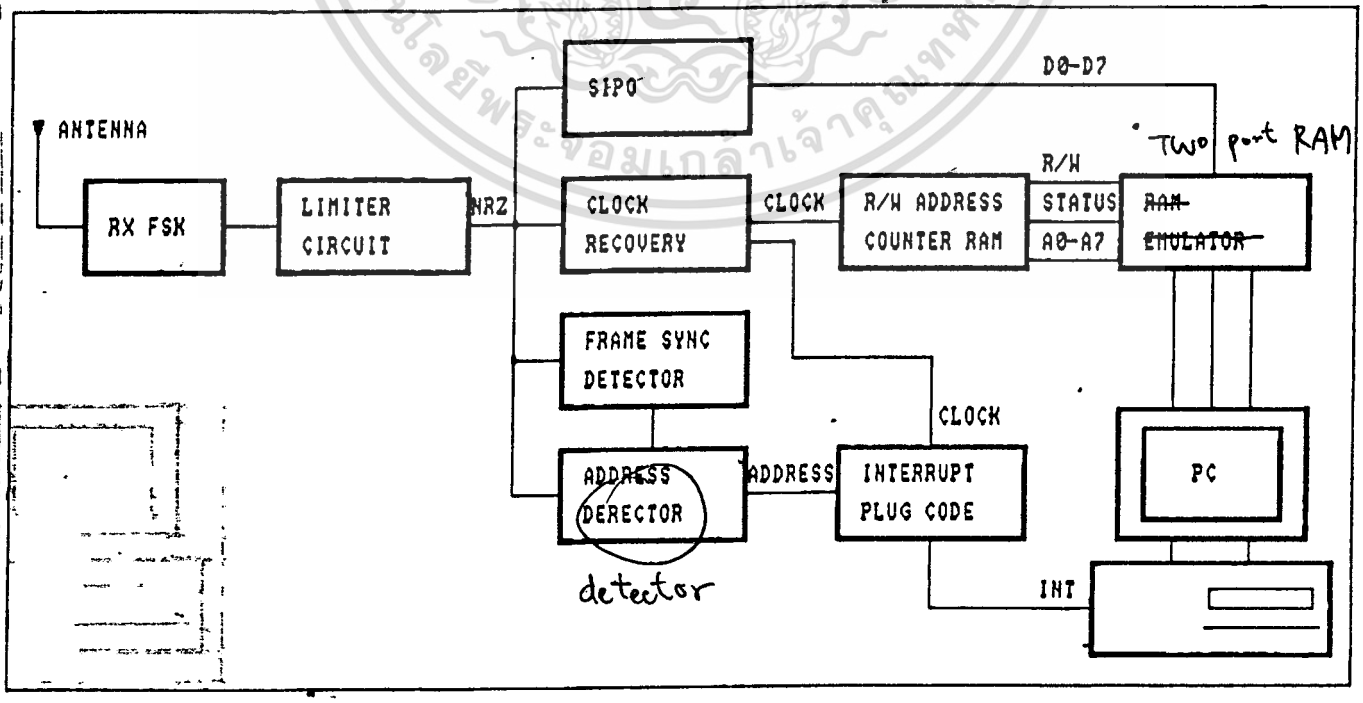
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ภาครับสัญญาณเพจเจอร์

ภาครับสัญญาณเพจเจอร์จะประกอบด้วนส่วนต่างๆ ดังแสดงเป็นบล็อกไดอะแกรมดังรูป 3.1

1. เครื่องรับสัญญาณวิทยุ (FSK Reciever)
2. วงจรลิมิเตอร์ (Limiter)
3. วงจรสัญญาณนาฬิกา (Clocl Recoverry)
4. วงจรตรวจจับสัญญาณเฟรมซิงค์ (Frame Sync Detector)
5. SIPO (Serie In Parallal Out)
6. ภาคกำเนิดสัญญาณอ่าน/เขียนแรมและตัวนับแอดเดรส
   (Read/Write Clock Generator & Address Counter)
7. ภาคแรมอิมูเลเตอร์ (RAM Emulator)
8. ภาคตรวจจับแอดเดรสปลักโคด (Address Plugcode Detector)
9. ภาคกำเนิดสัญญาณอินเตอร์รัพแอดเดรสปลักโคด (Interrupt Address Plug Code)
10. คอมพิวเตอร์ (PC)



รูปที่ 3.1 บล็อกไดอะแกรมภาครับสัญญาณรหัสวิทยุติดตามตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*หมายเหตุ ภาคส่งสัญญาณเพจเจอร์ที่ใช้ทดสอบจะใช้เครื่องทดสอบคลื่นวิทยุสื่อสาร (RF Communication Test Set) ในการทดลองส่ง โดยสามารถเปลี่ยนย่านความถี่ที่ส่งได้เพื่อหลีกเลี่ยงการรบกวนกันของความถี่ที่มีใช้กันอยู่ในปัจจุบันของบริษัทที่เปิดให้บริการ ทางด้านวิทยุติดตามตัว (Pager) ทั้งสองรุ่น

หลักการทั่วไปของภาครับเพจเจอร์

การรับส่งข้อมูลข่าวสารของเพจเจอร์ที่ใช้กันอยู่ในปัจจุบัน และในประเทศไทยใช้การมอดูเลทแบบ FSK (Frequency Shift Keying Modulation) ซึ่งมีย่านความถี่ต่างที่ใช้ดังนี้

1.VHF LOW BAND (33-37,40-50 MHz)

2.VHF HIGH BAND (138-174 MHz)

3.UFH LOW BAND (406-422,435-512 MHz)

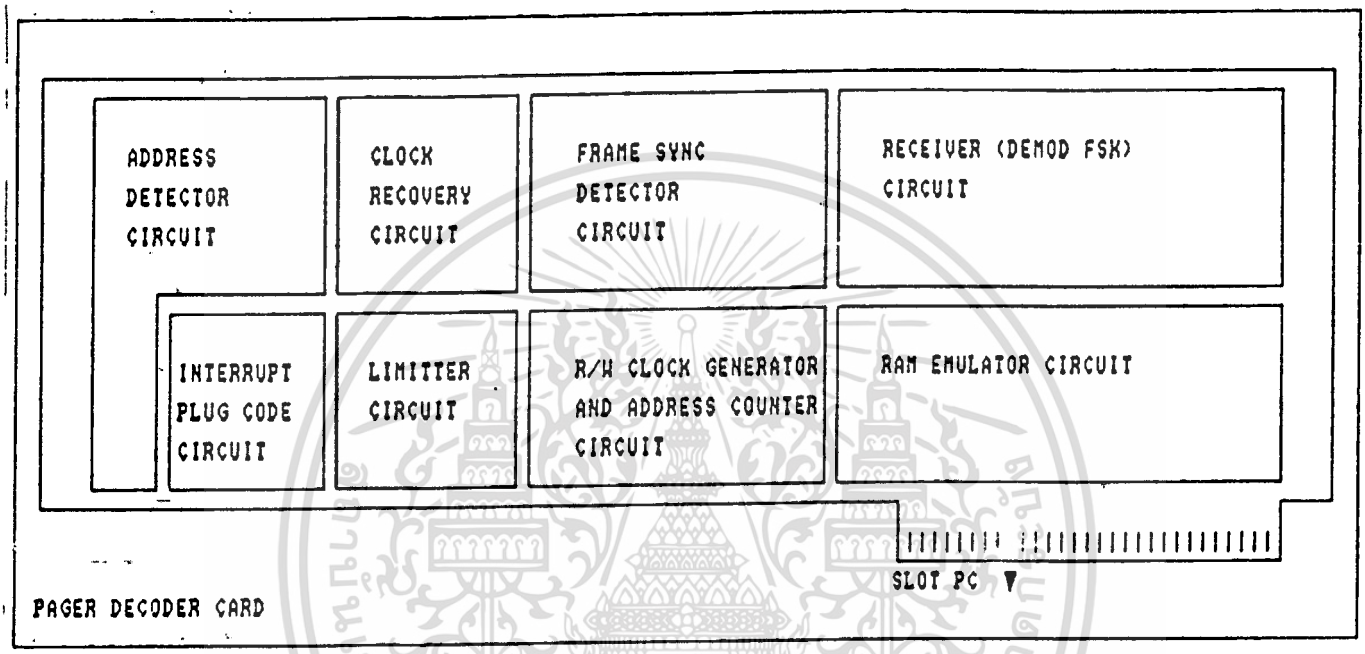
4.UFH HIGH BAND (920-932 MHZ)

สำหรับโครงงานนี้ได้ศึกษาและทดลองใช้ความถี่ 3 ความถี่ด้วยกัน ซึ่งในแต่ละความถี่จะใช้สำหรับในแต่ละรุ่นความถี่ 279.55 MHz ใช้ทดลองสำหรับรุ่นตัวเลข (Numeric) และความถี่ 279.8 MHz ใช้ทดลองสำหรับรุ่นตัวอักษรและตัวเลข (Alpha-Numeric) ส่วนความถี่ 279.748 MHz เป็นความถี่ในการใช้งานจริงยังไม่มีบริษัทไหนนำไปใช้ ดังนั้นจึงนำความถี่นี้ทดลองส่งด้วย RF Communication Test Set ซึ่งได้ใช้ทดสอบตัวโครงงานที่สร้างขึ้นมาว่าสามารถรับได้ทั้ง 2 รุ่นหรือเปล่า สาเหตุที่ใช้ความถี่ย่านนี้เนื่องจากถ้าส่งทำการทดสอบด้วย RF Communication Test Set ที่ข้อมูลที่รับได้ มักจะผิดพลาดเนื่องจากความถี่เดิมที่บริษัทยังมีการส่งอยู่ตลอดเวลา ฉะนั้นจึงเลี่ยงมาใช้ความถี่ใหม่เพื่อให้ง่ายแก่การทดสอบและทดลองและหลีกเลี่ยงปัญหาเนื่องจากความถี่รบกวนกัน

จากรูป 3.1 บล็อกไดอะแกรม สัญญาณข้อมูลข่าวสารที่มอดูเลทแบบ FSK ที่ความถี่ 279.748 MHz จะสามารถรับสัญญาณได้ที่ ภาค Rx ซึ่งในการทดลองได้ใช้เอาท์พุทจากเครื่องวัดความแรงของสัญญาณ(Field strength meter) แทน ซึ่งมีคุณสมบัติที่ใกล้เคียงกับเครื่องรับ FSK ที่ใช้งานจริง และเอาท์พุททที่ได้จะเป็นสัญญาณเบสแบนด์หรือสัญญาณข้อมูล ที่ผ่านการดีมอดูเลทแล้ว แต่สัญญาณที่ได้ยังมีความคมชัดไม่เพียงพอ ดังนั้นจึงต้องทำการปรับปรุงสัญญาณที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหม่ให้เป็นสัญญาณ NRZ ที่มีความคมชัดเพียงพอและมีความถูกต้องมากยิ่งขึ้น โดยการนำสัญญาณข้อมูลที่ได้นั้นไปผ่านภาค Limitter ซึ่งถูกสร้างจากออปแอมป์ที่ต่อวงจรแบบคอมพาราเตอร์ (Comparater) ซึ่งจะได้สัญญาณเอาท์พุทเป็นสัญญาณดิจิตอล



รูป 3.2 ภาครับสัญญาณรหัสวิทยุติดตามตัวที่ออกแบบบนคาร์ดคอมพิวเตอร์

ผลทำให้การตีความหมายของ TTL มีความถูกต้องมากยิ่งขึ้น เนื่องจากเพจเจอร์มีการส่งสัญญาณแบบ อะซิงโครนัส จึงต้องสร้างสัญญาณนาฬิกาขึ้นมาจากสัญญาณ NRZ ที่รับได้ ที่ภาค กู้สัญญาณนาฬิกา (Clock recovery) เพื่อใช้เป็นสัญญาณนาฬิกาควบคุมการทำงานของส่วนต่างๆ ในระบบภาครับสัญญาณ เพื่อให้เกิดการทำงานที่พร้อมเพียงกัน และซิงค์โครนัสกับสัญญาณ NRZ ที่รับได้ และสัญญาณนาฬิกาที่สร้างขึ้นจะต้องมีความถี่ที่ตรงกับบิทเรทของสัญญาณ NRZ ด้วย เนื่องจากเพจเจอร์มีการส่งสัญญาณแบบอนุกรม จึงต้องทำการแปลงสัญญาณจากอนุกรมให้เป็นขนานทีละ 8 บิทที่ภาคSIPO(Serie in parallal out) เพื่อให้สามารถทำการเขียนลง RAM ได้ และเหมาะสมกับไมโครคอมพิวเตอร์ที่มีการรับข้อมูลแบบขนาน ซึ่งใช้สัญญาณนาฬิกาเป็นตัวกำหนดการเลื่อนบิทของข้อมูล ส่วนที่ภาคตรวจจับเฟรมซิงค์จะทำการตรวจสอบว่าสัญญาณข้อมูลที่เข้ามานั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นสัญญาณที่เราต้องการหรือไม่ ซึ่งจะต้องทำการ Decode ในส่วนนี้ให้ตรงกับหมายเลขเครื่องที่เราต้องการ โดยรหัสที่ใช้ Decode ได้จากการนำเอาหมายเลขประจำตัวเครื่องของเพจเจอร์ทำการแปลงเป็นเลขฐานสอง และยังเป็นตัวบอกให้ทราบถึงจุดเริ่มต้นของสัญญาณข่าวสารในแต่ละแบทช์อีกด้วย เมื่อมีสัญญาณที่เราต้องการเข้ามา ภาค SIPO จะทำการแปลงสัญญาณให้เป็นแบบขนานและส่งไปที่ ภาค แรมอีมูเรเตอร์(RAM Emulater)เพื่อทำการเก็บข้อมูลลงใน RAM โดยมีภาค กำเนิดสัญญาณการอ่าน/เขียนและตัวนับแอดเดรส เป็นตัวควบคุมการทำงาน ซึ่งที่ภาคนี้จะใช้สัญญาณจากภาค ตรวจจับสัญญาณเฟรมซิงค์ และสัญญาณนาฬิกาเป็นตัวทริกให้เกิดการเขียนข้อมูลข่าวสารลงใน RAM ทุกๆ 8 บิท ของสัญญาณนาฬิกา อีกส่วนหนึ่งของภาคนี้จะเป็นตัวนับตำแหน่งแอดเดรสไปพร้อมๆ กับการเขียนข้อมูลลงใน RAM จากที่กล่าวมาในส่วนของภาค แรมอีมูเรเตอร์(RAM Emulater)จะทำหน้าที่ในการเก็บข้อมูลข่าวสาร เพื่อสะดวกแก่การอ่านข้อมูลโดยไมโครคอมพิวเตอร์ ในขณะเดียวกันนั้นสัญญาณรหัสคำหมายเลขเรียกขานจะถูกตรวจจับโดยวงจรตรวจจับสัญญาณหมายเลขเรียกขาน ซึ่งตัววงจรตรวจจับหมายเลขเรียกขานสามารถที่จะกำหนดหมายเลขที่ต้องการได้ โดยอาศัยสวิทช์เลือกสัญญาณที่ได้จะถูกส่งไปยังวงจรกำเนิดสัญญาณอินเตอร์รัพ (Interrupt Signal) สัญญาณอินเตอร์รัพที่เกิดขึ้นจะไปกระตุ้น PC (Personal Computer) กำเนิดเสียงหลังจากมีการตรวจจับได้หมายเลขเครื่องที่ต้องการที่เลือกเอาไว้ สาเหตุที่ต้องมีการเก็บข้อมูลข่าวสารลงใน RAM เพราะสัญญาณข้อมูลที่ส่งมานั้นจะทำการส่งมาเพียงครั้งเดียวต่อข่าวสารหนึ่งครั้งเท่านั้น โดยการส่งจะไม่ขึ้นอยู่กับระยะ เวลา เพราะเป็นการส่งแบบ อะซิงโครนัส ทำให้ไมโครคอมพิวเตอร์ไม่รู้ว่าจะมีการส่งข้อมูลข่าว สารมาเมื่อไหร่ อาจทำให้เกิดการรับข้อมูลข่าวสารที่ผิดพลาดได้ ส่วนที่ใช้ในการควบคุมการทำงานของ แรมอีมูเลเตอร์นั้น จะมีตัวเลือกแอดเดรสที่ทำหน้าที่ควบคุมการตัดต่อของบัสแอดเดรสระหว่าง RAM กับ ไมโครคอมพิวเตอร์หรือภาคกำเนิดสัญญาณการอ่าน/เขียนและตัวนับแอดเดรส โดยบัสแอดเดรสของ RAM จะต่อกับบัสแอดเดรสของภาค กำเนิดสัญญาณการอ่าน/เขียน เมื่อมีสัญญาณ status ที่เกิดจากสัญญาณเฟรมซิงค์มาทริกให้ตัวเลือกแอดเดรสทำการต่อบัสแอดเดรสให้ เมื่อสัญญาณ status หมดไปตัวเลือกแอดเดรสจะทำการต่อบัสแอดเดรสของ RAM เข้ากับบัสแอดเดรสของไมโครคอมพิวเตอร์ ทำให้ RAM พร้อมที่จะถูกอ่านโดยไมโครคอมพิวเตอร์ ตัวควบคุมอีกส่วนหนึ่งเป็นตัวตัดต่อบัสข้อมูล ที่จะทำหน้าที่ตัดต่อบัสข้อมูลระหว่าง RAM กับ ไมโครคอมพิวเตอร์หรือภาค SIPPO โดยตัวตัดต่อบัสข้อมูลจะทำการต่อบัสข้อมูลของภาค SIOP กับบัสข้อมูลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RAM เฉพาะเมื่อมี status. เพื่อให้เกิดการเก็บข้อมูลลงใน RAM เมื่อสัญญาณ status หมดไป ตัวตัดต่อบัสข้อมูลจะทำการต่อบัสข้อมูลของ RAM กับบัสข้อมูลของไมโครคอมพิวเตอร์แทน เพื่อให้ไมโครคอมพิวเตอร์ทำการอ่านข้อมูลใน RAM โดยผ่านทางบัสข้อมูลบนเมนบอร์ด ดังนั้นข้อมูลข่าวสารในบัสข้อมูลและบัสแอดเดรสของ RAM จะต่อทำงานสลับไปสลับมาระหว่างภาคใดภาคหนึ่งซึ่งโอกาสที่จะเกิดการทำงานพร้อมกันไม่มีเลย

1.วงจรภาครับสัญญาณ (Receiver Circuit)

สำหรับภาครับของสัญญาณเพจเจอร์ที่ใช้ทดสอบมีด้วยกัน 2 ชนิด คือ

1. Field Strength meter

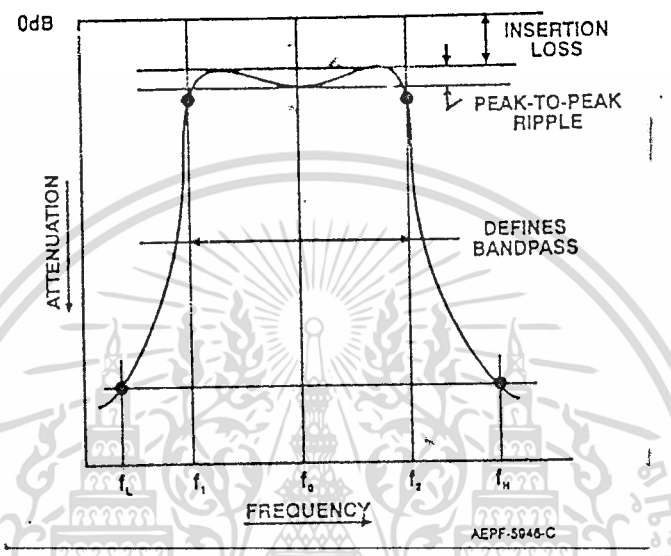
ประกอบด้วยส่วนต่าง ๆ ซึ่งคล้ายคลึงกับเครื่องรับวิทยุ FM ทั่ว ๆ ไปคือมีภาค RF amplifier mixer, bandpass filter, IF amplifier และ demodulator ข้อแตกต่างของเครื่องรับแบบ FM และ FSK ก็คือเครื่องรับ FM จะมีการดีมอดูเลทที่ค่อนข้างเป็นเชิงเส้น (linearity) มากกว่าแบบ FSK ทั้งนี้เนื่องจาก FSK มี ความถี่ของการส่งเพียงสองความถี่สำหรับระดับของแรงดันสองระดับ ("0"และ"1") เท่านั้น ถ้าพิจารณาจากรูปที่ 3.2 ความถี่ทั้งสองก็คือ f_1 และ f_2 การ ที่จะทำให้เครื่องรับมีคุณลักษณะดังกล่าวจะต้องใช้วงจรกรองความถี่ที่เหมาะสม ซึ่ง ในเครื่องรับเพจเจอร์จริง ๆ จะใช้วงจรกรองความถี่แบบผลึกสองชั้ว (Two-Pole Crystal Filter) ที่มีค่า Q สูงมากๆ และมีแบนด์วิดท์แคบมาก ๆ โดยมีความถี่กลาง (center frequency) หรือ IF ตรงกับ f_o ในรูปที่ 3.2 นั่นเอง วงจร กรองความถี่ชนิดนี้มีแบนด์วิดท์แคบมากประมาณ 12 KHz เท่านั้น และประกอบด้วยยอด สองยอดหรือสองชั้วดังนี้

$$f_1 = f_o - 6 \text{ KHz}$$

และ $$f_2 = f_o - 6 \text{ KHz}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ IF จะผ่านวงจรกรองและวงจรขยายหลายชั้นที่เดียว เพื่อเพิ่มความคมของการกรองความถี่ให้ดีขึ้นและแคบขึ้น สาเหตุที่จะต้องกรองความถี่อื่น ๆ ทิ้งก็เพื่อเป็นการ ลดสัญญาณรบกวนซึ่งแฝงมากับความถี่ต่างๆ ออกไปและช่วยให้สัญญาณข้อมูล (baseband signal) ที่ถูกดีมอดดูเลทออกมามีความถูกต้องมากขึ้น



รูปที่ 3.3 การตอบสนองของ Two-pole Crystal Bandpass Filter

อีกภาคหนึ่งที่มีความสำคัญมากก็คือภาคดีมอดูเลเตอร์ ภาคนี้จะต้องมีช่วงการตอบสนองความถี่ที่เหมาะสมไม่แคบจนเกินไป เพราะสัญญาณสแควร์เวฟที่ผ่านการดีมอดูเลทไม่ได้มีเพียงความถี่แรก (1st harmonic) เท่านั้นแต่จะต้องประกอบด้วย ความถี่อื่นๆ อีกบางช่วงเพื่อให้ได้สัญญาณสแควร์เวฟที่สมบูรณ์มากขึ้น ดังนั้นดีมอดูเลเตอร์จึงต้องมีคุณสมบัติที่ต่างจาก FM ดีมอดูเลเตอร์ทั่วไป

เนื่องจากเครื่องรับที่สร้างขึ้นมาเป็นแบบ Single Tune จึงไม่สามารถที่จะทดลองรับจากความถี่ในย่านอื่นได้ ดังนั้นจึงได้อาศัยเอาท์พุทจากเครื่องวัดความแรงของ สัญญาณ (field strength meter) แทน ในส่วนของเครื่องรับและได้ตั้งค่าต่างๆ ของเครื่องเพื่อให้ได้ใกล้เคียงกับคุณสมบัติของเครื่องรับ FSK ที่ใช้งานได้จริงโดยจะตั้งค่า BW (Band Width) ไว้ที่15 KHz และตั้งค่าสวิทช์ FM/AM ไว้ที่ FM โดยกำหนด ความถี่ทดสอบการรับส่งที่ 279.7 48 MHz การใช้เครื่องวัดความแรงของสัญญาณแทน Rx ข้อดีก็คือสามารถเลือกย่านความถี่ได้และยังสามารถทดสอบการรับข้อมูลจริงที่ส่งจากบริษัทต่างๆ ที่ทำการส่งเพจเจอร์ได้อีกด้วยเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รุ่น ตัวเลขของยี่ห้อ wordpage ความถี่ 279.550 MHz และรุ่นตัวอักษร-ตัวเลข ยี่ห้อ Easy call ความถี่ 279.8 MHz

2. FSK Demodulator (Single tune)

วงจรภาครับสัญญาณเป็นวงจรที่อยู่ในภาคแรกของเครื่องรับสัญญาณ pager ซึ่งทำหน้าที่รับสัญญาณในช่องความถี่ของการรับ และส่งสัญญาณ pager โดยใช้การมอดูเลทแบบ FSK (Frequency Shift Keying) ในปัจจุบันการรับส่งข้อมูลข่าวสารของ pager ส่วนใหญ่จะอยู่ในย่านความถี่ต่อไปนี้

1.VHF HIGH BAND (138 - 174 MHz)

2.UHF LOW BAND (406 - 422, 435 - 512 MHz)

คุณสมบัติในขั้นแรกของวงจรภาครับสัญญาณที่พึงจะมีได้แก่

1.มีความไวในการรับสัญญาณสูง (High Sensitivity)

2.ใช้แรงดันไฟในการทำงานต่ำ

3.มีการดีเทคจุดผ่านศูนย์ของสัญญาณ(Zero Crossing Detection)สำหรับใช้กับการรับส่งด้วยระบบ FM FSK ซึ่งมีอัตราเร็วข้อมูลสูง

4.วงจรควรมีขนาดเล็กเพื่อเป็นการประหยัดพื้นที่ในการต่อวงจรรวมกับภาคอื่นบนแผ่นวงจรพิมพ์เดียวกัน

เมื่อภาครับสัญญาณทำการรับสัญญาณที่ส่งมาในช่องความถี่ของ pagerได้แล้ว ก็จะทำการดีเทคเพื่อให้ได้เป็นสัญญาณ Digital Baseband และจะส่งสัญญาณนี้ไปทำการกู้ สัญญาณนาฬิกา และจัดเฟรมเพื่อถอดรหัสคำหมายเลขเรียกขานต่อไป อุปกรณ์ที่เป็นสิ่งสำคัญของภาครับสัญญาณนี้ก็คือ IC เบอร์ MC 3362 ของ MOTOROLA ซึ่งเป็น IC ภาครับ FM ระบบ Dual Conversion คือภาค IF มีการแปลงความถี่ 2 ครั้ง จากความถี่ ที่รับเข้ามาได้จะแปลงลงมาเป็น 10.7 MHz ก่อนแล้วค่อยลดลงเป็น 455 kHz อีกครั้งหนึ่งตาม ลำดับซึ่งเป็นวิธีที่ใช้กันโดยทั่วไปกับวิทยุรับ/ส่งในปัจจุบัน ภายในตัวของ MC 3362 ยังมีภาค Oscillator ซึ่งใช้กับความถี่ได้ 200 MHz แต่ถ้าใช้ภาค Oscillator ภายนอกจะใช้งานได้กับความถี่ 450 MHz แต่ความแรงของสัญญาณ output ของ Oscillator ที่จะนำมาใช้นั้นจะต้องไม่ต่ำกว่า 100 mVrms ภายใน MC 3362 ยังมีภาค Detector แบบ Quadrature และวงจรใช้ขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Meter เพื่อแสดงการรับสัญญาณให้ด้วย นอก จากนี้ยังมีส่วน Buffer ให้แก่ Oscillator ของ IF ทั้ง 2 ความถี่เพื่อความเที่ยงตรงในการ ทำงานรวมทั้งมีวงจรเปรียบเทียบสำหรับใช้การ Detection แบบ FSK MC 3362 นี้แรงดันไฟ เลี้ยงใช้ได้สูงสุด 8 Vdc และต่ำสุด 2 Vdc กินกระแสไฟประมาณ 3.6 mA ที่ 3 Vdc มีความ ไวในการรับ (Sensitivity Input) 0.7 uV ที่ 12 dB SINAD นอกจาก IC MC 3362 จะเป็นอุปกรณ์หลักของการทำงานในภาครับ แล้วก็ยังมีวงจรอื่นๆ ที่เป็นส่วนประกอบที่สำคัญได้แก่ วงจร Input Match, วงจร Colpits Oscillator, และวงจรขยายสัญญาณ Baseband (Baseband Amplifier) ซึ่งสาเหตุที่ต้องมีวงจรขยายสัญญาณ baseband ก็เพื่อที่จะใช้ในการ รับฟังสัญญาณ (Monitor) สำหรับการปรับแต่งวงจรภาครับว่าสามารถรับได้ตรงกับช่องสัญญาณ ของ Pager หรือไม่ โดยเราสามารถที่จะทำได้ด้วยการต่อ Input ของวงจรขยาย สัญญาณ Baseband เข้าที่ขา 13 ของMC 3362 ซึ่งเป็นขา Detector Output ขา 13 นี้จะให้สัญญาณออกมาเป็น Analog

การทำงานของภาครับสัญญาณ

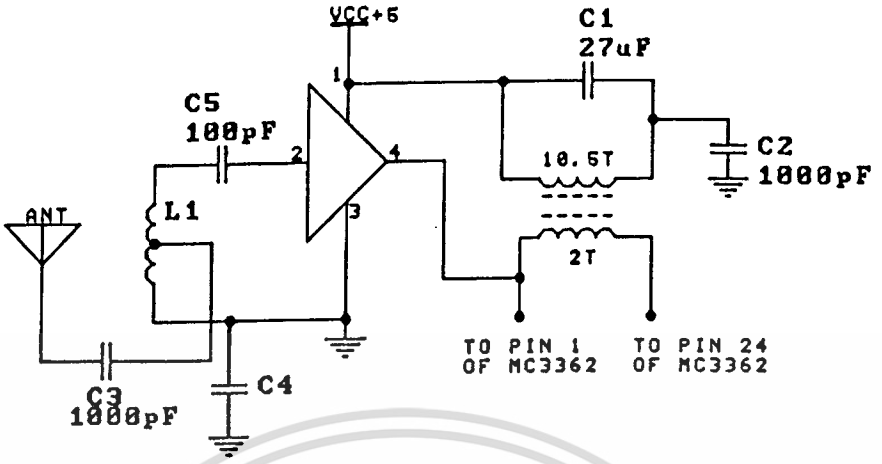
ในระบบการรับ/ส่ง Pager ทั่วๆไปโดยส่วนใหญ่แล้วจะอยู่ในย่านความถี่ของ VHF HIGH BAND ไปจนถึง UHF HIGH BAND ตามที่ได้อธิบายมาแล้วในข้างต้น แต่ในโครงงานวิจัยนี้จะขอยกตัว อย่างการรับสัญญาณ Pager ของ Essy Call ซึ่งใช้ความถี่ 279.80 MHz การทำงานของวงจรภาครับในขั้นแรกจะต้องทำการ Matching Input ที่สายอากาศเพื่อให้มี คุณสมบัติที่เหมาะสมกับย่านความถี่ที่ต้องการรับ จากนั้นจะทำการขยายสัญญาณที่รับเข้ามาได้ให้มี ความแรงขึ้นด้วยวงจร Wide Band Amplifierโดยใช้ IC ขยายความถี่สูงเบอร์ uPC 1651G ซึ่งมีกำลังขยาย 19 dB ที่ 500 MHz สามารถใช้งานได้ในย่านความถี่ตั้งแต่ HF ถึง UHF (1.2 GHz) แต่เนื่องจากความถี่ที่ต้องการรับนั้นเท่ากับ 279.80 MHz ดังนั้นการสร้าง Oscillator ที่ภาครับจะต้อง Oscillate ที่ความถี่ 290.50 MHz ซึ่งหาได้มาจากการนำค่า ความถี่ RF ที่ต้องการรับมาบวกเพิ่มกับค่าความถี่ของ IF 10.7 MHz ก่อนที่จะนำมาทำการ Beat กับความถี่ RF ที่ภาค Mixer ใน IC เบอร์ MC 3362 นั้นมีข้อจำกัดทางคุณสมบัติ ของวงจรภายในบางประการเกี่ยวกับการทำ Oscillate ด้วยวงจรภายใน ที่จะทำให้รับความ ถี่ได้เพียง 200 MHz เท่านั้น ดังนั้นเราต้องจัดวงจร Oscillator ตัวแรกด้วยวงจร Colpits

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

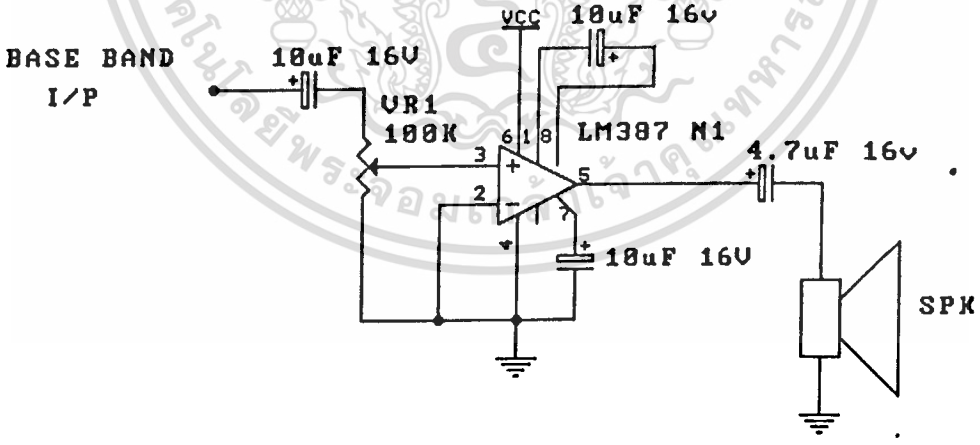
Oscillator ไว้ภายนอก เพื่อที่จะให้ได้ความถี่ 290.50 MHz และมีความแรงมาก กว่า 100 mVrms โดยปรับ LC Tank จึงจะส่งผลให้ภาครับใช้งานในย่านความถี่ VHF HIGH BAND ได้

ภาค Oscillator ตัวที่ 2 เป็นวงจรแบบ Colpits เช่นกัน ทำงานที่ความถี่ 10.245 MHz ควบคุมด้วยคริสตอล (X-TAL) ที่ MC 3362 มี Buffer Output ที่ขา 2 (ขา 2 และขา 3 ใช้งานสลับกันได้) สำหรับในส่วนของ Mixer จะจัดวางจรแบบสมดุล เพื่อลดผลของสัญญาณแปลกปลอม ซึ่ง Mixer ทั้งสองตัวมี Conversion Gain 18 dB และ 12 dB ตามลำดับ โดยมีเสถียรภาพการทำงานที่ไม่ขึ้นกับการเปลี่ยนแปลงแรงดันไฟเลี้ยง เพื่อให้ ออกแบบใช้งานได้ง่ายและมีราคาถูก ตำแหน่งขา IC และวงจรภายในจึงออกแบบมาให้ใช้กับ Ceramic Filter ในส่วนวงจรกรองความถี่ IF ได้ หลังจากผ่านวงจร Filter และ ขยาย IF ทั้ง 10.7 MHz และ 455 kHz แล้วสัญญาณที่ส่งกลับเข้าไปยังวงจร Limitter ซึ่งมีความไว 10 uV ที่ -3.0 dB Limitting ราบเรียบถึง 1.0 MHz จากวงจร Limitter สัญญาณจะถูกส่งมายัง Quadrature Detector ซึ่งต้องมีวงจรภายนอกเพิ่มเติมคือ LC Tank ระหว่าง Vcc กับขา 12 และ R ขนาน 68 kOHM เป็นกำหนดค่า Peek Separation ของ วงจร Detector ถ้าค่าต่ำจะได้ความเป็นเชิงเส้นดีแต่ความไวของวงจร Detector จะลดลง Output ที่ขา 13 จะต้องมีวงจรจัดรูปคลื่นเพื่อให้ได้สัญญาณเสียงที่ถูกต้องแล้วส่งมายัง IC เบอร์ LM 386 N-1 ทำหน้าที่ขยายสัญญาณเสียงสำหรับการ Monitor สัญญาณ Baseband ส่วนวงจร Comparator ที่ขา 14 , 15 ใช้สำหรับ Detect ผ่านจุดศูนย์ของสัญญาณ สำหรับใช้กับการรับและส่งด้วยระบบ FM FSK ซึ่งมีอัตราเร็วข้อมูลสูงตั้งแต่ 2,000 bpS ถึง 3,000 bpS ในส่วนของวงจรขับ Meter ที่ขา 10 ทำงานแบบ Active Low ต่อร่วมกับ Vcc ใช้แสดง ระดับความแรงของสัญญาณที่วัดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

