สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

อุปกรณ์เชื่อมต่อโทรศัพท์ผ่านโครงข่ายอินเตอร์เน็ต

Internet Telephony Interconnection Equipment

โดย

นางสาวศิริกุล กุลชุติสิน

นายอิทธนรรย์ สระทองฮัก

ปพ.
ต๓๔๘๐
2549

เลขหมู่.....
เลขทะเบียน 72854
วัน,เดือน,ปี 23 มิ.ย. 2550

b 11223327
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์เชื่อมต่อโทรศัพท์ผ่านโครงข่ายอินเตอร์เน็ต

Internet Telephony Interconnection Equipment

โดย

นางสาวศิริกุล กุลชุติสิน 46010780

นายอิทธนรรย์ สระทองฮัก 46010821

อาจารย์ที่ปรึกษา

ผศ.ดร.พิพัฒน์ พรหมมี

ผศ. ธเนศ พัฒนธาดาพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **อุปกรณ์เชื่อมต่อโทรศัพท์ผ่านโครงข่ายอินเตอร์เน็ต**

Internet Telephony Interconnection Equipment

ผู้จัดทำ

1. นางสาวศิริกุล กุลชุติสิน 46010780

2. นายอิทธนรรย์ สระทองอัก 46010821

..... อาจารย์ที่ปรึกษา

(ผศ.ดร. พิพัฒน์ พรหมมี)

..... อาจารย์ที่ปรึกษา

(ผศ. ธเนศ พัฒนธาดาพงษ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์เชื่อมต่อโทรศัพท์ผ่านโครงข่ายอินเตอร์เน็ต

Internet Telephony Interconnection Equipment

โดย นางสาวศิริกุล กุลชุติสิน 46010780
นายอิทธนรรย์ สระทองยัก 46010821

อาจารย์ที่ปรึกษา ผศ.ดร. พิพัฒน์ พรหมมี
ผศ. ธเนศ พัฒนธาดาพงษ์

บทคัดย่อ

โครงงานนี้นำเสนอชุดอุปกรณ์ส่งสัญญาณเสียงผ่านโครงข่ายอินเตอร์เน็ต ประกอบด้วยชุดแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิตอล และชุดแปลงสัญญาณดิจิตอลเป็นอนาลอก รวมทั้งมีการเชื่อมต่อเข้ากับโครงข่ายอินเตอร์เน็ตโดยใช้บอร์ดแรบบิทร่วมกับไมโครคอนโทรเลอร์ ซึ่งสามารถกำหนดเลขหมายปลายทางด้วย ไอพีแอดเดรส (IP Address) และใช้การกดแป้นตัวเลข ที่ตัวเครื่องเพื่อทำการติดต่อไปยังปลายทาง

ABSTRACT

This project presents an Internet Telephony Interconnection Equipment. It consists of an analog telephone to digital signal and a digital signal to analog telephone parts. The digital signal sent over internet using RABBIT module and microcontroller for Ethernet interfacing. The user can call to another by its keypad based on IP Address method.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน	1
1.4 ขั้นตอนในการทำโครงงาน	1
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 เครือข่ายท้องถิ่น (Local Area Network : LAN)	3
2.2 อีเธอร์เน็ต (Ethernet)	5
2.3 โปรโตคอลทีซีพี/ไอพี	9
2.4 หมายเลขไอพี (IP Addressing)	18
2.5 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	21
2.6 การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	23
2.7 ชุดพัฒนา RCM 3720	27
บทที่ 3 การคำนวณและการสร้าง	37
3.1 บล็อกไดอะแกรมรวมของโครงงานนี้	37
3.2 วงจรเพาเวอร์แอมป์	38
3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	38
3.4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	39
3.5 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter)	40
3.6 วงจรสร้างสัญญาณนาฬิกา	41
3.7 แอลซีดีและคีย์แพด	42
3.8 การส่งข้อมูลในแรบบิทโมดูล	43
3.4 การรับข้อมูลในแรบบิทโมดูล	44
บทที่ 4 การทดลองและผลการทดลอง	45
4.1 วงจรสร้างสัญญาณนาฬิกา	45
4.2 วงจรกรองสัญญาณความถี่ต่ำผ่าน (Low Pass Filter)	45
4.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (ด้านส่ง)	46
4.4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (ด้านรับ)	47
4.5 การรับส่งข้อมูลผ่านแพ็กเก็ตยูดีพีข้ามเน็ตเวิร์ค	48
4.6 การส่งข้อมูลต่าง ๆ ผ่านแพ็กเก็ตยูดีพีด้วยแรบบิทโมดูล	50
4.7 ด้านส่งส่งแพ็กเก็ต CALL ไปทางด้านรับ	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
4.8 ด้านรับตอบกลับแพ็กเก็ต ACK ไปทางด้านส่ง	52
4.9 ด้านส่งส่งแพ็กเก็ต DATA ไปทางด้านรับ	52
4.10 ด้านรับส่งแพ็กเก็ต END ไปทางด้านส่ง	56
บทที่ 5 วิจารณ์และสรุปผล	57
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 เครือข่ายท้องถิ่น	3
รูปที่ 2.2 เปรียบเทียบระหว่าง IEEE Ethernet และแบบอ้างอิง OSI	7
รูปที่ 2.3 ฟอร์แมตของอีเธอร์เน็ต	8
รูปที่ 2.4 ฟอร์แมตของแพ็กเก็ตไอพี	11
รูปที่ 2.5 ฟอร์แมตข้อมูลของแพ็กเก็ตทีซีพี	15
รูปที่ 2.6 ฟอร์แมตข้อมูลของแพ็กเก็ตยูดีพี	16
รูปที่ 2.7 การแบ่งส่วนของหมายเลขไอพีและด็อตเดซิมอลโนเตชัน	18
รูปที่ 2.8 การทำซับเน็ต	21
รูปที่ 2.9 การเข้ารหัสแบบ PCM	22
รูปที่ 2.10 การเกิด Quantization Error	23
รูปที่ 2.11 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ Linear Combination	24
รูปที่ 2.12 วงจรแปลงสัญญาณ Digital เป็น Analog แบบ R-2R Ladder	25
รูปที่ 2.13 วงจรเสมือนของรูปที่ 2.12	25
รูปที่ 2.14 การวิเคราะห์โดยใช้วงจรเสมือน Thevinin	26
รูปที่ 2.15 RCM 3720	27
รูปที่ 2.16 การใช้พอร์ตต่างๆ ของ Rabbit 3000	28
รูปที่ 2.17 แสดงรายละเอียดขาของ J1	28
รูปที่ 2.18 พอร์ต Ethernet RJ-45	31
รูปที่ 2.19 เอาท์พุตแคลเคส รีจิสเตอร์สำหรับ พอร์ตขนาน D และ E	34
รูปที่ 2.20 การกรองสัญญาณดิจิตอลของขาอินพุต	35
รูปที่ 2.21 ไทเมอร์	36
รูปที่ 3.1 บล็อกไดอะแกรมรวมของโครงงานนี้	37
รูปที่ 3.2 วงจรเพาเวอร์แอมป์	38
รูปที่ 3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	39
รูปที่ 3.4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	39
รูปที่ 3.5 วงจร (Low Pass Filter)	40
รูปที่ 3.6 วงจรสร้างสัญญาณนาฬิกา	41
รูปที่ 3.7โฟร์ลชาร์ตแสดงการทำงานของแอลซีดีและคีย์แพด	42
รูปที่ 3.8โฟร์ลชาร์ตแสดงการทำงานทางด้านส่ง	43
รูปที่ 3.9 โฟร์ลชาร์ตแสดงการทำงานทางด้านรับ	44
รูปที่ 4.1 สัญญาณเอาท์พุตจากวงจรสร้างสัญญาณนาฬิกา	45
รูปที่ 4.2 สัญญาณเอาท์พุตจากวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 สัญญาณเอาท์พุตจากวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก เมื่อทำการผ่านวงจรกรองความถี่ต่ำผ่าน 48
รูปที่ 4.4 แสดงผลที่ได้ทางด้านรับเมื่อทำการส่งข้อมูลผ่านแพ็กเก็ตยูดีพี 49
รูปที่ 4.5 แสดงผลที่ได้ทางด้านส่งเมื่อไม่สามารถทำการส่งข้อมูลผ่านแพ็กเก็ตยูดีพีได้ 49
รูปที่ 4.6 แสดงภาพของแรบบิทโมดูลขณะทำการส่งข้อมูล 50
รูปที่ 4.7 เมื่อทำการเชื่อมต่อแรบบิทโมดูลเข้ากับคอมพิวเตอร์ 50
รูปที่ 4.8 ผลที่ได้จากโปรแกรมฮีเธอร์เรียลเมื่อทำการส่งแพ็กเก็ตยูดีพีจากด้านส่งไปด้านรับ 51
รูปที่ 4.9 เมื่อด้านส่งทำการส่งแพ็กเก็ต CALL ไปทางด้านรับ 51
รูปที่ 4.10 เมื่อด้านรับทำการตอบกลับแพ็กเก็ต ACK ไปทางด้านส่ง 52
รูปที่ 4.11 เมื่อด้านส่งทำการส่งแพ็กเกต DATA ไปทางด้านรับ 53
รูปที่ 4.12 แสดงแพ็กเก็ต DATA ลำดับที่ 00 12 คือลำดับที่ 18 53
รูปที่ 4.13 แสดงแพ็กเก็ต DATA ลำดับที่ 00 13 คือลำดับที่ 19 54
รูปที่4.14 เอาท์พุตที่ได้จากด้านรับเมื่อกำหนดให้ด้านส่งส่งข้อมูลครั้งละ 200 ไบต์ 54
รูปที่ 4.15 เอาท์พุตที่ได้จากด้านรับเมื่อกำหนดให้ด้านส่งส่งข้อมูลครั้งละ 50 ไบต์ 55
รูปที่ 4.16 เอาท์พุตที่ได้จากด้านรับเมื่อกำหนดให้ด้านส่งส่งข้อมูลครั้งละ 10 ไบต์ 55
รูปที่ 4.17 เมื่อด้านรับทำการส่งแพ็กเก็ต END ไปทางด้านส่ง 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 หมายเลขพอร์ตของบางแอพพลิเคชัน	17
ตารางที่ 2.2 หมายเลขไอพีเครือข่ายส่วนบุคคล	20
ตารางที่ 2.3 ดีฟอลต์เน็ตมาส์คของไอพีแต่ละประเภท	21
ตารางที่ 2.4 แสดงขาต่างๆของ RCM 3720	29
ตารางที่ 2.4 แสดงขาต่างๆของ RCM 3720 (ต่อ)	30
ตารางที่ 4.1 ผลการทดลองวงจร Low Pass Filter (เมื่อป้อนอินพุท 1 Vp-p)	45
ตารางที่ 4.2 ผลการทดลองวงจร แปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล	46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เมื่ออินเตอร์เน็ตมีการใช้งานกว้างขวางขึ้น ความต้องการการประยุกต์แบบใหม่ ๆบนอินเตอร์เน็ตจึงได้รับการพัฒนา โดยเฉพาะอย่างยิ่งการใช้อินเตอร์เน็ตเป็นโครงสร้างพื้นฐานเพื่อรองรับการสื่อสารรูปแบบต่าง ๆ เช่น การใช้โทรศัพท์บนเครือข่าย การติดต่อด้วยเสียง ระบบวิดีโอคอนเฟอเรนซ์ การกระจายสัญญาณเสียงหรือภาพบนเครือข่าย และสิ่งหนึ่งที่มีการพัฒนาการประยุกต์จนสามารถใช้งานได้ดี คือ ระบบการสื่อสารด้วยเสียงผ่านเครือข่ายไอพี เครือข่ายไอพี (Internet Protocol) มีพัฒนามาจากรากฐานระบบการสื่อสารแบบแพ็กเก็ต โดยระบบมีการกำหนดแอดเดรส ที่เรียกว่า ไอพีแอดเดรส จากไอพีแอดเดรสหนึ่ง ถ้าต้องการส่งข่าวสารไปยังอีกไอพีแอดเดรสหนึ่ง ใช้หลักการบรรจุข้อมูลใส่ใน แพ็กเก็ต แล้วส่งไปในเครือข่าย ระบบการจัดส่งแพ็กเก็ตกระทำด้วยอุปกรณ์สื่อสารจำพวกเราเตอร์ มีหลักพื้นฐานการส่งแบบ ไปรษณีย์สมัยเก่า บางที่เราจึงเรียกการส่งแบบนี้ว่า ดาต้าแกรม