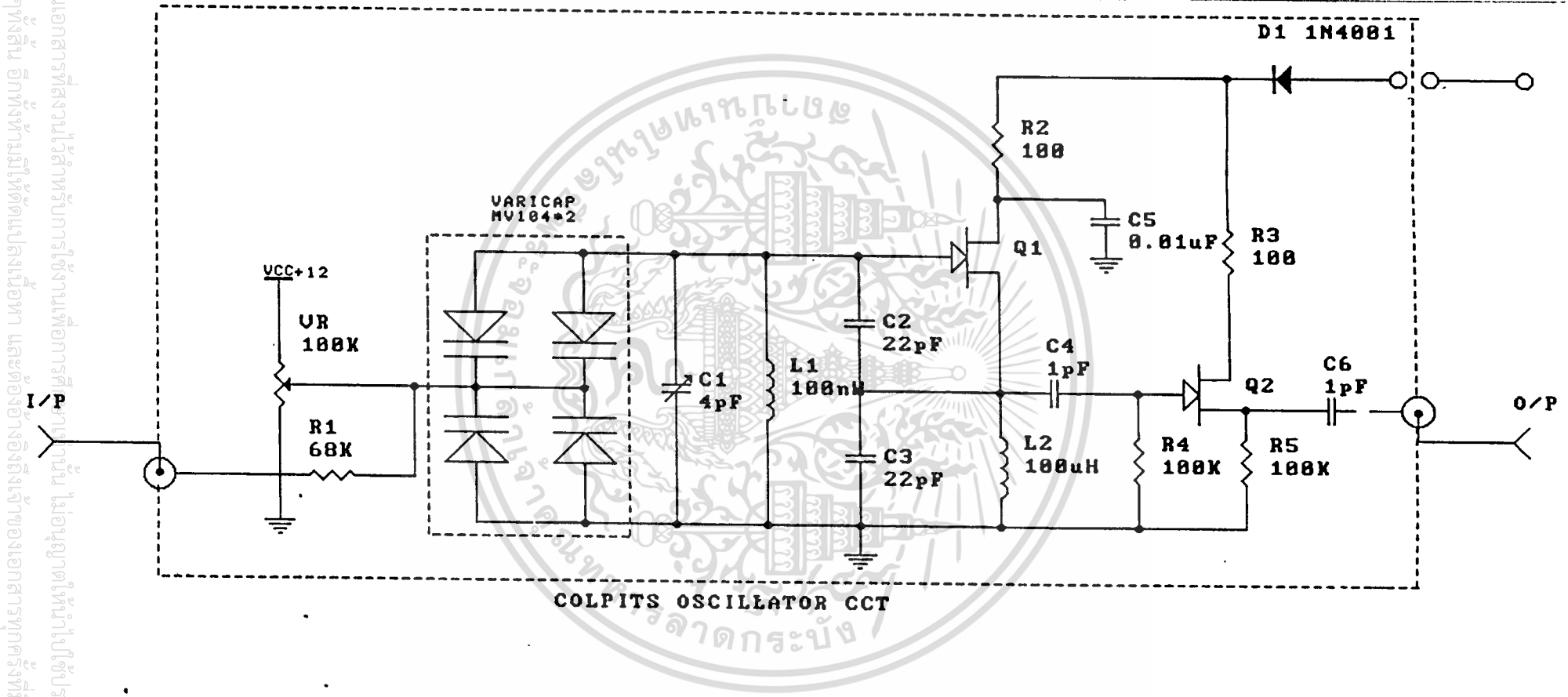


รูปที่ 3.3.1 วงจรแมทชิ่ง RF อินพุท และวงจรขยายความถี่แบนด์กว้าง

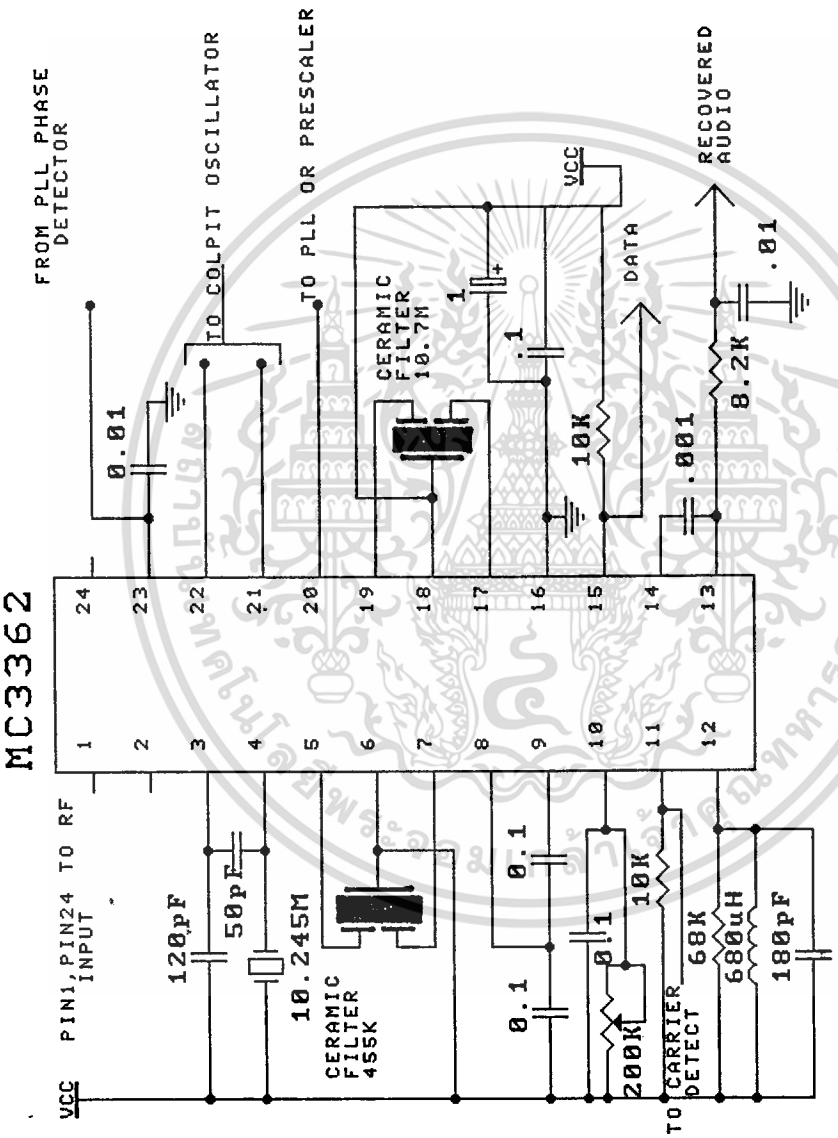


รูปที่ 3.3.2 วงจรขยายสัญญาณเบสแบนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.3 วงจรออสซิเลเตอร์แบบโคลปิกส์(Colpits Oscillator CCT)



รูปที่ 3.3.4 วงจรตัวอย่างภาครับของ MC3362 (Basic Receiver MC3362 CCT)

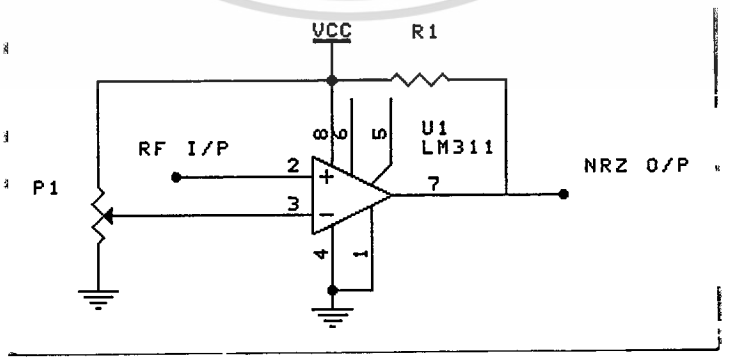
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.วงจรลิมิตเตอร์ (Limiter)

ธรรมชาติของสัญญาณสแควร์เวฟจะประกอบด้วยฮาร์โมนิกคนแรก (1st harmonic) และฮาร์โมนิคอื่นๆ อีกมากมายซึ่งเป็นฮาร์คโมนิคคี่ สามารถแสดงได้ในรูปของอนุกรมฟูเรียร์ (Fourier serie) ของสแควร์เวฟที่มี Duty cycle 50% ได้ดังนี้

$$V(t)=A[1/2 + \sin nt + 1/3\sin 3nt + 1/5\sin 5nt + \ldots\ldots]$$

แต่เครื่องรับโดยเฉพาะภาคขยายสัญญาณ, วงจรกรองต่างๆ และดีมอดูเลเตอร์จะลดทอนฮาร์โมนิกส์สูงๆ ออกไปอย่างมิอาจจะหลีกเลี่ยงได้ ดังนั้นสัญญาณ NRZ ที่ได้จากเครื่องรับจะมีรูป ร่างผิดไปจากความเป็นจริง จะทำให้การตีความของ TTL ผิดพลาดเนื่องจากไม่รู้ว่าเป็นระดับ สัญญาณสูงหรือต่ำ (High or Low logic) สิ่งนี้สามารถแก้ไขได้โดยให้สัญญาณที่ออกจากเครื่องรับผ่านวงจรลิมิเตอร์ก่อน โดยวงจรลิมิเตอร์จะทำหน้าที่เปลี่ยนระดับสัญญาณที่ได้ให้เป็นสัญญาณ NRZ ที่มี Duty cycle 50% ซึ่งวงจรลิมิเตอร์สร้างจากออปแอมป์ที่ถูกต่อแบบคอมพาราเตอร์ (Comparater) โดยอัตราขยายสูงมากๆ ไม่มีการป้อนกลับสัญญาณถูกต่อ เข้าขานอนอินเวอร์ติ้ง เพื่อเปรียบเทียบกับแรงดันอ้างอิง (Reference voltage) ที่ขาอินเวอร์ติ้ง ดังรูป 3.4



รูปที่ 3.4 วงจรลิมิตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

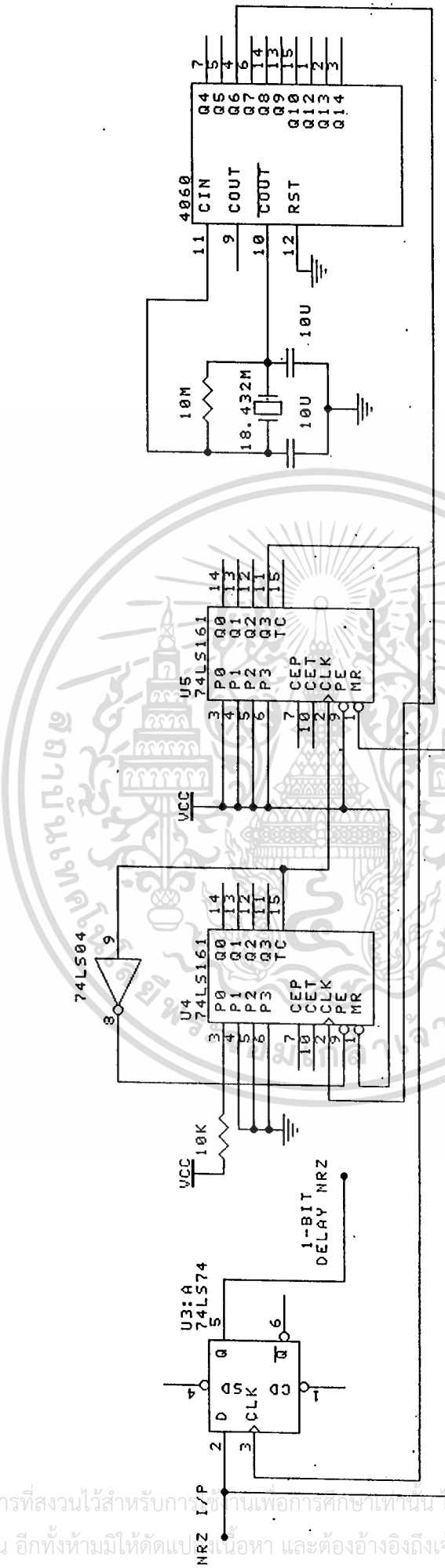
ออปแอมป์ที่ใช้ในวงจรนี้คือ LM 311 เนื่องจากเป็นออปแอมป์ที่ใช้แหล่งจ่ายไฟเดี่ยวซึ่งเหมาะสมกับวงจรโดยรวมที่มีเฉพาะแรงดัน TTL เท่านั้น จึงไม่ต้องหาแหล่งจ่ายไฟลบเพิ่มเติม

3.วงจรกู้สัญญาณนาฬิกา (Clock Recovery)

เนื่องจากการส่งข้อมูลของเพจเจอร์เป็นการส่งแบบอะซิงโครนัส จึงไม่มีสัญญาณนาฬิกาควบคู่มาด้วย ทางด้านรับจึงต้องสร้างสัญญาณนาฬิกาขึ้นมาใหม่จากสัญญาณ NRZ ที่ได้รับเพื่อใช้ควบคุมการทำงานของส่วนต่างๆในภาครับให้ทำงานพร้อมเพรียงกัน (Synchronouse) สัญญาณนาฬิกาที่สร้างขึ้นมาใหม่จะต้องมีความถี่ตรงกับบิทเรท (Bit rste) ของ NRZ และยังต้องซิงโครนัส (Synchronous) กับสัญญาณ NRZ ด้วย รูปที่ 3.5 เป็นวงจรกู้สัญญาณนาฬิกา เริ่มจากออสซิเลเตอร์ (Oscilator) อ้างอิงโดยใช้คริสตอล (Cycle) 18.432 MHz ต่อรวมกับตัวเก็บประจุและ 74HC4060 ทำหน้าที่ป้อนไฟให้คริสตอลรวมทั้งหารความถี่จากคริสตอลด้วย 64 ไปในตัว สัญญาณนาฬิกาจาก 4060 จะถูกหาร 15 ด้วย 74LS161(U_4)โดยใช้เทคนิคการตั้งค่าหารเริ่มต้น (resent) ที่ 1 แล้วดึงสัญญาณจากขาริปเปิล (ripple) ผ่านนอตเกทเพื่อโหลด (load) ตัวมันเอง ดังนั้น 74LS161(U_4) นี้จะเริ่มนับ 1 ถึง 15 แล้วนกลับมาเริ่มที่ 1 ใหม่ หลังจากนั้นสัญญาณนาฬิกาจากขาริปเปิลจะถูก 74LS161(U_5) หารด้วย 16 เพื่อให้ได้สัญญาณนาฬิกาความถี่ 1200 บิทต่อวินาที

สัญญาณ NRZ ที่ผ่านลิมิเตอร์มาแล้วถูกใช้เป็นตัวควบคุมให้สัญญาณนาฬิกาซิงโครนัสกับตัวเอง โดยนำ NRZ มาหน่วงเวลาด้วยน็อทเกท 3 ตัว เพื่อให้ขอบขาขึ้นและขอบขาลงของ NRZ ที่หน่วงไว้เหลื่อมกับ NRZ เดิม แล้วเทียบสัญญาณทั้งสองด้วยเอ็กซ์คลูซีฟออร์ (XOR) จึงทำให้เกิดพัลส์ "0" เล็กๆทุกๆการเปลี่ยนแปลงของสัญญาณ NRZ แต่ละครั้ง และจะเป็น "1" ในช่วงเวลาอื่น พัลส์ที่ได้นี้สามารถใช้เคลียร์ 74LS161(U_6) เพื่อให้สัญญาณนาฬิกาที่ได้ซิงค์กับ NRZ ตลอดเวลา พัลส์ที่ได้นี้สามารถใช้เคลียร์ ดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรกลับสัญญาณนาฬิกา(Clock Recovery)

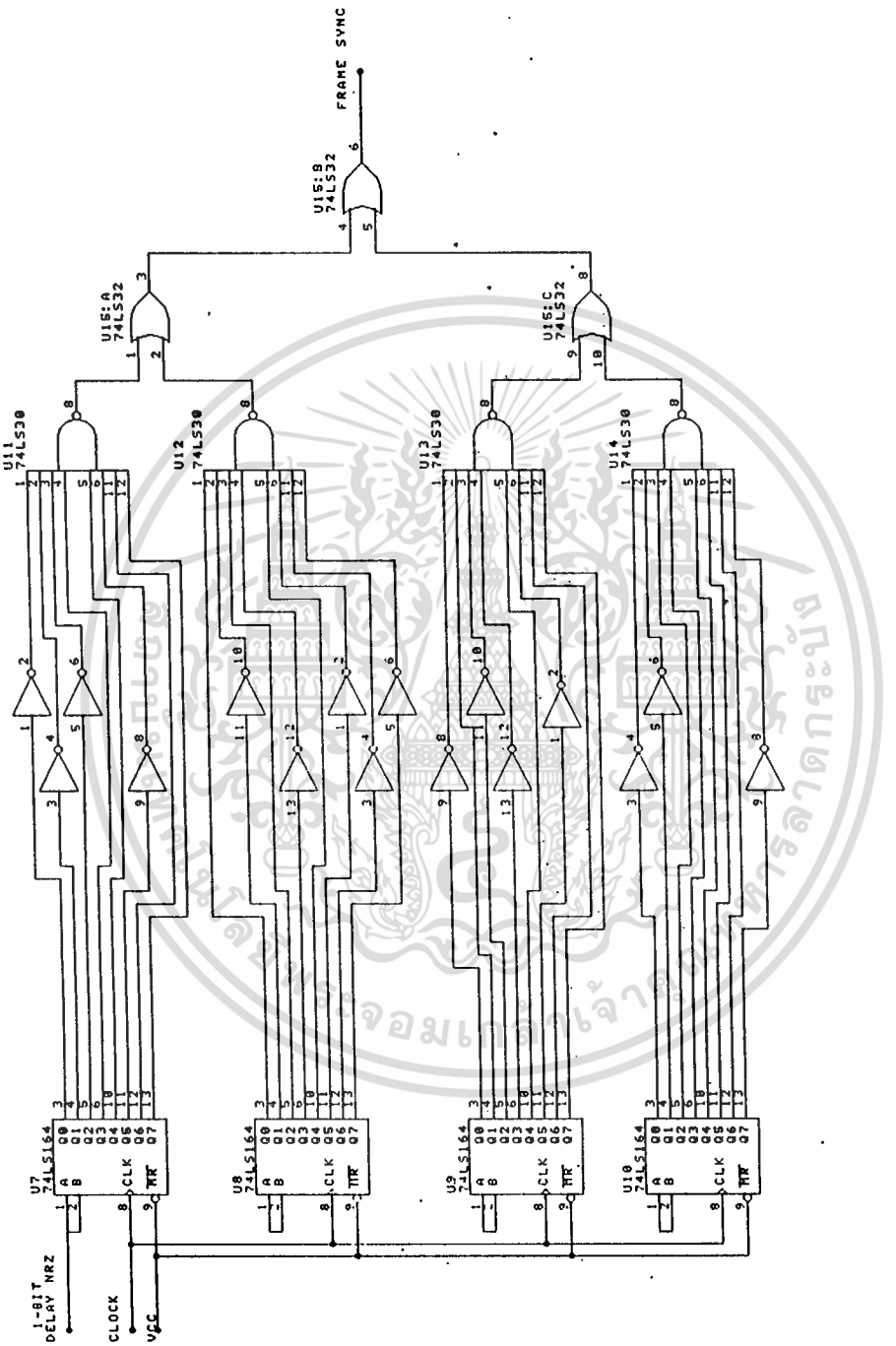
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.วงจรตรวจจับเฟรมซิงค์(Frame Sync Detector)

สัญญาณเฟรมซิงค์นับว่ามีความสำคัญในการส่งข้อมูลเพจเจอร์อย่างมากเพราะจะเป็นตัวบอกให้ทราบจุดเริ่มต้นของการส่งข้อมูลในแต่ละแบทช์ และเครื่องรับยังอาศัยสัญญาณนี้สำหรับใช้ในการตรวจสอบระยะเวลาการส่งว่าสิ้นสุดลงเมื่อใดอีกด้วย สัญญาณเฟรมซิงค์จะถูกสร้างจากรหัสคำการซิงค์ทั้ง 32 บิทซึ่งเป็นเอกลักษณะเฉพาะตัวอยู่แล้ว วิธีการตรวจจับสัญญาณเฟรมซิงค์จะเริ่มจากการนำสัญญาณ NRZ ผ่านดีฟลิบฟลอบ (D-Flipflop) เพื่อหน่วงสัญญาณไปหนึ่งบิท สาเหตุที่ต้องการหน่วงก็เพราะต้องการให้สัญญาณ NRZ ถูกเลื่อน (Shift) จากเดิม 32 บิท ไป 33 บิท (1 บิทดีฟลิบฟลอบ และ32 บิท ที่ชิฟริจิสเตอร์แบบ 8 บิท 4 ตัว) พร้อม ๆ กับที่บิทแรกซึ่งถัดจากรหัสคำการซิงค์เริ่มต้นนับแอดเดรส (Address)

จากตารางที่ 2.1 จะเห็นว่ารหัสคำการซิงค์ประกอบด้วย "0" 16 ตัว "1" 16 ตัว สัญญาณเฟรมซิงค์ถูกกำหนดให้เกิดขึ้นที่บิทสุดท้ายของรหัสคำการซิงค์ ดังนั้นจึงต้องต่อขาเอาท์พุทของ 74LS164 ที่คาดว่าจะเป็น "0" ในช่วงที่เกิดเฟรมซิงค์พอดี นำไปผ่านน็อทเกทแล้วนำไปเข้าแนนด์เกท 74LS30 ส่วนขาเอาท์พุทที่เหลือของ 74LS164 ถูกต่อเข้าแนนด์เกทโดยตรงดังรูป 3.6 หลังจากนั้นสัญญาณจากขา 8 ของแนนด์เกททั้ง 4 จะถูกออร์(OR) กันจนได้สัญญาณเฟรมซิงค์ออกมาทุกๆ บิทสุดท้ายของรหัสคำการซิงค์ ดังแสดงในรูปที่ 3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

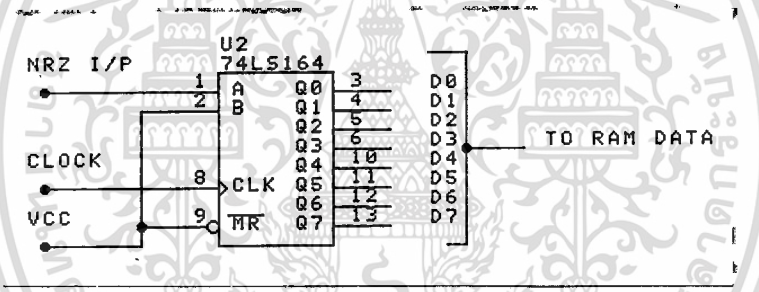


รูปที่ 3.6 วงจรตรวจจับเฟรมซิงค์(Frame Sync Detector)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.SIPO (Serie In Parallal Out)

เนื่องจากข้อมูลที่รับได้จากสัญญาณเพจเจอร์เป็นข้อมูลแบบอนุกรม จึงจำเป็นที่ต้องแปลงข้อมูลเหล่านั้นให้เป็นแบบขนาน 8 บิท เพื่อให้สามารถเขียนลงแรมได้ หลักการของ SIPO ก็คือการเลื่อนบิทของข้อมูลไปเรื่อยๆ และทุกๆ 8 บิท จะถูกอ่านจากแรม 1 ครั้ง ซึ่งวงจรที่ใช้ก็คือ 74LS164 (8 bit shift register) และใช้สัญญาณนาฬิกาเป็นตัวกำหนดการเลื่อนบิทของข้อมูลที่อินพุทขา 1 ของ 74LS164 จะเลื่อนไปที่ Q_0 (ขา 3) เมื่อสัญญาณนาฬิกาเป็นขอบขอขึ้น และข้อมูลจะถูกเลื่อนจาก Q_0ไป Q_1-Q_2-Q_3-Q_4-Q_5-Q_6-Q_7 แล้วจึงมีการอ่านข้อมูลจากแรม 1 ครั้ง ดังนั้นสัญญาณ จาก Q_0-Q_7 จะถูกต่อกับ D_0-D_7 ดังรูป 3.7



รูป 3.7 วงจรอินพุทอนุกรมเอาท์พุทขนาน (SIPO)

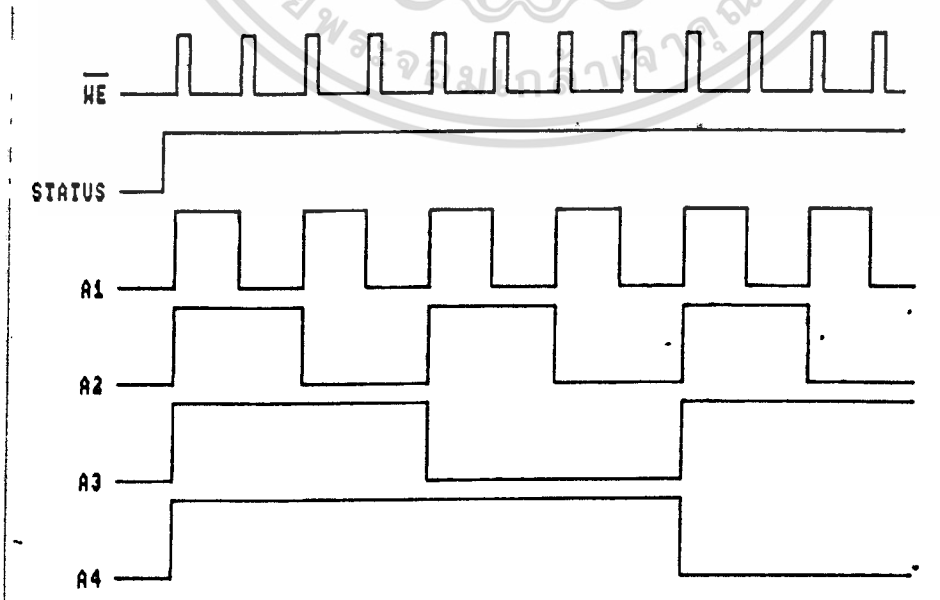
6.ภาคกำเนิดสัญญาณอ่าน/เขียน แรม และสัญญาณนับแอดเดรส

(Read/Write Clock Generator and Address Counter)

ในช่วงแรกของภาคนี้จะกำเนิดสัญญาณการอ่านและเขียนข้อมูลให้กับขา WE(ขา 27) ของแรม 6264 ซึ่งจะเป็นพัลส์เล็กๆ ซึ่งมีความกว้างเท่ากับพัลส์ของสัญญาณนาฬิกา โดยในขั้นแรกสัญญาณนาฬิกาที่มีบิทเรท 1200 บิทต่อวินาที จะถูกหารด้วย 2, 4, 8, 16 ที่ U16 ดังในรูปที่ 3.8 แล้วจึงนำเอาท์พุทจากขา Q_0, Q_1, Q_2, สัญญาณนาฬิกา และสัญญาณ STATUS มาทำการแนนด์กันหมด และจะได้พัลส์ "0" แคบๆ ทุกๆ 8 บิทของสัญญาณนาฬิกา หรือ สัญญาณ NRZ นั้นเอง ดังนั้นสัญญาณการเขียนนี้จะเกิดขึ้นเพื่อให้แรม ถูกเขียนข้อมูลทุกๆ 8 บิทหรือ 1 ไบท์ของเพจเจอร์ อีกส่วนหนึ่งของภาคนี้เป็นตัวนับแอดเดรส สำหรับแรม 6264 ซึ่งมีแอดเดรส

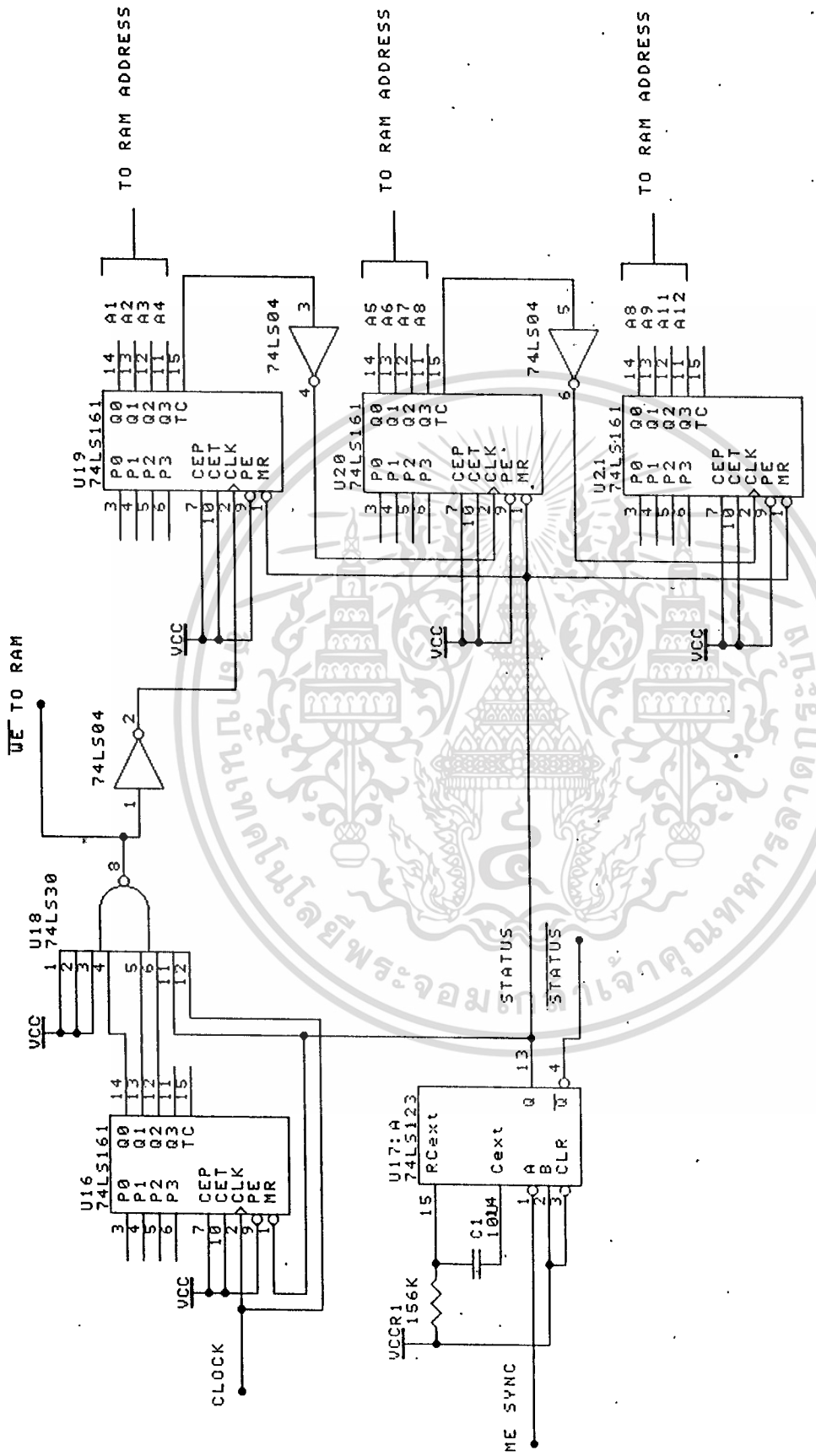
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมด 13 เส้น (A_0-A_{12})แต่เนื่องจากวงจรของแรมนี้ติดต่อ(Interface)กับคอมพิวเตอร์ผ่านทางบัสบนเมนบอร์ด(bus slot on main board) โดย ใช้บัสข้อมูล (data bus) เพียง 8 บิท (D_0-D_7) ขณะที่คอมพิวเตอร์ที่ใช้ในการทดลองเป็นแบบ 16 บิท (CPU 80286) คือ มีบัสข้อมูล 16 เส้น (D_0-D_{15}) จึงทำให้ CPU ไม่สามารถอ่านข้อมูลจาก D_8 -D_{15} ได้ (ดูรายละเอียดเกี่ยวกับบัสข้อมูลในภาคผนวก) วิธีการแก้ไขก็คือการใช้แอดเดรสคู่ของแรมในการบันทึกและอ่านข้อมูลเพจเจอร์เท่านั้น ซึ่ง หมายความว่า CPU จะต้องอ่านข้อมูลจาก D_0-D_7 (ไบท์ล่าง) ของบัสข้อมูลโดย ไม่สนใจข้อมูลจากD_8-D_{15} เลยในทางปฏิบัติจึงได้ทำการต่อขาA_0 (ขา10) ของ 6264 ลงกราวด์ และเหลือเฉพาะ A_1-A_{12} ที่จะเป็นตัวกำหนดแอดเดรส วงจรตัวนับแอดเดรส (Address Counter) จึงมีเอาท์พุทเหลือเพียง 12 เส้น สำหรับ A_1-A_{12} ของแรม การนับแอดเดรสของแรมจะเป็นคู่เท่านั้น (0,2,4,8,...) ในรูปที่ 3.8 แสดงถึงการสร้าง แอดเดรสโดยการใช้วงจรหารทำการหารพัลส์ อ่าน/เขียน ที่ออกจาก U_{10} โดยถูกกลับเฟสก่อนเพื่อให้ A_1 เปลี่ยนแปลงพร้อมๆกับที่มีการเขียนข้อมูลลงแรมดังรูป 3.9 จะสังเกตุเห็นว่าขา 1 ของ 74LS 161 ทุกตัวจะถูกต่อกับสัญญาณ STATUS เพื่อเริ่มการทำงาน เฉพาะตอนที่มีการส่งสัญญาณเพจเจอร์มาเท่านั้นและมีการนับ แอดเดรสทุกเส้นเริ่มต้นพร้อมๆกัน ดังนั้น U_{10} จะกำเนิดสัญญาณสำหรับแอดเดรส A_1-A_4, U_9 สำหรับ A_5-A_8 และ U_4 สำหรับ A_9-A_{12} ดังรูป 3.8



รูปที่ 3.9 Timming Diagram ของสัญญาณตัวนับแอดเดรส

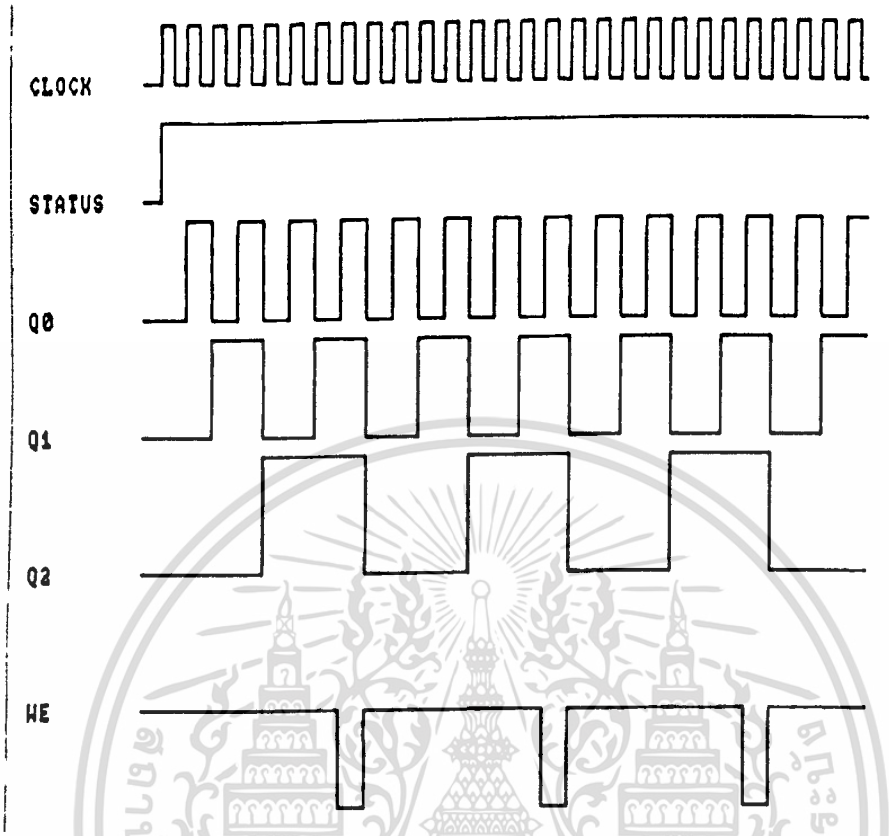
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรกำเนิดสัญญาณอ่าน/เขียนแรม และ ตัวนับแอดเดรส

(Read/Write Clock Generator and Address Counter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Timming Diagram ของสัญญาณอ่านและเขียนแรม

7.แรมอีมูเลเตอร์ (RAM Emulator)

แรมในส่วนนี้ทำหน้าที่เป็นที่พักข้อมูลเพจเจอร์ เพื่อให้คอมพิวเตอร์สะดวกในการอ่านข้อมูลโดย คอมพิวเตอร์จะมองว่า แรมอีมูเลเตอร์เป็นหน่วยความจำส่วนหนึ่งของเครื่อง และจะอ่านข้อมูล ผ่านทางบัสข้อมูลของสล็อทบนเมนบอร์ด

-สเตติกแรม (Static RAM)

ใช้แรม 6264 ที่มีความจุ 8 กิโลไบท์แต่คอมพิวเตอร์จะอ่านเฉพาะข้อมูลในแอดเดรสคู่เท่านั้นจึงต้องต่อ A_0 (ขา 10) ลงกราวด์ตลอด ทำให้เหลือความจุเพียงครึ่งหนึ่ง (4 กิโลไบท์) เท่านั้น แต่ก็เพียงพอสำหรับการบันทึกข้อมูลเพจเจอร์ในการส่งแต่ละครั้ง ขา WE (ขา 27) ได้รับสัญญาณการเขียนทุกๆ 8 บิท ของข้อมูลโดยใช้พัลส์ "0" สั่งให้แรมบันทึกข้อมูลส่วนช่วงเวลาอื่นสัญญาณที่ขา WE นี้เป็น "1" ตลอด