1.2 วัตถุประสงค์

การส่งเสียงบนเครือข่ายไอพี เป็นระบบที่นำสัญญาณข้อมูลเสียงมาบรรจุลงเป็นแพ็กเก็ตไอพีแล้วทำการส่งข้อมูลไปโดยที่เราเตอร์จะรับสัญญาณแพ็กเก็ตแล้วทำการจัดหาเส้นทางที่จะทำการส่งไปยังปลายทาง และยังแก้ปัญหาบางอย่างให้ เช่น การบีบอัดสัญญาณเสียง ให้มีขนาดเล็กลง การแก้ปัญหาเมื่อมีบางแพ็กเก็ตสูญหาย หรือได้มาล่าช้า

1.3 ขอบเขตของโครงงาน

ในโครงงานนี้จะทำการกำหนดให้มีการเชื่อมต่อระหว่างเลขหมายไอพีปลายทางได้แบบจุดต่อจุด โดยสามารถตรวจสอบการเชื่อมต่อและสามารถส่งสัญญาณเสียงผ่านไปในโครงข่ายไอพีได้

1.4 ขั้นตอนในการทำโครงงาน

1. ศึกษาระบบโครงข่ายและโปรโตคอลที่ใช้ในการเชื่อมต่อ
2. ศึกษาการใช้งานแรบบิทโมดูล และภาษาไดนามิก ซี
3. ออกแบบและสร้างบล็อกไดอะแกรมของโครงงานนี้ทั้งหมด
4. ออกแบบวงจรต่าง ๆ ให้ได้ตามเป้าหมายในแต่ละบล็อก
5. ทำบอร์ดและสายที่ใช้ในการลงโปรแกรมของแรบบิทโมดูล
6. สร้างวงจรต่าง ๆ ที่ได้ทำการออกแบบไว้
7. ทดลองทำการส่งข้อมูลระหว่างแรบบิทโมดูลผ่านโครงข่ายไอพี
8. ทดลองวงจรต่าง ๆ บันทึกผลการทดลองที่ได้
9. นำวงจรทั้งหมดมาต่อรวมกันแล้วทดสอบการทำงานร่วมกัน
10. สรุปวิเคราะห์ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

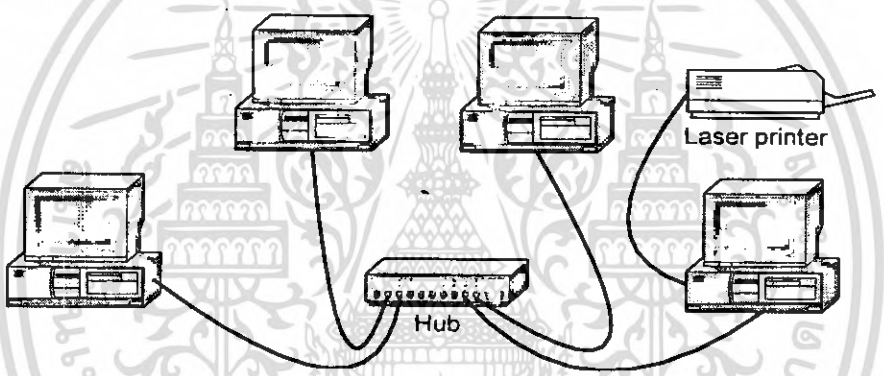
โครงงานนี้ให้ความรู้เกี่ยวกับระบบโครงข่ายไอพี และโปรโตคอลต่าง ๆ ที่ใช้ในการติดต่อสื่อสาร รวมทั้งการใช้งานแรบบิทโมดูล และคาดว่าสามารถที่จะนำไปพัฒนาเพื่อให้มีประสิทธิภาพในการใช้งานได้มากขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและการออกแบบ

2.1 เครือข่ายท้องถิ่น (Local Area Network : LAN)

เครือข่ายท้องถิ่นเป็นรากฐานของเครือข่ายคอมพิวเตอร์ทั่วไป กล่าวคือ เกือบทุกๆเครือข่ายต้องมีเครือข่ายท้องถิ่นเป็นองค์ประกอบ เครือข่ายท้องถิ่น อาจเป็นได้ตั้งแต่เครือข่ายแบบง่ายๆ เช่น มีคอมพิวเตอร์สองเครื่องเชื่อมต่อกันด้วยสายสัญญาณ ไปจนถึงเครือข่ายที่ซับซ้อน เช่น มีคอมพิวเตอร์เป็นร้อยๆเครื่องและมีอุปกรณ์เครือข่ายอื่นๆอีกมาก แต่ลักษณะสำคัญของเครือข่ายท้องถิ่นก็คือ เครือข่ายจะครอบคลุมพื้นที่จำกัด รูปที่ 2.1 แสดงเครือข่ายท้องถิ่นที่ประกอบด้วยเครื่องคอมพิวเตอร์สี่เครื่อง และมีเครื่องพิมพ์ที่แชร์กันใช้ เครื่องเซิร์ฟเวอร์ที่ใช้ในการจัดการเครือข่าย ซึ่งเครือข่ายจะรวมกันอยู่ในห้องปฏิบัติการ



รูปที่ 2.1 เครือข่ายท้องถิ่น

2.1.1 โครงสร้างของเครือข่าย (Network Topology)

โครงสร้างของเครือข่ายจะอธิบายถึงการเชื่อมต่อคอมพิวเตอร์ตามลักษณะทางกายภาพหรือทางตรรกะ ซึ่งจะแสดงถึงตำแหน่งของคอมพิวเตอร์และอุปกรณ์เครือข่ายอื่นๆ และเส้นทางการเชื่อมต่อของอุปกรณ์เหล่านี้ โครงสร้างของเครือข่ายจะมีผลต่อสมรรถนะของเครือข่าย การเลือกใช้โครงสร้างของเครือข่ายต้องมีการวางแผนที่ดี เพราะโครงสร้างจะมีผลต่อชนิดของสายนำสัญญาณที่ใช้และเป็นตัวกำหนดลักษณะการสื่อสารกันระหว่างคอมพิวเตอร์ด้วย

- **โครงสร้างแบบดาว (Star Topology)** เป็นโครงสร้างที่เชื่อมต่อคอมพิวเตอร์แต่ละตัวเข้ากับคอมพิวเตอร์ศูนย์กลาง การรับส่งข้อมูลทั้งหมดจะต้องผ่านคอมพิวเตอร์ศูนย์กลางเสมอ มีข้อดีคือ การเชื่อมต่อคอมพิวเตอร์เครื่องใหม่สามารถทำได้ง่ายและไม่กระทบกับเครื่องอื่นในระบบเลย แต่ข้อเสียคือมีค่าใช้จ่ายเกี่ยวกับสายสูงและถ้าคอมพิวเตอร์ศูนย์กลางเสียระบบเครือข่ายจะหยุดชะงักทั้งหมดทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **โครงสร้างแบบบัส (Bus Topology)** เป็นโครงสร้างที่เชื่อมคอมพิวเตอร์แต่ละตัวด้วยสายเคเบิลที่ใช้ร่วมกัน ซึ่งสายเคเบิลหรือบัสนี้เปรียบเสมือนกับถนนที่ข้อมูลจะถูกส่งผ่านไปมาระหว่างแต่ละเครื่องได้ตลอดเวลา โดยไม่ต้องผ่านไปที่ศูนย์กลางก่อน โครงสร้างแบบนี้มีข้อดีที่ใช้สายน้อย และถ้ามีเครื่องเสียก็ไม่มีผลอะไรต่อระบบโดยรวม ส่วนข้อเสียก็คือตรวจหาจุดที่เป็นปัญหาได้ยาก
- **โครงสร้างแบบวงแหวน (Ring Topology)** เป็นโครงสร้างที่เชื่อมคอมพิวเตอร์ทั้งหมดเข้าเป็นวงแหวน ข้อมูลจะถูกส่งต่อๆกันไปในวงแหวนจนกว่าจะถึงเครื่องผู้รับที่ถูกต้อง ข้อดีของโครงสร้างแบบนี้คือ ใช้สายเคเบิลน้อยและสามารถตัดเครื่องที่เสียออกจากระบบได้ ทำให้ไม่มีผลต่อระบบเครือข่าย ข้อเสียคือหากมีเครื่องที่มีปัญหาอยู่ในระบบจะทำให้เครือข่ายไม่สามารถทำงานได้เลย และการเชื่อมต่อเครื่องเข้าสู่เครือข่ายอาจต้องหยุดระบบทั้งหมดลงก่อน

2.1.2 เทคโนโลยีเครือข่าย

เทคโนโลยีเครือข่ายท้องถิ่นมีหลายประเภท เช่น อีเธอร์เน็ต (Ethernet), เอทีเอ็ม (ATM), โทเคนริง(Token Ring),เอฟดีดีไอ (FDDI) เป็นต้น แต่ที่นิยมกันมากที่สุดในปัจจุบันคือ อีเธอร์เน็ต ซึ่งในอีเธอร์เน็ตเองยังจำแนกออกได้หลายประเภทขึ้นอยู่กับความเร็ว โครงสร้างของเครือข่ายและสายสัญญาณที่ใช้ เทคโนโลยีเครือข่ายท้องถิ่นแต่ละประเภทมีทั้งข้อดีข้อเสียที่แตกต่างกัน การเลือกใช้เทคโนโลยีเหล่านี้ควรเลือกให้เหมาะกับลักษณะการใช้งานเครือข่าย

- อีเธอร์เน็ต (Ethernet)