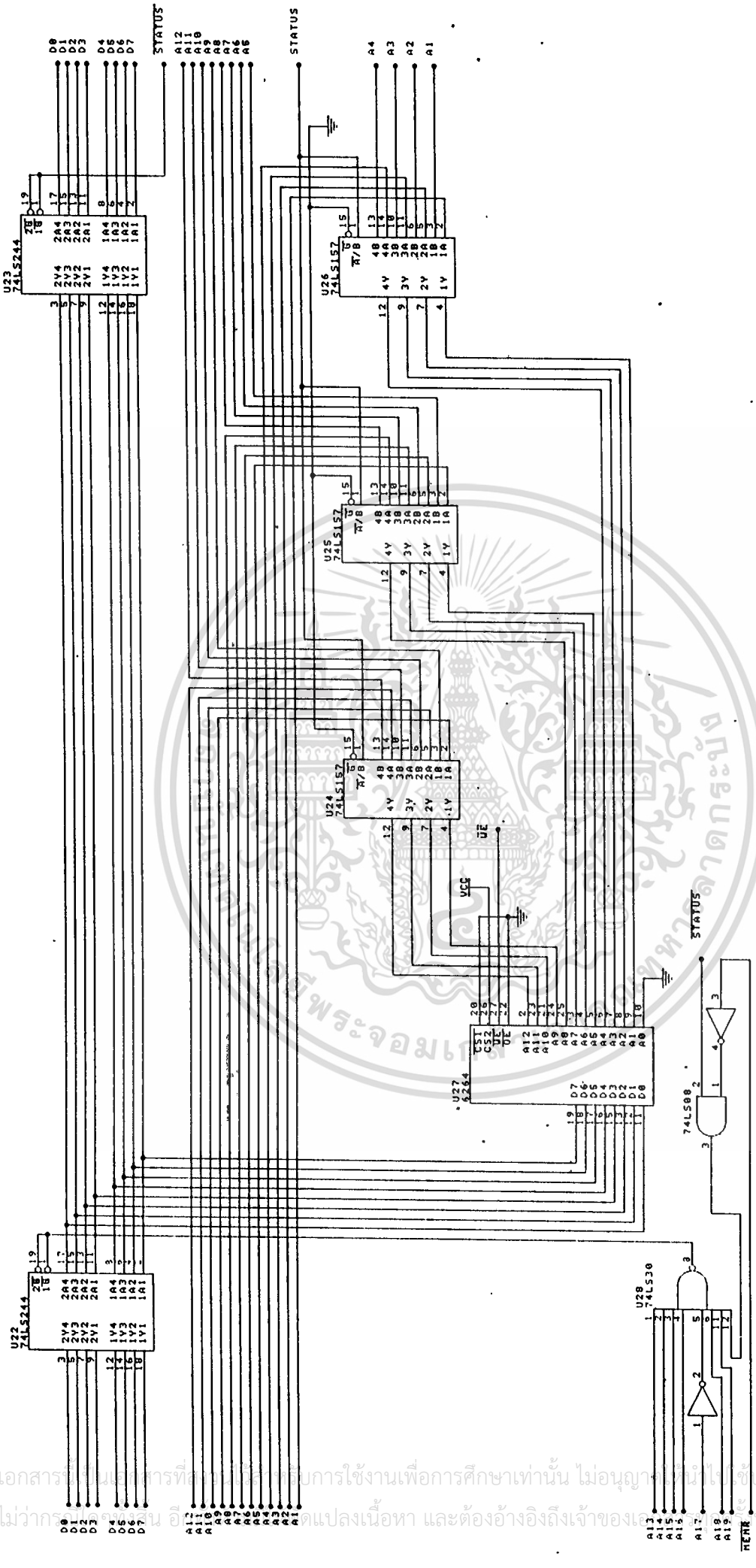
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ขา A_1-A_{12} ของแรมถูกตัดต่อ (Switch) กับแอดเดรสจากคอมพิวเตอร์ และจากวงจรนับแอดเดรสที่ขึ้นอยู่กับสัญญาณ STATUS ในช่วงที่เกิดการเขียนข้อมูลของเพจเจอร์ลงแรม A_1-A_{12} จะถูกต่อกับแอดเดรสที่สร้างจากสัญญาณนาฬิกาในภาคตัวนับแอดเดรส แต่ในช่วงจังหวะที่คอมพิวเตอร์ทำการอ่านข้อมูลจากแรม 6240 ขาแอดเดรส A_1-A_{12} จะถูกต่อกับ A_1-A_{12} ของบัสแอดเดรสบนเมนบอร์ดตัวเลือกแอดเดรส (Address Selector) จะใช้ 74LS157 เป็นตัวเลือกที่ถูกควบคุมด้วยสัญญาณ STATUS ที่ขา A/B (ขา 1) เมื่อมีสัญญาณ เพจเจอร์ส่งเข้ามา STATUS จะเป็น "1" ทำให้ 74LS157 ต่อ A_1-A_{12} ของ แรมเข้ากับ A_1-A_{12} ของตัวนับแอดเดรส แต่ในช่วงที่ไม่มีสัญญาณเพจเจอร์ STATUS จะเป็น "0" ทำให้ 74LS157 ต่อขาแอดเดรส A_1-A_{12}ของแรมกับขาแอดเดรส A_1-A_{12}ของคอมพิวเตอร์แทน และจึงทำให้แรม พร้อมที่จะถูกอ่านข้อมูลโดยคอมพิวเตอร์ ตัวตัดต่อบัสข้อมูลใช้ 74LS244 จำนวน 2 ตัว โดยตัวแรก (U_{23}) ทำหน้าที่ต่อบัสข้อมูลจากภาค SIPO เข้ากับแรมซึ่งจะต่อบัสเฉพาะตอนที่มีการส่งข้อมูลเพจเจอร์เท่านั้น สัญญาณที่ใช้ควบคุมคือ STATUS ซึ่งต่อกับ 1G และ 2G (ขา 1 และขา 19) ของ U_{23} ถ้าขา G1 และ G2 เป็น "0" ข้อมูลจะผ่านไปได้ แต่ถ้าเป็น "1" เอาท์พุทของ U_{23} จะเสมือนเปิดวงจร (high impedance) ส่วน 244 อีกตัวหนึ่งทำหน้าที่ตัดต่อบัสข้อมูลของแรมกับคอมพิวเตอร์ โดยอาศัย สัญญาณควบคุมจากการถอดรหัส A_{13}-A_{19}, MEMR และ STATUS มาแนนด์กันสาเหตุที่นำ A_{13}-A_{19} มาต่อไว้ก็เพื่อเป็นการถอดรหัสให้ 74LS244 ต่อบัสข้อมูลของแรม เข้ากับของคอมพิวเตอร์เฉพาะตอนที่คอมพิวเตอร์ทำการอ่านหน่วยความจำส่วนนี้เท่านั้น โดยที่แรมอีมูเลเตอร์ถูกตั้งให้มีแอดเดรสดังตารางที่ 3.1

Address	A_{19} A_{18} A_{17} A_{16} A_{15} A_{14} A_{13} A_{12} A_{11} .....A_0
Logic	1 1 0 1 1 1 1 X X X

ตารางที่ 3.1 บัสแอดเดรสของแรมอีมูเลเตอร์ (DE00:0000)

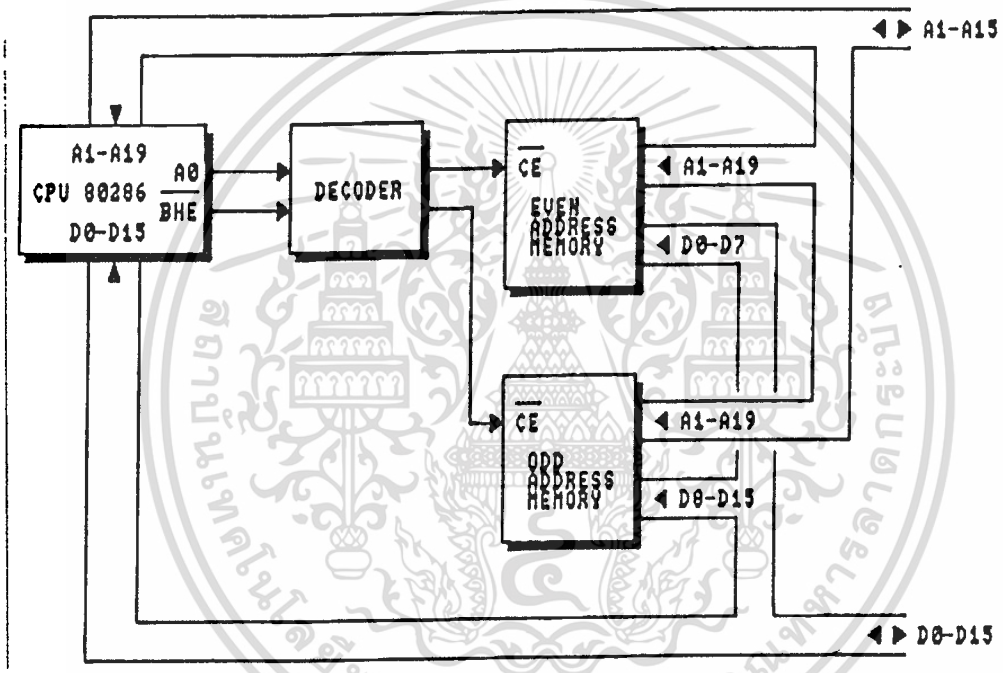
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



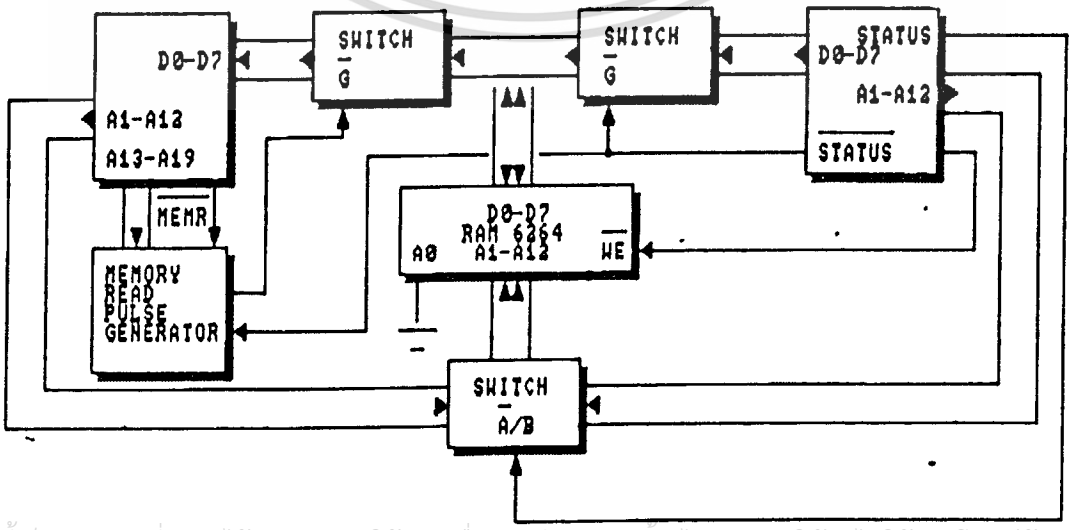
รูปที่ 3.11 การต่อแอดเดรสบัส(RAM Emulator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่บัสแอดเดรสเส้นใดเป็น "0" ต้องผ่าน not gate ก่อน จึงจะเข้าวงจรถอดรหัสได้ ส่วนสัญญาณ MEMR จะต้องมีไว้เพื่อให้ U_{ee} ต่อบัสข้อมูลเฉพาะตอนที่คอมพิวเตอร์ต้องการอ่านมิฉะนั้นข้อมูลจะไหลเข้าบัสข้อมูลของเมนบอร์ดคอมพิวเตอร์ในช่วงที่ไม่ต้องการ และอาจจะทำให้คอมพิวเตอร์หยุดทำงาน (hank) ได้ เนื่องจากบัสข้อมูลถูกใช้ร่วมกันทั้งจากแรมอีมูเลเตอร์และจากส่วนอื่นๆ ของคอมพิวเตอร์ด้วย สัญญาณอีกเส้นที่จำเป็นด้วยคือ STATUS เพื่อที่จะทำให้คอมพิวเตอร์อ่านข้อมูลเฉพาะตอนที่ไม่มีการส่งข้อมูลเพจเจอร์มา ดังนั้นข้อมูลในบัสข้อมูลของแรมจะมีเพียงอย่างใดอย่างหนึ่งในเวลาเดียวกันเท่านั้น



รูปที่ 3.12 การแบ่งหน่วยความจำของ CPU 80286 (16 Bits)



รูปที่ 3.13 บล็อกไดอะแกรมของแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.ภาคตรวจจับหมายเลขเรียกขาน (Address Codeword Detector)

สัญญาณหมายเลขเรียกขานเกิดจากรหัสหมายเลขเฉพาะที่กำหนดให้เพจเจอร์แต่ละเครื่อง ซึ่งมีความสำคัญอย่างมาก เพราะเพจเจอร์แต่ละเครื่องสามารถที่จะรับข่าวสารจากผู้ส่งได้อย่างถูกต้องและตรงได้จากการตรวจสอบสัญญาณหมายเลขเรียกขาน และยังเป็นสัญญาณที่บ่งบอกถึงจุดเริ่มต้นของข่าวสาร(MASSAGE)ที่ส่งมาให้เพจเจอร์แต่ละเครื่องด้วย

การตรวจจับสัญญาณหมายเลขเรียกขาน จากวงจรในรูปที่ 3.14 จากสัญญาณ NRZ ที่ถูก DELAY ไป 1 บิท ถูกป้อนให้กับ 74LS164 U_7 โดยที่กำหนดให้สัญญาณหมายเลขเรียกขานเกิด OUTPUT ที่บิทสุดท้ายของรหัสคำหมายเลขเรียกขาน (ซึ่งหมายถึงจะเกิด OUTPUT หลังจากบิทที่ 19 ของรหัสคำหมายเลขเรียกขาน) ดังนั้นจึงต้องต่อขา OUTPUT ของ 74LS164 ที่คาดว่าจะเป็น "0" ในช่วงเวลาที่เกิดสัญญาณหมายเลขเรียกขานพอดีผ่าน NOT GATE ซึ่งจากวงจรสามารถที่จะเลือกตั้งค่าได้โดยใช้ DIP SW. จากนั้นสัญญาณ OUTPUT ที่ได้จาก NOT GATE แต่ละตัว รวมทั้งสัญญาณ OUTPUT ของ 74LS164 ที่เหลือด้วย จะถูกต่อเข้ากับ NAND GATE ที่มี INPUT 8 ขา สัญญาณ OUTPUT ที่ได้จาก NAND GATE จะถูกต่อเข้ากับ OR GATE ซึ่งจะได้ OUTPUT เป็นสัญญาณหมายเลขเรียกขานออกมาทุกครั้งที่มีการรับข่าวสารต่อการส่งสัญญาณ 1 ครั้ง

9.ภาคสร้างสัญญาณอินเตอร์รัพแอดเดรสปลักโคด (Interrupt Addrees Plugcode)

จากการที่มีการกำหนดให้หมายเลขเรียกขานของเครื่องรับเพจเจอร์แต่ละเครื่องนั้นแตกต่างกันมีผลทำให้สัญญาณหมายเลขเรียกขานของแต่ละเครื่องถูกส่งมาอยู่ใน เฟรม (FRAME) ที่แตกต่างกันด้วย ซึ่งจะสามารถทราบตำแหน่งของ เฟรม (FRAME) ที่ส่งหมายเลขเรียกขานได้โดยการนำเอาหมายเลข PLUGCODE ที่เป็นหมายเลขประจำตัวเครื่องรับเพจเจอร์ ซึ่งเป็นเลขฐาน 10 แปลงให้อยู่ในรูปเลขฐาน 2 จะได้ทั้งหมด 21 บิท แต่ 3 บิทหลังที่มีความสำคัญน้อย (LSB) คือ บิทที่ 19-21 ที่ไม่ได้ทำการส่ง จะเป็นตัวกำหนด เฟรม (FRAME) ที่ใช้ในการส่งหมายเลขเรียกขานของแต่ละเครื่องรับเพจเจอร์

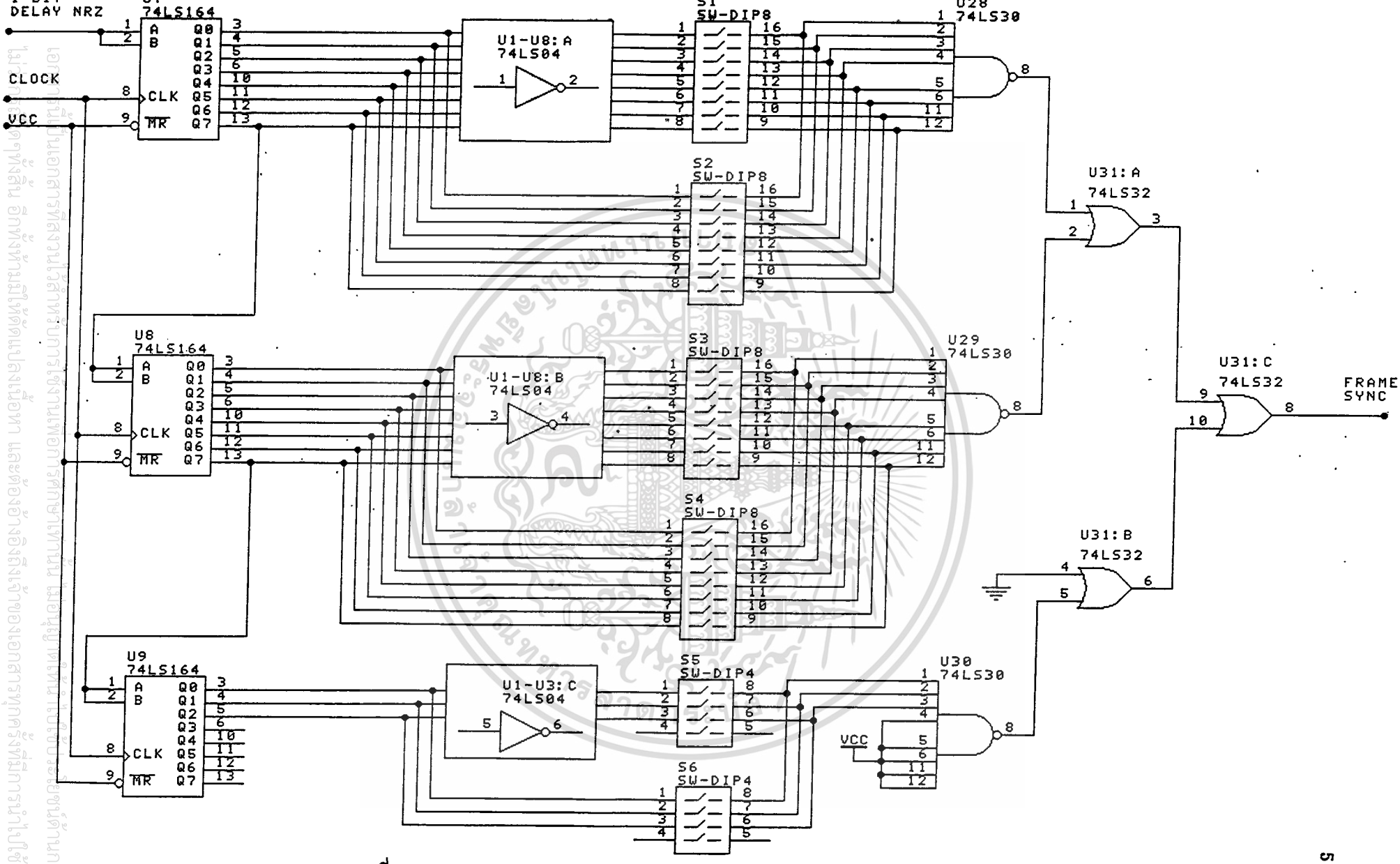
วงจรนับเฟรมของหมายเลขเรียกขาน จากวงจรในรูปที่ 3.15 สัญญาณนาฬิกาที่มีความถี่ 1200 bps และสัญญาณ FRAME SYNC จะถูกป้อนเข้าที่ 74LS161(U_{12}) ซึ่งสัญญาณนาฬิกาจะถูกนับ 16 ได้ OUTPUT ออกมา 1 ครั้ง จากนั้นสัญญาณ OUTPUT ที่ได้จะถูกต่อผ่าน NOT GATE และป้อนเข้าที่ 74LS161 (U_{13}) เพื่อทำการนับ 2 ได้ OUTPUT ออกมา 1 ครั้ง จากนั้นนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

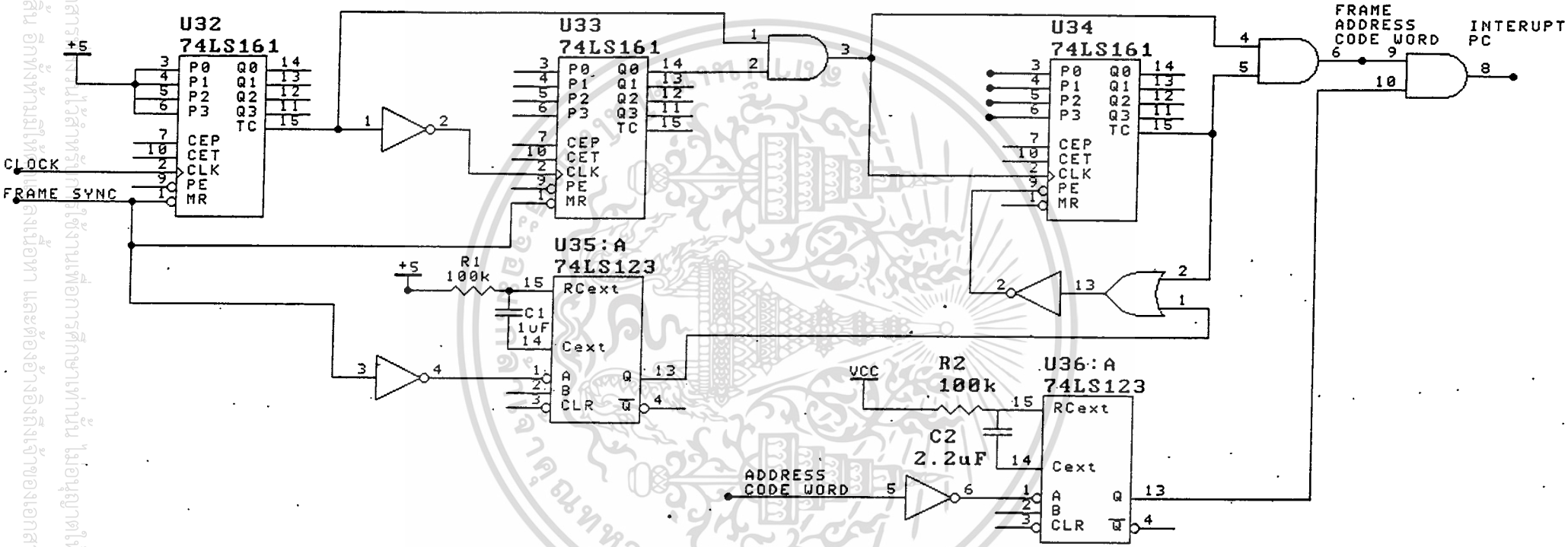
เอา OUTPUT ที่ได้ของ 74LS164 (U_{32}) และ (U_{33}) มาผ่าน AND GATE เพื่อให้ได้สัญญาณ OUTPUT เป็น "1" ทุก ๆ 32 บิท หรือ เท่ากับ 1 CODEWORD ส่วน 74LS161 (U_{34}) จะนำเอา OUTPUT ที่ได้จาก AND GATE มาทำการนับอีกครั้ง เพื่อให้ได้สัญญาณ OUTPUT ออกมาในช่วงจังหวะที่มีการส่งสัญญาณหมายเลขเรียกขาน ซึ่งสามารถเลือกตั้งค่านับ ได้โดยใช้ DIP SW และค่าที่เลือกใช้ในการนับ เพื่อให้เกิด OUTPUT ในช่วงที่มีการส่งสัญญาณหมายเลขเรียกขานได้ตรงกับเครื่องรับเพจเจอร์ของแต่ละเครื่อง มีค่าเท่ากับค่าของ บิทที่ 19-21 ของหมายเลขเรียกขาน และจะถูกทำการ RESET ให้ทำการนับใหม่ที่ขา 9 ด้วยสัญญาณที่เกิดจากขาที่ 15 ของ 74LS161 (U_{34}) และสัญญาณ FRAME SYNC จะทำการ RESET ค่าการนับของ 74LS161 ทุก ๆ ตัว เพื่อทำให้เกิดการนับที่เป็นไปอย่างถูกต้อง

จากนั้นนำเอาสัญญาณ OUTPUT ที่ได้จากการตั้งค่านับ และสัญญาณ OUTPUT ที่ได้จากภาคตรวจจับสัญญาณหมายเลขเรียกขานมาผ่าน AND GATE จะได้ OUTPUT เป็นสัญญาณที่ใช้ในการ INTERRUPT PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 วงจรตรวจจับแอดเดรสปลักโคด(Address Plugcode Detector)



รูปที่ 3.15 วงจรกำเนิดสัญญาณอินเตอร์รัพท์แอดเดรสปลักโคด

(Interrupt Address Plugcode)

10.คอมพิวเตอร์ (PC)

ถือเป็นส่วนสำคัญซึ่งถูกใช้สำหรับการอ่านและแสดงข่าวสารของสัญญาณเพจเจอร์ การทดลองโครงงานนี้ใช้คอมพิวเตอร์แบบ 16 บิท CPU 80286 ดังนั้นบัสข้อมูลของระบบจึงมี 16 เส้น แบ่ง เป็นไบท์บน 8 เส้น (D_0-D_{15}) และไบท์ล่าง 8 เส้น (D_0-D_7) ซึ่งแต่ละกลุ่มของบัสข้อมูลจะต้อเข้ากับกลุ่มของหน่วยความจำ (memory bank) ที่แตกต่างกัน ในเครื่อง AT แบบ 16 บิท จะแบ่งหน่วยความจำเป็น 2 กลุ่ม ซึ่งเรียกว่าแบงค์ โดยมีขนาดเท่ากันทั้ง 2 แบงค์ หน่วยความจำแบงค์แรกจะต่อกับไบท์ล่างของบัสข้อมูล (D_0-D_7) โดยมีแอดเดรสเป็นคู่ทั้งหมด (0, 2, 4, 6, ...) ส่วนอีกแบงค์หนึ่งจะต่อกับไบท์บนของบัสข้อมูล (D_0-D_{15}) ซึ่งมีแอดเดรสเป็นคี่ทั้งหมด (1, 3, 5, 7, ...) เช่นกัน ดังรูปที่ 3.12 สัญญาณที่ถูกใช้สำหรับการถอดรหัสเพื่อเลือกแบงค์หน่วยความจำก็คือสัญญาณ A_0 และBHE ของ CPU 80286 ในกรณีที่ A_0 ="0" แสดงว่ามีการเลือกแบงค์คู่ แต่ถ้า A_0="1" จะเป็นการเลือกแบงค์คี่ ส่วนสัญญาณ BHE จะเป็นตัวบอกว่า CPU ยอมให้มีการใช้ไบท์บนของบัสข้อมูลหรือไม่ในกรณีที่ BHE ="0" แสดงว่ายอมให้ใช้ D_0-D_{15} แต่ถ้า BHE ="1" แสดงว่าไม่ยอมให้ใช้ D_0-D_{15} เมื่อเรานำสัญญาณ A_0 กับ BHE มาใช้ร่วมกันก็จะก่อให้เกิด 4 สถานะขึ้นตามตารางที่ 3.2

คำสั่งการอ่านหน่วยความจำในภาษา C	BHE	A_0	หน้าที่การทำงาน
Peek (แอดเดรสคู่)	0	0	รับ-ส่งข้อมูลเป็นเวิร์ด (D_0-D_{15})
Peekb (แอดเดรสคี่)	0	1	รับ-ส่งข้อมูลทางไบท์บน (D_0-D_{15})
Peekb (แอดเดรสคู่)	1	0	รับ-ส่งข้อมูลทางไบท์บน (D_0-D_7)
	1	1	

ตารางที่ 3.2 การอ่านข้อมูลจากหน่วยความจำด้วยวิธีต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาษา C มีฟังก์ชั่นสำหรับการอ่านข้อมูลจากหน่วยความจำอยู่ 2 ชนิด คือ peek สำหรับอ่านหน่วยความจำทีละเวิร์ด (16 บิท) และ peekb สำหรับอ่านหน่วยความจำทีละไบท์ (8 บิท) ในกรณีที่ใช้ฟังก์ชั่น peek CPU จะอ่านข้อมูลจากหน่วยความจำทั้ง 16 เส้น เช่นถ้าสั่ง peek(0XDE 00,0)CPU ก็จะอ่านข้อมูลจาก DE00:0 ผ่านทาง D_0-D_7 และอ่านข้อมูลจาก DE00:1 ผ่านทาง D_8-D_{15} รวมเป็น 1 เวิร์ด แต่ถ้าใช้ peek(แอดเดรสคี่) เช่น peek(0XDE00,1) CPU จะทำการนำข้อมูลจาก DE00:1 ผ่านทาง D_0-D_{15} และอ่านข้อมูลของ DE00:2 ผ่านทาง D_0-D_7 รวมเป็น 1 เวิร์ดเช่นกัน ส่วน peekb เป็นการอ่านหน่วยความจำทีละไบท์ตามแอดเดรสที่กำหนดให้โดยตรงเช่น peekb(0XDE00,0) CPU ก็จะอ่านข้อมูลจาก DE00:0 มาหนึ่งไบท์ผ่านทาง D_0 -D_7 แต่ถ้าสั่ง Peekb(0XDE00.1) ข้อมูลจะถูกอ่านจาก DE00:1 ผ่านทาง D_0-D_{15} ข้อจำกัดนี้จะถูกกำหนดไว้ด้วยคุณสมบัติของเครื่อง (hard waare) ดังนั้นไม่ว่าเราจะใช้ภาษาใด ก็ตามข้อจำกัดนี้ก็ยังคงมีอยู่ แม้แต่ภาษา (assembly) ก็ไม่มีข้อยกเว้น เช่นถ้าเราสั่ง MOV AX,WORD [DE00H:1] ก็จะมีผลเช่นเดียวกับ peek(0XDE00, 1) นั่นคือ การอ่านข้อมูลจาก DE0 0:1 ผ่านทาง D_0-D_{15} และอ่านข้อมูลจาก DE00:2 ผ่านทาง D_0-D_7 เช่นกัน ด้วยข้อจำกัดนี้เองเราจึงไม่สามารถอ่านข้อมูลจากหน่วยความจำที่มีแอดเดรสคี่ผ่านทาง D_0-D_7 ซึ่ง ต่ออยู่กับแรมอีมูเลเตอร์ได้โดยตรง ทางออกสำหรับปัญหานี้มี 2 วิธีคือ

1.บันทึกข้อมูลลงทุกแอดเดรสของแรมอีมูเลเตอร์ และดัดแปลงวงจรเพื่อถอดรหัสบัสแอดเดรสของคอมพิวเตอร์ โดยทำการหารแอดเดรสคอมพิวเตอร์ด้วย 2 ก่อนที่จะต่อกับแอดเดรสของแรมอีมูเลเตอร์ดังตารางที่ 3.3

แอดเดรสจากคอมฯ	แอดเดรสของแรมอิมูเลเตอร์
DE00:0	DE00:0
DE00:2	DE00:1
DE00:4	DE00:2
.....	.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้องสมุดขอสงวนสิทธิ์ในการไม่รับผิดชอบต่อการละเมิดลิขสิทธิ์ที่เกิดจากการนำเอกสารนี้ไปใช้

ตารางที่ 3.3 การเปลี่ยนรหัสแอดเดรสของคอมพิวเตอร์มาเป็นแอดเดรสของแรมอิมูเลเตอร์

แม้วิธีการนี้จะใช้แรมได้อย่างมีประสิทธิภาพ แต่ก็เพิ่มความยุ่งยากในการออกแบบ และการสร้างวงจรมากขึ้น อีกทั้งข้อมูลเพจเจอร์ที่รับเข้ามาจากการส่งแต่ละครั้งก็ไม่มากจนถึงกับต้องใช้แรมให้ครบ 8 กิโลไบท์ ดังนั้นจึงนำมาสู่วิธีแก้ถัดมาคือ

2.บันทึกข้อมูลลงเฉพาะแอดเดรสคู่ของแรมอีมูเลเตอร์เท่านั้น และให้คอมพิวเตอร์อ่านเฉพาะข้อมูลแอดเดรสคู่เหล่านั้นผ่าน D_0-D_7 ได้ ซึ่งวิธีการนี้ถูกใช้อยู่ในวงจรการทดลองจริง ดังจะเห็นได้จากรูปที่ 3.11 ว่าขา A_0 ของ 6264 ถูกต่อลงกราวด์หรือเป็น "0" ตลอด และขา A_0 จากคอมพิวเตอร์จึงไม่จำเป็นต้องใช้ สำหรับโปรแกรมที่ใช้ถูกเขียนและคอมไพล์ (compile) บนเทอร์โบซี (Turbo C Version 2.0) สาเหตุที่เลือกใช้ภาษาซีในการเขียนโปรแกรม ก็เนื่องจากเป็นภาษาที่มีโครงสร้างกระชับ และมีฟังก์ชั่นต่างๆ ให้ใช้งานมาก อีกทั้งมีคุณสมบัติพิเศษในการทำงานกับข้อมูลที่เป็นบิท อาศัยหลักการของ บิทฟิลด์ (bit fild) ได้ ซึ่งเหมาะสมกับโครงงานนี้ เนื่องจากข้อมูลเพจเจอร์ที่ถูกอ่านเข้ามานั้นตัวโปรแกรมจะต้องถอดรหัสจาก 1 บิทของข้อมูลบ้าง, 4 บิทบ้างหรือ 18 บิทบ้าง ซึ่งภาษาคอมพิวเตอร์บางตัวไม่สามารถทำได้ หรือถ้าได้ก็อาจจะยุ่งยากกว่า ลักษณะเด่นอีกประการหนึ่งของภาษาซีก็คือ มีการทำงานที่รวดเร็วใกล้เคียงกับภาษาแอสเซมบลี้มากกว่าภาษาอื่นๆ ซึ่งนับว่าจำเป็นมากสำหรับการทำงานในลักษณะ "ตามเวลาจริง" (Real time) นั่นคือโปรแกรมจะต้องอ่านข้อมูลจากแรมอีมูเลเตอร์ให้เสร็จทัน ก่อนที่การส่งสัญญาณเพจเจอร์ครั้งต่อไปจะมาถึง และการแสดงข่าวสารจะต้องเกิดขึ้นเกือบจะทันทีที่สัญญาณส่วนที่เป็นข่าวสารนั้นๆ มาถึง และเพื่อความสะดวกในการเขียนโปรแกรม ได้มีการสร้างเฮดเดอร์ไฟล์ (header file) ขึ้นมาต่างหากสำหรับการกำหนดตัวแปรชนิดพิเศษ อาทิเช่น บิทฟิลด์ รวมทั้งค่าคงที่ต่างๆ ซึ่งถูกเรียกจากโปรแกรมหลักเป็นประจำ

โปรแกรมหลักที่ใช้สำหรับการแสดงผลนี้มี 4 โปรแกรมโดยโปรแกรมที่ 1 และ 2 นั้นจะใช้สำหรับเพจเจอร์รุ่นตัวเลข (Numeric) และ โปรแกรมที่ 3 และ 4 สำหรับเพจเจอร์รุ่นตัวอักษร (Alpha-Numeric) ซึ่งในแต่ละโปรแกรมจะมีรายละเอียดดังนี้คือ

1.Demons.c เป็นโปรแกรมที่ใช้สำหรับการแสดงข้อมูลทั้งหมดของเพจเจอร์รุ่น ตัวเลข ที่ได้รับทั้งข้อมูลที่เป็นรหัสคำข่าวสาร, รหัสหมายเลขเรียกขาน, รหัสคำการซิงค์ และรหัสคำเทียม การแสดงข้อมูลจะแบ่งเป็นรหัสคำละ 5 อักขระ แล้วรวม 17 รหัสคำเป็น 1 แบทช์ การแสดงข้อมูลจะเริ่มจากแอดเดรสแรกของแรมอีมูเลเตอร์ (DE00:0000) ไปจนจบหนึ่งหน้าจอ (DE00:00DC) ซึ่งก็คือหนึ่งหน้าจอแสดงได้ 476 ไบท์หรือ 7 แบทช์ ในกรณีที่การรับข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจเจอร์ถูกต้องรหัสคำต่างๆ จะตรงกับตารางที่ 3.4

ชนิดของรหัสคำ	ตัวอักษรที่ปรากฏ
Frame sync	F9524,
Idle	FA841,
EOT	55555
Address	xxxx,
Message	xxxxx

ตารางที่ 3.4 รูปแบบของรหัสคำชนิดต่างๆ

เนื่องจากโปรแกรมนี้แสดงข้อมูลเฉพาะบิท 1 ถึง 20 ของรหัสคำ เราจึงไม่ทราบว่ารหัสคำที่แสดงเป็นรหัสคำข่าวสารหรือรหัสของหมายเลข (ขึ้นต้นด้วยบิท "0" หรือ "1") จึงเขียนโปรแกรมให้มีเครื่องหมาย "," อยู่ท้ายรหัสคำที่มีบิทแรกเป็น "0"