อีเธอร์เน็ตได้ถูกคิดค้นขึ้นตั้งแต่ทศวรรษ 1970 และยังคงเป็นเทคโนโลยีชั้นนำของเครือข่ายท้องถิ่น อีเธอร์เน็ตตั้งอยู่บนมาตรฐานการส่งข้อมูลหรือโปรโตคอลซีเอสเอ็มเอซีดี (CSMA/CD : Carrier Sense Multiple Access with Collision Detection) โปรโตคอลนี้ถูกใช้สำหรับการเข้าใช้สื่อกลางในการส่งสัญญาณที่แชร์กันระหว่างสถานีหรือโนด (Node) ต่างๆ ซึ่งมีขั้นตอนดังนี้ เมื่อโนดใดๆต้องการที่จะส่งข้อมูลจะต้องคอยฟังก่อน (Carrier Sense) ว่ามีโนดอื่นกำลังส่งข้อมูลอยู่หรือไม่ ถ้ามีให้รอจนกว่าโนดนั้นจะส่งข้อมูลเสร็จก่อน แล้วค่อยเริ่มส่งข้อมูล และในขณะที่กำลังส่งข้อมูลอยู่นั้นต้องคอยตรวจสอบว่ามีการชนกันของข้อมูลเกิดขึ้นหรือไม่ (Collision Detection) ถ้ามีการชนกันของข้อมูลเกิดขึ้นให้หยุดทำการส่งข้อมูลทันที และส่งสัญญาณแจ้งเตือนไปยังโนดอื่นๆว่ามีการชนกันของข้อมูลเกิดขึ้น แล้วค่อยเริ่มกระบวนการส่งข้อมูลใหม่อีกครั้ง เนื่องจากอีเธอร์เน็ตจะใช้สื่อกลางร่วมกัน ซึ่งเรียกว่า "บัส (BUS)" ฉะนั้นจึงมีโนดที่ส่งข้อมูลได้แค่โนดเดียวในขณะใดขณะหนึ่ง การชนกันของข้อมูลเกิดขึ้นเนื่องจากมีมากกว่าหนึ่งโนดที่ทำการส่งข้อมูลไปบนสื่อกลางในเวลาเดียวกัน ผลที่ได้คือ ข้อมูลจะกลายเป็นขยะหรืออ่านไม่ได้ทันที เมื่อมีจำนวนโนดเพิ่มมากขึ้นความน่าจะเป็นที่ข้อมูลจะชนกันก็เพิ่มขึ้นตามลำดับ

ตามมาตรฐานแล้วอีเธอร์เน็ตจะมีอัตราการส่งข้อมูลหรือแบนด์วิธที่ 10 Mbps (สิบล้านบิตต่อวินาที) ในขณะที่ฟาสต์อีเธอร์เน็ต (Fast Ethernet) มีการทำงานคล้ายๆกัน เพียงแต่มีอัตราข้อมูลที่สูงกว่า 10 เท่าหรือ 100 Mbps ส่วนกิกะบิตอีเธอร์เน็ต (Gigabit Ethernet) มีอัตราการส่งข้อมูลสูงสุดคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1,000 Mbps หรือ 1 Gbps และยังได้มีการพัฒนาอีเธอร์เน็ตที่ความเร็ว 10 Gbps ซึ่งเรียกว่า เท็นกิกะบิตอีเธอร์เน็ต (10G Ethernet)

นอกจากข้อแตกต่างในเรื่องของความเร็วแล้ว อีเธอร์เน็ตยังแบ่งย่อยออกเป็นแชร์อีเธอร์เน็ต (Shared Ethernet) และสวิตช์อีเธอร์เน็ต (Switched Ethernet) โดยแชร์อีเธอร์เน็ตมีการใช้ตัวกลางร่วมกัน คล้ายๆกับถนนมีเลนเดียว ดังนั้นจึงมีรถวิ่งบนถนนได้แค่คันเดียวในขณะใดขณะหนึ่ง ในความหมายเครือข่ายก็คือ ในขณะใดขณะหนึ่งจะมีแค่สถานีเดียวที่สามารถส่งข้อมูลได้ อุปกรณ์เครือข่ายที่ใช้สำหรับแชร์อีเธอร์เน็ตคือ ฮับ (Hub) ส่วนสวิตช์อีเธอร์เน็ตจะเปรียบได้กับถนนที่มีหลายเลน ดังนั้นจึงมีรถหลายคันที่สามารถวิ่งบนถนนได้ในเวลาเดียวกัน ส่วนอุปกรณ์ที่ใช้ในสวิตช์อีเธอร์เน็ตก็คือสวิตช์นั่นเอง

- โทเคนริง (Token Ring)

เครือข่ายแบบโทเคนริงซึ่งจะมีลักษณะการเชื่อมต่อแบบวงแหวนนี้ ถือได้ว่าเป็นเครือข่ายที่กำลังล้าสมัยเพราะมีการใช้งานน้อยลง โทเคนริงเป็นเครือข่ายที่นิยมใช้กันมากในการสร้างเครือข่ายช่วงสมัยแรกๆ เนื่องจากข้อดีของการส่งข้อมูลในเครือข่ายแบบนี้จะไม่มีการชนกันของข้อมูล เหมือนกับเครือข่ายแบบอีเธอร์เน็ต แต่ข้อเสียของเครือข่ายประเภทนี้อยู่ที่ความสามารถในการขยายเครือข่าย (Scalability) และการบริหารและจัดการเครือข่ายจะค่อนข้างยาก เครือข่ายประเภทนี้ยังมีใช้อยู่กับอุปกรณ์คอมพิวเตอร์ของบริษัทไอบีเอ็มที่เป็นระบบเมนเฟรมและมินิคอมพิวเตอร์

- เอทีเอ็ม (ATM :Asynchronous Transfer Mode)

ATM ย่อมาจาก "Asynchronous Transfer Mode" ไม่ได้หมายถึงตู้ ATM (Automatic Teller Machine) ที่เราใช้ถอนเงินสดจากธนาคาร แต่บางทีตู้ ATM ที่ใช้ถอนเงินสดอาจจะเชื่อมต่อกับศูนย์กลางด้วยระบบเครือข่ายแบบ ATM ก็ได้ ATM เป็นมาตรฐานการรับส่งข้อมูลที่กำหนดโดย ITU-T (International Telecommunication Union-Telecommunication Standard Sector) ซึ่งจะรวมบริการต่างๆ เช่น ข้อมูล เสียง วีดีโอ เข้าด้วยกันแล้วส่งเป็นเซลล์ (Cell) ข้อมูลที่มีขนาดเล็กและคงที่ เป็นเครือข่ายที่รองรับแบนด์วิธ (band width) ตั้งแต่ Mbps จนถึง Gbps ปัจจุบันยังมีการใช้ ATM ไม่มากเท่ากับอีเธอร์เน็ต แต่มีแนวโน้มว่า ATM อาจจะเป็นอีกทางเลือกหนึ่งที่นิยมในเครือข่ายในอนาคต

2.2 อีเธอร์เน็ต (Ethernet)

ปัจจุบันเป็นที่ยอมรับกันว่าอีเธอร์เน็ต เป็นเทคโนโลยีเครือข่ายที่เป็นฐานหลักของเทคโนโลยีสารสนเทศทั้งหมด เนื่องจากเป็นเทคโนโลยีเครือข่ายท้องถิ่นที่เป็นที่นิยมมากที่สุด อีเธอร์เน็ตมีอายุกว่า 30 ปีแล้ว และได้มีการพัฒนาอย่างต่อเนื่อง ด้วยเหตุนี้จึงเป็นการยากที่จะพัฒนาเทคโนโลยีใหม่มาแทนที่ได้ เทคโนโลยีนี้ได้ถูกพัฒนาและปรับปรุงภายใต้ความดูแลและรับผิดชอบของสถาบัน IEEE (Institute of Electrical and Electronics Engineer) โดยสิ่งที่สำคัญอย่างหนึ่งในการเปลี่ยนแปลงและปรุงคือการเพิ่มความเร็วในการรับส่งข้อมูลหรือแบนด์วิธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการปรับปรุงครั้งแรกนั้นเป็นการปรับจากความเร็วเดิมที่ 10 Mbps เป็น 100 Mbps ซึ่งในการปรับปรุงครั้งนั้นได้มีการพัฒนาฟิสิคอลเลเยอร์ (Physical Layer) เพื่อให้สามารถรับส่งข้อมูลได้ที่ความเร็ว 100 Mbps และในการปรับปรุงฟิสิคอลเลเยอร์นี้ทำให้ต้องมีการปรับเปลี่ยนดาต้าลิงค์เลเยอร์ (Data Link Layer) เช่นกัน มาตรฐานใหม่นี้เรียกว่า "อีเธอร์เน็ตความเร็วสูง หรือฟาสต์อีเธอร์เน็ต" และได้รับความนิยมเหนือ ATM

2.2.1 สถาปัตยกรรม IEEE 802.3 อีเธอร์เน็ต

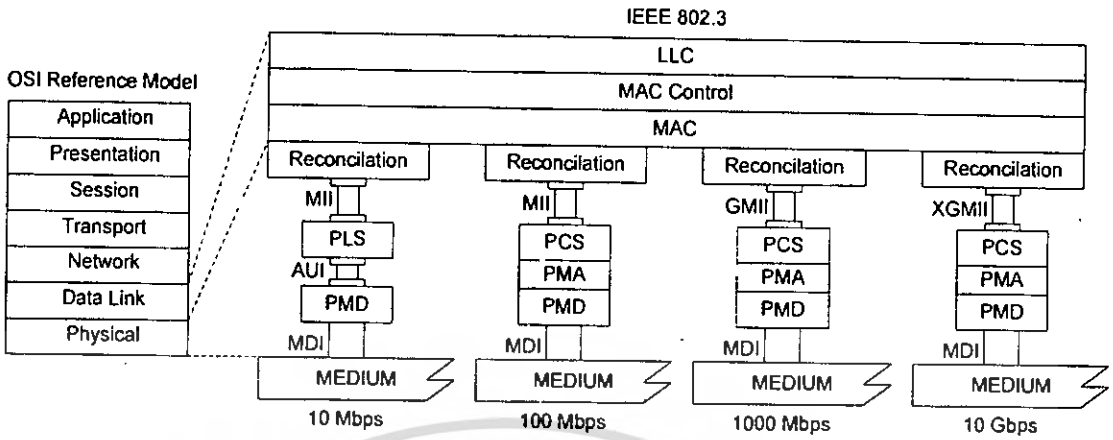
ในสมัยแรกคำว่า "อีเธอร์เน็ต กับ ซีเอสเอ็มเอซีดี" มักจะหมายถึงระบบเครือข่ายชนิดเดียวกัน เนื่องจากอีเธอร์เน็ตจะใช้โปรโตคอลซีเอสเอ็มเอซีดีในการเข้าถึงสื่อกลางในการรับส่งข้อมูล แต่ปัจจุบันความหมายของอีเธอร์เน็ตได้เปลี่ยนไปเพราะได้มีการปรับปรุงเทคโนโลยี เช่น อีเธอร์เน็ตความเร็วสูงได้พัฒนาโปรโตคอลในฟิสิคอลเลเยอร์ใหม่ และมีการปรับเปลี่ยนกลไกในการเข้าใช้สื่อกลางเล็กน้อย สิ่งที่สำคัญที่สุดคือ การเพิ่มการรับส่งข้อมูลแบบฟูลล์ดูเพล็ก (Full Duplex) หรือการสื่อสารข้อมูลที่สามารถรับส่งข้อมูลได้ในเวลาเดียวกัน ซึ่งการรับส่งข้อมูลแบบนี้จะใช้สายคู่เกลียวบิดหนึ่งคู่ในการส่งข้อมูลและอีกหนึ่งคู่ในการรับข้อมูล เมื่อใช้เทคโนโลยีสวิตชิ่ง (Switching) ทำให้กำจัดปัญหาในการเข้าใช้สื่อกลางได้ เนื่องจากไม่จำเป็นต้องแชร์สื่อนำสัญญาณร่วมกัน อุปกรณ์เครือข่ายสามารถรับส่งข้อมูลได้ในอัตราที่สูงขึ้น ซึ่งประสิทธิภาพของเครือข่ายจะถูกจำกัดโดยสายสัญญาณที่ใช้มากกว่า ดังนั้นคำว่าซีเอสเอ็มเอซีดีก็จะใช้แทนคำว่าอีเธอร์เน็ตไม่ได้อีกต่อไป

อีกคำหนึ่งคือ บรอดแบนด์ (Broadband) กับเบสแบนด์ (Baseband) อีเธอร์เน็ตเกือบทุกประเภทจะเป็นแบบเบสแบนด์ มีบางประเภทที่เป็นบรอดแบนด์ เช่น 10Broad36 แต่มีการใช้เครือข่ายประเภทนี้จริงน้อยมาก และช่วงหลังๆ แทบจะไม่มีมาตรฐานอีเธอร์เน็ตที่เป็นแบบบรอดแบนด์เลย เพราะฉะนั้นเมื่อกล่าวถึงอีเธอร์เน็ตมักจะหมายถึงการส่งข้อมูลแบบเบสแบนด์

เพื่อไม่เป็นการสับสนกับชื่อต่างๆ จึงขอสรุปการเรียกชื่อดังนี้ เมื่อกล่าวถึงอีเธอร์เน็ตจะหมายถึงอีเธอร์เน็ตแบบดั้งเดิมที่มีความเร็วที่ 10 Mbps ส่วนคำว่าอีเธอร์เน็ตความเร็วสูงหรือฟาสต์อีเธอร์เน็ตจะหมายถึงอีเธอร์เน็ตที่ความเร็ว 100 Mbps ส่วนกิกะบิตอีเธอร์เน็ตจะหมายถึงอีเธอร์เน็ตที่มีความเร็วที่ 1,000 Mbps หรือ 1 Gbps และสุดท้ายเทนกิกะบิตอีเธอร์เน็ตนั้น จะหมายถึงอีเธอร์เน็ตที่ความเร็ว 10 Gbps หรือบางทีก็เรียกว่า 10 GbE ก็ได้เช่นกัน

คณะชุดทำงานของสถาบัน IEEE ได้ออกแบบอีเธอร์เน็ตโดยการแบ่งแยกหน้าที่ หรือฟังก์ชันของเครือข่ายท้องถิ่น ออกเป็นส่วนย่อยๆหรือเลเยอร์ (Layer) ตามลำดับขั้นตอนของเหตุการณ์ต่างๆที่เกิดขึ้นระหว่างการสื่อสารผ่านเครือข่าย รูปที่ 2.2 แสดงการเปรียบเทียบเลเยอร์ต่างๆของอีเธอร์เน็ตกับแบบอ้างอิง OSI ซึ่งจะเห็นได้ว่ามีทั้งเลเยอร์ย่อยและมีโมดูลในบางเลเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 เปรียบเทียบระหว่าง IEEE Ethernet และแบบอ้างอิง OSI

ดาต้าลิงค์เลเยอร์ (Data Link Layer)

IEEE ได้แบ่งชั้นเชื่อมโยงข้อมูลหรือดาต้าลิงค์เลเยอร์ออกเป็น 2 เลเยอร์ย่อย คือ LLC (Logical Link Control) และ MAC (Medium Access Control) ทั้งสองเลเยอร์ย่อยนี้ถือได้ว่าเป็นหัวใจของอีเธอร์เน็ต เนื่องจากเป็นเลเยอร์ที่สร้างเฟรมข้อมูลและที่อยู่ (Address) เป็นชั้นที่ทำให้ข้อมูลส่งถึงปลายทางได้อย่างถูกต้องและในสองเลเยอร์นี้ยังรับผิดชอบเกี่ยวกับกลไกการตรวจสอบข้อผิดพลาดที่อาจเกิดขึ้นระหว่างการรับส่งข้อมูล และถ้ามีข้อผิดพลาดเกิดขึ้นก็จะเตรียมการในการส่งข้อมูลใหม่ โดยสรุปก็คือเป็นเลเยอร์ที่ควบคุมการรับส่งข้อมูล ถึงแม้ว่าจะไม่ใช่เลเยอร์ที่ส่งข้อมูลจริงๆก็ตาม เลเยอร์ที่ทำการรับส่งข้อมูลจริงๆก็คือ ฟิสิคอลเลเยอร์นั่นเอง

- โลจิคอลลิงค์คอนโทรล (Logical Link Control:LCC)

โลจิคอลลิงค์คอนโทรลเป็นเลเยอร์ที่อยู่ด้านบนของดาต้าลิงค์เลเยอร์ ซึ่งจะให้บริการกับโปรโตคอลของเลเยอร์บนในการเข้าใช้สื่อกลางหรือสายสัญญาณในการรับส่งข้อมูล ตามมาตรฐาน IEEE 802.3 แล้ว จะอนุญาตให้สถาปัตยกรรมของเครือข่ายท้องถิ่นที่ต่างกันสามารถทำงานร่วมกันได้ กล่าวคือ โปรโตคอลเลเยอร์บนไม่จำเป็นต้องทราบว่าฟิสิคอลเลเยอร์ใช้สายสัญญาณประเภทใดในการรับส่งข้อมูล เพราะ LLC จะรับผิดชอบแทนในการปรับเฟรมข้อมูลให้สามารถส่งไปได้ในสายสัญญาณประเภทนั้นๆ LLC เป็นเลเยอร์ที่แยกเน็ตเวิร์คเลเยอร์ (Network Layer) ออกจากการเปลี่ยนแปลงบ่อยๆ ของสถาปัตยกรรมเครือข่ายท้องถิ่น โดยโปรโตคอลของเลเยอร์ที่สูงกว่าไม่จำเป็นต้องสนใจว่าแพ็กเก็ตนั้นจะส่งผ่านเครือข่ายแบบอีเธอร์เน็ต โทเคนริง หรือ ATM และไม่จำเป็นต้องรู้ว่าการส่งผ่านข้อมูลในฟิสิคอลเลเยอร์จะใช้การรับส่งข้อมูลแบบใด ชั้น LLC จะจัดการเรื่องเหล่านี้ให้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีเดียแอคเซสคอนโทรล (Media Access Control:MAC)

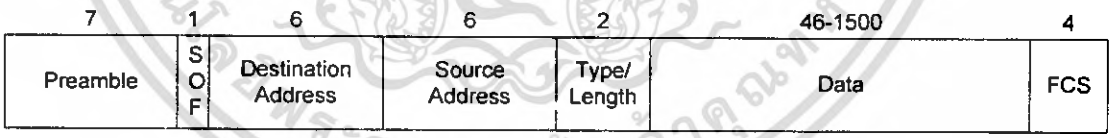
มีเดียแอคเซสคอนโทรลเป็นเลเยอร์ที่อยู่ล่างสุดของดาต้าลิงค์เลเยอร์ ซึ่งจะทำหน้าที่เชื่อมต่อกับฟิสิคอลเลเยอร์ (Physical Layer) และรับผิดชอบในการรับส่งข้อมูลให้สำเร็จและถูกต้อง โดยจะแบ่งหน้าที่ออกเป็นสองส่วนคือ การส่งข้อมูลและการรับข้อมูล

MAC จะทำหน้าที่ห่อหุ้มข้อมูลที่ส่งผ่านจากชั้น LLC และทำให้อยู่ในรูปของเฟรมข้อมูล ซึ่งเฟรมข้อมูลนี้จะประกอบด้วยที่อยู่และข้อมูลต่างๆที่จำเป็นสำหรับการส่งข้อมูลให้ถึงปลายทาง ชั้น MAC ยังรับผิดชอบในการสร้างกลไกสำหรับตรวจสอบข้อผิดพลาดของข้อมูลในเฟรมนั้นๆ ในระหว่างการรับส่งเฟรมด้วย นอกจากนี้ MAC ยังต้องตรวจสอบฟิสิคอลเลเยอร์ว่าช่องสัญญาณพร้อมสำหรับการส่งข้อมูลหรือไม่ ถ้าพร้อมเฟรมก็จะถูกส่งต่อไปยังฟิสิคอลเลเยอร์เพื่อทำการส่งไปตามสายสัญญาณต่อไป แต่ถ้ายังไม่พร้อมชั้น MAC ก็จะรอจนกว่าจะว่าง แล้วค่อยทำการส่งข้อมูล

หน้าที่สุดท้ายของชั้น MAC คือการตรวจสอบสถานภาพของเฟรมที่กำลังส่ง ว่ามีการชนกันของข้อมูลเกิดขึ้นหรือไม่ ถ้าหากมีการชนกันเกิดขึ้นก็จะหยุดส่งข้อมูล และเข้าสู่กลไกการรอด้วยช่วงเวลาที่เป็นเลขสุ่มเพื่อทำการส่งข้อมูลใหม่อีกครั้ง ซึ่งจะทำเช่นนี้ไปเรื่อยๆ จนกว่าจะทำการส่งข้อมูลได้สำเร็จ กระบวนการส่งข้อมูลที่ว่านี้เป็นทั้งข้อดีและข้อเสียของอีเธอร์เน็ต ข้อดีก็คือเป็นการรับรองให้แก่โปรโตคอลชั้นที่อยู่เหนือกว่ามั่นใจว่าข้อมูลจะถูกส่งไปถึงปลายทางอย่างแน่นอน แต่ในขณะเดียวกันข้อเสียก็คือการส่งข้อมูลอาจใช้เวลานานมากถ้ามีการใช้เครือข่ายมาก ๆ

2.2.2 ฟอร์แมตเฟรมข้อมูล

ข้อมูลที่อยู่ในเลเยอร์ที่สองนั้นจะถูกจัดให้อยู่ในรูปของเฟรม ซึ่งตามมาตรฐาน IEEE 802.3 แล้วรูปแบบของเฟรมจะเป็นดังรูปที่ 2.3 โดยตัวเลขที่แสดงข้างบนเป็นจำนวนไบต์ของฟิลด์นั้นๆ



รูปที่ 2.3 ฟอร์แมตของอีเธอร์เน็ต

รายละเอียดของฟิลด์ต่างๆ ของเฟรมอีเธอร์เน็ตแสดงในรูปที่ 2.3 ดังนี้

- Preamble : เป็นฟิลด์ที่มีบิตสลับกันระหว่าง 1 กับ 0 ซึ่งเป็นสัญญาณบอกสถานีฝ่ายรับว่ากำลังมีข้อมูลส่งมา ฟิลด์นี้มีความยาว 8 ไบต์ โดยรวมเอาไบต์ของ SOF เข้าด้วย
- SOF (Start-of-Frame) : เป็นไบต์สุดท้ายของพรีแอมเบิล (Preamble) ซึ่งไบต์นี้จะแตกต่างจากไบต์อื่นๆคือ 2 บิตสุดท้ายจะเป็น 1 ทั้งคู่ เพื่อเป็นสัญญาณสำหรับบอกจุดเริ่มต้นของเฟรมจริงๆ
- Destination/Source Address : หมายเลข หรือที่อยู่ของสถานีปลายทางและต้นทางซึ่งมีความยาวส่วนละ 48 บิต ส่วนใหญ่จะเรียกว่า "แมคแอดเดรส (MAC Address)" การกำหนดหมายเลขนี้จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมโดย IEEE ซึ่งมีเกณฑ์คือ 24 บิตแรกเป็นหมายเลขที่กำหนดให้กับบริษัทผู้ผลิต NIC (Network Interface Card) และบริษัทนั้นจะเป็นผู้กำหนดอีก 24 บิตที่เหลือ การทำเช่นนี้เพื่อให้แน่ใจว่าในเครือข่ายหนึ่งๆจะไม่มีหมายเลขนี้ซ้ำกัน