2.Pager.C เป็นโปรแกรมที่ใช้งานจริงเพื่อแสดงข่าวสารเฉพาะของหมายเลขที่ต้องการเท่านั้นของเพจเจอร์รุ่นตัวเลข โดยในโปรแกรม pager.c เริ่มจากกำหนดการรับหมายเลขเครื่องเข้าทางแป้นพิมพ์ก่อนแล้วสลับบิทของหมายเลข ให้สามารถเปรียบเทียบกับข้อมูลที่รับเข้ามาได้ให้ง่ายขึ้น โดยการใช้ฟังก์ชั่น convert_bits หลังจากนั้นโปรแกรมจะดึงข้อมูลจากแรมอีมูเลเตอร์เข้ามาทีละ 4 ไบท์ แล้วตรวจดูว่าตรงกับหมายเลขเครื่องที่กำหนดไว้หรือไม่ ถ้าไม่โปรแกรมก็จะอ่านข้อมูลจาก 4 ไบท์ถัดไปเข้ามาอีกแล้วทำซ้ำลักษณะเดิมไปเรื่อยๆ แต่เมื่อโปรแกรมตรวจพบข้อมูลที่เป็นหมายเลขเครื่องที่ต้องการ ข้อมูลจากแรมอีก 4 ไบท์ถัดไปจะถูกอ่าน และเปลี่ยนรหัสจาก POCSAG เป็น ASCII ด้วยฟังก์ชั่น changecode แล้วจึงพิมพ์ออกหน้าจอด้วยฟังก์ชั่น POCSAG ข่าวสารจะถูกอ่าน และพิมพ์ออกมาเรื่อยๆ จนกว่าจะพบกับรหัสคำหมายเลขเรียกขานอื่น หรือรหัสคำแสดงการจบของการส่ง (EOT) แต่โปรแกรมจะเว้นการแสดงรหัสคำการชิงค์ออกหน้าจอโดยการข้ามรหัสคำส่วนนั้นไป

3.Binary.c เป็นโปรแกรมสำหรับแสดงข้อมูลทั้งหมดของเพจเจอร์รุ่นตัวอักษร-ตัวเลข ซึ่งจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงข้อ มูลออกทางจอภาพโดยจะแสดงข้อมูลทั้งหมดเป็นเลขฐานสอง และเลขฐานสิบ โดยจะแสดงทั้งหมด 17 รหัสคำ หรือ 1 แบทต์ โดยเริ่มอ่านข้อมูลจากแรมที่แอดเดรส DE00:0000 ก่อนซึ่งข้อมูลที่แสดงจะเป็นทั้งรหัสคำซิงโครนัส ,รหัสคำหมายเลขเรียกขาน ,รหัสคำข่าวสาร และรหัสคำเทียม ทั้งหมดเป็นเลขฐานสองเพื่อง่ายต่อการวิเคราห์จึงแสดงเป็นค่าประจำรหัสคำด้วยเลขฐานสิบ และยังสามารถแสดงเฉพาะข่าวสาร (Message) ที่ส่งมาโดยจะแสดงที่ละ 20 ตัวจนกระทั่งถึงตัวสุดท้ายที่ส่งมา โดยมีลักษณะการทำงานก็คือ โดยเริ่มจากรับหมายเลขเครื่อง (Address Plugcode) และทำการเปลี่ยนเป็นเลขฐานสองจากนั้นทำการสลับบิทเฉพาะ 21 บิทแรกที่อ่านได้เพื่อเก็บไว้ในตัวแปรสตรัจเจอร์สำหรับรอการเปรียบเทียบ จากนั้นก็ทำการรับจำนวนแบทต์ที่ต้องการ เพื่อกำหนดช่วงการแสดงผลที่หน้าจอ เมื่อเสร็จขั้นตอนนี้แล้วโปรแกรมจะทำการอ่านแรมที่แอดเดรส แรกคือ DE00:0000 แล้วนำมาแสดงที่หน้าจอเมื่อตรวจสอบพบหมายเลขที่ต้องการก็จะมีเสียงเตือน จากนั้นก็ทำการอ่านแอดเดรสต่อไปเพื่อนำมาแสดงผลข้อมูลที่อ่านมาได้ถ้าเป็นข่าวสารก็จะ แสดงผลที่เป็นเฉพาะข่าวสารแยกต่างหาก โดยจะแสดงในเพจเจอร์รุ่นตัวอักษร-ตัวเลข โดยจะแสดงผลจนครบจำนวนแบทต์ที่ได้กำหนดเอาไว้

4.Pager1.c เป็นโปรแกรมที่ใช้งานจริงสำหรับแสดงข่าวสารเฉพาะหมายเลขเครื่องที่ผู้ใช้ต้องการเท่านั้นโดยจะมีลักษณะการทำงานดังนี้คือ โดยเริ่มจากรับหมายเลขเครื่อง (Address Plug Code) และทำการเปลี่ยนเป็นเลขฐานสองจากนั้นทำการสลับบิทเฉพาะ 21 บิทแรกที่อ่านได้เพื่อเก็บไว้ในตัวแปรสตรัจเจอร์สำหรับรอการเปรียบเทียบ เมื่อเสร็จขั้นตอนนี้แล้วโปรแกรมจะทำการอ่านแรมที่แอดเดรสแรกคือ DE00:0000 โดยจะอ่านเข้ามาที่ละ 4 ไบท์ (32 บิท) หรือ 1 รหัสคำ และนำมาทำการตรวจตรวจสอบว่าเป็นรหัสคำหมายเลขเรียกขานหรือไม่ถ้าไม่ใช่ก็จะอ่านข้อมูลที่แอดเดรสต่อไปที่ละ 4 ไบท์เช่นกันจนกว่าจะพบรหัสคำหมายเลขเรียกขานที่ต้องการ หรือมีการกดคีย์บอร์ดเพื่อหยุดการทำงาน ถ้าตรวจสอบพบรหัสคำหมายเลขเรียกขานที่ต้องการก็จะมีเสียงเตือนเพื่อบอกให้ผู้ใช้ทราบว่าพบหมายเลขที่ต้องการแล้ว หลังจากนั้นก็จะอ่านแอดเดรสต่อไปรหัสคำที่ต่อจากรหัสคำหมายเลขเรียกขานนั้นจะเป็นรหัสคำข่าวสารโดยตัวโปรแกรมจะทำการอ่านและตรวจสอบก่อนว่ารหัสคำที่อ่านได้เป็นรหัสคำข่าวสาร ถ้าใช่ก็จะทำการโอนถ่ายค่าลงสู่ตัวแปรสตรัจเจอร์ (Structure Variable) จากนั้นก็จะทำการอ่านแรมที่แอดเดรสต่อไปและกระทำเช่นเดียวกันกับที่กล่าวมาแล้ว เมื่อรหัสข่าวสารที่อ่านมาได้ครบ 20 ตัวอักษรก็จะทำการดีความและแสดงผลมาทางหน้าจอ หรือจะแสดงผลในกรณีที่สิ้นสุดการส่งรหัสคำข่าวสารเปรียบเทียบจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้นำเอกสารนี้ไปเผยแพร่ต่อ

รหัสคำต่อไปเมื่อเป็นรหัสคำเทียม นั้นหมายความว่าสิ้นสุดการส่งข้อมูลแล้ว ดังนั้นจึงทำการตีความรหัสข่าวสารที่เหลือไม่ถึง 20 ตัวอักษร และทำการแสดงผลทางหน้าจอ เหตุผลที่ทำไมต้องแสดงผลที่ละ 20 ตัวอักษรเนื่องมาจากความไม่ลงตัวของจำนวนบิทข่าวสาร หรือกล่าวอีกในหนึ่งว่าบิทข่าวสารมีการเลื่อนไปในทุกๆ รหัสคำข่าวสารอันเนื่องมาจากการจำนวนบิทของ 1 อักษรมีจำนวน 7 บิท แต่จำนวนของบิทข่าวสาร (Message Bits) มีเพียง 20 บิท จึงสามารถบรรจุตัวอักษร หรือ อักขระได้ 2 ตัว กับอีก 6 บิทของตัวที่ 3 จะสังเกตุได้ว่าเมื่ออักษร 3 ตัวต่อมาก็จะมีการเลื่อนไปอีกดังนั้น ลักษณะการเลื่อนดังกล่าวจากจำนวนตัวอักษร 20 ตัวจะมีลักษณะการเลื่อนวนกลับมาที่เดิมดังนั้นในตัวโปรแกรมเองจึงได้สร้างตัวแปรสรัคเจอร์ที่มีขนาด 20 ตัวอักษรมารองรับเพื่อง่ายต่อการนำรหัสที่ได้ไปตีความสำหรับการแสดงผลต่อไปนั้นเอง และสำหรับโปรแกรมอื่นๆ ที่ไม่ใช้ใช้สำหรับแสดงผลที่หน้าจอก็คือ

5.Plugcode.c เป็นโปรแกรมบริการอินเตอร์รัพท์สำหรับกำเนิดเสียงในกรณีที่ฮาร์ดแวร์สามารถตรวจสอบจับหมายเลขเรียกขานที่กำหนดโดยดิปสวิทช์ได้ โดยมีลักษณะการทำงานดังนี้คือเมื่อมีสัญญาณอินเตอร์รัพท์มาจากทางสล็อทของ PC CPU จะเรียกโปรแกรมบริการอินเตอร์รัพท์ออกมาทำงานโดยตัวโปรแกมจะรอการโดยทำการฝังตัวเองลงในหน่วยความจำ หรือเป็นโปรแกรมประเภท TSR (Terminate Stay and Resident Program) และจึงกำเนิดเสียงเตือน 3 ครั้งเพื่อให้ผู้ใช้ทราบว่ามีการส่งข้อมูลเข้ามา

เพื่อให้ความสะดวกในการเขียนโปรแกรมจึงได้ทำเฮดเดอร์ไฟล์ (Heaad files) โดยมีเฮดเดอร์ไฟล์ที่ใช้อยู่ 2 ไฟล์ คือ Pager.h สำหรับโปรแกรมแสดงผลเพจเจอร์รุ่นตัวเลข และ N_pager.h สำหรับโปรแกรมแสดงผลเพจเจอร์รุ่นตัวอักษร-ตัวเลข รายละเอียดของโปรแกรมจะแสดงในบทที่ 4 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

โปรแกรมสำหรับแสดงผลข้อมูลเพจเจอร์ (Programs Display Data Pager)

สำหรับในบทนี้จะแสดงรายละเอียดของโปรแกรมที่ใช้ในการแสดงผลข้อมูลเพจเจอร์ทั้ง 2 รุ่น คือ เฮดเดอร์ไฟล์ และ เอกซ์คิวไฟล์ โดยเริ่มจากโปรแกรมแสดงผลของเพจเจอร์รุ่นตัวเลข (Numeric) และโปรแกรมแสดงผลของเพจเจอร์รุ่นตัวอักษร-ตัวเลข (ALpha-Numeric)

ไฟล์ Pager.h และ N_pager.h เป็นเฮดเดอร์ไฟล์โดยจะกำหนดค่าหลักๆ ที่โปรแกรมเรียกใช้บ่อยครั้ง และตัวแปรโครงสร้าง (Structure Variable) รวมทั้งตัวแปรที่เป็นบิทฟิวด์ (Bits Field) สำหรับเพจเจอร์รุ่นตัวเลขจะใช้ไฟล์ Pager.h และเพจเจอร์รุ่นตัวอักษร-ตัวเลขจะใช้ไฟล์ N_pager.h ตามลำดับ

ไฟล์ Demons.EXE และ Pager.EXE เป็นเอกซ์คิวไฟล์สำหรับแสดงข้อมูลของเพจเจอร์รุ่นตัวเลขโดยโฟลว์ชาร์ท การทำงานของทั้ง 2 โปรแกรมจะแสดงไว้ดังรูปที่ 4.1 และ 4.2 ตามลำดับ ส่วนไฟล์ Binary.EXE และ Pager1.EXE เป็นเอกซ์คิวไฟล์สำหรับแสดงข้อมูลของเพจเจอร์รุ่นตัวอักษร-ตัวเลข โดยโฟลว์ชาร์ทการทำงานของทั้ง 2 โปรแกรมจะแสดงไว้ดังรูปที่ 4.3 และ 4.4 ตามลำดับ ไฟล์ Plugcode.EXE เป็นเอกซ์คิวไฟล์ที่รองรับการอินเตอร์รัพท์จากฮาร์ดแวร์โดยจะทำหน้าที่เป็นโปรแกรมบริการอินเตอร์รัพท์ที่ให้กำเนิดเสียงเตือน เมื่อมีการตรวจพบหมายเลขเครื่องที่ต้องการ

สำหรับซอร์ดโปรแกรมของทั้ง 4 โปรแกรมที่ใช้แสดงข้อมูล และ 1 โปรแกรมอินเตอร์รัพท์ รวมทั้ง 2 เฮดเดอร์ไฟล์ จะมีแสดงรายละเอียดตามลำดับดังนี้ดังนี้คือ

1.Pager.h

2.N_pager.h

3.Demons.c

4.Pager.c

5.Binary.c

6.Pager1.c

7.Plugcode.c

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของเฮดเดอร์ไฟล์ Pager.h

```
/* Spectail heaader file for Pager Code Receiver Card (Numeric) */
#include <stdio.h>
#include <dos.h>

#define SEGMENT    0xDE00
#define FRAME_SYN  0x1BA84B3E
#define CLEAR      0xFFFFFFFF
#define IDLE_WORD  0xE98391AE
#define EOT        0x55555555

typedef struct
{
 unsigned b0      :1;
 unsigned b1_4    :4;
 unsigned b5_8    :4;
 unsigned b9_12   :4;
 unsigned b13_16  :4;
 unsigned b17_20  :4;
 unsigned b21_30  :10;
 unsigned b31     :1;
 }BIT_1;

typedef struct
{
 unsigned b0      :1;
 unsigned b1_4    :4;
 unsigned b5_8    :4;
 unsigned b9_12   :4;
 unsigned b13_16  :4;
 unsigned b17_20  :4;
 unsigned b21_30  :10;
 unsigned b31     :1;
}BIT_2;

typedef struct
{
 unsigned b1 :1;
 unsigned b2 :1;
 unsigned b3 :1;
 unsigned b4 :1;
 unsigned b5 :1;
 unsigned b6 :1;
 unsigned b7 :1;
 unsigned b8 :1;
 unsigned b9 :1;
 unsigned b10 :1;
 unsigned b11 :1;
 unsigned b12 :1;
 unsigned b13 :1;
 unsigned b14 :1;
 unsigned b15 :1;
 unsigned b16 :1;
 unsigned b17 :1;
 unsigned b18 :1;
```

ลิขสิทธิ์สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 unsigned b19 :1;
 unsigned b20 :1;
 unsigned b21_31 :11;
}BIT_3;

typedef union
{
 unsigned  long CODEWORD;
 unsigned BIT_1  MESSAGE;
 unsigned BIT_2  ADDRESS;
 unsigned BIT_3  BUFFER;
 char BYTE[4];
}DOUBLE_WORD;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของเฮดเดอร์ไฟล์ N_pager.h

```
#include <stdio.h>
#include <dos.h>
#define SEGMENT    0xDE00
#define FRAME_SYN 464014142
#define IDLE_WORD 3917713758
#define EOT       3540460221
#define CLEAR     0xFFFFFFFF
#define END       0
typedef struct
{
 unsigned b1      :1;
 unsigned b2_5    :4;
 unsigned b6_9    :4;
 unsigned b10_13  :4;
 unsigned b14_17  :4;
 unsigned b18_21  :4;
 unsigned b22_31  :10;
 unsigned b32     :1;
}BIT_1;

typedef struct
{
 unsigned b1      :1;
 unsigned b2_5    :4;
 unsigned b6_9    :4;
 unsigned b10_13  :4;
 unsigned b14_17  :4;
 unsigned b18_21  :4;
 unsigned b22_31  :10;
 unsigned b32     :1;
}BIT_2;

typedef struct
{
 unsigned b1      :1;
 unsigned b2      :1;
 unsigned b3      :1;
 unsigned b4      :1;
 unsigned b5      :1;
 unsigned b6      :1;
 unsigned b7      :1;
 unsigned b8      :1;
 unsigned b9      :1;
 unsigned b10     :1;
 unsigned b11     :1;
 unsigned b12     :1;
 unsigned b13     :1;
 unsigned b14     :1;
 unsigned b15     :1;
 unsigned b16     :1;
 unsigned b17     :1;
 unsigned b18     :1;
 unsigned b19     :1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 unsigned b20     :1;
 unsigned b21     :1;
 unsigned b21_31 :11;
}BIT_3;

typedef struct
{
 unsigned b1 :1;
 unsigned b2 :1;
 unsigned b3 :1;
 unsigned b4 :1;
 unsigned b5 :1;
 unsigned b6 :1;
 unsigned b7 :1;
 unsigned b8 :1;
 unsigned b9 :1;
 unsigned b10 :1;
 unsigned b11 :1;
 unsigned b12 :1;
 unsigned b13 :1;
 unsigned b14 :1;
 unsigned b15 :1;
 unsigned b16 :1;
 unsigned b17 :1;
 unsigned b18 :1;
 unsigned b19 :1;
 unsigned b20 :1;
 unsigned b21 :1;
 unsigned b22 :1;
 unsigned b23 :1;
 unsigned b24 :1;
 unsigned b25 :1;
 unsigned b26 :1;
 unsigned b27 :1;
 unsigned b28 :1;
 unsigned b29 :1;
 unsigned b30 :1;
 unsigned b31 :1;
 unsigned b32 :1;
}BIT_4;

typedef struct
{
 unsigned b1      :1;
 unsigned b2_10   :9;
 unsigned b11_19 :9;
 unsigned b20_32 :13;
}BIT_5;

typedef union
{
 unsigned  long CODEWORD;
 unsigned BIT_1 MESSAGE;
 unsigned BIT_2 ADDRESS;
 unsigned BIT_3 BUFFER;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 unsigned BIT_4 BINARY;
 unsigned BIT_5 BLUGCODE;
 char BYTE[4];
}DOUBLE_WORD;

typedef struct
{
 unsigned W1_SB  :1;
 unsigned W1_M1  :7;
 unsigned W1_M2  :7;
 unsigned W1_M3  :6;
 unsigned W1_EB  :11;

 unsigned W2_SB  :1;
 unsigned W2_M3  :1;
 unsigned W2_M4  :7;
 unsigned W2_M5  :7;
 unsigned W2_M6  :5;
 unsigned W2_EB  :11;

 unsigned W3_SB  :1;
 unsigned W3_M6  :2;
 unsigned W3_M7  :7;
 unsigned W3_M8  :7;
 unsigned W3_M9  :4;
 unsigned W3_EB  :11;

 unsigned W4SBb  :1;
 unsigned W4_M9  :3;
 unsigned W4_M10 :7;
 unsigned W4_M11 :7;
 unsigned W4_M12 :3;
 unsigned W4_EB  :11;

 unsigned W5_SB  :1;
 unsigned W5_M12 :4;
 unsigned W5_M13 :7;
 unsigned W5_M14 :7;
 unsigned W5_M15 :2;
 unsigned W5_EB  :11;

 unsigned W6_SB  :1;
 unsigned W6_M15 :5;
 unsigned W6_M16 :7;
 unsigned W6_M17 :7;
 unsigned W6_M18 :1;
 unsigned W6_EB  :11;

 unsigned W7_SB  :1;
 unsigned W7_M18 :6;
 unsigned W7_M19 :7;
 unsigned W7_M20 :7;
 unsigned W7_EB  :11;
}MCW_BIT1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
typedef union
{
 unsigned long      CODE_WORD[6];
 unsigned MCW_BIT1 MESSAGE_7CW;
}ENTRY_MESSAGE;

typedef struct
{
 unsigned W1_M1     :7;
 unsigned Insert1   :1;
 unsigned W1_M2     :7;
 unsigned Insert2   :1;
 unsigned W1_M3     :6;
 unsigned W2_M3     :1;
 unsigned Insert3   :1;
 unsigned W2_M4     :7;
 unsigned Insert4   :1;
 unsigned W2_M5     :7;
 unsigned Insert5   :1;
 unsigned W2_M6     :5;
 unsigned W3_M6     :2;
 unsigned Insert6   :1;
 unsigned W3_M7     :7;
 unsigned Insert7   :1;
 unsigned W3_M8     :7;
 unsigned Insert8   :1;
 unsigned W3_M9     :4;
 unsigned W4_M9     :3;
 unsigned Insert9   :1;
 unsigned W4_M10    :7;
 unsigned Insert10  :1;
 unsigned W4_M11    :7;
 unsigned Insert11  :1;
 unsigned W4_M12    :3;
 unsigned W5_M12    :4;
 unsigned Insert12  :1;
 unsigned W5_M13    :7;
 unsigned Insert13  :1;
 unsigned W5_M14    :7;
 unsigned Insert14  :1;
 unsigned W5_M15    :2;
 unsigned W6_M15    :5;
 unsigned Insert15  :1;
 unsigned W6_M16    :7;
 unsigned Insert16  :1;
 unsigned W6_M17    :7;
 unsigned Insert17  :1;
 unsigned W6_M18    :1;
 unsigned W7_M18    :6;
 unsigned Insert18  :1;
 unsigned W7_M19    :7;
 unsigned Insert19  :1;
 unsigned W7_M20    :7;
 unsigned Insert20  :1;
}CHECK_BIT1;
```

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
typedef union
{
 unsigned char        CHECK_MCW[19];
 unsigned CHECK_BIT1 CHECK_MCW1;
}CHECK_MESSAGE;

typedef struct
{
 unsigned b1_4:4;
 unsigned b5_8:4;
}BIT_6;

typedef union
{
 short int       MARK_CODE;
 unsigned BIT_6 MARK_CODE1;
}CODE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

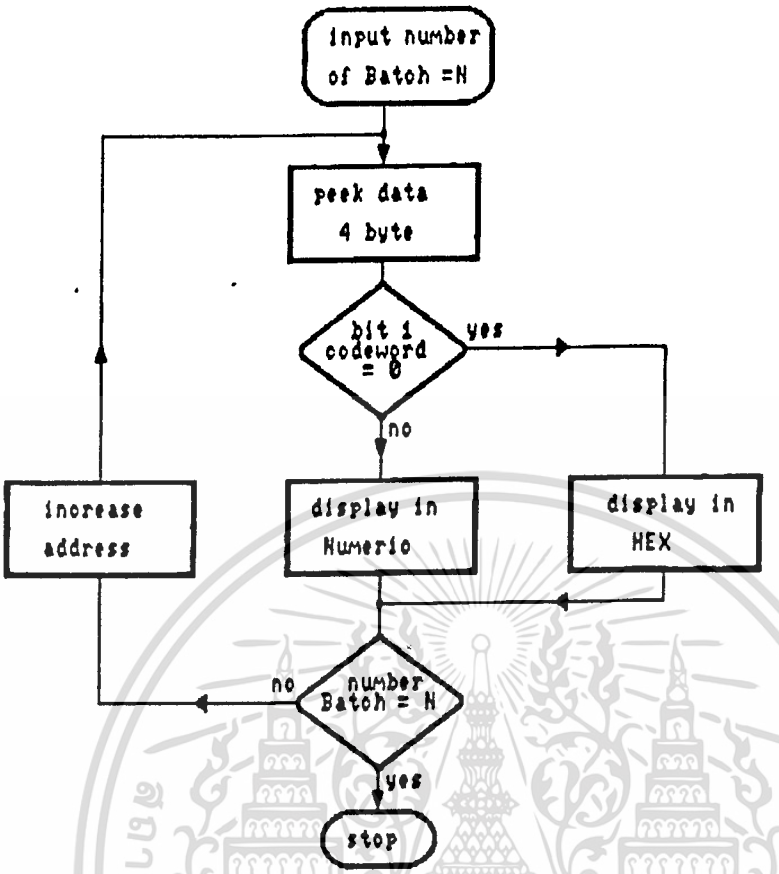
รายละเอียดของโปรแกรม Demons.c

```
/* Program displaying all pager data received */

#include<pager.h>

void POCSAG(DOUBLE_WORD *mes);                  /* types characters */
char changecode(register char z);               /* changes pager data to
                                                   ASCII */

main()
 {
  DOUBLE_WORD data;
  unsigned i, j, k;
  clrscr();
  printf("                                Message of Pager\n\n");
  while(!kbhit())
  {
   for(i=0;i<7;i++)
   {
    for(j=0;j<17;j++)
     {
      for(k=0;k<4;k++)
        data.BYTE[k] = peekb(SEGMENT, i*17*4*2+j*4*2+k*2);
      POCSAG(&data);
      if(data.CODEWORD == CLEAR)
        break;
     }
    if(data.CODEWORD == CLEAR)
      break;
    printf("\n\n");
   }
   gotoxy(1,3);
  }
  gotoxy(1,24);
 }
/*-------------------------------------------------------------*/
void POCSAG(DOUBLE_WORD *mes)
 {
   char a[5];
   int i;
   a[0] = changecode((char)mes->MESSAGE.b1_4);
   a[1] = changecode((char)mes->MESSAGE.b5_8);
   a[2] = changecode((char)mes->MESSAGE.b9_12);
   a[3] = changecode((char)mes->MESSAGE.b13_16);
   a[4] = changecode((char)mes->MESSAGE.b17_20);
   for(i=0;i<5;i++)
     printf("%c",a[i]);
   if(mes->MESSAGE.b0 == 0)
      printf(",  ");
    else
      printf("   ");
 }
/*-------------------------------------------------------------*/
char changecode(register char z)
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  switch (z)
  {
   case 0xa: z = 'A';
             break;
   case 0xb: z = 'B' ;
             break;
   case 0xc: z = 'C';
             break;
   case 0xd: z = 'D';
             break;
   case 0xe: z = 'E';
             break;
   case 0xf: z = 'F';
             break;
   default : z |= 0x30;
             break;
  }
   return z;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART PROGRAM DISPLAY ALL DATA PAGER IN MODE NUMERIC

รูปที่ 4.1 โฟลว์ชาร์ตโปรแกรม Demons.c แสดงข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของโปรแกรม Pager.c

```
* Program displaying pager message with address definition produced by
  35-M and developtment by 36-M 1995 of Pager Reciver Card Project
  Group,in advisory by Kitdakorn Klomkarn, 1994.-1995.*/

include<n_pager.h>      /* a special include file for pager programs */
include<stdlib.h>

* Define variable */
OUBLE_WORD   .data,address,address1;
nt i,j,k;
hort int m1,m2,m3,m4,m5,m6,m7,m8,m9,m10;
hort int m11,m12,m13,m14,m15,m16,m17,m18,m19,m20;
hort int m21,m22,m23,m24,m25,m26,m27,m28,m29,m30,m31,m32;
hort int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
hort int n11,n12,n13,n14,n15,n16,n17,n18,n19,n20;
hort int n21,n22,n23,n24,n25,n26,n27,n28,n29,n30,n31,n32;

* Prototype of function */
* void display(DOUBLE_WORD *addr);*/ /* function for displaying pager

oid display(void);
OUBLE_WORD convert_bits(DOUBLE_WORD address); /* converts address bits */
oid POCSAG(DOUBLE_WORD *dat);   /* types characters */
har changecode(register char z); /* changes POCSAG code to ASCII  */

ain()
{
 DOUBLE_WORD address;
 char dummy[10];
 do
   {
    clrscr();
    printf("                    <<<<< READ MESSAGE PAGER (Numeric) >>>>>\n\n");
    display();
    gotoxy(3,24);printf("Would you like to exit program");
    printf(":(press Y = yes,others = no)? ");
   }while(tolower(getche()) != 'y');
 gotoxy(1,24);
}
*-------------------------------------------------------------------------*/
oid display(void)
{
 gotoxy(1,2);printf("Enter blugcode of pager -->");
 gotoxy(29,2);scanf("%lu",&address);
 address1.BINARY.b1 =address.BINARY.b20;
 address1.BINARY.b2 =address.BINARY.b19;
 address1.BINARY.b3 =address.BINARY.b18;
 address1.BINARY.b4 =address.BINARY.b17;
 address1.BINARY.b5 =address.BINARY.b16;
 address1.BINARY.b6 =address.BINARY.b15;
 address1.BINARY.b7 =address.BINARY.b14;
 address1.BINARY.b8 =address.BINARY.b13;
 address1.BINARY.b9 =address.BINARY.b12;
 address1.BINARY.b10=address.BINARY.b11;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ddress1.BINARY.b11=address.BINARY.b10;
ddress1.BINARY.b12=address.BINARY.b9;
ddress1.BINARY.b13=address.BINARY.b8;
ddress1.BINARY.b14=address.BINARY.b7;
ddress1.BINARY.b15=address.BINARY.b6;
ddress1.BINARY.b16=address.BINARY.b5;
ddress1.BINARY.b17=address.BINARY.b4;
ddress1.BINARY.b18=address.BINARY.b3;
2 =address1.BINARY.b2;n3 =address1.BINARY.b3;n4 =address1.BINARY.b4;
5 =address1.BINARY.b5;n6 =address1.BINARY.b6;n7 =address1.BINARY.b7;
8 =address1.BINARY.b8;n9 =address1.BINARY.b9;n10=address1.BINARY.b10;
1=address1.BINARY.b11;n12=address1.BINARY.b12;n13=address1.BINARY.b13;
4=address1.BINARY.b14;n15=address1.BINARY.b15;n16=address1.BINARY.b16;
7=address1.BINARY.b17;n18=address1.BINARY.b18;n19=address1.BINARY.b19;
toxy(1,3);printf("binary of blugcode=[%d%d%d%d%d%d%d%d%d%d%d",n2,n3,n4,n5,n6
,n9,n10,n11);
rintf("%d%d%d%d%d%d%d%d]",n12,n13,n14,n15,n16,n17,n18,n19);
rintf("\nMessage of Pager is    ");
= -4;
hile(!kbhit())
 {
   i += 4;for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
   if((data.MESSAGE.b1 == 0) &&
      (address1.BLUGCODE.b2_10  == data.BLUGCODE.b2_10)&&
      (address1.BLUGCODE.b11_19 == data.BLUGCODE.b11_19))
     /* in case of matching address */
     {
       printf("  ");
       gotoxy(25,4);
       i += 4;for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
       while((data.MESSAGE.b1 == 1) && (data.CODEWORD != EOT))
        {
         POCSAG(&data);          /* type message */
         JUMP:                   /* label to which is jumped */
         i += 4;
         for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
        }
       if(data.CODEWORD == FRAME_SYN)
         goto JUMP;              /* go on typing message */
     }
   else if(data.MESSAGE.b1 == 0)
     continue;                 /* continue peeking data from next address */
   else if(data.CODEWORD == EOT)
     do

       {
         i = 0;
         for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
       }while((data.CODEWORD != CLEAR) && !kbhit());
   while((data.CODEWORD == CLEAR) && !kbhit())
     {
      i = 0;
      for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
     }
 }
getch();
```

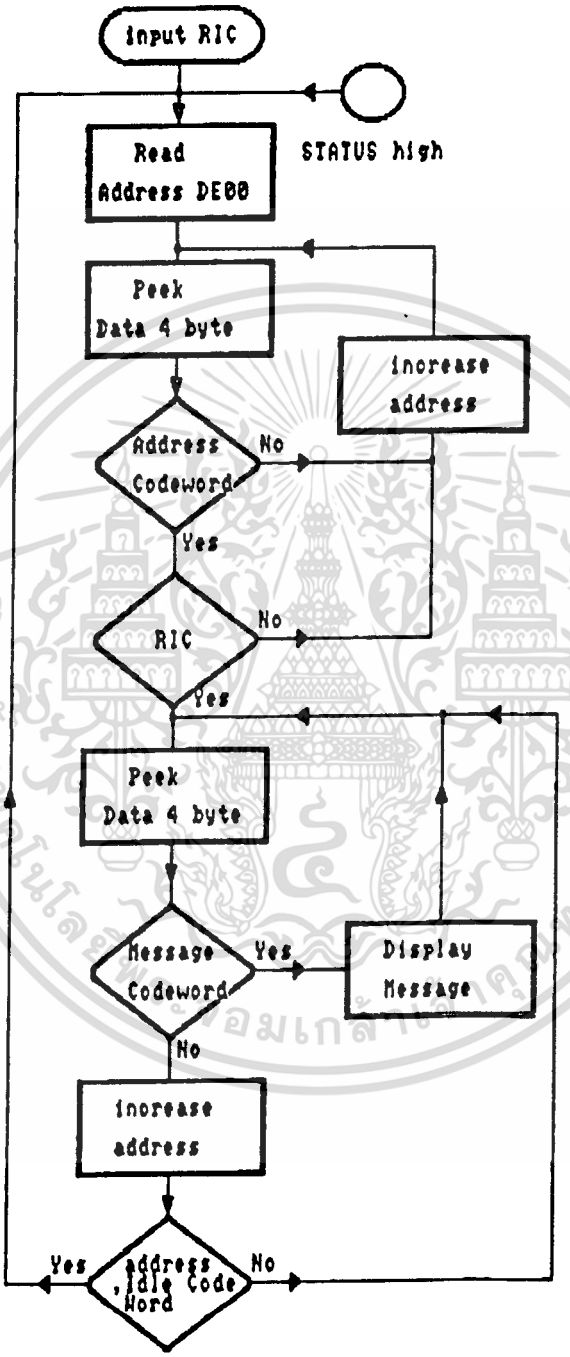
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-----------------------------------------------------------------*/
void POCSAG(DOUBLE_WORD *dat)
 {
   register char a[5];
   int i;
   a[0] = changecode((char)dat->MESSAGE.b2_5);    /* change POCSAG */
   a[1] = changecode((char)dat->MESSAGE.b6_9);    /* code into     */
   a[2] = changecode((char)dat->MESSAGE.b10_13);  /* ASCII code    */
   a[3] = changecode((char)dat->MESSAGE.b14_17);
   a[4] = changecode((char)dat->MESSAGE.b18_21);
   for(i=0;i<5;i++) printf("%c",a[i]);
 }
/*-----------------------------------------------------------------*/
DOUBLE_WORD convert_bits(DOUBLE_WORD address)
 {
  DOUBLE_WORD b;
  b.CODEWORD = 0;
  b.BUFFER.b1  = address.BUFFER.b4;
  b.BUFFER.b2  = address.BUFFER.b3;
  b.BUFFER.b3  = address.BUFFER.b2;
  b.BUFFER.b4  = address.BUFFER.b1;
  b.BUFFER.b5  = address.BUFFER.b8;
  b.BUFFER.b6  = address.BUFFER.b7;
  b.BUFFER.b7  = address.BUFFER.b6;
  b.BUFFER.b8  = address.BUFFER.b5;
  b.BUFFER.b9  = address.BUFFER.b12;
  b.BUFFER.b10 = address.BUFFER.b11;
  b.BUFFER.b11 = address.BUFFER.b10;
  b.BUFFER.b12 = address.BUFFER.b9;
  b.BUFFER.b13 = address.BUFFER.b13;
  b.BUFFER.b14 = address.BUFFER.b12;
  b.BUFFER.b15 = address.BUFFER.b11;
  b.BUFFER.b16 = address.BUFFER.b10;
  b.BUFFER.b17 = 0;
  b.BUFFER.b18 = 0;
  b.BUFFER.b19 = address.BUFFER.b18;
  b.BUFFER.b20 = address.BUFFER.b17;
  return b;
 }
/*-----------------------------------------------------------------*/
char changecode(register char z)
 {
   switch (z)
   {
    case 0xa: z = 0xf0;
              break;
    case 0xb: z = 'U' ;
              break;
    case 0xc: z = 0x20;
              break;
    case 0xd: z = 0x2d;
              break:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 0xe: z = 0x5d;
              break;
    case 0xf: z = 0x5b;
              break;
    default : z |= 0x30;
              break;
   }
    return z;
}
/*-------------------------------------------------------------------*/
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART DECODE DATA PAGER IN MODE NUMERIC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.2 โฟลว์ชาร์ตโปรแกรม Pager.c สำหรับแสดงข้อมูลเฉพาะข่าวสาร

รายละเอียดของโปรแกรม Binary.c

```
/* This program used  for analysis data of pager type Alpha-Numeric and
   display message product by   Udomporn Sununachaikul 36-M 36012087 in
   avisory by Kitdrakorn Khomkan (Pager Code Recive Card) */

#include<n_pager.h>     /* a special include file for pager programs */
#include<stdlib.h>
#include<conio.h>

/*Photo type function */
void display(void);
void type_bits(DOUBLE_WORD data1,int count1);
CHECK_MESSAGE transfer_codeword(ENTRY_MESSAGE entry_message);
void POCSAG(CHECK_MESSAGE check_message);
char changecode1(short int codein);
char changecode2(short int codein);
char changecode3(short int codein);
char changecode4(short int codein);
char changecode5(short int codein);