- Type/Length : เป็นฟิลด์ที่บอกประเภทของโปรโตคอลของเลเยอร์ที่อยู่เหนือกว่า เช่น IP, IPX, IPv6, ARP, AppleTalk เป็นต้น
- Data : ส่วนนี้จะเป็นฟิลด์ที่เก็บข้อมูลซึ่งมีความยาวอย่างน้อยต้องไม่ต่ำกว่า 46 ไบต์ ถ้าต่ำกว่านี้จะต้องมีฟิลด์เสริม (Padding) เพื่อให้ข้อมูลมีอย่างน้อย 46 ไบต์ เหตุที่ต้องกำหนดความยาวขั้นต่ำนี้ก็เพื่อสำหรับการตรวจเช็คการชนกันของข้อมูล (Collision) ในระหว่างการรับส่งข้อมูล ส่วนความยาวสูงสุดคือ 1,500 ไบต์
- FCS (Frame Check Sequence) : ฟิลด์นี้มีความยาว 4 ไบต์ ซึ่งเป็นโค้ดสำหรับตรวจสอบข้อผิดพลาดแบบ CRC (Cyclic Redundancy Check) ของข้อมูลในเฟรม

2.3 โปรโตคอล ทีซีพี/ไอพี

ปัจจุบันโปรโตคอล ทีซีพี/ไอพี เป็นโปรโตคอลที่นิยมใช้ในเครือข่ายมากที่สุด เหตุผลหนึ่งที่โปรโตคอลชุดนี้เป็นที่นิยมมาก เนื่องจากหลายบริษัทที่ผลิตอุปกรณ์หรือซอฟท์แวร์ของเครือข่ายนำมาใช้เป็นมาตรฐาน และอีกอย่างทีซีพีไอพีเป็นโปรโตคอลพื้นฐานของเครือข่ายอินเตอร์เน็ต (Internet) ซึ่งเป็นเครือข่ายที่ใหญ่ที่สุดในโลก และเป็นเครือข่ายที่ทำให้คอมพิวเตอร์กลายเป็นส่วนที่สำคัญในชีวิตประจำวันของเราในปัจจุบัน ดังนั้นทีซีพีไอพีจึงได้กลายโปรโตคอลมาตรฐานที่ใช้ในองค์กรธุรกิจและรัฐบาล

ชุดโปรโตคอลทีซีพี/ไอพีได้ถูกพัฒนามานานแล้วกว่า 30 ปี ซึ่งเริ่มจากการวิจัยที่สนับสนุนโดยกระทรวงกลาโหมสหรัฐฯ จุดประสงค์ของการวิจัยนี้ก็เพื่อเชื่อมต่อคอมพิวเตอร์ที่ต่างแพลตฟอร์มกันให้สามารถสื่อสารกันผ่านเครือข่ายได้ ซึ่งสามารถทำได้โดยการแบ่งโปรโตคอลเป็นชั้นและเป็นการแยกการทำงานของแอพพลิเคชันของผู้ใช้ออกจากฮาร์ดแวร์ที่ใช้รับส่งข้อมูลผ่านเครือข่าย ชุดโปรโตคอลนี้จะมีการจัดรูปแบบที่แตกต่างจากแบบอ้างอิง OSI เล็กน้อย

เครือข่ายคอมพิวเตอร์ในปัจจุบันประกอบด้วยหลากหลายอุปกรณ์ และฮาร์ดแวร์ที่ผลิตโดยบริษัทต่างๆแต่อุปกรณ์เครือข่ายเหล่านี้มีระบบการสื่อสารข้อมูลที่เหมือนกัน เครือข่ายที่ใช้อุปกรณ์จากหลายบริษัทนี้สามารถทำงานร่วมกันได้ เนื่องจากอุปกรณ์แต่ละชิ้นผลิตตามมาตรฐานที่กำหนดโดยองค์กรกลาง โปรโตคอลทีซีพีไอพีเป็นมาตรฐานที่ได้รับความนิยมมากที่สุดสำหรับเครือข่ายในปัจจุบันเนื่องจากหลายเหตุผลดังนี้

- เป็นโปรโตคอลระบบเปิด (Open System) ที่ไม่มีบริษัทใดบริษัทหนึ่งเป็นเจ้าของลิขสิทธิ์และข้อกำหนดของโปรโตคอลจะถูกพัฒนาโดยองค์กรสาธารณะและมีการตีพิมพ์ให้ทราบ
- ทีซีพี/ไอพีถูกออกแบบมาเพื่อให้แพลตฟอร์มต่างกันสามารถสื่อสารกันได้ โปรแกรมบริการต่างๆ เช่น FTP (File Transfer Protocol) และเทลเน็ต เป็นโปรแกรมที่ไม่ขึ้นต่อระบบ เพียงแต่บริษัทนั้นๆ พัฒนาระบบของตัวเองให้สามารถรองรับทีซีพีไอพีได้ ก็สามารถสื่อสารกับระบบอื่นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรโตคอลทีซีพี/ไอพีได้ถูกพิสูจน์แล้วว่าเป็นโปรโตคอลที่แข็งแกร่ง มีประสิทธิภาพสูง และมีความสามารถในการขยายตัวสูง ด้วยการใช้งานในเครือข่ายอินเตอร์เน็ตซึ่งเป็นเครือข่ายที่ใหญ่ที่สุดในโลก
- โปรโตคอลทีซีพี/ไอพีได้กลายเป็นโปรโตคอลมาตรฐานกลางในการสื่อสารข้อมูลของคอมพิวเตอร์เนื่องจากเป็นภาษาที่ใช้ในระบบอินเตอร์เน็ต

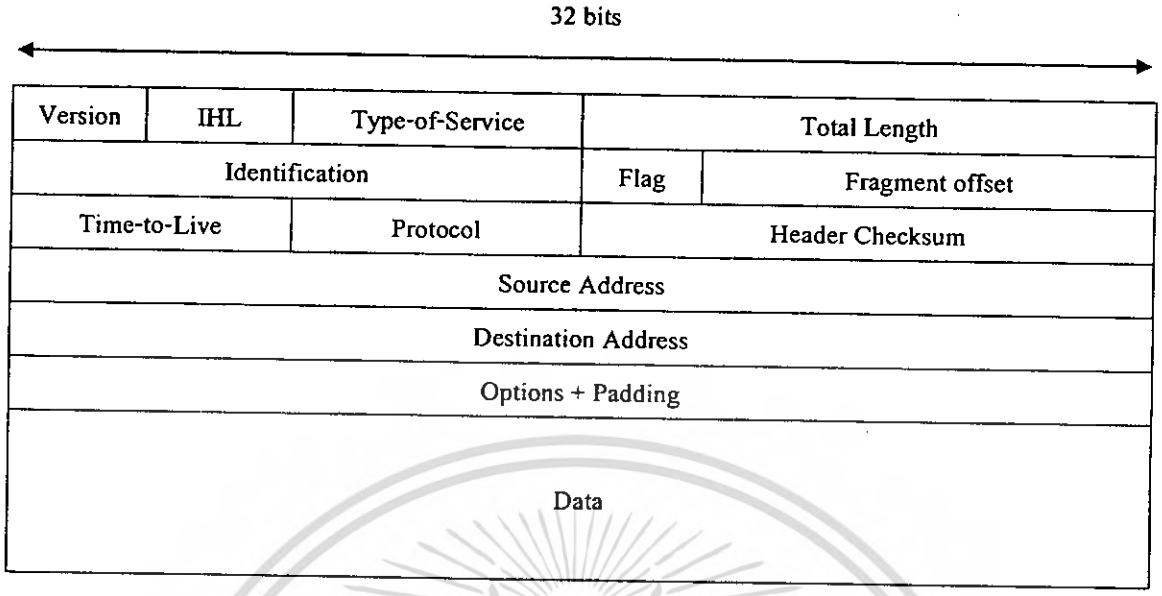
ไม่ว่าจะด้วยเหตุผลอะไรก็แล้วแต่ที่ทำให้โปรโตคอลทีซีพีไอพีเป็นโปรโตคอลสุดยอดนิยมในปัจจุบันแต่สิ่งที่สำคัญที่สุดคือโปรโตคอลนี้ได้กลายเป็นมาตรฐานเครือข่ายๆไปโดยปริยาย ดังนั้นเครือข่ายสมัยใหม่จึงจำเป็นที่ต้องสร้างให้สามารถรองรับโปรโตคอลนี้

2.3.1 โปรโตคอลไอพี (Internet Protocol : IP)

โปรโตคอลไอพี (IP) ทำหน้าที่เสมือนกับที่ทำการไปรษณีย์ กล่าวคือ โปรโตคอลไอพี จะทำหน้าที่จัดการเกี่ยวกับการรับส่งแพ็กเก็ต หรือบางทีก็เรียกว่า "ดาต้าแกรม (Datagram)" คือหน่วยข้อมูลที่รับมาจากโปรโตคอลที่อยู่ในเลเยอร์ที่สูงกว่า เช่น ทีซีพี (TCP) และยูดีพี (UDP) ถ้าโฮสต์ปลายทางอยู่คนละเครือข่ายกับโฮสต์ที่ส่งข้อมูล โปรโตคอลไอพีจะรับผิดชอบในการจัดเส้นทาง (Routing) ให้แพ็กเก็ตส่งไปยังเครือข่ายที่โฮสต์นั้นอยู่ ซึ่งในการจัดส่งแพ็กเก็ตข้ามเครือข่ายนั้น จะใช้เราท์เตอร์ (Router) ในการเชื่อมต่อเครือข่ายเหล่านั้นโดยทั่วไปแล้วอุปกรณ์ที่ทำหน้าที่รับส่งข้อมูลระหว่างเครือข่ายจะเรียกว่า "เราท์เตอร์" แต่บางทีอุปกรณ์ตัวนี้ก็จะเรียกว่า "เกตเวย์ (Gateway)" ซึ่งทำหน้าที่เป็นเสมือนประตูไปยังเครือข่ายอื่นๆ อย่างไรก็ตามทั้งเราท์เตอร์และเกตเวย์เป็นอุปกรณ์ที่ทำหน้าที่ในเลเยอร์ที่ 3 เหมือนกัน

โปรโตคอลไอพี เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันเลสส์ (Connectionless) ซึ่งทำให้มีความน่าเชื่อถือน้อย เนื่องจากไม่มีการสร้างการเชื่อมต่อก่อนที่จะทำการรับส่งข้อมูล กล่าวคือในการส่งข้อมูลแต่ละครั้ง โฮสต์ที่ต้องการส่งข้อมูลจะไม่ทำการติดต่อโฮสต์ปลายทางเพื่อตกลงเกี่ยวกับการรับส่งข้อมูลก่อน แต่โฮสต์ที่ต้องการส่งข้อมูลจะทำการส่งแพ็กเก็ตออกไปทันที โดยที่คาดหวังว่าโฮสต์ปลายทางจะได้รับแพ็กเก็ตนั้นในที่สุด ดังนั้นความน่าเชื่อถือในการส่งข้อมูลจึงมีน้อยเพราะแพ็กเก็ตข้อมูลอาจสูญหายระหว่างทางหรือถ้าข้อมูลประกอบด้วยหลายแพ็กเก็ต แต่ละแพ็กเก็ตอาจเดินทางมาถึงปลายทางไม่เป็นลำดับได้ หรือมีการส่งแพ็กเก็ตซ้ำกันหรือแพ็กเก็ตส่งถึงล่าช้า การแก้ปัญหานี้จะปล่อยให้เป็นหน้าที่ของโปรโตคอลที่อยู่ในเลเยอร์ที่สูงกว่ารับผิดชอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ฟอร์แมตของแพ็กเก็ตไอพี

ฟอร์แมตของแพ็กเก็ตไอพี ประกอบด้วยหลายฟิลด์ดังแสดงในรูปที่ 2.4 ข้อมูลในส่วนหัวของแพ็กเก็ตไอพี มีดังนี้

- Version (4 บิต) : ข้อมูล 4 บิตแรกจะเป็นข้อมูลที่บอกถึงเวอร์ชันของโปรโตคอลทีซีพี/ไอพี ที่ใช้อยู่ ซึ่งในปัจจุบันจะใช้เวอร์ชัน 4 หรือเรียกสั้นๆว่า IPv4 ในอนาคตอันใกล้อาจจะมีการเปลี่ยนแปลงไปใช้เวอร์ชันใหม่คือ เวอร์ชัน 6 หรือ IPv6 เนื่องจากเวอร์ชัน 4 กำลังมีปัญหาเกี่ยวกับหมายเลขไอพีไม่เพียงพอต่อการใช้งาน
- Internet Header Length หรือ IHL (4 บิต) : เป็นตัวเลขที่บอกความยาวของข้อมูลในส่วนหัว (Header)
- Type of Service (8 บิต) : ในแต่ละบิตของข้อมูลส่วนนี้จะเป็นแฟล็ก (Flag) ที่แสดงถึงลำดับความสำคัญ (Precedence), ความล่าช้า (Delay), อัตราการส่งผ่าน (Throughput) และค่ากำหนดความเชื่อถือได้ของแพ็กเก็ตข้อมูลนี้
- Total Length (16 บิต) : ข้อมูลส่วนนี้จะบอกถึงความยาวของแพ็กเก็ตทั้งหมดซึ่งมีหน่วยเป็นไบต์ ซึ่งความยาวของแพ็กเก็ตนี้เป็นไปได้ตั้งแต่ 576 ถึง 65,536 ไบต์
- Identifier (16 บิต) : ถ้าดาต้าแกรมประกอบด้วยหลายแพ็กเก็ต หมายเลขนี้จะถูกกำหนดให้กับแต่ละแพ็กเก็ตย่อย ซึ่งแพ็กเก็ตย่อยแต่ละแพ็กเก็ตจะมีหมายเลขนี้ที่ไม่ซ้ำกันในช่วงเวลาหนึ่งๆ
- Flag (3 บิต) : เป็นฟิลด์ที่ใช้ในการจัดการเกี่ยวกับการแบ่งข้อมูลเป็นแพ็กเก็ตย่อย
- Fragment Offset (13 บิต) : เป็นค่าที่บอกจุดเริ่มต้นในส่วนของข้อมูลย่อย (Fragmented Content) ซึ่งเป็นตัวเลขที่บ่งบอกว่าแพ็กเก็ตย่อยนี้อยู่ห่างจากจุดเริ่มต้นของดาต้าแกรมทั้งหมดเท่าใด โดยจำนวนนี้มีหน่วยวัดเป็น 64 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Time to Live หรือ TTL (8 บิต) : แพ็กเก็ตจะไหลเวียนอยู่ในเครือข่ายได้ในช่วงเวลาหนึ่งเท่านั้น การกำหนดว่าแพ็กเก็ตแต่ละแพ็กเก็ตจะอยู่ได้ในเครือข่ายนานเท่าใดนั้น จะบอกเป็นจำนวนของ Hop หรือจำนวนครั้งที่ผ่านเราท์เตอร์ ทุกครั้งที่ผ่านเราท์เตอร์ค่า TTL จะลดลงทีละหนึ่ง เมื่อค่านี้เป็นศูนย์แพ็กเก็ตนี้ก็จะถูกละทิ้งไป
- Protocol (8 บิต) : เป็นข้อมูลที่บอกโปรโตคอลของชั้นที่เหนือกว่าเช่น ทีซีพี, ยูดีพี เป็นต้น
- Header Checksum (16 บิต) : เป็นข้อมูลส่วนที่ใช้ในการตรวจสอบข้อผิดพลาดในส่วนหัวของแพ็กเก็ต ซึ่งเมื่อผ่านอุปกรณ์เครือข่ายแต่ละครั้งจะทำการเช็คข้อผิดพลาดทุกครั้ง
- Source IP Address (32 บิต) : หมายเลขไอพีของเครื่องที่ส่งข้อมูล
- Destination IP Address (32 บิต) : หมายเลขไอพีของเครื่องปลายทาง
- Padding : เป็นเลข 0 ที่เพิ่มให้กับส่วนหัวของแพ็กเก็ตเพื่อให้ส่วนหัวมีความยาวที่หารด้วย 32 บิตลงตัว หรือเป็นข้อมูลที่เกี่ยวกับฟีเจอร์อื่นๆ เช่น การรักษาความปลอดภัย
- Data : ข้อมูลของโปรโตคอลที่อยู่สูงกว่าซึ่งความยาวจะไม่คงที่

หน้าที่หลักๆของโปรโตคอลไอพี มีอยู่ 3 อย่างดังต่อไปนี้

- การระบุหมายเลข (Addressing)

หน้าที่นี้หมายถึงการให้บริการในการติดตั้ง "ลอจิคัลแอดเดรส (Logical Address)"ให้กับเครื่องคอมพิวเตอร์ต่างๆที่ใช้โปรโตคอลไอพี เนื่องจาก ลอจิคัลแอดเดรสนี้จะไม่ได้ถูกกำหนดมาตายตัวหรือฝังมากับเน็ตเวิร์คการ์ด ดังนั้นมันจึงเป็นแอดเดรสที่ผู้ออกแบบหรือบริหารระบบเครือข่ายเป็นผู้ตั้งขึ้นมาเอง และสามารถเปลี่ยนแปลงแก้ไขได้ตามใจชอบ ข้อดีของการมีลอจิคัลแอดเดรสหรือแอดเดรสในเน็ตเวิร์คเลเยอร์ (Network Layer) ก็คือ

- ทำให้เราสามารถออกแบบระบบเน็ตเวิร์คได้ง่ายขึ้น
- ทำให้ระบบเน็ตเวิร์คสามารถขยายเพิ่มเติมได้โดยง่าย
- ทำให้การแก้ไขปัญหาทำได้โดยง่าย

- การจัดแพ็กเก็ต (Packaging)

เป็นการจัดเตรียมแพ็กเก็ตไอพีให้อยู่ในสภาพที่พร้อมส่งไปยังเครื่องปลายทางโดยการนำเอาทีซีพีเซ็กเมนต์ (TCP Segment) หรือยูดีพีเซ็กเมนต์ (UDP Segment) จากเลเยอร์บนมาบรรจุไว้ในฟิลด์ดาต้า (Data) ของแพ็กเก็ตไอพี หากขนาดของเซ็กเมนต์ใหญ่เกินกว่าจะส่งได้ภายในแพ็กเก็ตไอพี แพ็กเก็ตเดียวมันจะต้องแบ่งเซ็กเมนต์ ซอยย่อยออกและส่งไปในหลายๆแพ็กเก็ต จากนั้นก็ใส่ค่าฟิลด์ Destination Address และ Source Address ให้เป็นหมายเลขไอพีแอดเดรสปลายทางและต้นทางตามลำดับ และที่สำคัญก็คือ มันจะใส่ค่าฟิลด์ Protocol Number ลงไปด้วยตัวเลขค่าหนึ่งที่ระบุว่าเลเยอร์บนเป็นทีซีพีหรือยูดีพี (หมายเลข 6 สำหรับทีซีพีและ หมายเลข 17 สำหรับยูดีพี) แพ็กเก็ตไอพีหนึ่งๆ บางครั้งถูกเรียกว่าดาต้าแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การจัดหาเส้นทาง (Routing)

ความหมายของ Routing ก็คือ การหาเส้นทางในการส่งแพ็กเก็ตไปให้ถึงเครื่องปลายทางให้ได้ หลักสำคัญของการส่งแพ็กเก็ตโดยโปรโตคอลไอพีก็คือ มันจะส่งให้ดีที่สุด (Best Effort) โดยไม่รับประกันว่าข้อมูลจะถึงปลายทางหรือไม่ และจะปล่อยให้เป็นหน้าที่ของโปรโตคอลในระดับสูงกว่า (คือ ทีซีพี) เป็นผู้รับประกันให้

2.3.2 โปรโตคอลทีซีพี (Transmission Control Protocol :TCP)

โปรโตคอลทีซีพี เป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันโอเรียนเต็ด (Connection-Oriented) ซึ่งเป็นการส่งข้อมูลที่เชื่อถือได้ โปรโตคอลทีซีพีจะส่งข้อมูลทั้งหมดจนสำเร็จ ซึ่งถ้าข้อมูลมีขนาดใหญ่ก็จะถูกแบ่งย่อยเป็นหลายแพ็กเก็ต โปรโตคอลทีซีพี จะทำหน้าที่ควบคุมการรับส่งแพ็กเก็ตข้อมูลย่อยๆเหล่านี้ สำหรับกลไกในการควบคุมการไหลของข้อมูลมีรายละเอียดดังนี้

- การจัดการเกี่ยวกับเซสชั่น (Session)

เนื่องจากทีซีพีเป็นโปรโตคอลที่ให้บริการแบบคอนเน็กชันโอเรียนเต็ด ดังนั้นก่อนที่จะมีการส่งข้อมูล จำเป็นที่ต้องสร้างเซสชั่นเพื่อเชื่อมต่อกับโฮสต์ปลายทางก่อน เซสชั่นเป็นการสร้างการสนทนาอย่างเป็นรูปแบบระหว่างทั้งสองโฮสต์เพื่อใช้สำหรับการกู้คืนข้อมูลเมื่อเกิดข้อผิดพลาดระหว่างการรับส่งข้อมูล ขั้นตอนในการสร้างเซสชั่นนี้จะมีอยู่ 3 ขั้นตอนซึ่งบางทีก็เรียกว่า "ทรีเวย์แฮนเช็ค (Three-Way Handshake)"

1. โฮสต์ที่ต้องการส่งข้อมูลจะส่งแพ็กเก็ตไปยังโฮสต์ปลายทางเพื่อแจ้งให้ทราบว่าต้องการส่งข้อมูล
2. โฮสต์ปลายทางก็จะตอบตกลงกลับมาพร้อมทั้งรหัสที่จะใช้ในการรับส่งข้อมูล
3. โฮสต์ต้นทางก็จะส่งแพ็กเก็ตพร้อมรหัสที่ได้รับ เพื่อเป็นการยืนยันการเชื่อมต่อ

หลังจากที่ได้มีการสร้างเซสชั่นสำเร็จแล้วจึงจะเริ่มกระบวนการรับส่งข้อมูลจริงๆ ซึ่งการรับส่งข้อมูลแต่ละครั้งก็จะมีการยืนยันการรับข้อมูลจากโฮสต์ปลายทางทุกครั้ง เมื่อรับส่งข้อมูลเสร็จก็เป็นขั้นตอนการยกเลิกการเซสชั่น ซึ่งจะคล้ายๆกับการสร้างเซสชั่น