/*Define varuiabled */
DOUBLE_WORD   data,data1,address,address1; /* for change blugcode        */
ENTRY_MESSAGE entry_message,entry_message1;/* for get message codeword   */
CHECK_MESSAGE check_message,check_message1;
int i,j,k;
int num_code,count,count1,batch,cw,cc,c,a,oldx;
short codein,codein1;
short int m1,m2,m3,m4,m5,m6,m7,m8,m9,m10;
short int m11,m12,m13,m14,m15,m16,m17,m18,m19,m20;
short int m21,m22,m23,m24,m25,m26,m27,m28,m29,m30,m31,m32;
short int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
short int n11,n12,n13,n14,n15,n16,n17,n18,n19,n20;
short int n21,n22,n23,n24,n25,n26,n27,n28,n29,n30,n31,n32;

void main(void)
{
 do
 {
  clrscr();
  gotoxy(20,1);printf("<<<<< READ MESSAGE PAGER (Alpha-Numeric) >>>>>");
  display();
  gotoxy(1,25);printf("Would you like to exit program);
;
 }while(tolower(getche()) != 'y');
 gotoxy(1,25);
}

/*void display(DOUBLE_WORD address)*/

void display(void)
{
 gotoxy(1,2);printf("Enter PLUGCODE of Pager -->");
 gotoxy(29,2);scanf("%lu",&address);
 gotoxy(1,3);printf("Enter number of Batch ---->");
 gotoxy(29,3);scanf("%d",&batch);

 address1.BINARY.b2 =address.BINARY.b21;
 address1.BINARY.b3 =address.BINARY.b20;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 address1.BINARY.b4 =address.BINARY.b19;
 address1.BINARY.b5 =address.BINARY.b18;
 address1.BINARY.b6 =address.BINARY.b17;
 address1.BINARY.b7 =address.BINARY.b16;
 address1.BINARY.b8 =address.BINARY.b15;
 address1.BINARY.b9 =address.BINARY.b14;
 address1.BINARY.b10=address.BINARY.b13;
 address1.BINARY.b11=address.BINARY.b12;
 address1.BINARY.b12=address.BINARY.b11;
 address1.BINARY.b13=address.BINARY.b10;
 address1.BINARY.b14=address.BINARY.b9;
 address1.BINARY.b15=address.BINARY.b8;
 address1.BINARY.b16=address.BINARY.b7;
 address1.BINARY.b17=address.BINARY.b6;
 address1.BINARY.b18=address.BINARY.b5;
 address1.BINARY.b19=address.BINARY.b4;
 n2 =address1.BINARY.b2;n3 =address1.BINARY.b3;n4 =address1.BINARY.b4;
 n5 =address1.BINARY.b5;n6 =address1.BINARY.b6;n7 =address1.BINARY.b7;
 n8 =address1.BINARY.b8;n9 =address1.BINARY.b9;n10=address1.BINARY.b10;
 n11=address1.BINARY.b11;n12=address1.BINARY.b12;n13=address1.BINARY.b13;
 n14=address1.BINARY.b14;n15=address1.BINARY.b15;n16=address1.BINARY.b16;
 n17=address1.BINARY.b17;n18=address1.BINARY.b18;n19=address1.BINARY.b19;
 gotoxy(1,4);printf("binary of PLUGCODE=[%d%d%d%d%d%d%d%d%d%d%d
,n8,n9,n10,n11);
 printf("%d%d%d%d%d%d%d%d]",n12,n13,n14,n15,n16,n17,n18,n19);

 printf("\nMessage of Pager is   ");

 i=-4;
 num_code=0;
 count=0;
 cw=0;
 oldx=0;

 while((!kbhit())&&(num_code<((batch*17))))
      {

       LOOP1:

       i+=4;for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
       count++;num_code++;
       type_bits(data,count);
       if(count==17)
         {
          count=0;
          getch();
         }

       if((address1.BLUGCODE.b2_10  == data.BLUGCODE.b2_10)&&
          (address1.BLUGCODE.b11_19 == data.BLUGCODE.b11_19))
         {

          printf("\a"); /* Beep sound */
          cw--;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP2:

i+=4;for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
count++;num_code++;
type_bits(data,count);
/* check proability of b1 */
if((cw!=5)&&
   (count<17)&&
   (data.CODEWORD!=FRAME_SYN)&&
   (data.CODEWORD!=IDLE_WORD)&&
   (data.CODEWORD!=EOT))
   data.MESSAGE.b1=1;

if(count==17)
  {
   count=0;
   getch();
   goto LOOP2;
  }
if((data.MESSAGE.b1==1)&&(data.CODEWORD!=EOT))
  {
   cw++;
   gotoxy(1,7);printf("%d  ",cw);printf("%lu",data.CODEWORD);
   entry_message.CODE_WORD[cw]=data.CODEWORD;
   gotoxy(1,18);printf("%d  ",cw);printf("%lu",entry_message.CODE_WORD
   [cw]);
   gotoxy(1,19);printf("Transfers complet ");
   if(cw==6)
     {
      for(a=0;a<20;a++) check_message.CHECK_MCW[a]=0;

            (1,22);for(a=0;a<20;a++)printf("%d ",check_message.CHECK_M
            CW[a];
      gotoxy(1,8);printf("Transfers complet");
      gotoxy(1,9);printf("7 message codeword");
      cc=1;
      for(cc=0;cc<7;cc++)
         {
          gotoxy(1,(cc+10));
          printf("%lu",entry_message.CODE_WORD[cc]);
         }
      getch();
      POCSAG(check_message); /* Mark & display message */
      for(c=1;c<8;c++)entry_message.CODE_WORD[c]=0; /* Reset variable
      */
      cw=-1;
     }
   goto LOOP2;
  }

if(data.CODEWORD==FRAME_SYN)
  {
   goto LOOP2;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          if((((data.CODEWORD==IDLE_WORD)||
               (data.CODEWORD==EOT)||
               (data.CODEWORD==END))&&
               (cw!=7))||
               ((data.MESSAGE.b1==0)&&(cw!=7)&&(data.CODEWORD!=IDLE_WORD
            {
             for(a=0;a<20;a++) check_message.CHECK_MCW[a]=0;
             check_message=transfer_codeword(entry_message);
             gotoxy(1,22);for(a=0;a<20;a++)printf("%d ",check_message.
             CHECK_MCW[a]);
             gotoxy(1,8);printf("Transfers uncomplet");
             gotoxy(1,9);printf("7 message codeword");
             cc=1;
             for(cc=1;cc<8;cc++)
                {
                 gotoxy(1,(cc+9));
                 printf("%lu",entry_message.CODE_WORD[cc]);
                }
             getch();
             POCSAG(check_message); /* Mark & display message */
             for(c=1;c<8;c++)entry_message.CODE_WORD[c]=0; /
             gotoxy(1,25);printf("End of Messagge:Press Any Keys");
             goto LOOP3;
            }
          oldx=0;
          goto LOOP1;
         }
        oldx=0;
        if(count==17)
          {
           count=0;
           getch();
          }
        }
 LOOP3:
 getch();
}

void type_bits(DOUBLE_WORD data1,int count1)
{
  m1=data1.BINARY.b1;
  m2=data1.BINARY.b2;
  m3=data1.BINARY.b3;
  m4=data1.BINARY.b4;
  m5=data1.BINARY.b5;
  m6=data1.BINARY.b6;
  m7=data1.BINARY.b7;
  m8=data1.BINARY.b8;
  m9=data1.BINARY.b9;
  m10=data1.BINARY.b10;
  m11=data1.BINARY.b11;
  m12=data1.BINARY.b12;
  m13=data1.BINARY.b13;
  m14=data1.BINARY.b14;
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   m15=data1.BINARY.b15;
   m16=data1.BINARY.b16;
   m17=data1.BINARY.b17;
   m18=data1.BINARY.b18;
   m19=data1.BINARY.b19;
   m20=data1.BINARY.b20;
   m21=data1.BINARY.b21;
   m22=data1.BINARY.b22;
   m23=data1.BINARY.b23;
   m24=data1.BINARY.b24;
   m25=data1.BINARY.b25;
   m26=data1.BINARY.b26;
   m27=data1.BINARY.b27;
   m28=data1.BINARY.b28;
   m29=data1.BINARY.b29;
   m30=data1.BINARY.b30;
   m31=data1.BINARY.b31;
   m32=data1.BINARY.b32;
   gotoxy(22,4+count1);printf("%d",count1);
   gotoxy(25,4+count1);
   printf("%d %d%d%d%d%d%d%d %d%d",m1,m2,m3,m4,m5,m6,m7,m8,m9,m10);
   printf("%d%d%d%d%d %d%d%d%d%d%d",m11,m12,m13,m14,m15,m16,m17,m18,m19,m20);
   printf("%d %d%d%d%d%d%d%d%d%d%d",m21,m22,m23,m24,m25,m26,m27,m28,m29,m30);
   printf("%d%d",m31,m32);
   printf(" %lu",data1.CODEWORD);
}


CHECK_MESSAGE transfer_codeword(ENTRY_MESSAGE entry_message1)
{
 check_message1.CHECK_MCW1.W1_M1  = entry_message1.MESSAGE_7CW.W1_M1;
 check_message1.CHECK_MCW1.W1_M2  = entry_message1.MESSAGE_7CW.W1_M2;
 check_message1.CHECK_MCW1.W1_M3  = entry_message1.MESSAGE_7CW.W1_M3;

 check_message1.CHECK_MCW1.W2_M3  = entry_message1.MESSAGE_7CW.W2_M3;
 check_message1.CHECK_MCW1.W2_M4  = entry_message1.MESSAGE_7CW.W2_M4;
 check_message1.CHECK_MCW1.W2_M5  = entry_message1.MESSAGE_7CW.W2_M5;
 check_message1.CHECK_MCW1.W2_M6  = entry_message1.MESSAGE_7CW.W2_M6;

 check_message1.CHECK_MCW1.W3_M6  = entry_message1.MESSAGE_7CW.W3_M6;
 check_message1.CHECK_MCW1.W3_M7  = entry_message1.MESSAGE_7CW.W3_M7;
 check_message1.CHECK_MCW1.W3_M8  = entry_message1.MESSAGE_7CW.W3_M8;
 check_message1.CHECK_MCW1.W3_M9  = entry_message1.MESSAGE_7CW.W3_M9;

 check_message1.CHECK_MCW1.W4_M9  = entry_message1.MESSAGE_7CW.W4_M9;
 check_message1.CHECK_MCW1.W4_M10 = entry_message1.MESSAGE_7CW.W4_M10;
 check_message1.CHECK_MCW1.W4_M11 = entry_message1.MESSAGE_7CW.W4_M11;
 check_message1.CHECK_MCW1.W4_M12 = entry_message1.MESSAGE_7CW.W4_M12;

 check_message1.CHECK_MCW1.W5_M12 = entry_message1.MESSAGE_7CW.W5_M12;
 check_message1.CHECK_MCW1.W5_M13 = entry_message1.MESSAGE_7CW.W5_M13;
 check_message1.CHECK_MCW1.W5_M14 = entry_message1.MESSAGE_7CW.W5_M14;
 check_message1.CHECK_MCW1.W5_M15 = entry_message1.MESSAGE_7CW.W5_M15;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 check_message1.CHECK_MCW1.W6_M15 = entry_message1.MESSAGE_7CW.W6_M15;
 check_message1.CHECK_MCW1.W6_M16 = entry_message1.MESSAGE_7CW.W6_M16;
 check_message1.CHECK_MCW1.W6_M17 = entry_message1.MESSAGE_7CW.W6_M17;
 check_message1.CHECK_MCW1.W6_M18 = entry_message1.MESSAGE_7CW.W6_M18;

 check_message1.CHECK_MCW1.W7_M18 = entry_message1.MESSAGE_7CW.W7_M18;
 check_message1.CHECK_MCW1.W7_M19 = entry_message1.MESSAGE_7CW.W7_M19;
 check_message1.CHECK_MCW1.W7_M20 = entry_message1.MESSAGE_7CW.W7_M20;
 return check_message1;
}


void POCSAG(CHECK_MESSAGE check_message1)
{
 int t,l=0;
 char alphanumeric;
 gotoxy(1,23);for(a=0;a<20;a++) printf("%d ",check_message.CHECK_MCW[a])
 for(l=0;l<20;l++)
 {
  codein=0;
  codein=check_message1.CHECK_MCW[l];
  if((codein>=48)&&(codein<=63))t=1;
  if((codein>=64)&&(codein<=79))t=2;
  if((codein>=80)&&(codein<=95))t=3;
  if((codein>=96)&&(codein<=111))t=4;
  if((codein>=112)&&(codein<=127))t=5;
  if(!((codein>=48)&&(codein<=127)))t=6;
  switch(t)
        {
         case 1:
              alphanumeric=changecode1(codein); /*1100*/
              break;
         case 2:
              alphanumeric=changecode2(codein); /*0010*/
              break;
         case 3:
              alphanumeric=changecode3(codein); /*1010*/
              break;
         case 4:
              alphanumeric=changecode4(codein); /*0110*/
              break;
         case 5:
              alphanumeric=changecode5(codein); /*1110*/
              break;
         case 6:
              alphanumeric=' ';
              break;
         default:
              alphanumeric=' ';
              break;
        }
  gotoxy(1+oldx+1,24);printf("%c",alphanumeric);
 }
 oldx=(wherex()-1);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char changecode1(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 48:
             d = '0';
             break;
        case 49:
             d = '1';
             break;
        case 50:
             d = '2';
             break;
        case 51:
             d = '3';
             break;
        case 52:
             d = '4';
             break;
        case 53:
             d = '5';
             break;
        case 54:
             d = '6';
             break;
        case 55:
             d = '7';
             break;
        case 56:
             d = '8';
             break;
        case 57:
             d = '9';
             break;
        case 58:
             d = ':';
             break;
        case 59:
             d = ';';
             break;
        case 60:
             d = '<';
             break;
        case 61:
             d = '=';
             break;
        case 62:
             d = '>';
             break;
        case 63:
             d = '?';
             break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }.
 return d;
}


char changecode2(short int codein1)
{
 char d;
 switch(codein1)
        {
         case 64:
              d = '@';
              break;
         case 65:
              d = 'A';
              break;
        . case 66:
              d = 'B';
              break;
         case 67:
              d = 'C';
              break;
         case 68:
              d = 'D';
              break;
         case 69:
              d = 'E';
              break;
         case 70:
              d = 'F';
              break;
         case 71:
              d = 'G';
              break;
         case 72:
              d = 'H';
              break;
         case 73:
              d = 'I';
              break;
         case 74:
              d = 'J';
              break;
         case 75:
              d = 'K';
              break;
         case 76:
              d = 'L';
              break;
         case 77:
              d = 'M';
              break;
         case 78:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            d = 'N';
            break;
        case 79:
            d = 'O';
            break;
       }
 return d;
}

char changecode3(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 80:
            d = 'P';
            break;
        case 81:
            d = 'Q';
            break;
        case 82:
            d = 'R';
            break;
        case 83:
            d = 'S';
            break;
        case 84:
            d = 'T';
            break;
        case 85:
            d = 'U';
            break;
        case 86:
            d = 'V';
            break;
        case 87:
            d = 'W';
            break;
        case 88:
            d = 'X';
            break;
        case 89:
            d = 'Y';
            break;
        case 90:
            d = 'Z';
            break;
        case 91:
            d = '[';
            break;
        case 92:
            d = ' ';
            break;
        case 93:
            d = ']';
            break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 94:
             d = '^';
             break;
        case 95:
             d = '-';
             break;
       }
 return d;
}


char changecode4(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 96:
             d = '"';
             break;
        case 97:
             d = 'a';
             break;
        case 98:
             d = 'b';
             break;
        case 99:
             d = 'c';
             break;
        case 100:
             d = 'd';
             break;
        case 101:
             d = 'e';
             break;
        case 102:
             d = 'f';
             break;
        case 103:
             d = 'g';
             break;
        case 104:
             d = 'h';
             break;
        case 105:
             d = 'i';
             break;
        case 106:
             d = 'j';
             break;
        case 107:
             d = 'k';
             break;
        case 108:
             d = 'l';
             break;
```

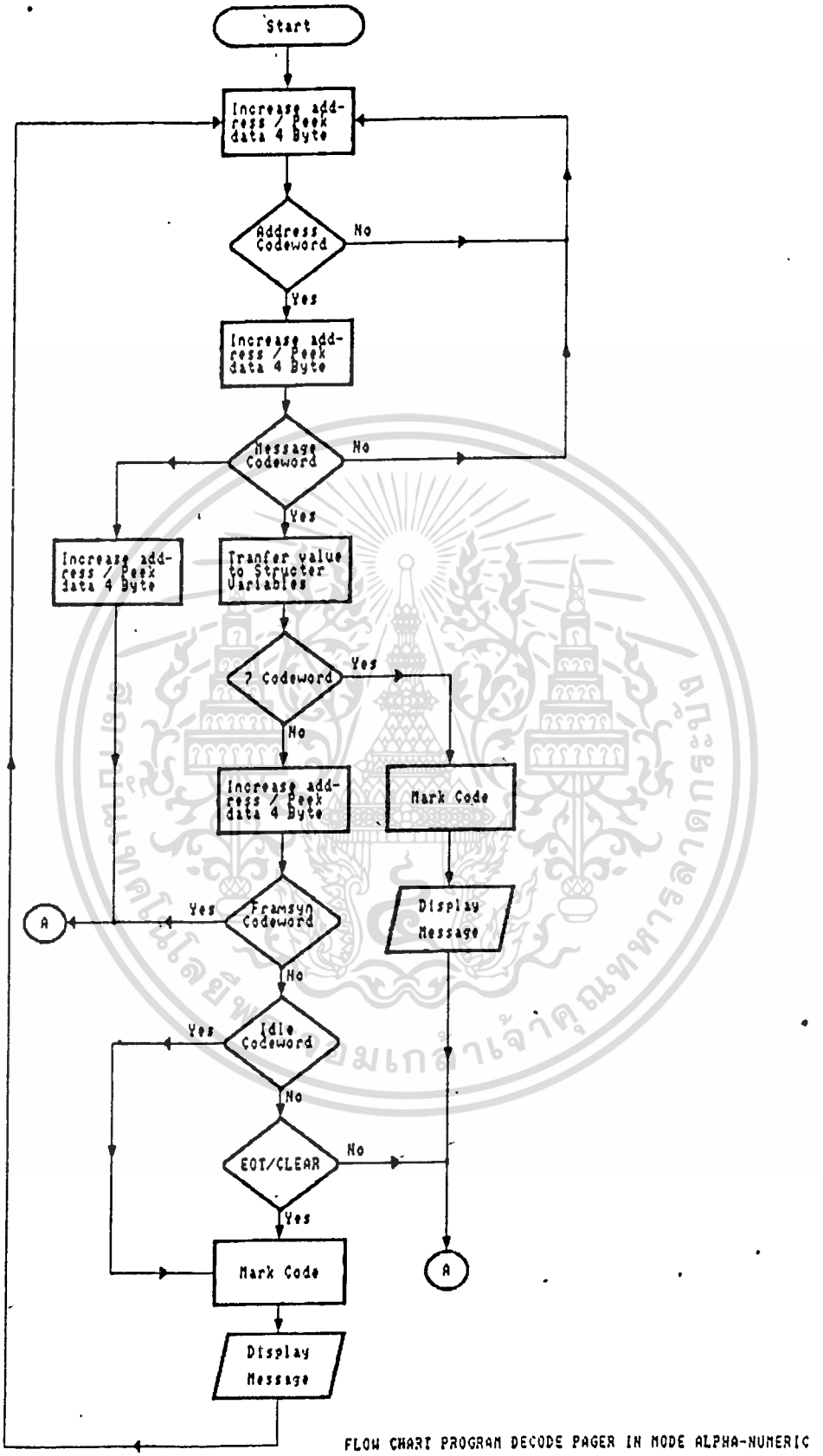
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 109:
             d = 'm';
             break;
        case 110:
             d = 'n';
             break;
        case 111:
             d = 'o';
             break;
       }
 return d;
}

char changecode5(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 112:
             d = 'p';
             break;
        case 113:
             d = 'q';
             break;
        case 114:
             d = 'r';
             break;
        case 115:
             d = 's';
             break;
        case 116:
             d = 't';
             break;
        case 117:
             d = 'u';
             break;
        case 118:
             d = 'v';
             break;
        case 119:
             d = 'w';
             break;
        case 120:
             d = 'x';
             break;
        case 121:
             d = 'y';
             break;
        case 122:
             d = 'z';
             break;
        case 123:
             d = '{';
             break;
        case 124:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            d = '|';
            break;
        case 125:
            d = '}';
            break;
        case 126:
            d = '~';
            break;
        case 127:
            d = '#';
            break;
    }
return d;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART PROGRAM DECODE PAGER IN MODE ALPHA-NUMERIC

รูปที่ 4.3 โฟลว์ชาร์ตโปรแกรม Binary.c สำหรับแสดงข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของโปรแกรม Pager1.c

```
/* This program used  for analysis data of pager type Alpha-Numeric and
   display message product by   Udomporn Sununachaikul 36-M 36012087 in
   avisory by Kitdrakorn Khomkan (Pager Code Recive Card) */

#include<n_pager.h>     /* a special include file for pager programs */
#include<stdlib.h>
#include<conio.h>

/*Photo type function */
void display(void);
CHECK_MESSAGE transfer_codeword(ENTRY_MESSAGE entry_message);
void POCSAG(CHECK_MESSAGE check_message);
char changecode1(short int codein);
char changecode2(short int codein);
char changecode3(short int codein);
char changecode4(short int codein);
char changecode5(short int codein);

/*Define varuiabled */
DOUBLE_WORD   data,data1,address,address1; /* for change blugcode
ENTRY_MESSAGE entry_message,entry_message1;/* for get message codeword
CHECK_MESSAGE check_message,check_message1;/*
int i,j,k;
int num_code,count,count1,batch,cw,cc,c,a,oldx;
short codein,codein1;
short int m1,m2,m3,m4,m5,m6,m7,m8,m9,m10;
short int m11,m12,m13,m14,m15,m16,m17,m18,m19,m20;
short int m21,m22,m23,m24,m25,m26,m27,m28,m29,m30,m31,m32;
short int n1,n2,n3,n4,n5,n6,n7,n8,n9,n10;
short int n11,n12,n13,n14,n15,n16,n17,n18,n19,n20;
short int n21,n22,n23,n24,n25,n26,n27,n28,n29,n30,n31,n32;

void main(void)
{
 do
 {
  clrscr();
  gotoxy(20,1);printf("<<<<< READ MESSAGE PAGER (Alpha-Nunmeric) >>>>>");
  display();
  gotoxy(1,25);printf("Would you like to exit program);
;
 }while(tolower(getche()) != 'y');
 gotoxy(1,25);
}

/*void display(DOUBLE_WORD address)*/

void display(void)
{
 gotoxy(1,2);printf("Enter PLUGCODE of Pager -->");
 gotoxy(29,2);scanf("%lu",&address);
 gotoxy(1,3);printf("Standard [Post Office Code Standardization Advisory
(POCSAG)]");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 address1.BINARY.b2 =address.BINARY.b21;
 address1.BINARY.b3 =address.BINARY.b20;
 address1.BINARY.b4 =address.BINARY.b19;
 address1.BINARY.b5 =address.BINARY.b18;
 address1.BINARY.b6 =address.BINARY.b17;
 address1.BINARY.b7 =address.BINARY.b16;
 address1.BINARY.b8 =address.BINARY.b15;
 address1.BINARY.b9 =address.BINARY.b14;
 address1.BINARY.b10=address.BINARY.b13;
 address1.BINARY.b11=address.BINARY.b12;
 address1.BINARY.b12=address.BINARY.b11;
 address1.BINARY.b13=address.BINARY.b10;
 address1.BINARY.b14=address.BINARY.b9;
 address1.BINARY.b15=address.BINARY.b8;
 address1.BINARY.b16=address.BINARY.b7;
 address1.BINARY.b17=address.BINARY.b6;
 address1.BINARY.b18=address.BINARY.b5;
 address1.BINARY.b19=address.BINARY.b4;
 n2 =address1.BINARY.b2;n3 =address1.BINARY.b3;n4 =address1.BINARY.b4;
 n5 =address1.BINARY.b5;n6 =address1.BINARY.b6;n7 =address1.BINARY.b7;
 n8 =address1.BINARY.b8;n9 =address1.BINARY.b9;n10=address1.BINARY.b10;
 n11=address1.BINARY.b11;n12=address1.BINARY.b12;n13=address1.BINARY.b13;
 n14=address1.BINARY.b14;n15=address1.BINARY.b15;n16=address1.BINARY.b16;
 n17=address1.BINARY.b17;n18=address1.BINARY.b18;n19=address1.BINARY.b19;
 gotoxy(1,4);printf("Binary of PLUGCODE=[%d%d%d%d%d%d%d%d%d%d%d",n2,n3,n4
,n8,n9,n10,n11);
 printf("%d%d%d%d%d%d%d%d]",n12,n13,n14,n15,n16,n17,n18,n19);

 printf("\nMessage of Pager is    ");

 i=-4;
 num_code=0;
 count=0;
 cw=0;
 oldx=0;

 while(!kbhit())
      {

       LOOP1:

       i+=4;for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
       count++;num_code++;

       if(count==17)
         {
          count=0;
         }

       if((address1.BLUGCODE.b2_10  == data.BLUGCODE.b2_10)&&
          (address1.BLUGCODE.b11_19 == data.BLUGCODE.b11_19))
         {

          printf("\a"); /* Beep sound */
          cw--;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LOOP2:

i+=4;for(j=0;j<4;j++) data.BYTE[j] = peekb(SEGMENT,(i+j)*2);
count++;num_code++;

/* check proability of bl */
if((cw!=5)&&
   (count<17)&&
   (data.CODEWORD!=FRAME_SYN)&&
   (data.CODEWORD!=IDLE_WORD)&&
   (data.CODEWORD!=EOT))
   data.MESSAGE.bl=1;

if(count==17)
  {
   count=0;
   goto LOOP2;
  }

if((data.MESSAGE.bl==1)&&(data.CODEWORD!=EOT))
  {
   cw++;
   entry_message.CODE_WORD[cw]=data.CODEWORD;
   if(cw==6)
     {
      for(a=0;a<20;a++) check_message.CHECK_MCW[a]=0;
      check_message=transfer_codeword(entry_message);
      POCSAG(check_message); /* Mark & display message */
      for(c=1;c<8;c++)entry_message.CODE_WORD[c]=0;

      cw=-1;
     }
   goto LOOP2;
  }

if(data.CODEWORD==FRAME_SYN)
  {
   goto LOOP2;
  }

if((((data.CODEWORD==IDLE_WORD)||
     (data.CODEWORD==EOT)||
     (data.CODEWORD==END))&&
     (cw!=7))||
     ((data.MESSAGE.bl==0)&&(cw!=7)&&(data.CODEWORD!=IDLE_WORD)))
  {
   for(a=0;a<20;a++) check_message.CHECK_MCW[a]=0;
   check_message=transfer_codeword(entry_message);
   POCSAG(check_message); /* Mark & display message */
   for(c=1;c<8;c++)entry_message.CODE_WORD[c]=0; /* Reset variable */
   gotoxy(1,25);printf("End of Messagge:Press Any Keys to continue");
   goto LOOP3;
  }
oldx=0;
goto LOOP1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
          oldx=0;
          if(count==17)
            {
             count=0;
            }
          }
  LOOP3:
  getch();
 }


 CHECK_MESSAGE transfer_codeword(ENTRY_MESSAGE entry_message1)
 {
  check_message1.CHECK_MCW1.W1_M1  = entry_message1.MESSAGE_7CW.W1_M1;
  check_message1.CHECK_MCW1.W1_M2  = entry_message1.MESSAGE_7CW.W1_M2;
  check_message1.CHECK_MCW1.W1_M3  = entry_message1.MESSAGE_7CW.W1_M3;

  check_message1.CHECK_MCW1.W2_M3  = entry_message1.MESSAGE_7CW.W2_M3;
  check_message1.CHECK_MCW1.W2_M4  = entry_message1.MESSAGE_7CW.W2_M4;
  check_message1.CHECK_MCW1.W2_M5  = entry_message1.MESSAGE_7CW.W2_M5;
  check_message1.CHECK_MCW1.W2_M6  = entry_message1.MESSAGE_7CW.W2_M6;

  check_message1.CHECK_MCW1.W3_M6  = entry_message1.MESSAGE_7CW.W3_M6;
  check_message1.CHECK_MCW1.W3_M7  = entry_message1.MESSAGE_7CW.W3_M7;
  check_message1.CHECK_MCW1.W3_M8  = entry_message1.MESSAGE_7CW.W3_M8;
  check_message1.CHECK_MCW1.W3_M9  = entry_message1.MESSAGE_7CW.W3_M9;

  check_message1.CHECK_MCW1.W4_M9  = entry_message1.MESSAGE_7CW.W4_M9;
  check_message1.CHECK_MCW1.W4_M10 = entry_message1.MESSAGE_7CW.W4_M10;
  check_message1.CHECK_MCW1.W4_M11 = entry_message1.MESSAGE_7CW.W4_M11;
  check_message1.CHECK_MCW1.W4_M12 = entry_message1.MESSAGE_7CW.W4_M12;

  check_message1.CHECK_MCW1.W5_M12 = entry_message1.MESSAGE_7CW.W5_M12;
  check_message1.CHECK_MCW1.W5_M13 = entry_message1.MESSAGE_7CW.W5_M13;
  check_message1.CHECK_MCW1.W5_M14 = entry_message1.MESSAGE_7CW.W5_M14;
  check_message1.CHECK_MCW1.W5_M15 = entry_message1.MESSAGE_7CW.W5_M15;

  check_message1.CHECK_MCW1.W6_M15 = entry_message1.MESSAGE_7CW.W6_M15;
  check_message1.CHECK_MCW1.W6_M16 = entry_message1.MESSAGE_7CW.W6_M16;
  check_message1.CHECK_MCW1.W6_M17 = entry_message1.MESSAGE_7CW.W6_M17;
  check_message1.CHECK_MCW1.W6_M18 = entry_message1.MESSAGE_7CW.W6_M18;

  check_message1.CHECK_MCW1.W7_M18 = entry_message1.MESSAGE_7CW.W7_M18;
  check_message1.CHECK_MCW1.W7_M19 = entry_message1.MESSAGE_7CW.W7_M19;
  check_message1.CHECK_MCW1.W7_M20 = entry_message1.MESSAGE_7CW.W7_M20;
  return check_message1;
 }


 void POCSAG(CHECK_MESSAGE check_message1)
 {
  int t,l=0;
  char alphanumeric;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 for(i=0;i<20;i++)
 {
  codein=0;
  codein=check_message1.CHECK_MCW[i];
  if((codein>=48)&&(codein<=63))t=1;
  if((codein>=64)&&(codein<=79))t=2;
  if((codein>=80)&&(codein<=95))t=3;
  if((codein>=96)&&(codein<=111))t=4;
  if((codein>=112)&&(codein<=127))t=5;
  if(!((codein>=48)&&(codein<=127)))t=6;
  switch(t)
        {
         case 1:
              alphanumeric=changecode1(codein); /*1100*/
              break;
         case 2:
              alphanumeric=changecode2(codein); /*0010*/
              break;
         case 3:
              alphanumeric=changecode3(codein); /*1010*/
              break;
         case 4:
              alphanumeric=changecode4(codein); /*0110*/
              break;
         case 5:
              alphanumeric=changecode5(codein); /*1110*/
              break;
         case 6:
              alphanumeric=' ';
              break;
         default:
              alphanumeric=' ';
              break;
        }
  gotoxy(1+oldx+1,7);printf("%c",alphanumeric);
 }
 oldx=(wherex()-1);
}


char changecode1(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 48:
             d = '0';
             break;
        case 49:
             d = '1';
             break;
        case 50:
             d = '2';
             break:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 51:
             d = '3';
             break;
        case 52:
             d = '4';
             break;
        case 53:
             d = '5';
             break;
        case 54:
             d = '6';
             break;
        case 55:
             d = '7';
             break;
        case 56:
             d = '8';
             break;
        case 57:
             d = '9';
             break;
        case 58:
             d = ':';
             break;
        case 59:
             d = ';';
             break;
        case 60:
             d = '<';
             break;
        case 61:
             d = '=';
             break;
        case 62:
             d = '>';
             break;
        case 63:
             d = '?';
             break;
       }
 return d;
}


char changecode2(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 64:
             d = '@';
             break;
        case 65:
             d = 'A';
             break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 66:
             d = 'B';
             break;
        case 67:
             d = 'C';
             break;
        case 68:
             d = 'D';
             break;
        case 69:
             d = 'E';
             break;
        case 70:
             d = 'F';
             break;
        case 71:
             d = 'G';
             break;
        case 72:
             d = 'H';
             break;
        case 73:
             d = 'I';
             break;
        case 74:
             d = 'J';
             break;
        case 75:
             d = 'K';
             break;
        case 76:
             d = 'L';
             break;
        case 77:
             d = 'M';
             break;
        case 78:
             d = 'N';
             break;
        case 79:
             d = 'O';
             break;
      }
 return d;
}

char changecode3(short int codein1)
{
 char d;
 switch(codein1)
       {
        case 80:
             d = 'P';
             break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 81:
             d = 'Q';
             break;
        case 82:
             d = 'R';
             break;
        case 83:
             d = 'S';
             break;
        case 84:
             d = 'T';
             break;
        case 85:
             d = 'U';
             break;
        case 86:
             d = 'V';
             break;
        case 87:
             d = 'W';
             break;
        case 88:
             d = 'X';
             break;
        case 89:
             d = 'Y';
             break;
        case 90:
             d = 'Z';
             break;
        case 91:
             d = '[';
             break;
        case 92:
             d = ' ';
             break;
        case 93:
             d = ']';
             break;
        case 94:
             d = '^';
             break;
        case 95:
             d = '_';
             break;
      }
 return d;
}

char changecode4(short int codein1)
{
```

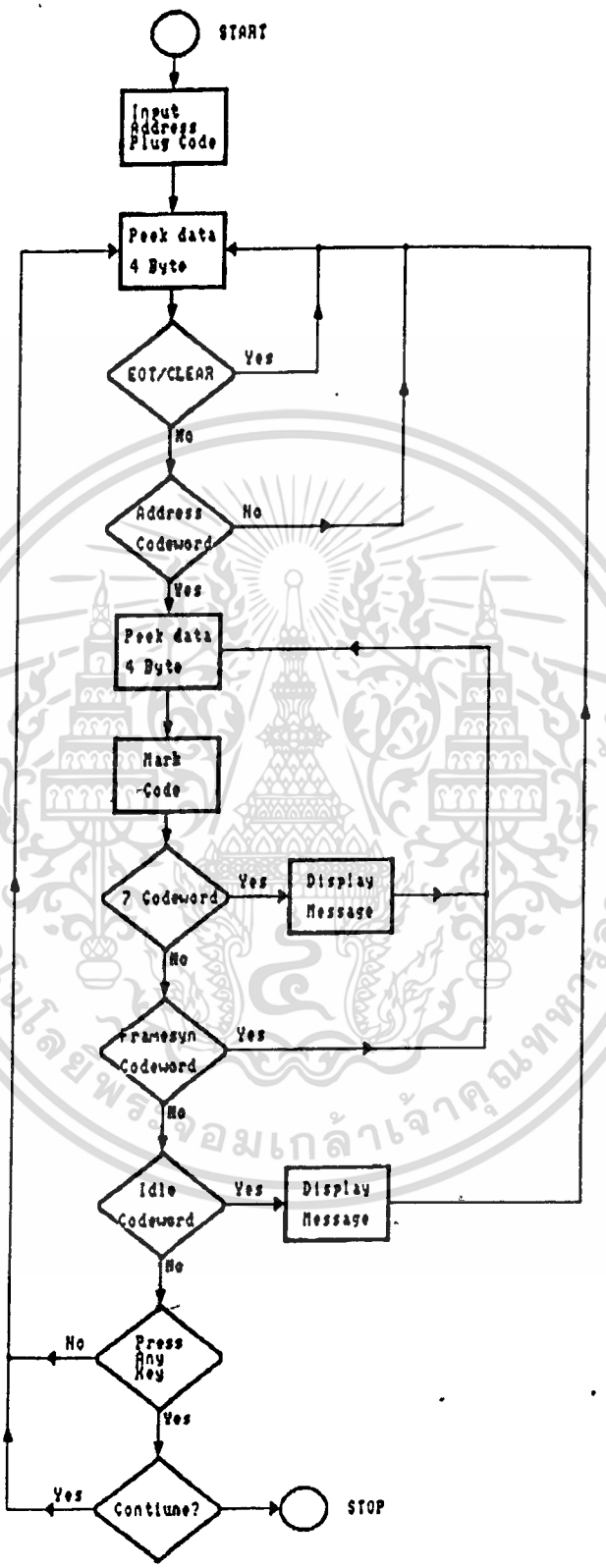
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char d;
switch(codein1)
     {
        case 96:
             d = '"';
             break;
        case 97:
             d = 'a';
             break;
        case 98:
             d = 'b';
             break;
        case 99:
             d = 'c';
             break;
        case 100:
             d = 'd';
             break;
        case 101:
             d = 'e';
             break;
        case 102:
             d = 'f';
             break;
        case 103:
             d = 'g';
             break;
        case 104:
             d = 'h';
             break;
        case 105:
             d = 'i';
             break;
        case 106:
             d = 'j';
             break;
        case 107:
             d = 'k';
             break;
        case 108:
             d = 'l';
             break;
        case 109:
             d = 'm';
             break;
        case 110:
             d = 'n';
             break;
        case 111:
             d = 'o';
             break;
     }
return d;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char changecode5(short int codein1)
{
 char d;
 switch(codein1)
        {
         case 112:
              d = 'p';
              break;
         case 113:
              d = 'q';
              break;
         case 114:
              d = 'r';
              break;
         case 115:
              d = 's';
              break;
         case 116:
              d = 't';
              break;
         case 117:
              d = 'u';
              break;
         case 118:
              d = 'v';
              break;
         case 119:
              d = 'w';
              break;
         case 120:
              d = 'x';
              break;
         case 121:
              d = 'y';
              break;
         case 122:
              d = 'z';
              break;
         case 123:
              d = '{';
              break;
         case 124:
              d = '¦';
              break;
         case 125:
              d = '}';
              break;
         case 126:
              d = '~';
              break;
         case 127:
              d = '#';
              break;
        }
 return d;
}
```