- การควบคุมการไหลและการกู้คืนข้อมูล

ในแต่ละเซสชั่น โฮสต์ฝ่ายรับต้องตอบกลับทุกๆแพ็กเก็ตที่ได้รับภายในเวลาที่กำหนด เพื่อเป็นการยืนยันการรับข้อมูลทุกๆแพ็กเก็ตที่ส่ง ฝ่ายรับจะทำการเช็คความถูกต้องของแพ็กเก็ตข้อมูลทุกครั้ง และแจ้งให้ทราบถึงผลการตรวจสอบนั้น ถ้าฝ่ายส่งไม่ได้รับการตอบกลับจากฝ่ายรับภายในเวลาที่กำหนดก็จะคาดเดาว่าแพ็กเก็ตสูญหายระหว่างทาง ฝ่ายส่งก็จะทำการส่งแพ็กเก็ตนั้นใหม่อีกครั้ง เพื่อทำให้มั่นใจได้ว่าข้อมูลทุกๆแพ็กเก็ตส่งถึงปลายทางอย่างสมบูรณ์ นอกจากนี้การแบ่งข้อมูลขนาดใหญ่ออกเป็นแพ็กเก็ตย่อยๆ ทีซีพีก็จะกำหนดหมายเลขลำดับ (Sequence Number) ให้แต่ละแพ็กเก็ต เพื่อใช้สำหรับการจัดรวมแพ็กเก็ตย่อยๆเหล่านั้นให้เป็นข้อมูลเหมือนเดิม นอกจากนี้หมายเลขลำดับยังใช้สำหรับการตรวจสอบว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

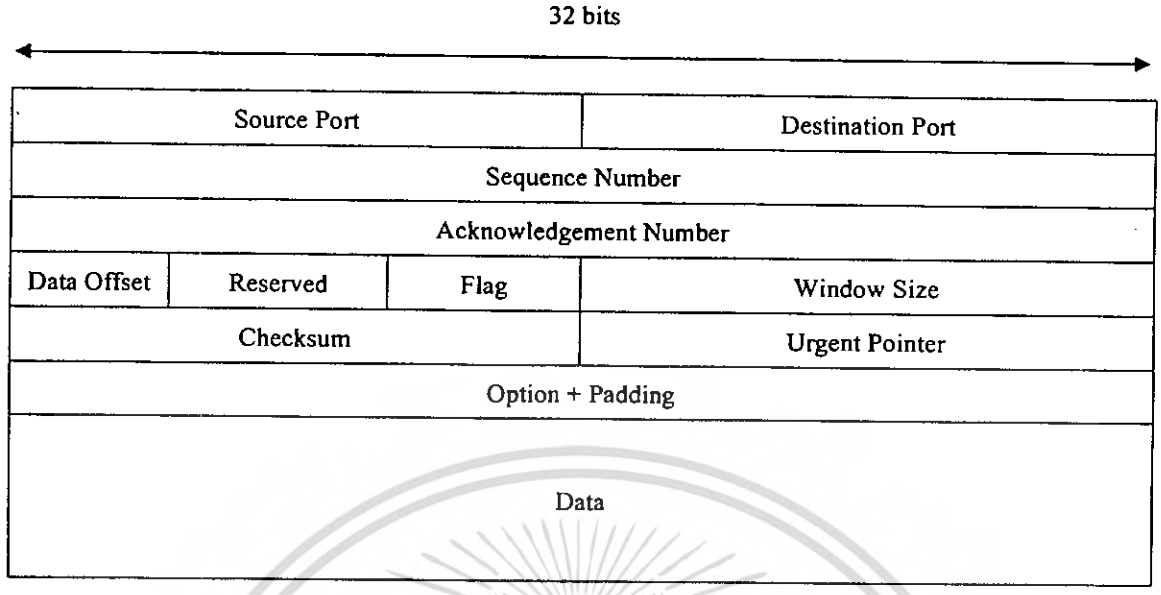
ข้อมูลส่งถึงปลายทางครบทุกแพ็กเก็ตหรือไม่ กลไกการตอบกลับแพ็กเก็ตนั้นมีอยู่ 2 ประเภท ประเภทแรกคือ PAR (Positive Acknowledgement and Retransmission) กลไกการทำงานก็คือ เมื่อฝ่ายส่งทำการส่งแพ็กเก็ตหนึ่งก็จะรอการตอบกลับจากฝ่ายรับ แล้วค่อยส่งแพ็กเก็ตต่อไป ถ้าไม่ได้รับการตอบกลับภายในเวลาที่กำหนดก็จะส่งแพ็กเก็ตนั้นอีกครั้ง ปัญหาของกลไกนี้ก็คือ ถ้าข้อมูลประกอบด้วยหลายๆแพ็กเก็ต และการที่ฝ่ายรับต้องส่งแพ็กเก็ตตอบกลับต่อทุกๆแพ็กเก็ตที่ได้รับนั้นอาจเป็นการสิ้นเปลืองแบนด์วิธ และเป็นกระบวนการที่ไร้ประสิทธิภาพเนื่องจากฝ่ายส่งจะใช้เวลาในการรอมากกว่าการส่งข้อมูล กลไกที่สองจะแก้ปัญหานี้ซึ่งกลไกนี้จะเรียกว่า "สไลดิ้งวินโดว์ (Sliding Window)" กลไกการทำงานคือ ฝ่ายรับสามารถยืนยันการได้รับแพ็กเก็ตโดยส่งแพ็กเก็ตเดียวเพื่อแสดงการยืนยันการได้รับหลายๆแพ็กเก็ต วิธีนี้ช่วยลดจำนวนแพ็กเก็ตที่ต้องไหลเวียนในเครือข่าย และฝ่ายส่งสามารถส่งทีละหลายๆแพ็กเก็ตก่อนที่จะรอการตอบกลับ

เมื่อสร้างเซสชั่นสำเร็จ ขั้นตอนต่อไปคือการต่อรองเกี่ยวกับขนาดของวินโดว์ (Window Size) ขนาดของวินโดว์คือ จำนวนไบต์ที่ฝ่ายรับได้รับก่อนที่จะทำการตอบกลับหรือจำนวนไบต์ที่ฝ่ายส่งสามารถส่งได้ก่อนที่จะรอการตอบกลับ การทำงานของสไลดิ้งวินโดว์มีขั้นตอนดังนี้

1. เมื่อโฮสต์ต้องการที่จะส่งข้อมูล ทีซีพีจะย้ายข้อมูลไปไว้ที่บัฟเฟอร์ที่จะใช้ส่งข้อมูล ซึ่งข้อมูลส่วนนี้จะเรียกว่า "เซ็กเมนต์" ซึ่งแต่ละเซ็กเมนต์อาจจะถูกแบ่งย่อยเป็นหลายแพ็กเก็ตซึ่งแต่ละแพ็กเก็ตก็จะถูกกำหนดหมายเลขลำดับ
2. ทุกๆแพ็กเก็ตในเซ็กเมนต์จะถูกส่งต่อไปให้ส่วนไอพี เพื่อทำการส่งไปยังโฮสต์ปลายทาง
3. เซ็กเมนต์ข้อมูลจะยังคงถูกเก็บไว้ที่บัฟเฟอร์จนกว่าจะได้รับการตอบกลับจากโฮสต์ฝ่ายรับก่อนและโฮสต์ฝ่ายส่งจะตั้งเวลาเพื่อรอการตอบกลับ ถ้าโฮสต์ฝ่ายรับไม่ตอบกลับภายในเวลาที่กำหนดข้อมูลที่อยู่ในบัฟเฟอร์ก็จะถูกส่งใหม่อีกครั้ง
4. เมื่อแพ็กเก็ตเดินทางมาถึงฝ่ายรับ โฮสต์ฝ่ายรับก็จะใช้หมายเลขลำดับในการเรียบเรียงแพ็กเก็ตให้ได้เป็นเซ็กเมนต์เหมือนเดิม
5. เมื่อโฮสต์ฝ่ายรับได้รับแพ็กเก็ตครบและตรวจสอบแล้วว่าไม่มีข้อผิดพลาดใดๆก็จะส่งแพ็กเก็ตตอบกลับไปยังโฮสต์ฝ่ายส่งว่าได้รับข้อมูลครบหมดแล้ว
6. เมื่อโฮสต์ฝ่ายส่งได้รับการตอบกลับ เซ็กเมนต์ในบัฟเฟอร์ก็จะถูกลบทิ้งไปแล้วทำการส่งเซ็กเมนต์ถัดไป จนกว่าข้อมูลจะถูกส่งทั้งหมด

กระบวนการส่งข้อมูลแบบนี้จะทำให้มั่นใจได้ว่าข้อมูลจะส่งถึงปลายทางอย่างแน่นอนและถูกต้อง ซึ่งการให้บริการแบบนี้จะเรียกว่า "คอนเน็กชันโอเรียนเต็ด (Connection-Oriented)" นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ฟอร์แมตข้อมูลของแพ็กเก็ตทีซีพี

ข้อมูลในส่วนหัวของโปรโตคอลทีซีพี จะประกอบด้วยข้อมูลมากที่สุด 20 ไบต์และประกอบด้วยส่วนต่างๆ ดังแสดงในรูปที่ 2.5 ซึ่งแต่ละฟิลด์มีความหมายดังนี้

- TCP Source Port (16 บิต) : ส่วนนี้จะเป็นหมายเลขพอร์ตที่เป็นจุดเริ่มต้นการสื่อสาร หมายเลขพอร์ตเมื่อรวมกับหมายเลขไอพี จะเป็นที่อยู่ของการส่งข้อมูลกลับไป
- TCP Destination Port (16 บิต) : เป็นหมายเลขพอร์ตของเครื่องรับ ซึ่งพอร์ตนี้จะเป็นพอร์ตที่ใช้เชื่อมต่อกับแอพพลิเคชันที่จะนำข้อมูลที่ส่งไปให้นี้ไปประมวลผลต่อไป
- TCP Sequence Number (32 บิต) : เป็นหมายเลขที่บอกลำดับแพ็กเก็ตที่จะใช้โดยฝั่งเครื่องรับในการเรียงข้อมูลให้อยู่ในรูปแบบเดิม ในการส่งข้อมูลผ่านเครือข่ายที่สลับซับซ้อนนั้นแพ็กเก็ตแต่ละชุดอาจจะถูกส่งไปบนเส้นทางที่ต่างกัน ดังนั้นจึงเป็นไปได้ที่แพ็กเก็ตจะเดินทางมาถึงปลายทางไม่เป็นไปตามลำดับที่ส่ง หมายเลขนี้จะใช้ในการจัดเรียงแพ็กเก็ตเหล่านี้ให้อยู่ในลำดับเดิม
- TCP Acknowledgement Number (32 บิต) : เป็นหมายเลขลำดับแพ็กเก็ตถัดไปที่ทางฝั่งรับคาดหวัง ซึ่งเป็นการบอกเป็นนัยว่าแพ็กเก็ตที่มีหมายเลขลำดับก่อนหน้านี้ได้รับหมดแล้วนั่นเอง
- Data Offset (4 บิต) : เป็นตัวเลขที่บอกขนาดของข้อมูลส่วนหัว (TCP Header) ซึ่งมีหน่วยเป็น 32 บิต หรือ word
- Reserved (6 บิต) : ส่วนนี้จะถูกกำหนดให้เป็นศูนย์ตลอด ซึ่งข้อมูลส่วนนี้ไม่มีความหมายอะไร เพียงแต่เป็นการสงวนไว้ใช้ในอนาคตเมื่อมีการปรับปรุงโปรโตคอล
- Flag (6 บิต) : เป็นข้อมูลที่ใช้สำหรับควบคุมการรับส่งแพ็กเก็ต เช่น บิต SYN และ ACK ใช้สำหรับการสร้างการเชื่อมต่อ ส่วนบิต FIN เป็นการแจ้งการยกเลิกการเชื่อมต่อ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Window Size (16 บิต) : เป็นตัวเลขที่เครื่องปลายทางบอกให้เครื่องต้นทางทราบขนาดของวินโดวส์ที่เครื่องปลายทางสามารถรับข้อมูลได้
- Checksum (16 บิต) : เป็นข้อมูลที่ใช้ในการตรวจสอบข้อผิดพลาดของข้อมูลในส่วนหัว โดยเครื่องส่งจะทำการคำนวณค่าเช็คซัม (Checksum) ของข้อมูลส่วนหัว เมื่อเครื่องปลายทางได้รับข้อมูลก็จะทำการคำนวณเช็คซัมด้วยวิธีเดียวกัน แล้วทำการเปรียบเทียบค่าที่คำนวณได้กับค่าที่อยู่ในฟิลด์นี้ ถ้าเหมือนกันแสดงว่าไม่มีข้อผิดพลาดในข้อมูลที่ได้รับ
- Padding : เป็นข้อมูลส่วนเพิ่มเติมเพื่อให้ข้อมูลส่วนหัวมีจำนวนบิตที่หารด้วย 32 ลงตัว