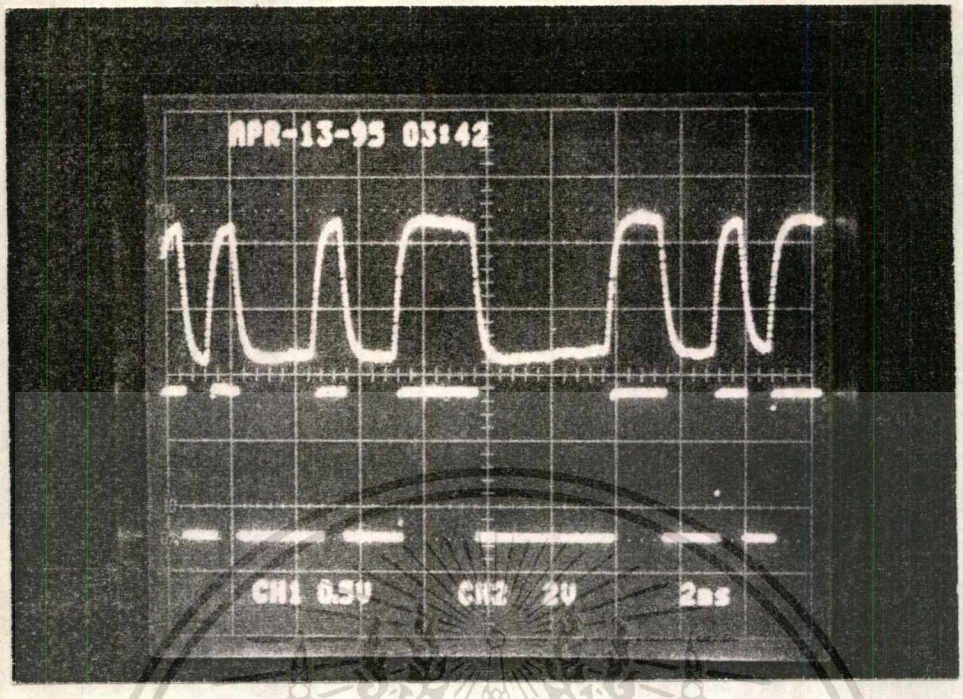
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



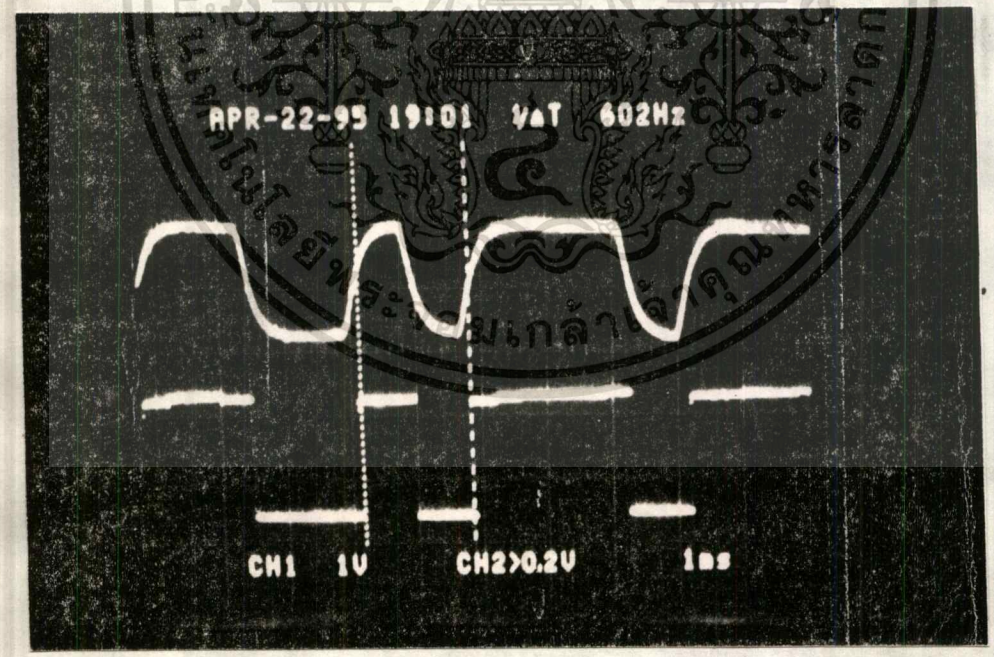
FLOW CHART PROGRAM DISPLAY MESSAGE PAGER IN MODE ALPHA-NUMERIC

รูปที่ 4.4 โฟลว์ชาร์ตโปรแกรม Pager1.c สำหรับแสดงผลเฉพาะข่าวสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 สัญญาณ NRZ ของข้อมูลเทียบกับสัญญาณที่รับได้



รูปที่ 5.3 สัญญาณ NRZ เทียบกับขบวนสัญญาณที่รับได้

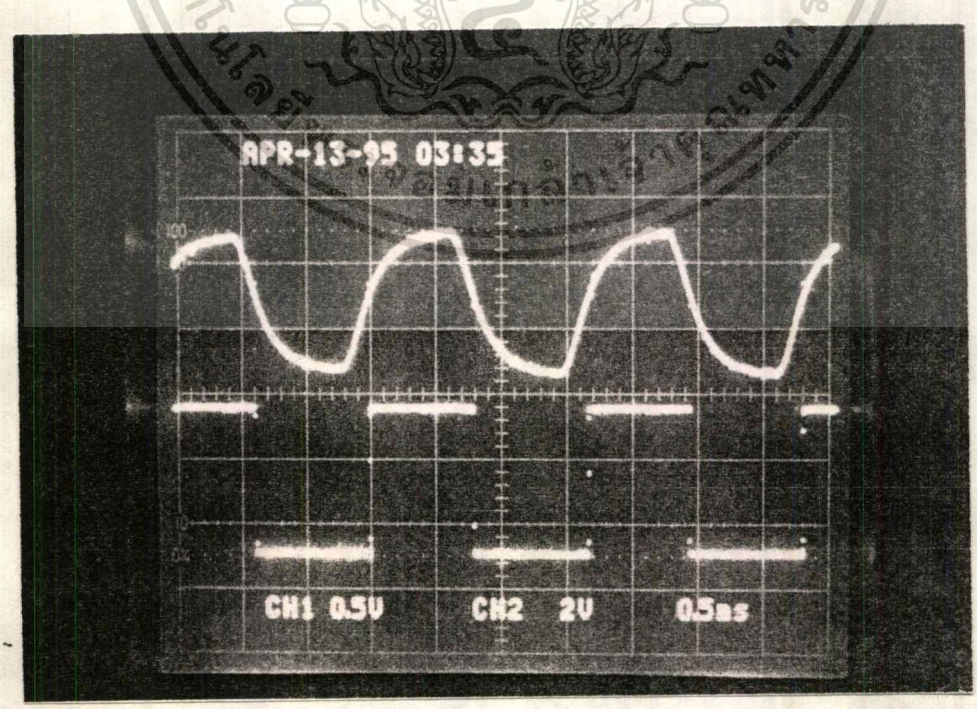
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

ในบทนี้ จะกล่าวถึงผลการทดลองของการรับสัญญาณเพจเจอร์ ทั้งจากภายนอกที่มีการส่งจากบริษัทที่เปิดให้บริการในปัจจุบัน และ จากเครื่องทดสอบคลื่นวิทยุ (RF Communication Test Set) โดยได้วัดรูปสัญญาณเอาท์พุทจากภาคต่างๆ ของโครงงานที่สร้างเช่น ภาคลิมิตเตอร์ ภาคกู้สัญญาณนาฬิกา ภาคตรวจจับเฟรมซิงค์ และภาคกำเนิดสัญญาณเยียนแรมเป็นต้น รูปสัญญาณที่ได้จะถูกนำมาเปรียบเทียบ เพื่อทำการวิเคราะห์ และแก้ไขวงจรจนได้รูปสัญญาณที่เป็นไปตามทฤษฎี ตลอดจนได้ทดสอบโปรแกรมแสดงผลข้อมูลที่หน้าจอคอมพิวเตอร์ที่ความถี่ต่างๆ กันคือ 279.55 MHz 279.8 MHz และ 279.748 MHz โดยทดลองทั้ง 2 รุ่นคือ เพจเจอร์รุ่นตัวเลข และเพจเจอร์รุ่นตัวอักษร โดย มีผลการทดลองดังนี้

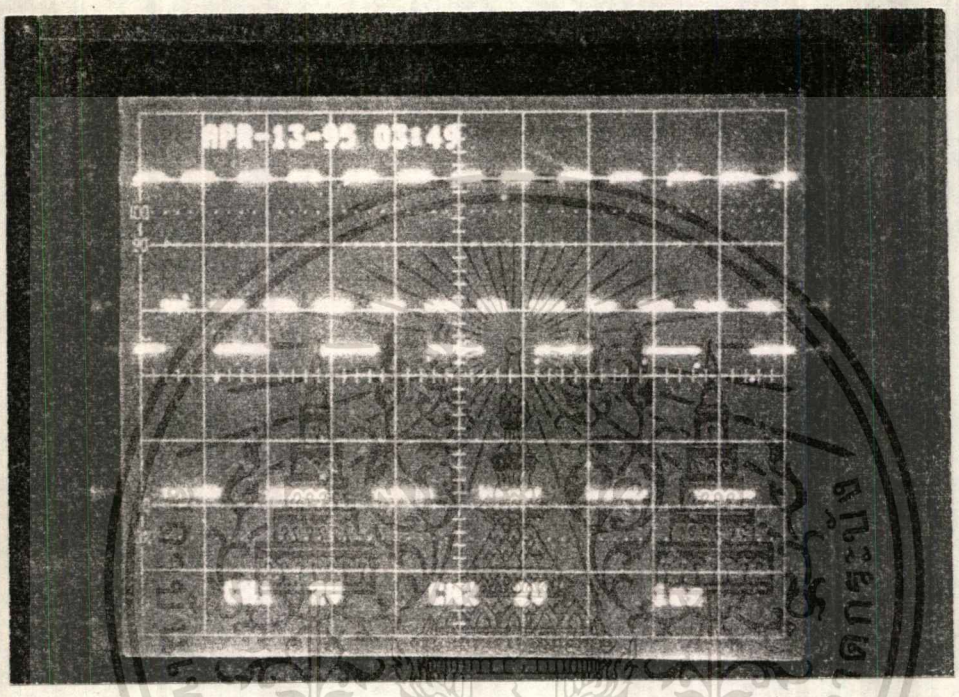
ผลการทดลองภาคลิมิเตอร์(Limitter) สามารถสร้างสัญญาณ NRZ ที่ถูกต้องได้โดยมี duty clycle 50 % ที่สัญญาณอินพุทเป็นพัลส์ส่วนหน้า(Preamble) โดยแสดงไว้ดังรูปที่ 5.1 , 5.2 และ 5.3



รูปที่ 5.1 สัญญาณ NRZ ของพัลส์ส่วนหน้า (Preamble Pulse)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองภาคกู้สัญญาณนาฬิกา (Clock Recovery) สามารถสร้างสัญญาณนาฬิกาที่ความถี่บิทเรท 1200 บิทต่อวินาที ซึ่งเป็นบิทเรทที่เครื่องรับเพจเจอร์ใช้กันอยู่ในปัจจุบัน แสดงดังรูปที่ 5.4

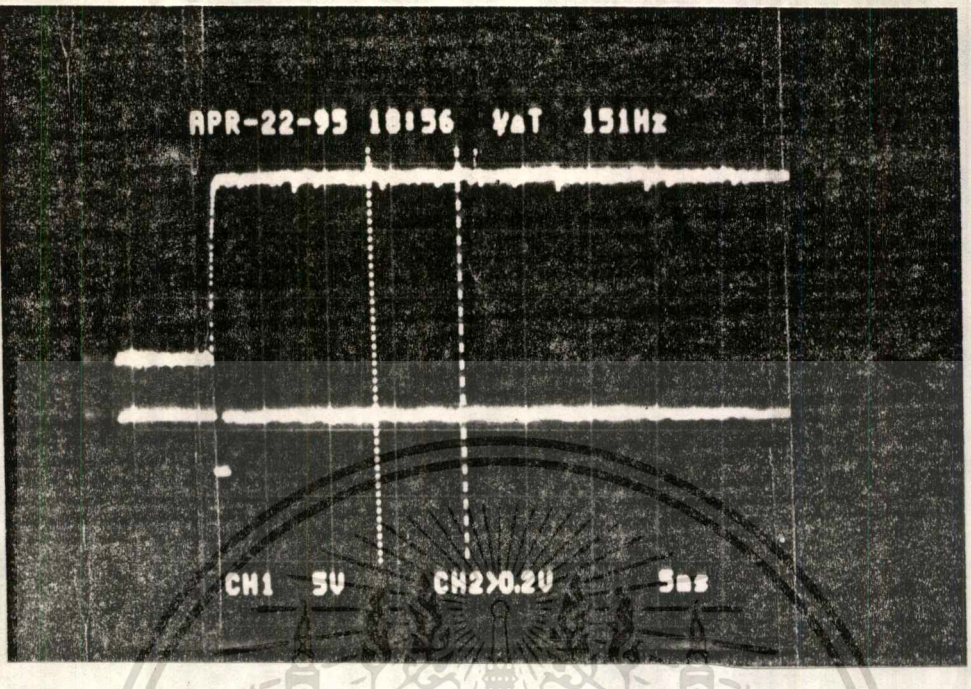


รูปที่ 5.4 สัญญาณนาฬิกาที่สร้างขึ้นเทียบกับสัญญาณ NRZ

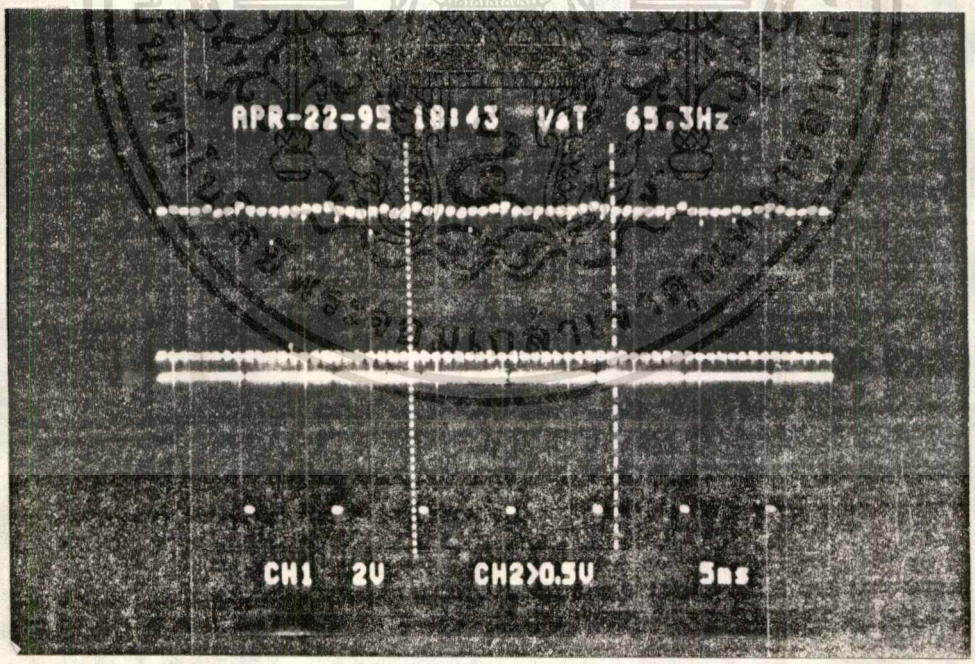
ผลการทดลองภาคตรวจจับเฟรมซิงค์ (Frame Sync Detector) สามารถถอดรหัสสัญญาณเฟรมซิงค์ได้โดยจะกำเนิดพัลส์แคบๆ หลังจากที่ตรวจจับได้ แสดงดังรูปที่ 5.5

ผลการทดลองภาคกำเนิดสัญญาณเขียนแรม (Read/Write Clock Generator) จะเกิดสัญญาณเขียนแรมหลังจากที่ตรวจจับเฟรมซิงค์ได้โดยจะเป็นกำเนิดพัลส์การเขียนแรม ทุกๆ 8 บิท ซึ่งแสดงในรูปที่ 5.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



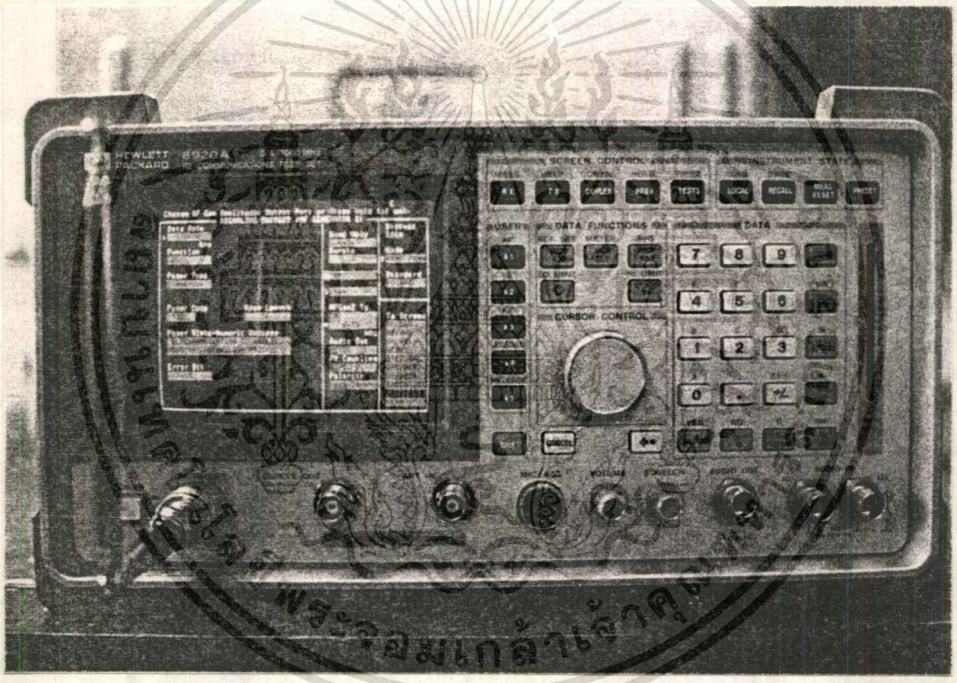
รูปที่ 5.5 สัญญาณเฟรมซิงค์เทียบกับชุดบานสัญญาณที่รับได้



รูปที่ 5.6 สัญญาณการเข้ารหัสแรมทุก 8 บิทเทียบกับสัญญาณนาฬิกา

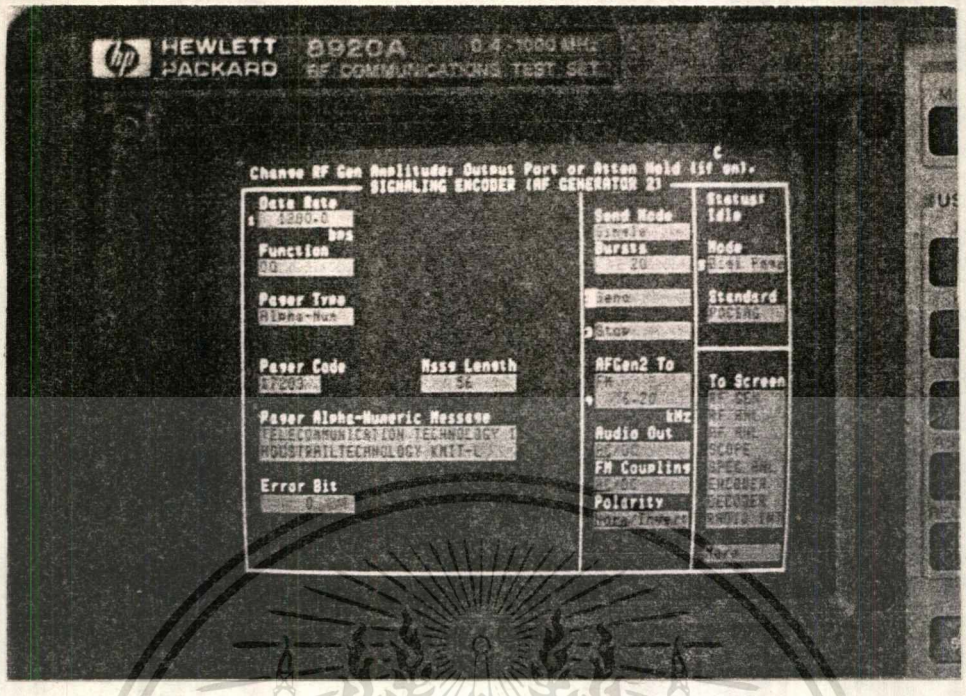
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองนั้นได้ทดลองรับสัญญาณภายนอก และทดลองส่งเองโดยใช้เครื่องทดสอบความถี่วิทยุสื่อสาร (RF Communication Test Set) โดยจะตั้งความถี่ 279.748 MHz และโหมดการ ทำงานเป็น Digital Page ที่ความแรงของสัญญาณ 15-18 dB โดยใช้เสาสไลด์แบบตั้ง รายละเอียดของการกำหนดค่าจะแสดงไว้ดังรูปที่ 5.7 และ 5.8 สำหรับสเปกตรัมของการส่งที่มอดูเลทแบบเลื่อนความถี่ (FSK Moodulating) และภาครับที่ใช้จะใช้ Field Strength Meter เป็นเครื่องรับ ซึ่งมีข้อดีตรงสามารถเลือกความถี่ที่ต้องการรับโดยตั้งความถี่ 279.748 MHz ซึ่งตรงกับภาคส่ง และตั้งแบนด์วิทไว้ที่ 15 KHz โดยมอดูเลทแบบ FM โดยใช้เสาสไลด์ ดังแสดงไว้ดังรูปที่ 5.9

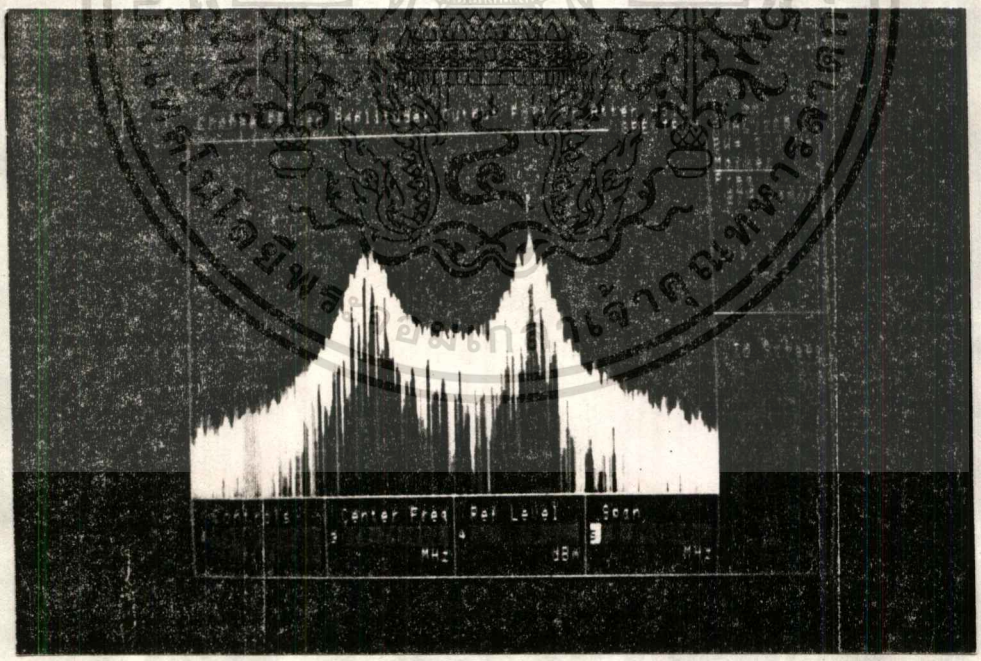


รูปที่ 5.7 เครื่องทดสอบความถี่วิทยุสื่อสาร (RF Communication Test Set)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

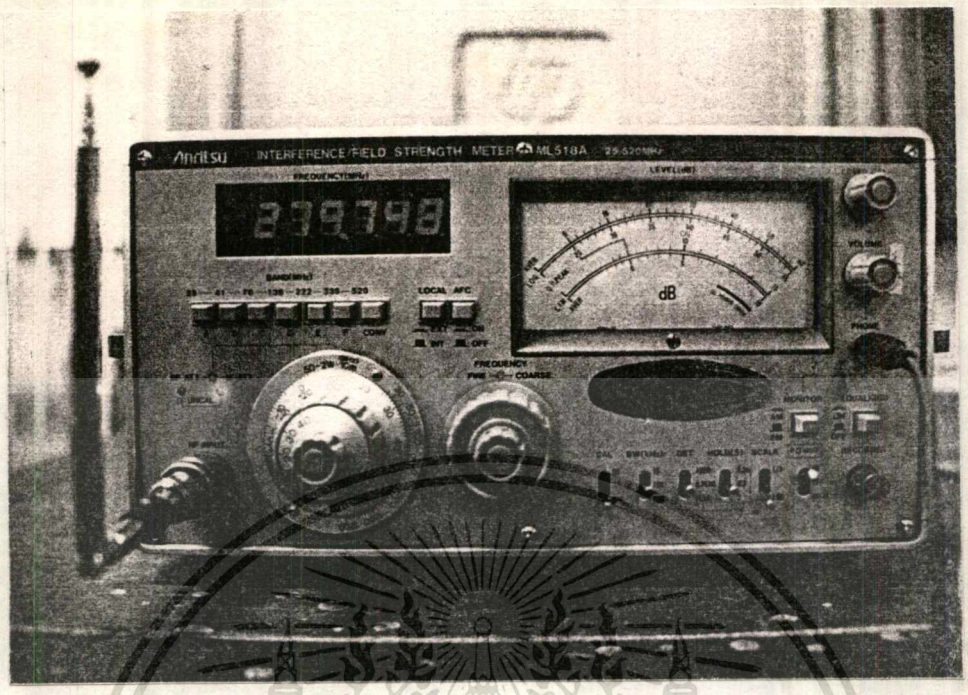


รูปที่ 5.8 การกำหนดค่าต่างๆ บนเครื่องทดสอบความถี่วิทยุสื่อสาร



รูปที่ 5.9 สเปกตรัมของ FSK Modulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

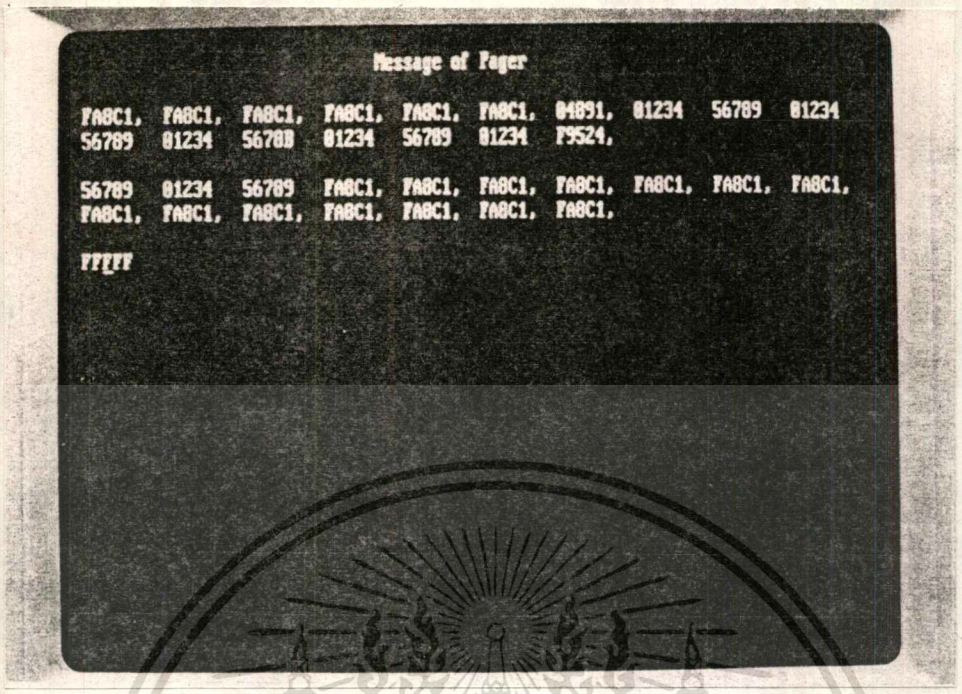


รูปที่ 5.10 เครื่องรับ(Field Strength Meter)

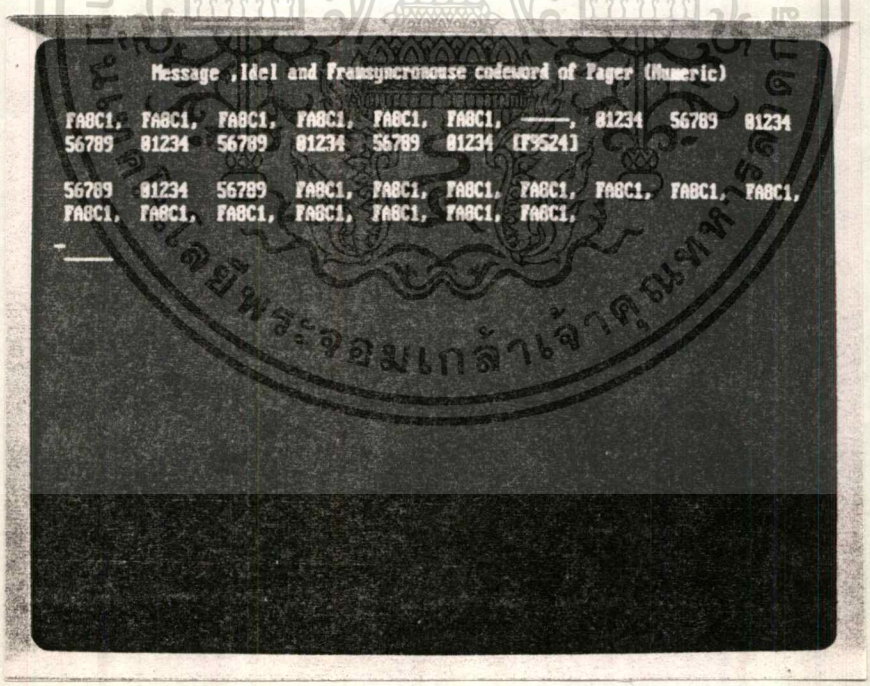
การทดสอบการทำงานของโปรแกรมทั้ง 2 รุ่น ได้ทำการทดลองส่งหมายเลขโทรศัพท์ และข้อความต่างๆ โดยใช้หมายเลขเครื่องที่ทดสอบคือ 17203 ซึ่งจะตรงกับเพจเจอร์รุ่น Wordpage ของบริษัท TAC เป็นรุ่นตัวเลข และความถี่ที่ใช้คือ 279.55 MHz และยังใช้หมายเลขเดียวกันนี้ทดสอบเพจเจอร์รุ่นตัวอักษร-ตัวเลข โดยเปลี่ยนความถี่มาเป็น 279.748 MHz เพื่อไม่ให้เกิดการรบกวนกันของความถี่ เนื่องมาจากความถี่ 279.8 MHz ที่เคยใช้ทดสอบของรุ่นตัวอักษร-ตัวเลขของบริษัท Packlink นั้นได้ถูกยกเลิกไป ผลการทดลองแสดงในรูปที่ 5.11 ,5.12 ,5.13 สำหรับเพจเจอร์รุ่นตัวเลข และ 5.14 ,5.15 สำหรับเพจเจอร์รุ่นตัวอักษร-ตัวเลข

สำหรับโครงงานนี้ได้ทำตัวต้นแบบมา 2 ชุด คือชุดที่ใช้ทดลอง และชุดที่ออกแบบบนการ์ดคอมพิวเตอร์โดยแสดงไว้ดังรูปที่ 5.16 และ5.17 ตามลำดับ สำหรับรูปการต่อใช้งาน และโครงงานทั้งหมดจะแสดงไว้ดังรูปที่ 5.18 และ 5.19 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

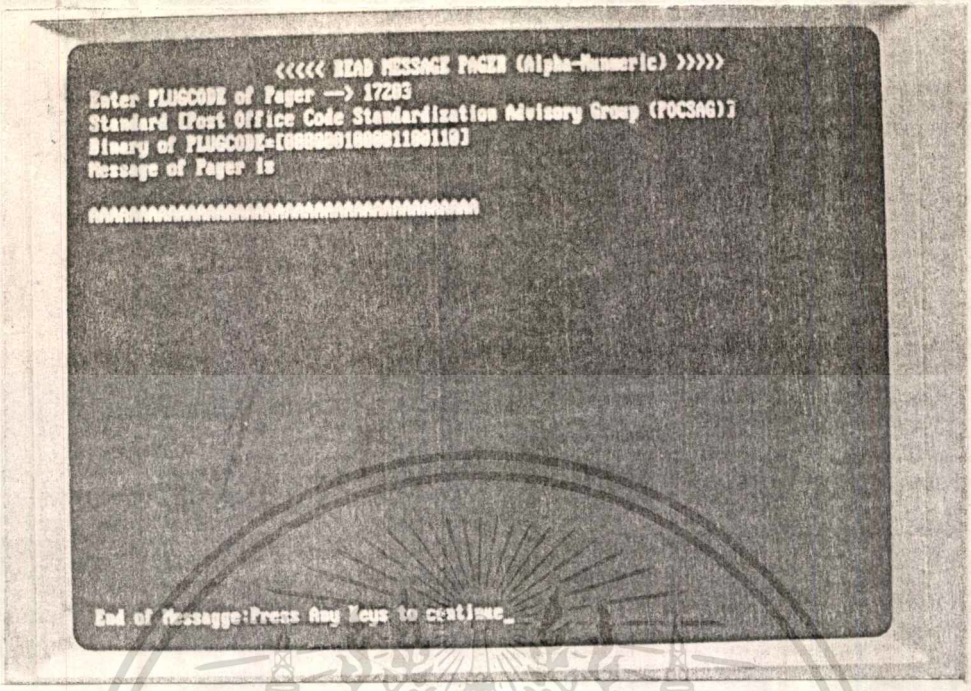


รูปที่ 5.11 แสดงข้อมูลเพจเจอร์รุ่นตัวเลข

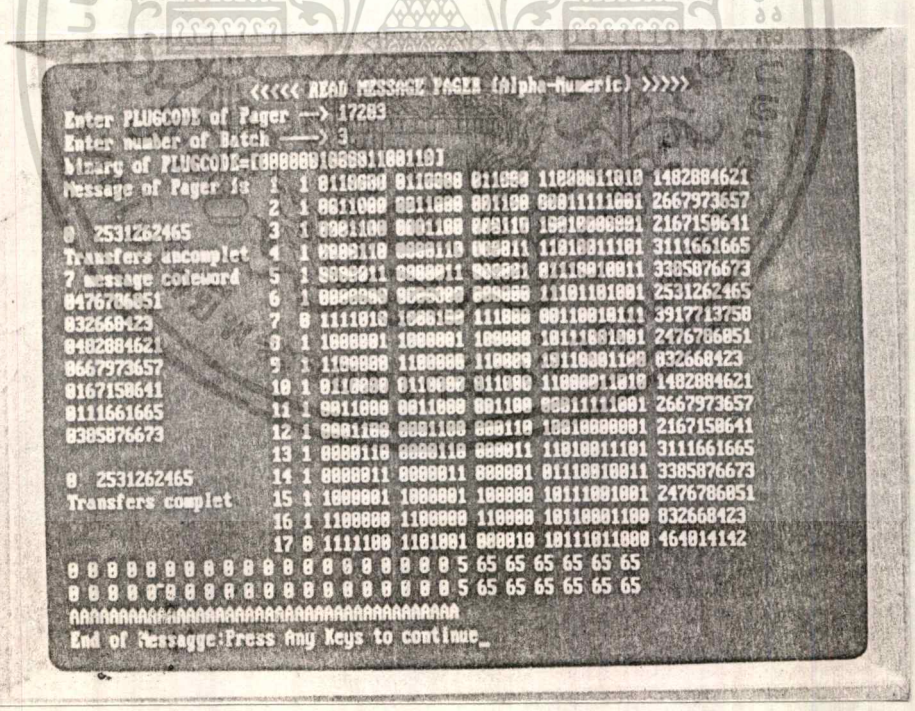


รูปที่ 5.12 แสดงข้อมูลบางส่วนของเจอร์รุ่นตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

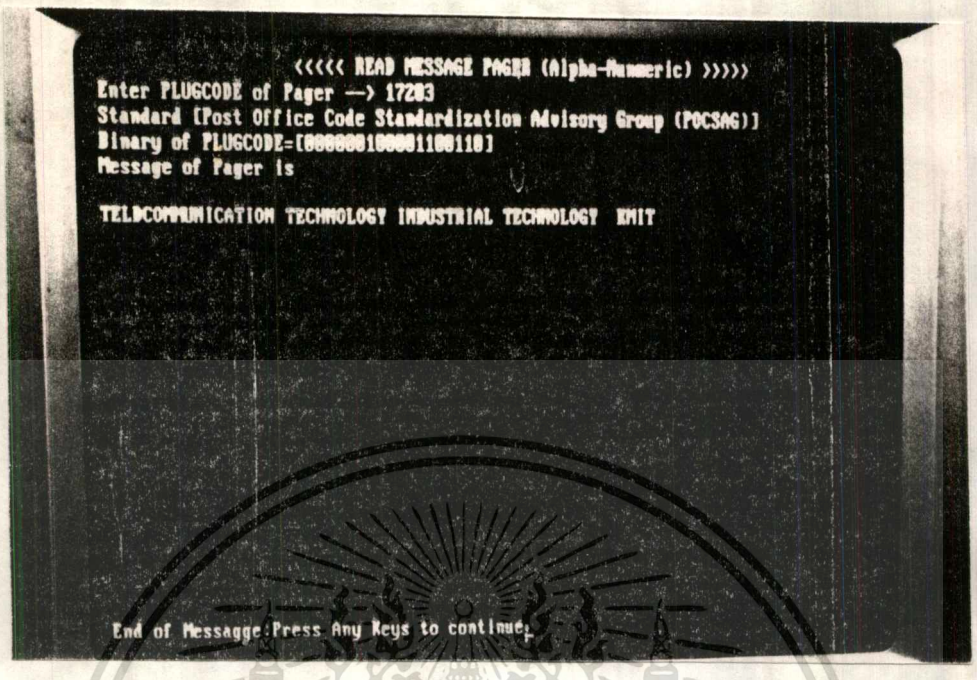


รูปที่ 5.13 แสดงข่าวสารเพจเจอร์รุ่นตัวเลข



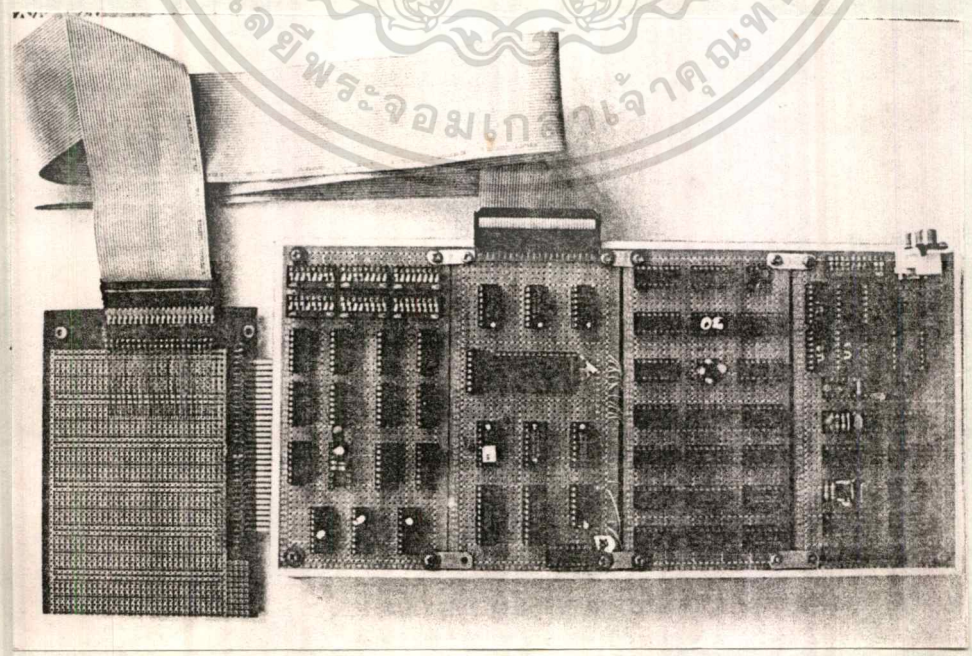
รูปที่ 5.14 แสดงข้อมูลเพจเจอร์รุ่นตัวอักษร-ตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



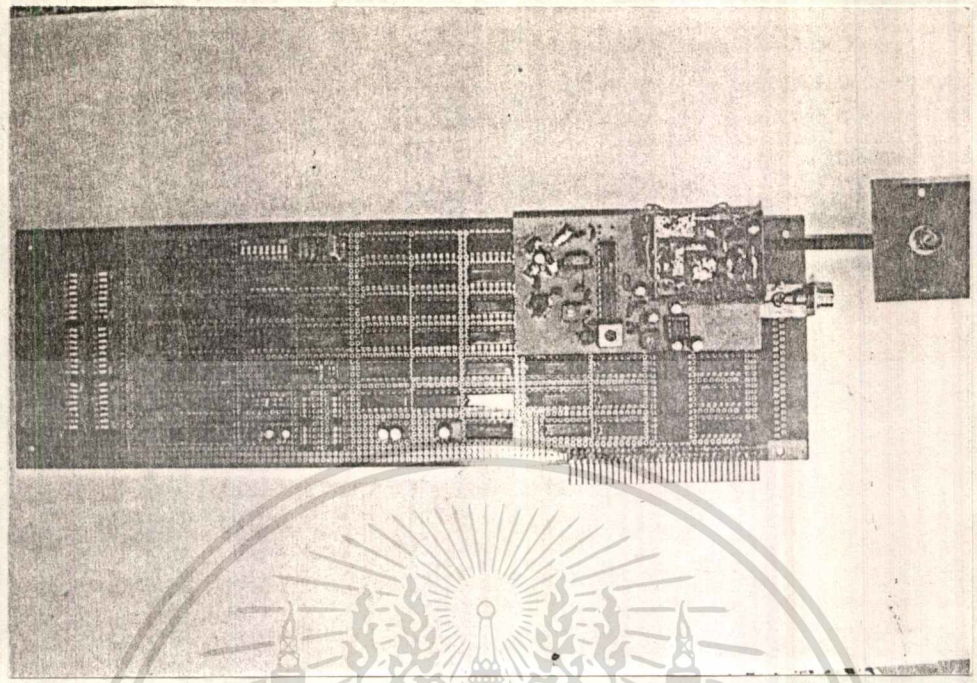
รูปที่ 5.15 แสดงข่าวสารเพจเจอร์รุ่นตัวอักษร-ตัวเลข

สำหรับโครงงานนี้ได้ทำชุดต้นแบบไว้ 2 ชุด คือชุดทดลอง และชุดที่ออกแบบบนการ์ดคอมพิวเตอร์ซึ่งแสดงไว้ในรูปที่ 5.16 และ5.17 ตามลำดับ ส่วนรูปที่ 5.18 และ 5.19 เป็นรูปที่แสดงการต่อใช้งานจริงของโครงงานนี้นั้นเอง

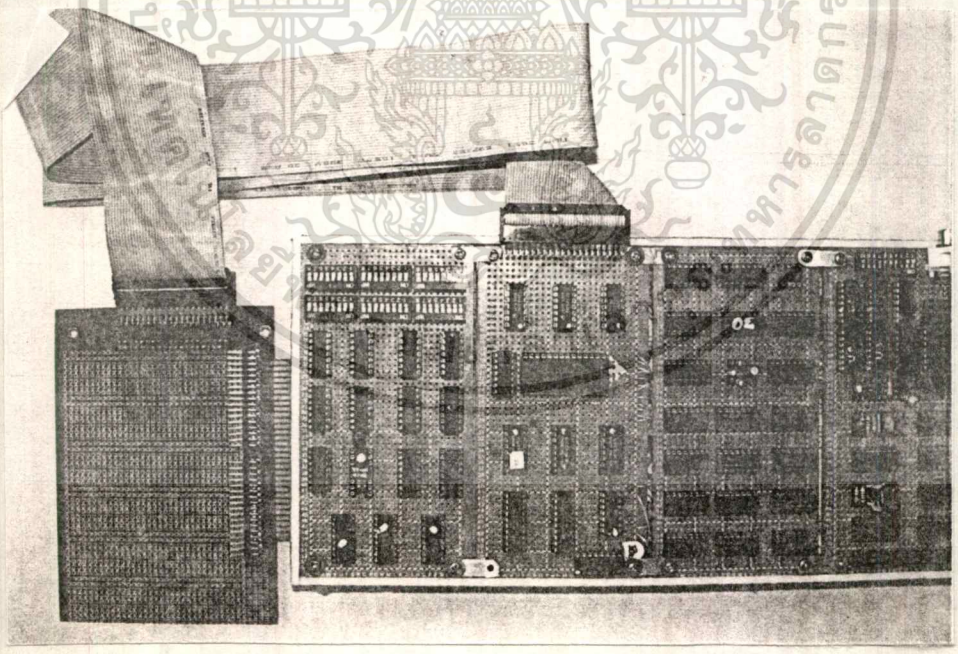


รูปที่ 5.16 โครงงานชุดทดลองต้นแบบเมื่อประกอบเสร็จเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.17 การ์ดเพจเจอร์ต้นแบบ



รูปที่ 5.18 ลักษณะการต่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.19 โครงงานเพจเจอร์ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Hewlet Packard."HP 8920 RF Communicaion Test Set User's Guide",1991

2. British Telecom. "A Standard Code for Radio Paging Report of Post Office Code Standard Advisory Group(POCSAG)" June 1978.

3. IBM, "Personal Computer AT Technicaiil Reference" ,1985.

4. Anrisu, "Interference Strength Meter MI518A" ,1990.

5. ฮาร์ดแวร์ไมโครคอมพิวเตอร์ 8088,80286,80386

6. คู่มือไอซี ทีทีแอล

7. คู่มือไอซี ซีมอส

8. คู่มือลิเนียร์ไอซี โมโตโรลา

9. ปริญญานิพนธ์เรื่อง เครื่องรับวิทยุติดตามตัว ภาคเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

10. หลักสื่อการเขียนภาษาซีสำหรับวิศวกรรม ของ ธันวา ศรีประโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCM6264

8K × 8 Bit Fast Static RAM

The MCM6264 is a 65,536 bit static random access memory organized as 8192 words of 8 bits, fabricated using Motorola's high-performance silicon-gate CMOS technology. Static design eliminates the need for external clocks or timing strobes, while CMOS circuitry reduces power consumption which provides greater reliability.

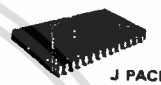
The chip enable pins ($\overline{E1}$ and E2) are not clocks. Either pin, when asserted false, causes the part to enter a low power standby mode. The part will remain in standby mode until both pins are asserted true again. The availability of active high and active low chip enable pins provides more system design flexibility than single chip enable devices.

The MCM6264 is available in 300 and 600 mil, 28 pin plastic dual-in-line packages and 300 and 400 mil, 28 pin plastic SOJ packages. All packages feature the JEDEC standard pinout.

- Single 5 V Supply, ±10%
- 8K × 8 Organization
- Fully Static—No Clock or Timing Strobes Necessary
- Fast Access Time—30, 35, 45, 55 ns (Maximum)
- Low Power Operation—105, 100, 90, 80 mA (Maximum, Active)
- Three State Outputs
- All Inputs and Outputs are TTL Compatible
- Output Enable ($\overline{G}$) Feature for Increased System Flexibility and to Eliminate Bus Contention Problems

P PACKAGE
300 MIL PLASTIC
CASE 710A

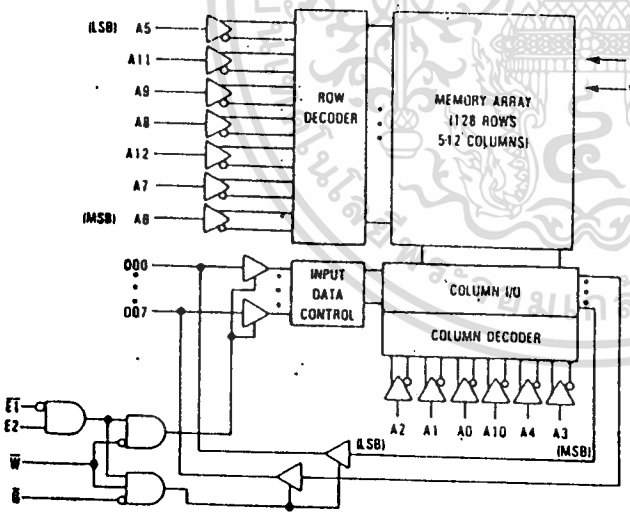
WP PACKAGE
600 MIL PLASTIC
CASE TBD



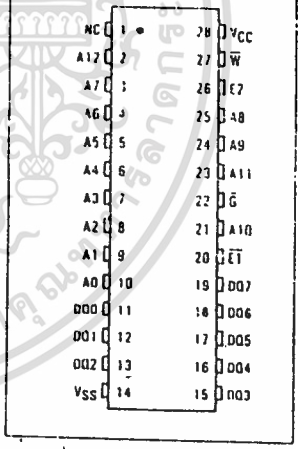
J PACKAGE
400 MIL SOJ
CASE 810

NJ PACKAGE
300 MIL SOJ
CASE 810B

BLOCK DIAGRAM



PIN ASSIGNMENT



PIN NAMES	
A0-A12	Address
$\overline{W}$	Write Enable
$\overline{E1}$, E2	Chip Enable
$\overline{G}$	Output Enable
DQ0-DQ7	Data Input/Output
V_{CC}	+5 V Power Supply
V_{SS}	Ground
NC	No Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRUTH TABLE

$\overline{E1}$	E2	$\overline{G}$	$\overline{W}$	Mode	Supply Current	I/O Pin
H	X	X	X	Not Selected	I_{SB}	High Z
X	L	X	X	Not Selected	I_{SB}	High Z
L	H	H	H	Output Disabled	I_{CC}	High Z
L	H	L	H	Read	I_{CC}	D_{out}
L	H	X	L	Write	I_{CC}	D_{in}

X = don't care

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high-impedance circuit.

ABSOLUTE MAXIMUM RATINGS (See Note)

Rating	Symbol	Value	Unit
Power Supply Voltage	V_{CC}	−0.5 to +7.0	V
Voltage Relative to V_{SS} for Any Pin Except V_{CC}	V_{in}, V_{out}	−0.5 to V_{CC} +0.5	V
Output Current (per I/O)	I_{out}	±20	mA
Power Dissipation (T_A = 25°C)	P_D	1.0	W
Temperature Under Bias	T_{bias}	−10 to +85	°C
Operating Temperature	T_A	0 to +70	°C
Storage Temperature	T_{stg}	−55 to +125	°C

NOTE: Permanent device damage may occur if ABSOLUTE MAXIMUM RATINGS are exceeded. Functional operation should be restricted to RECOMMENDED OPERATING CONDITIONS. Exposure to higher than recommended voltages for extended periods of time could affect device reliability.

DC OPERATING CONDITIONS AND CHARACTERISTICS

(V_{CC} = 5.0 V ±10%, T_A = 0 to 70°C, Unless Otherwise Noted)

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Typ	Max	Unit
Supply Voltage (Operating Voltage Range)	V_{CC}	4.5	5.0	5.5	V
Input High Voltage	V_{IH}	2.2	–	V_{CC} +0.3	V
Input Low Voltage	V_{IL}	−0.3*	–	0.8	V

*V_{IL} (min) = −0.3 V dc; V_{IL} (min) = −3.0 V ac (pulse width ≤20 ns)

DC CHARACTERISTICS

Parameter		Symbol	Min	Max	Unit
Input Leakage Current (All Inputs, V_{in} = 0 to V_{CC})		$I_{lkg(I)}$	–	±1.0	μA
Output Leakage Current ($\overline{E1}$ = V_{IH}, E2 = V_{IL}, or $\overline{G}$ = V_{IH}, V_{out} = 0 to V_{CC})		$I_{lkg(O)}$	–	±1.0	μA
Power Supply Current ($\overline{E1}$ = V_{IL}, E2 = V_{IH}, I_{out} = 0)	(t_{AVAV} = 30 ns)	I_{CC}	–	105	mA
	(t_{AVAV} = 35 ns)		–	100	
	(t_{AVAV} = 45 ns)		–	90	
	(t_{AVAV} = 55 ns)		–	80	
Standby Current ($\overline{E1}$ = V_{IH} or E2 = V_{IL})		I_{SB1}	–	10	mA
Standby Current ($\overline{E1}$ ≥ V_{CC} −0.2 V or E2 ≤0.2 V, V_{in} = V_{IH} or V_{in} = V_{IL})		I_{SB2}	–	5	mA
Output Low Voltage (I_{OL} = 8.0 mA)		V_{OL}	–	0.4	V
Output High Voltage (I_{OH} = −4.0 mA)		V_{OH}	2.4	–	V

CAPACITANCE (f = 1.0 MHz, dV = 3.0 V, T_A = 25°C, Periodically Sampled Rather Than 100% Tested)

Characteristic		Symbol	Max	Unit
Input Capacitance	All Inputs Except DQ	C_{in}	6	pF
I/O Capacitance	DQ	$C_{I/O}$	8	pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC OPERATING CONDITIONS AND CHARACTERISTICS

(V_{CC} = 5.0 V ±10%, T_A = 0 to 70°C, Unless Otherwise Noted)

Input Pulse Levels	0 to 3.0 V
Input Rise/Fall Time	5 ns
Input Timing Measurement Reference Levels	1.5 V
Output Timing Measurement Reference Levels	0.8 and 2.0 V
Output Load	See Figure 1

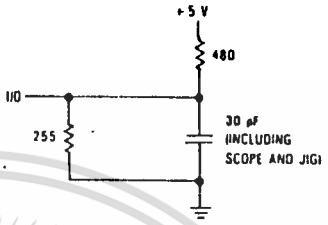
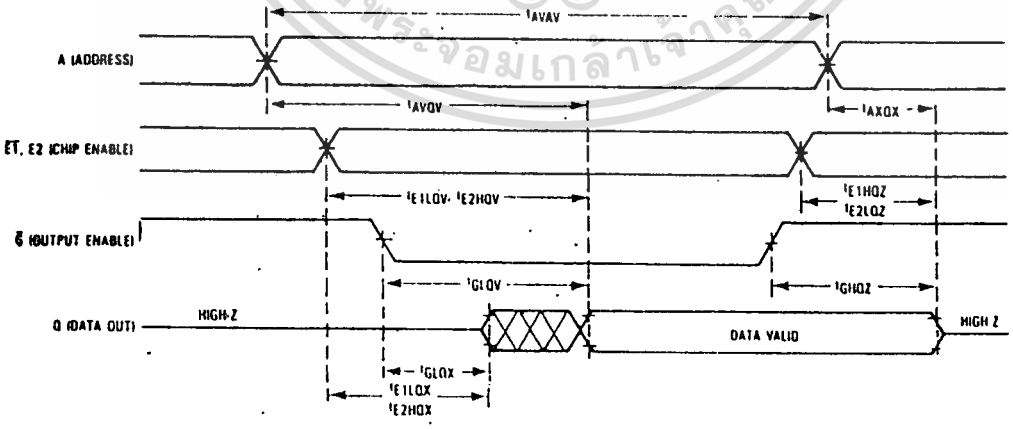


Figure 1. Test Load

READ CYCLE (See Note 1)

Parameter	Symbol	Alt Symbol	MCM6264-30		MCM6264-35		MCM6264-45		MCM6264-55		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
Read Cycle Time	t_{AVAV}	t_{RC}	30	–	35	–	45	–	55	–	ns	–
Address Cycle Time	t_{AVQV}	t_{AA}	–	30	–	35	–	45	–	55	ns	–
$\overline{E1}$ Access Time	t_{E1LQV}	t_{AC1}	–	30	–	35	–	45	–	55	ns	–
E2 Access Time	t_{E2HQV}	t_{AC2}	–	30	–	35	–	45	–	55	ns	–
$\overline{G}$ Access Time	t_{GLQV}	t_{OE}	–	12.5	–	15	–	20	–	25	ns	–
Output Hold from Address Change	t_{AXQX}	t_{OH}	5	–	5	–	5	–	5	–	ns	–
Chip Enable to Output Low-Z	t_{E1LQX}, t_{E2HQX}	t_{CLZ}	5	–	5	–	5	–	5	–	ns	2, 3
Output Enable to Output Low-Z	t_{GLQX}	t_{OLZ}	0	–	0	–	0	–	0	–	ns	2, 3
Chip Enable to Output High-Z	t_{E1HQZ}, t_{E2LQZ}	t_{CHZ}	0	15	0	15	0	15	0	15	ns	2, 3
Output Enable to Output High-Z	t_{GHQZ}	t_{OHZ}	0	15	0	15	0	15	0	15	ns	2, 3

NOTES:
1. $\overline{W}$ is high at all times for read cycles.
2. All high-Z and low-Z parameters are considered in a high or low impedance state when the output has made a 500 mV transition from the previous steady state voltage.
3. These parameters are periodically sampled and not 100% tested.



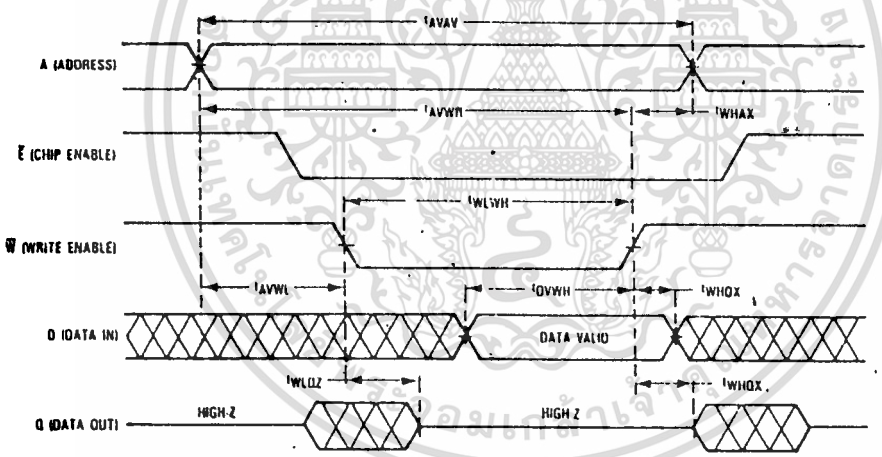
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE CYCLE 1 (W CONTROLLED) (See Note 1)

Parameter	Symbol	Alt Symbol	MCM6264-30		MCM6264-35		MCM6264-45		MCM6264-55		Unit	Notes
			Min	Max	Min	Max	Min	Max	Min	Max		
Write Cycle Time	t_{AVAV}	t_{WC}	30	—	35	—	45	—	55	—	ns	—
Address Setup Time	t_{AVWL}	t_{AS}	0	—	0	—	0	—	0	—	ns	—
Address Valid to End of Write	t_{AVWH}	t_{AW}	22.5	—	25	—	35	—	45	—	ns	—
Write Pulse Width	t_{WLWH}	t_{WP}	17.5	—	20	—	25	—	30	—	ns	3
Data Valid to End of Write	t_{DVWH}	t_{DW}	12.5	—	15	—	20	—	25	—	ns	—
Data Hold Time	t_{WHDX}	t_{DH}	0	—	0	—	0	—	0	—	ns	3
Write Low to Output in High-Z	t_{WLQZ}	t_{WHZ}	0	15	0	15	0	15	0	15	ns	4, 5
Write High to Output Low-Z	t_{WHQX}	t_{OW}	5	—	5	—	5	—	5	—	ns	4, 5
Write Recovery Time	t_{WHAX}	t_{WR}	0	—	0	—	0	—	0	—	ns	—

NOTES:

1. A write cycle starts at the latest transition of a low $\overline{E1}$, low $\overline{W}$, or high E2. A write cycle ends at the earliest transition of a high $\overline{E1}$, high $\overline{W}$, or low E2.
2. If $\overline{W}$ goes low coincident with or prior to $\overline{E1}$ low or E2 high then the outputs will remain in a high impedance state.
3. During this time the output pins may be in the output state. Signals of opposite phase to the outputs must not be applied at this time.
4. All high-Z and low-Z parameters are considered in a high or low impedance state when the output has made a 500 mV transition from the previous steady state voltage.
5. These parameters are periodically sampled and not 100% tested.



TIMING PARAMETER ABBREVIATIONS

t X X X X

- signal name from which interval is defined
- transition direction for first signal
- signal name to which interval is defined
- transition direction for second signal

The transition definitions used in this data sheet are:

- H = transition to high
- L = transition to low
- V = transition to valid
- X = transition to invalid or don't care
- Z = transition to off (high impedance)

TIMING LIMITS

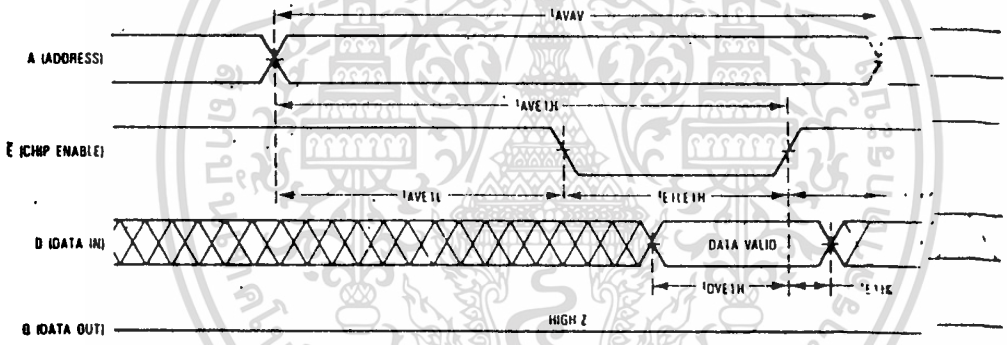
The table of timing values shows either a minimum or a maximum limit for each parameter. Input requirements are specified from the external system point of view. Thus, address setup time is shown as a minimum since the system must supply at least that much time (even though most devices do not require it). On the other hand, responses from the memory are specified from the device point of view. Thus, the access time is shown as a maximum since the device never provides data later than that time.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WRITE CYCLE 2 (ENABLE CONTROLLED) (See Note 1)

Parameter	Symbol	Alt Symbol	MCM6264-30		MCM6264-35		MCM6264-45		MCM6264-55		Notes
			Min	Max	Min	Max	Min	Max	Min	Max	
Write Cycle Time	t_{AVAV}	t_{WC}	30	—	35	—	45	—	55	—	
Address Setup Time	t_{AVE1L}	t_{AS}	0	—	0	—	0	—	0	—	
Address Valid to End of Write	t_{AVE1H}	t_{AW}	22.5	—	25	—	35	—	45	—	
Chip Enable to End of Write	t_{E1LE1H}	t_{CW}	22.5	—	25	—	35	—	45	—	
Data Valid to End of Write	t_{DVE1H}	t_{DW}	12.5	—	15	—	20	—	25	—	
Data Hold Time	t_{E1HDX}	t_{DH}	0	—	0	—	0	—	0	—	
Write Recovery Time	t_{E1HAX}	t_{WR}	0	—	0	—	0	—	0	—	

NOTES:
1. A write cycle starts at the latest transition of a low $\overline{E1}$, low $\overline{W}$, or high E2. A write cycle ends at the earliest transition of a high $\overline{E1}$, high $\overline{W}$, or low E2.
2. $\overline{E1}$ and E2 timings are identical when E2 signals are inverted.
3. If $\overline{W}$ goes low coincident with or prior to $\overline{E1}$ low or E2 high then the outputs will remain in a high impedance state.
4. During this time the output pins may be in the output state. Signals of opposite phase to the outputs must not be applied.



ORDERING INFORMATION
(Order by Full Part Number)

MCM 6264 X XX XX

- MCM — Motorola Memory Prefix
- 6264 — Part Number
- XX — Speed (30 = 30 ns, 35 = 35 ns, 45 = 45 ns, 55 = 55 ns)
- XX — Shipping Method (R2 = Tape & Reel)
- X — Package (P = 300 mil Plastic DIP, WP = 600 mil Plastic DIP, J = 400 mil Plastic SOJ, NJ = 300 mil Plastic SOJ)

Full Part Numbers—

MCM6264P30	MCM6264WP30	MCM6264J30	MCM6264J30R2	MCM6264NJ30	MCM6264NJ30R2
MCM6264P35	MCM6264WP35	MCM6264J35	MCM6264J35R2	MCM6264NJ35	MCM6264NJ35R2
MCM6264P45	MCM6264WP45	MCM6264J45	MCM6264J45R2	MCM6264NJ45	MCM6264NJ45R2
MCM6264P55	MCM6264WP55	MCM6264J55	MCM6264J55R2	MCM6264NJ55	MCM6264NJ55R2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS240
SN54/74LS241
SN54/74LS244

DESCRIPTION — The SN54LS/74LS240, 241 and 244 are Octal Buffers and Line Drivers designed to be employed as memory address drivers, clock drivers and bus-oriented transmitters/receivers which provide improved PC board density.

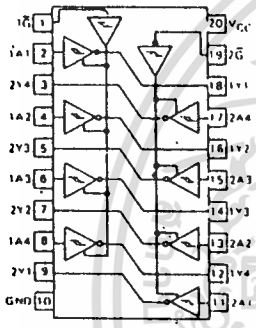
- HYSTERESIS AT INPUTS TO IMPROVE NOISE MARGINS
- 3-STATE OUTPUTS DRIVE BUS LINES OR BUFFER MEMORY ADDRESS REGISTERS
- INPUT CLAMP DIODES LIMIT HIGH-SPEED TERMINATION EFFECTS

OCTAL BUFFER/LINE DRIVER WITH 3-STATE OUTPUTS

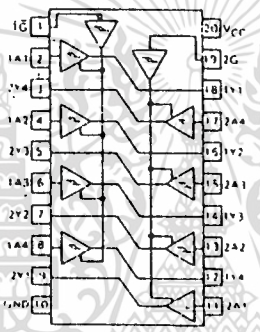
LOW POWER SCHOTTKY

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

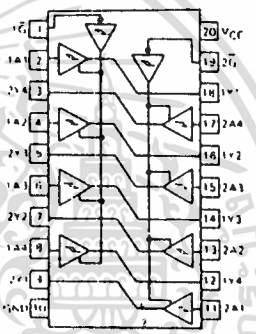
SN54LS/74LS240



SN54LS/74LS241



SN54LS/74LS244



TRUTH TABLES

SN54LS/74LS240

INPUTS		OUTPUT
1G,2G	D	
L	L	H
L	H	L
H	X	(Z)

SN54LS/74LS244

INPUTS		OUTPUT
1G,2G	D	
L	L	L
L	H	H
H	X	(Z)

SN54LS/74LS241

INPUTS		OUTPUT	INPUTS		OUTPUT
1G	D		2G	D	
L	L	L	H	L	L
L	H	H	H	H	H
H	X	(Z)	L	X	(Z)

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial
Z = HIGH Impedance

J Suffix — Case 732-03 (Ceramic)
N Suffix — Case 738-03 (Plastic)

5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54,74			−3.0	mA
		54 74			−12 −15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74			0.8		
$V_{T+} - V_{T-}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA
V_{OH}	Output HIGH Voltage	54,74	2.4	3.4		V	V_{CC} = MIN, I_{OH} = −3.0 mA
		54,74	2.0			V	V_{CC} = MIN, I_{OH} = MAX
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 12 mA; V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA; V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V
I_{OZL}	Output Off Current LOW				20	μA	V_{CC} = MAX, V_{OUT} = 0.4 V
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V
I_{IL}	Input LOW Current				−0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Output Short Circuit Current		−40		−225	mA	V_{CC} = MAX
I_{CC}	Power Supply Current Total, Output HIGH				27	mA	V_{CC} = MAX
	Total, Output LOW	LS240			44		
		LS241/244			46		
	Total at HIGH Z	LS240			50		
		LS241/244			54		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC CHARACTERISTICS: $T_A = 25°C$, $V_{CC} = 5.0$ V

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS240		9.0 12	14 18	ns	$C_L = 45$ pF, $R_L = 667\ \Omega$
t_{PLH} t_{PHL}	Propagation Delay, Data to Output LS241, '244		12 12	18 18	ns	
t_{PZH}	Output Enable Time to High Level					
	Output Disable Time from Low Level					
t_{PHZ}	Output Disable Time from High Level		10	18	ns	$C_L = 5.0$ pF, $R_L = 667\ \Omega$

AC WAVEFORMS

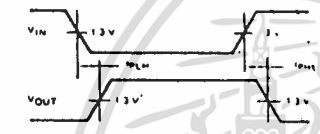


Fig. 1

Fig. 2

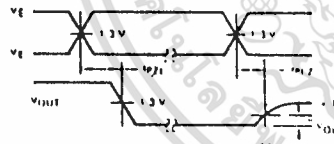


Fig. 3

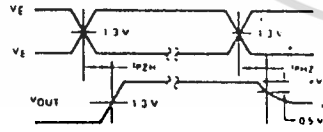


Fig. 4

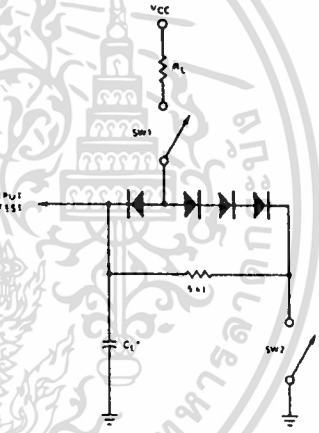


Fig. 5

SWITCH POSITIONS

SYMBOL	SW1	SW2
t_{PZH}	Open	Closed
t_{PZL}	Closed	Open
t_{PLZ}	Closed	Closed
t_{PHZ}	Closed	Closed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54123/74123 Dual Retriggerable Monostable Multivibrator with Clear

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF
T.I.											SN54LS123		J①		W①	SN54123	J①			W①	SN54L123	J①			
											SN74LS123		J	N		SN74123	J①	N①			SN74L123	J①	N①		
FAIRCHILD																FJ54123/FJ54L123	J①								
																FC74123/FC54L123	J①	P①							
MOTOROLA																					DM54L123A		①		
																MC74123		P①			DM74L123A		①		
N.S.C.											DM54LS123		①												
											DM74LS123		①			DM74123		N①							
PHILIPS																									
											N74LS123		J			N74123		①							
SIGNETICS																S54123	F①	B①		W①					
																N74123	F①	B①							
SIEMENS																									
																FLK121		①							
FUJITSU											74LS123			MJ①		MB440	①	MJ①							
HITACHI											HD74LS123			PJ		HD74123/HD02516	①	P①							
MITSUBISHI											M74LS123			PJ		M53323		P①							
NEC																μPB2123	D①								
TOSHIBA																									

Electrical Characteristics SN54LS123 SN74LS123

absolute maximum ratings operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		65°C to 150°C

recommended operating conditions

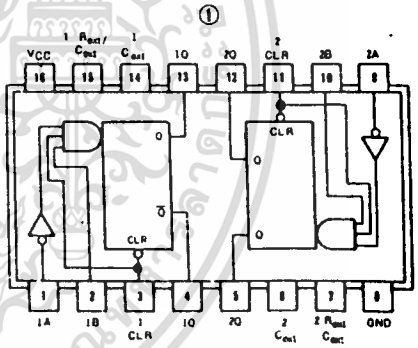
	SN54LS123 MIN	NOM	MAX	SN74LS123 MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-400			-400	μA
Low-level output current, I_{OL}			4			8	mA
Pulse width, t_w	40			40			ns
External timing resistance R_{ext}	5		180	5		260	kΩ
External timing capacitance C_{ext}	No restriction			No restriction			
Timing capacitance at R_{ext} C_{ext} terminal			10	0		10	pF
Operating free-air temperature T_A	55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = -18mA				-1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, I_{OH} = MAX, See Note 1		2.7	3.5		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, I_{OL} = 8mA, See Note 1			0.35	0.5	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V				-0.4	mA
I_{OS}	Short-circuit output current ◆	V_{CC} = MAX, See Note 1		-20		-100	mA
I_{CC}	Supply current (quiescent or triggered)	V_{CC} = MAX, See Notes 2 and 3			12	20	mA
t_{PLH}	from A to output Q	V_{CC} = 5V, T_A = 25°C, C_L = 15pF, R_L = 2kΩ	C_{ext} 0, R_{ext} 5kΩ		22	33	ns
t_{PLH}	from B to output Q				23	44	
t_{PHL}	from A to output Q				32	45	ns
t_{PHL}	from B to output Q				34	56	
t_{PLH}	from Clear to output Q				20	27	ns
t_{PHL}	from Clear to output Q				28	45	
t_w(min)	from A or B to output Q				116	200	ns
t_w	from A or B to output Q		C_{ext} 1000pF, R_{ext} 10kΩ	4	4.5	5	μs

† For conditions shown as MIN or MAX, use the value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5V, T_A = 25°C.
◆ Not more than one output should be shorted at a time.
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output

Pin Assignment (Top View)



FUNCTION TABLE

123 LS123, L123 (See Note 4)

INPUTS CLEAR	A	B	OUTPUTS Q	Q
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⎍	⎎
H	↓	H	⎍	⎎
↑	L	H	⎍	⎎

NOTES: 1. Ground C_{ext} to measure V_{OH} at Q, V_{OL} at Q or I_{OS} at Q. C_{ext} is open to measure V_{OH} at Q, V_{OL} at Q, or I_{OS} at Q.
2. Quiescent I_{CC} is measured (after clearing) with 2.4V applied to all clear and A inputs, B inputs grounded, all outputs open, C_{ext} = 0.02 μF, and R_{ext} = 25kΩ.
3. I_{CC} is measured in the triggered state with 2.4V applied to all clear and B inputs, A inputs grounded, all outputs open, C_{ext} = 0.02 μF, and R_{ext} = 25 kΩ.
4. A ↓ = high level (steady state), L = low level (steady state), ↑ = transition from low to high level, ↓ = transition from high to low level, ⎍ = one high-level pulse, ⎎ = one low-level pulse, X = irrelevant (any input, including transitions).
B. An external timing capacitor may be connected between and R_{ext}/C_{ext} (positive).
C. For accurate repeatable pulse widths, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{ti} open-circuited.
D. To obtain variable pulse widths, connect external resistance between R_{ext} or R_{ext}/C_{ext} and V_{CC}.

ศูนย์วิทยทรัพยากร จุฬาลงกรณ์มหาวิทยาลัย
ห้ามดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54157/74157 Quadruple 2-Line-to-1-Line Data Selector/Multiplexer

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	C
T.I.	SN54S157	J①			W①						SN54LS157	J①			W①	SN54157	J①			W①	SN54L157	J①			
	SN74S157	J①	N①								SN74LS157	J①	N①			SN74157	J①	N①			SN74L157	J①	N①		
FAIRCHILD		9①									FJ54LS157/FW54LS157	9①			F①		9①								
	FC74S157	9①	P①								FC74LS157/FC74LS157	9①	P①		F①	FC74157	9①	P①							
MOTOROLA											MC74LS157		P①			MC74157		P①							
N.S.C.											DM54LS157		①								DM54L157A				
	DM74S157		N①								DM74LS157		①			DM74157		N①			DM74L157A				
PHILIPS																									
	N74S157		①								N74LS157		①			N74157									
SIGNETICS	S54S157															S54157	F①	B①		W①					
	N74S157										N74LS157		A①			N74157	F①	B①							
SIEMENS																FLY171		①							
FUJITSU											74LS157		M①												
HITACHI	HD74S157		P①								HD74LS157		P①			HD74157	①	P①							
MITSUBISHI	M5S157		P①								M74LS157		P①			M53357		P①							
NEC											74LS157		C①			74157		C①							
AMD	AM54S157		①								AM54LS157		①												
	AM74S157		①								AM74LS157		①												

Electrical Characteristics SN54LS157/SN74LS157

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

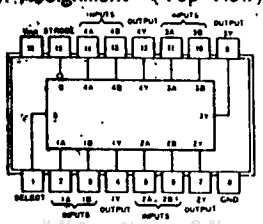
recommended operating conditions

	SN54LS157 MIN	NOM	MAX	SN74LS157 MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC}=MIN, I_I=−18mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC}=MIN, V_{IH}=2V, V_{IL}=0.8V, I_{OH}=−400μA		2.7	3.4		V
V_{OL}	Low-level output voltage	V_{CC}=MIN, V_{IH}=2V, V_{IL}=0.8V, I_{OL}=8mA			0.35	0.5	V
I_I	Input current at maximum input voltage	S or G input	V_{CC}=MAX, V_I=7V			0.2	mA
		A or B input				0.1	
I_{IH}	High-level input current	S or G input	V_{CC}=MAX, V_I=2.7V			40	μA
		A or B input				20	
I_{IL}	Low-level input current	S or G input	V_{CC}=MAX, V_I=0.4V			−0.8	mA
		A or B input				−0.4	
I_{OS}	Short-circuit output current‡	V_{CC}=MAX	SN54LS	−20		−100	mA
			SN74LS	−20		−100	
I_{CC}	Supply current	V_{CC}=MAX See Note			9.7	16	mA
t_{PLH}	from data input	V_{CC}=5V, T_A=25°C, C_L=15pF, R_L=2kΩ			8	14	ns
t_{PHL}					9	14	
t_{PLH}	from Strobe				14	20	ns
t_{PHL}					14	21	
t_{PLH}	from Select				18	23	ns
t_{PHL}					18	27	

Pin Assignment (Top View)



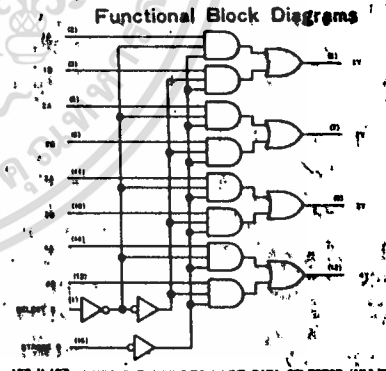
Function Table

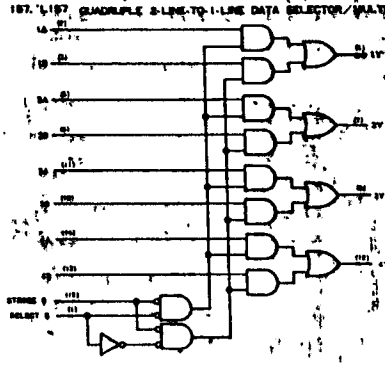
'157, 'S157, 'LS157 'LS158

INPUTS STROBE	SELECT	A	B	OUTPUT Y
H	X	X	X	L
L	L	L	X	L
L	L	H	X	H
L	H	X	L	L
L	H	X	H	H

H = high level, L = low level, X = irrelevant

Functional Block Diagrams





NOTE : I_{CC} is measured with 4.5V applied to all inputs and all outputs open.

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
‡ All typical values are at V_{CC}=5V, T_A=25°C.
§ Not more than one output should be shorted at a time.
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output

เอกสารนี้ใช้ประโยชน์ด้านการศึกษาเท่านั้น ไม่ว่ากรณีใดๆ ห้ามมิให้ทำซ้ำ ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54121/74121. Monostable Multivibrator with Schmitt-Trigger Input

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Dexice Type	Package C	P	M	CF
																SN54121	J①			W①	SN54L121	J①	N①	T①	
																SN74121	J①	N①			SN74L121	J①	N①	T①	
AIRCHILD																FM54121/FM9603	D①			F①					
																FC74121/FC9603	D①	P①							
OTOROLA																MC54121	L①			F①					
																MC74121	L①	P①		F①					
.S.C.																DM54121	J①	N①		W①					
																DM74121	J①	N①							
ILIPS																FJK101/74121		①							
GNETICS																S54121	F①	A①		W①					
																N74121	F①	A①		W①					
EMENS																FLK101		①							
UITSU																									
ITACHI																HD74121/HD2543	①	P①							
ITSUBISHI																M53321		P①							
EC																									
OSHIBA																TD34121A		P①							

Electrical Characteristics SN54S121 SN74S121

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range: SN54S	55°C to 125°C
Input voltage	5.5V	SN74S	0°C to 70°C
		Storage temperature range	65°C to 150°C

recommended operating conditions

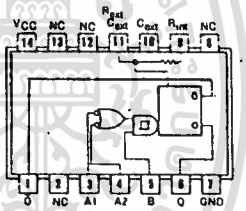
		SN54S121 MIN	NOM	MAX	SN74S121 MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}							400	μA
Low-level output current, I_{OL}							16	mA
Rate of rise or fall of input pulse, dv/dt	Schmitt input B	1			1			V/s
	Logic input A1, A2	1			1			V/s
Input pulse width, t_w		50			50			ns
External timing capacitance, C_{ext}		14		40	14		40	pF
External timing capacitance, C_{ext}		0		1000	0		1000	pF
Duty cycle	B1 (ALL)			67			67	%
	B1 (MAX Rint)			90			90	%
Operating free-air temperature, T_A		55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

	PARAMETER	TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{T+}	Positive-going threshold voltage	V_{CC} = MIN	A input		1.4	2	V
V_{T-}	Negative-going threshold voltage	V_{CC} = MIN	A input	0.8	1.4		V
V_{T+}	Positive-going threshold voltage	V_{CC} = MIN	B input		1.55	2	V
V_{T-}	Negative-going threshold voltage	V_{CC} = MIN	B input	0.8	1.35		V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −12mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, I_{OH} = MAX		2.4	3.4		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, I_{OL} = MAX			0.2	0.4	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 5.5V				1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.4V	A1 or A2			40	μA
	High-level input current		B			80	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V	A1 or A2			−1.6	mA
			B			−3.2	mA
I_{OS}	Short-circuit output current#	V_{CC} = MAX	54 Family	−20		−55	mA
			74 Family	−18		−55	mA
I_{CC}	Supply current	V_{CC} = MAX	Quiescent		13	25	mA
			Triggered		23	40	mA
t_{PLH}	Propagation delay time, low-to-high-level Q output from either A input	V_{CC} = 5V, T_A = 25°C, C_L = 15pF, R_L = 400Ω	C_T = 80 pF, Rint to V_{CC}		45	70	ns
t_{PHL}	Propagation delay time, low-to-high-level Q output from B input				35	55	ns
t_{PLH}	Propagation delay time, high-to-low-level Q output from either A input				50	80	ns
t_{PHL}	Propagation delay time, high-to-low-level Q output from B input				40	65	ns
$t_{w(out)}$	Pulse width obtained using internal timing resistor	V_{CC} = 5V, T_A = 25°C, C_L = 15pF, R_L = 400Ω	C_T = 80pF, Rint to V_{CC}	70	110	150	ns
$t_{w(out)}$	Pulse width obtained with zero timing capacitance		C_T = 0, Rint to V_{CC}		30	50	ns
$t_{w(out)}$	Pulse width obtained using external timing resistor	C_T = 100pF, R_T = 10kΩ		600	700	800	ns
		C_T = 1μF, R_T = 10kΩ		6	7	8	ms

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC} = 5V, T_A = 25°C.
# Not more than one output should be shorted at a time.

Pin Assignment (Top view)



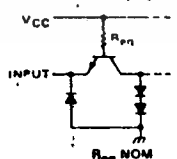
NC—No internal connection
'121···R_{int} = 2 kΩ NOM
L121···R_{int} = 4 kΩ NOM

Function Table

'121, 'L121 (See Note)

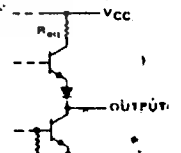
INPUTS A1	A2	B	OUTPUTS Q	$\bar{Q}$
L	X	H	L	H
X	L	H	L	H
X	X	L	L	H
H	H	X	L	H
H	↓	H	⊓	⊔
↓	H	H	⊓	⊔
↓	↓	H	⊓	⊔
L	X	↑	⊓	⊔
X	L	↑	⊓	⊔

schematics of inputs and outputs



INPUT	'121	'L121
A1	4 kΩ	8 kΩ
A2	4 kΩ	8 kΩ
B	2 kΩ	4 kΩ

EQUIVALENT OF EACH INPUT



'121: R_{eq} = 130 Ω NOM
'L121: R_{eq} = 260 Ω NOM

TYPICAL OF BOTH OUTPUTS

NOTES: A. H = high level (steady state), L = low level (steady state), ↑ = transition from low to high level, ↓ = transition from high to low level, ⊓ = one high-level pulse, ⊔ = one low-level pulse, X = irrelevant (any input, including transitions).
B. To use the internal timing resistor connect R_{int} to V_{CC}.
C. An external timing capacitor may be connected between C_{ext} (positive) and R_{ext}/C_{ext}.
D. For accurate repeatable pulse widths, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{int} open-circuited.
E. To obtain variable pulse widths, connect external values resistance between R_{int} or R_{ext}/C_{ext} and V_{CC}.

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อาทิ ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MC3362

Advance Information

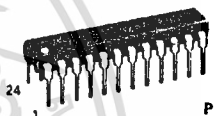
LOW-POWER NARROWBAND FM RECEIVER

. . . includes dual FM conversion with oscillators, mixers, quadrature discriminator, and meter drive carrier detect circuitry. The MC3362 also has buffered first and second local oscillator outputs and a comparator circuit for FSK detection.

- Complete Dual Conversion Circuitry
- Low Voltage: V_{CC} = 2.0 to 6.0 Vdc
- Low Drain Current (3.6 mA (Typ) @ V_{CC} = 3.0 Vdc)
- Excellent Sensitivity: Input Limiting Voltage — (−3.0 dB) = 0.7 μV (Typ)
- Externally Adjustable Carrier Detect Function
- Low Number of External Parts Required
- Manufactured in Motorola's MOSAIC Process Technology
- See AN980 for Additional Design Information

LOW-POWER DUAL CONVERSION FM RECEIVER

SILICON MONOLITHIC INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 724

DW SUFFIX
PLASTIC PACKAGE
CASE 751E
(SO-24L)

8

FIGURE 1 — TYPICAL APPLICATION IN A PLL FREQUENCY SYNTHESIZED RECEIVER

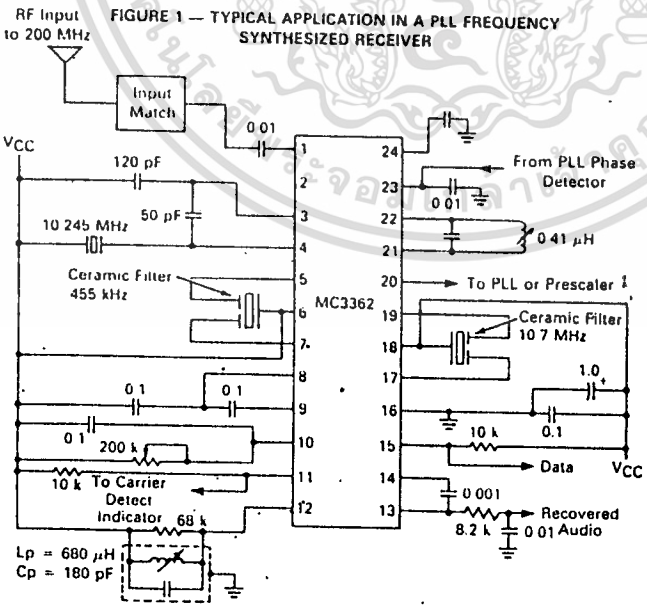
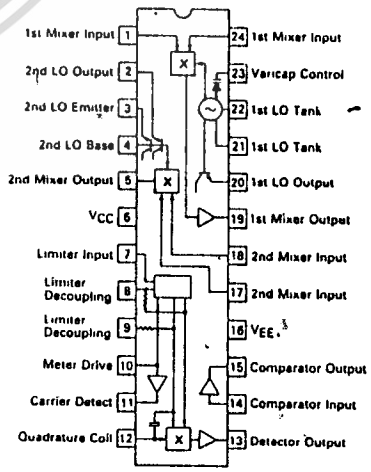


FIGURE 2 — PIN CONNECTIONS AND FUNCTIONAL BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

MAXIMUM RATINGS (T_A = 25°C, unless otherwise noted)

Rating	Pin	Symbol	Value	Unit
Power Supply Voltage (See Diagram)	6	$V_{CC(max)}$	7.0	Vdc
Operating Supply Voltage Range (Recommended)	6	V_{CC}	2.0 to 6.0	Vdc
Input Voltage (V_{CC} = 5.0 Vdc)	1, 24	V_{1-24}	1.0	Vrms
Junction Temperature	—	T_J	150	°C
Operating Ambient Temperature Range	—	T_A	−40 to +85	°C
Storage Temperature Range	—	T_{stg}	65 to +150	°C

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 Vdc, f_o = 49.7 MHz, Deviation = 3.0 kHz, T_A = 25°C, Test Circuit of Figure 3 unless otherwise noted)

Characteristic	Pin	Min	Typ	Max	Units
Drain Current (Carrier Detect Low — See Figure 5)	6	—	4.5	7.0	mA
Input for 3.0 dB Limiting	—	—	0.7	2.0	μVrms
Recovered Audio (RF signal level = 10 mV)	13	—	350	—	mVrms
Noise Output (RF signal level = 0 mV)	13	—	250	—	mVrms
Carrier Detect Threshold (below V_{CC})	10	—	0.64	—	Vdc
Meter Drive Slope	10	—	100	—	nA/dB
Input for 20 dB (S + N) N (See Figure 7)	—	—	0.7	—	μVrms
First Mixer 3rd Order Intercept (Input)	—	—	22	—	dBm
First Mixer Input Resistance (R_P)	—	—	690	—	Ω
First Mixer Input Capacitance (C_P)	—	—	7.2	—	pF
First Mixer Conversion Voltage Gain	—	—	18	—	dB
Second Mixer Conversion Voltage Gain	—	—	21	—	dB
Detector Output Resistance	13	—	1.4	—	kΩ

FIGURE 3 — TEST CIRCUIT

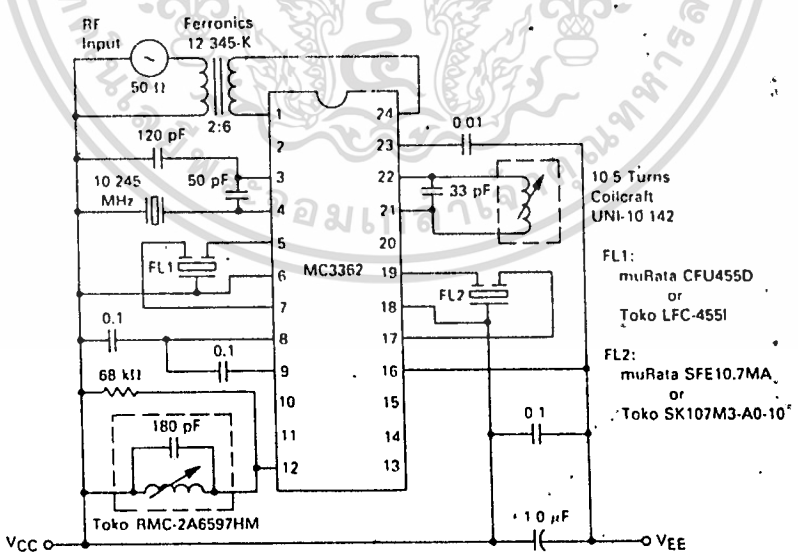


FIGURE 4 — I_{METER} versus INPUT

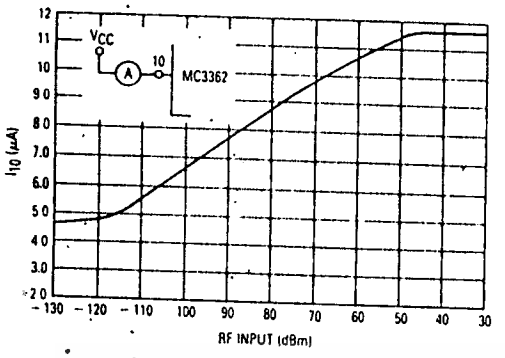


FIGURE 5 — DRAIN CURRENT, RECOVERED AUDIO versus SUPPLY

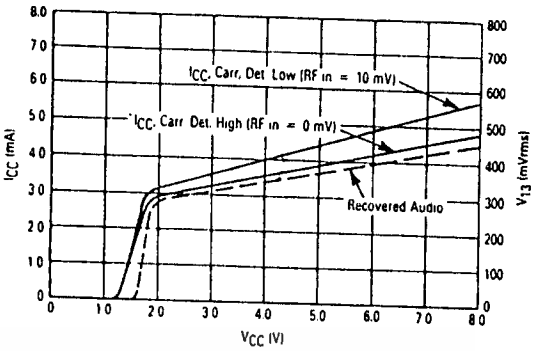


FIGURE 6 — SIGNAL LEVELS

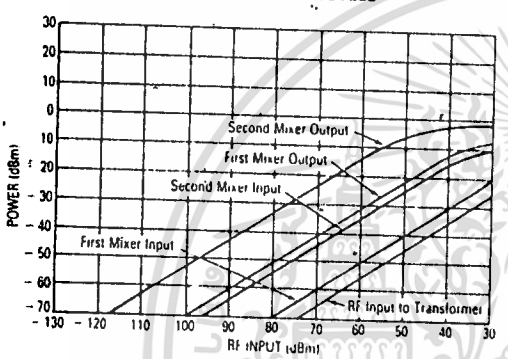


FIGURE 7 — S + N, N, AMR versus INPUT

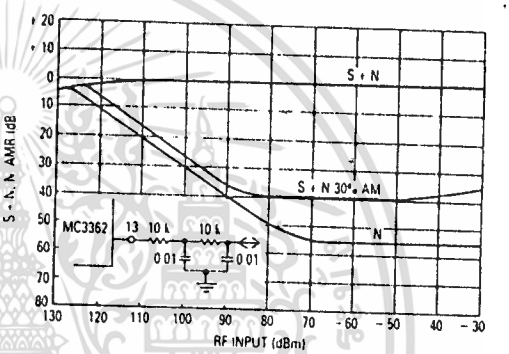


FIGURE 8 — 1ST MIXER 3RD ORDER INTERMODULATION

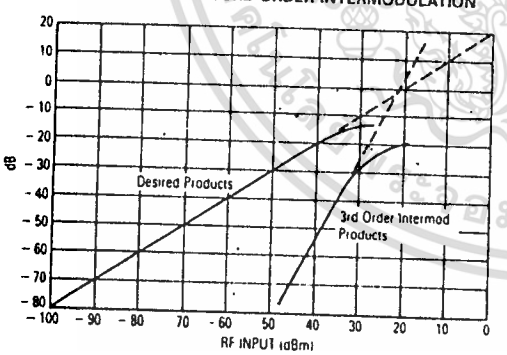
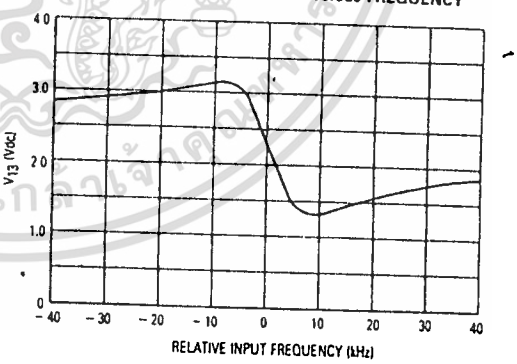


FIGURE 9 — DETECTOR OUTPUT versus FREQUENCY



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 10 — PC BOARD TEST CIRCUIT
(LC Oscillator Configuration Used in PLL Synthesized Receiver)

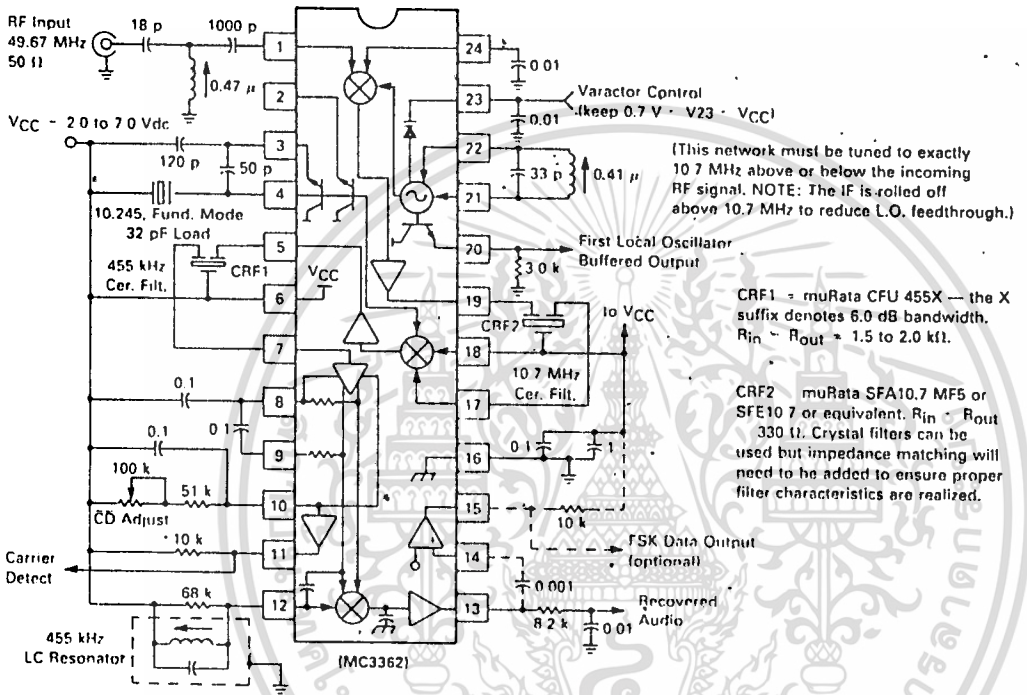
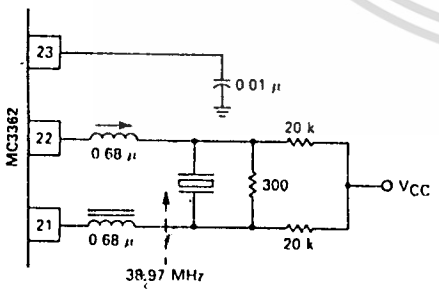


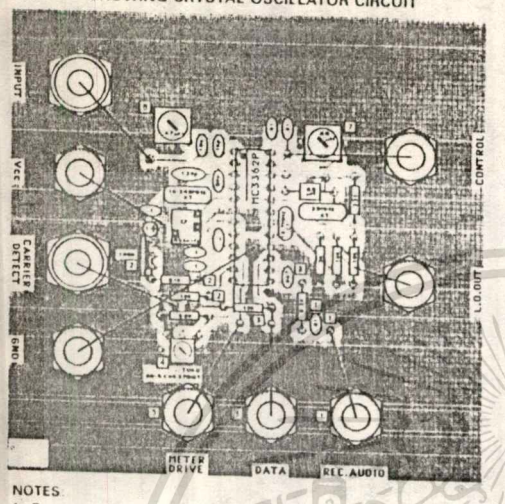
FIGURE 10A — CRYSTAL OSCILLATOR CONFIGURATION FOR SINGLE CHANNEL APPLICATION



Crystal used is series mode resonant (no load capacity specified), 3rd overtone. This method has not proven adequate for fundamental mode, 5th or 7th overtone crystals. The inductor and capacitor will need to be changed for other frequency crystals. See AN980 for further information.

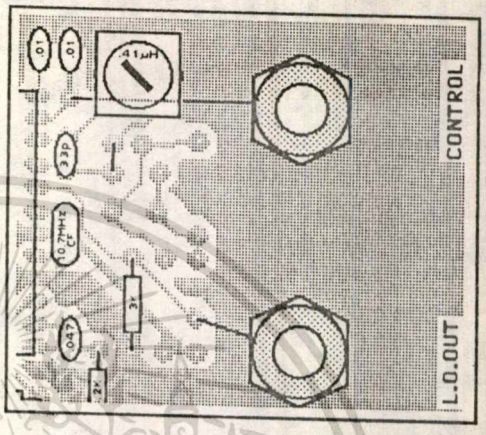
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

FIGURE 11 — COMPONENT PLACEMENT VIEW SHOWING CRYSTAL OSCILLATOR CIRCUIT



NOTES:
1. Recovered Audio components may be deleted when using data output.
2. Carrier Detect components must be deleted in order to obtain linear Meter Drive output. With these components in place the Meter Drive outputs serve only to trip the Carrier Detect indicator.
3. Data Output components should be deleted in applications where only audio modulation is used. For combined audio data applications, the 0.047 μF coupling capacitor will add distortion to the audio so a pull down resistor at pin 13 may be required.
4. Toko 5SVLC0617BG1 5mm quadrature coil for miniaturization. Use Toko RMC2A6597HM high quality 10mm coil for maximum recovered signal at pin 13.

FIGURE 11A — LC OSCILLATOR COMPONENT VIEW



5. Meter Drive cannot be used simultaneously with Carrier Detect output. For analog meter drive, remove components labelled "2" and measure meter current (4–12 μA) through ammeter to V_{CC}.
6. Either type of oscillator circuit may be used with any output circuit configuration.
7. LC Oscillator Coil: Coilcraft UNI 10-42 10.5 turns, 0.41 μH Crystal Oscillator circuit, trim coil, 0.68 μH, Toko B199SNT1051Z.
8. 0.47 μH, Toko B199SNT1050Z. Input LC network used to match first mixer input impedance to 50 Ω.

8

CIRCUIT DESCRIPTION

The MC3362 is a complete FM narrowband receiver from antenna input to audio preamp output. The low voltage dual conversion design yields low power drain, excellent sensitivity and good image rejection in narrowband voice and data link applications.

In the typical application (Figure 1), the first mixer amplifies the signal and converts the RF input to 10.7 MHz. This IF signal is filtered externally and fed into the second mixer, which further amplifies the signal and converts it to a 455 kHz IF signal. After external band pass filtering, the low IF is fed into the limiting amplifier and detection circuitry. The audio is recovered using a conventional quadrature detector. Twice-IF filtering is provided internally.

The input signal level is monitored by meter drive circuitry which detects the amount of limiting in the limiting amplifier. The voltage at the meter drive pin determines the state of the carrier detect output, which is active low.

APPLICATION

The first local oscillator can be run using a free-running LC tank, as a VCO using PLL synthesis, or driven from an external crystal oscillator. It has been run to 190 MHz.* A buffered output is available at Pin 20. The second local oscillator is a common base Colpitts type which is typically run at 10.245 MHz under crystal control. A buffered output is available at Pin 2. Pins 2 and 3 are interchangeable.

The mixers are doubly balanced to reduce spurious responses. The first and second mixers have conversion gains of 18 dB and 22 dB (typical), respectively, as seen in Figure 6. Mixer gain is stable with respect to supply voltage. For both conversions, the mixer impedances and pin layout are designed to allow the user to employ low cost, readily available ceramic filters. Overall sensitivity and AM rejection are shown in Figure 7. The input level for 20 dB (S + N)/N is 0.7 μV using the two-pole post-detection filter pictured.

*If the first local oscillator (Pins 21 and/or 22) is driven from a strong external source (100 mVrms), the mixer can be used to over 450 MHz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Following the first mixer, a 10.7 MHz ceramic bandpass filter is recommended. The 10.7 MHz filtered signal is then fed into one second mixer input pin, the other input pin being connected to V_{CC}. Pin 6 (V_{CC}) is treated as a common point for emitter-driven signals.

The 455 kHz IF is typically filtered using a ceramic bandpass filter then fed into the limiter input pin. The limiter has 10 μV sensitivity for 3.0 dB limiting, flat to 1.0 MHz.

The output of the limiter is internally connected to the quadrature detector, including a quadrature capacitor. A parallel LC tank is needed externally from Pin 12 to V_{CC}. A 68 kΩ shunt resistance is included which determines the peak separation of the quadrature detector; a smaller value will increase the spacing and linearity but decrease recovered audio and sensitivity.

A data shaping circuit is available and can be coupled to the recovered audio output of Pin 13. The circuit is a comparator which is designed to detect zero crossings of FSK modulation. Data rates are typically limited to 1200 baud to ensure data integrity and avoid adjacent channel "splatter." Hysteresis is available by connecting a high valued resistor from Pin 15 to Pin 14. Values below 120 kΩ are not recommended as the input signal cannot overcome the hysteresis.

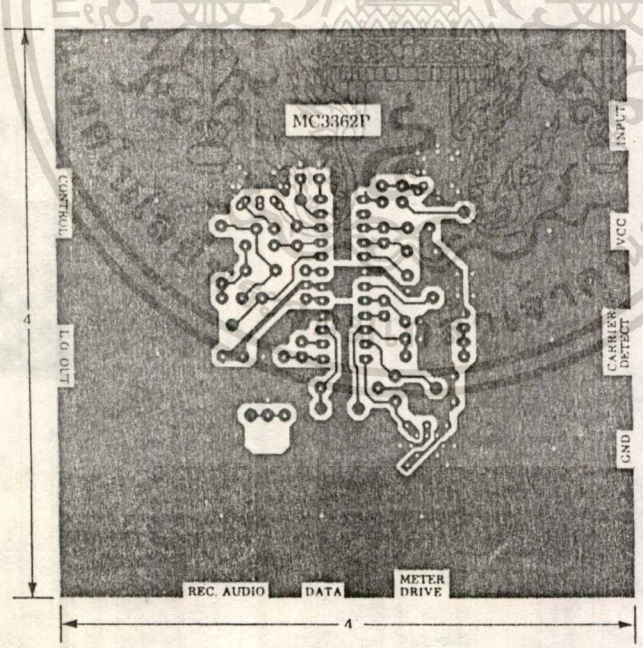
The meter drive circuitry detects input signal level by monitoring the limiting amplifier stages. Figure 4 shows the unloaded current at Pin 10 versus input power. The meter drive current can be used directly (RSSI) or can be used to trip the carrier detect circuit at a specified input power. To do this, pick an RF trip level in dBm. Read the corresponding current from Figure 4 and pick a resistor such that:

$$R_{10} \quad 0.64\ \text{Vdc} \quad I_{10}$$

Hysteresis is available by connecting a high valued resistor R_H between Pins 10 and 11. The formula is:

$$\text{Hysteresis} \quad V_{CC}\ (R_H \times 10^{\ 7})\ \text{dB}$$

FIGURE 12 — CIRCUIT SIDE VIEW



8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

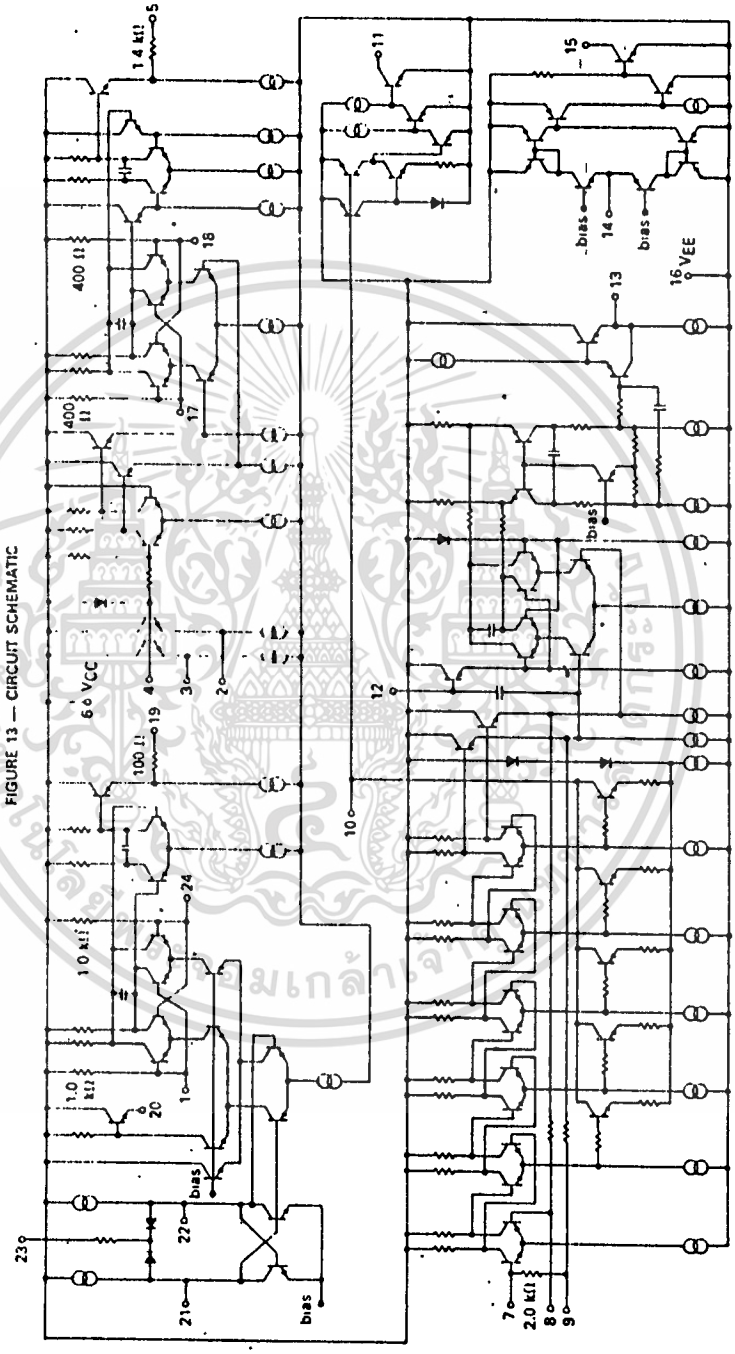


FIGURE 13 — CIRCUIT SCHEMATIC

8