2.3.3 โปรโตคอล ยูดีพี (User Datagram Protocol :UDP)

โปรโตคอลยูดีพี มีคุณสมบัติที่แตกต่างจาก ทีซีพี โดยที่การรับส่งข้อมูลผ่านโปรโตคอล ยูดีพีจะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล ทีซีพี และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลนั้นๆด้วยเนื่องจากโปรโตคอล ยูดีพี ไม่มีสัญญาณสอบทานข้อมูล (acknowledgement) ในการส่งข้อมูลแต่ละครั้ง และไม่มีการส่งข้อมูลใหม่อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอพพลิเคชั่นหรือโปรเซสใดที่ต้องอาศัยโปรโตคอล ยูดีพี ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างขบวนการตรวจสอบข้อมูลขึ้นมาเอง การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูลที่มีการรับส่งจะไปทำในขั้นตอนของโปรโตคอลชั้นที่สูงกว่าแทน

Source Port	Destination Port
length	Checksum
Data	

รูปที่ 2.6 ฟอร์แมตข้อมูลของแพ็กเก็ตยูดีพี

- UDP Source Port (16 บิต) : ส่วนนี้จะเป็นหมายเลขพอร์ตที่เป็นจุดเริ่มต้นการสื่อสาร หมายเลขพอร์ตเมื่อรวมกับหมายเลขไอพี จะเป็นที่อยู่ของการส่งข้อมูลกลับไป
- UDP Destination Port (16 บิต) : เป็นหมายเลขพอร์ตของเครื่องรับ ซึ่งพอร์ตนี้จะเป็นพอร์ตที่ใช้เชื่อมต่อกับแอพพลิเคชันที่จะนำข้อมูลที่ส่งไปให้นี้ไปประมวลผลต่อไป
- UDP length (16 บิต) : บอกถึงความยาวของดาต้าแกรม ทั้งส่วนหัว (Header) และ ข้อมูล (data) นั่นหมายความว่า ค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของส่วนหัว (Header)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Checksum (16 บิต) : เป็นตัวตรวจสอบความถูกต้องของยูดีพี ดาตาแกรม (UDP datagram) และจะนำข้อมูลบางส่วนในไอพี เฮดเดอร์ (IP Header) มาคำนวณด้วย

จากรูปที่ 2.6 รูปแบบของ ยูดีพีแพ็คเก็ต จะมีฟิลด์ข้อมูลส่วน เฮดเดอร์น้อยมากและไม่มีข้อมูล ส่วนการตรวจสอบข้อมูล ทำให้ ยูดีพี แพ็คเก็ต มีขนาดเล็ก และใช้หน่วยความจำหรือทรัพยากรของระบบน้อย

- หมายเลขพอร์ต

คอมพิวเตอร์ที่ใช้โปรโตคอลทีซีพี/ไอพี ส่วนใหญ่จะมีแอพพลิเคชันหลายตัวที่ใช้โปรโตคอลทีซีพี/ไอพีในการสื่อสารกับเครื่องอื่น ซึ่งโปรโตคอลทีซีพี/ไอพี จะจัดส่งข้อมูลไปยังแอพพลิเคชันที่เหมาะสม เพื่อให้โปรโตคอลทีซีพี/ไอพีสามารถรองรับแอพพลิเคชันหลายแอพพลิเคชันในเครื่องเดียว จึงมีการใช้พอร์ตและซ็อกเก็ต (Socket) เพื่อช่วยในการแยกแยะแอพพลิเคชันต่างๆ

แอพพลิเคชันแต่ละตัวที่จะรับส่งข้อมูลผ่านเครือข่ายจะใช้หมายเลขพอร์ตตั้งแต่ 0 ถึง 65,535 ดังนั้นเพื่อให้การรับส่งข้อมูลถูกต้อง แอพพลิเคชันที่ทำงานในเครื่องเดียวกันจะต้องใช้หมายเลขพอร์ตที่ต่างกันเพื่อช่วยลดความสับสน แอพพลิเคชันที่นิยมใช้กันทั่วไปส่วนใหญ่จะถูกกำหนดให้ใช้หมายเลขพอร์ตใดพอร์ตหนึ่ง ซึ่งองค์กรที่ทำหน้าที่กำหนดหมายเลขนี้คือ IANA (Internet Assigned Numbers Authority) หมายเลขพอร์ตเหล่านี้จะถูกตีพิมพ์ใน RFC 1700 ซึ่งพอร์ตสำหรับแอพพลิเคชันที่นิยมใช้ทั่วไปได้แสดงในตารางที่ 2.1

ตารางที่ 2.1 หมายเลขพอร์ตของบางแอพพลิเคชัน

พอร์ต	โปรโตคอล	แอพพลิเคชัน
20	TCP	FTP (Data)
21	TCP	FTP (Control)
23	TCP	Telnet
25	TCP	SMTP (Email)
53	TCP/ UDP	DNS (Domain Name Server)
80	TCP	HTTP (Web Server)
110	TCP	POP3 (Email)
161	UDP	SNMP (Simple Network Management Protocol)

โปรโตคอลจะแยกแยะแอพพลิเคชันที่ทำงานในแต่ละโฮสต์โดยใช้ข้อมูล 3 ส่วนต่อไปนี้

1. หมายเลขไอพีของโฮสต์นั้น
2. ประเภทของโปรโตคอลในชั้นทรานสปอร์ตเลเยอร์ (ทีซีพีหรือยูดีพี)
3. หมายเลขพอร์ตที่แอพพลิเคชันนั้นใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
72854

ตารางที่ 2.1 แสดงหมายเลขพอร์ตที่ถูกใช้โดยแอพพลิเคชันที่เป็นที่รู้จักกันโดยทั่วไป ส่วนเครื่องไคลเอนท์ (Client) ที่เชื่อมต่อกับเซิร์ฟเวอร์ (Server) จะใช้หมายเลขพอร์ตที่ต่างจากเซิร์ฟเวอร์ ซึ่งหมายเลขพอร์ตที่ใช้บนเครื่องไคลเอนท์จะถูกจัดการโดยระบบปฏิบัติการของเครื่องนั้น กล่าวคือถ้าไคลเอนท์จะทำการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ หมายเลขพอร์ตที่เว็บเซิร์ฟเวอร์ใช้ก็คือ พอร์ต 80 แต่เครื่องไคลเอนท์จะใช้หมายเลขพอร์ตอื่นที่ว่างอยู่

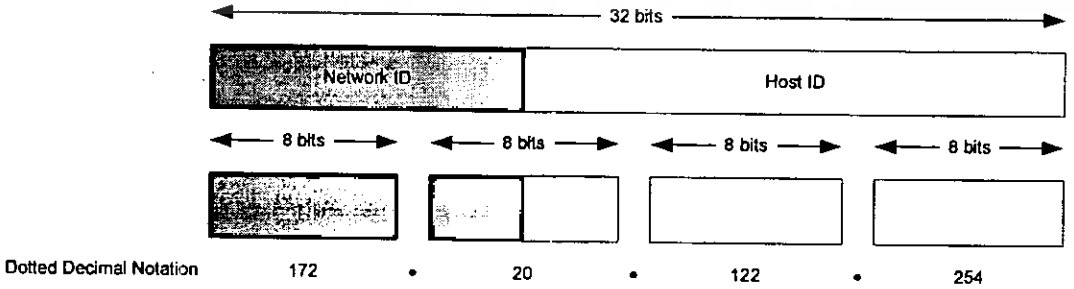
- พอร์ตที่นิยมใช้งาน (Well Known Port)

คำว่า "Well Known Port" จะหมายถึงหมายเลขพอร์ตที่ทางองค์กรกลางที่ชื่อ IETF (Internet Engineering Task Force) เป็นผู้กำหนดให้กับเซอร์วิสที่รู้จักกันดีและนิยมใช้กันแพร่หลายในเครือข่ายอินทราเน็ต (Intranet) และอินเตอร์เน็ต ค่าของ Well Known Port จะอยู่ระหว่าง 1 ถึง 1,024 ดังนั้นแอพพลิเคชันหรือเซอร์วิสอื่นที่เขียนขึ้นมาทำงานบนโปรโตคอลทีซีพี/ไอพี จึงสามารถตั้งค่าพอร์ตของตนเองได้ เป็นค่าที่มากกว่า 1,024 ขึ้นไป ตัวอย่างของ Well Known Port เช่น พอร์ตหมายเลข 80 เป็นของ HTTP พอร์ตหมายเลข 25 เป็นของ SMTP เป็นต้น หากต้องการทราบค่าของ Well Known Port ต่างๆที่นิยมใช้กัน สามารถเปิดดูได้จากไฟล์ชื่อ SERVICES สำหรับระบบปฏิบัติการวินโดวส์ NT 4.0, 2000, XP และ 2003 ไฟล์ SERVICES จะอยู่ที่โฟลเดอร์ C:\<winnt_root>\system32\driver\etc

2.4 หมายเลขไอพี (IP Addressing)

หมายเลขไอพี (IP Address) คือเลขที่บอกที่อยู่ของโหนดหรือโฮสต์ที่อยู่ในเครือข่าย รวมถึงคอมพิวเตอร์และเราท์เตอร์ ที่อยู่บนระบบเครือข่าย หมายเลขนี้จะเป็นที่อยู่ในเลเยอร์ที่ 3 หรือเน็ตเวิร์คเลเยอร์ หมายเลขไอพีของแต่ละเครื่องที่อยู่ในเครือข่ายเดียวกันจะต้องไม่ซ้ำกัน

ปัจจุบันโปรโตคอลไอพีที่ใช้งานอยู่ในเครือข่ายอินเตอร์เน็ตจะเป็นเวอร์ชัน 4 หรือเรียกสั้นๆว่า IPv4 ซึ่งในเวอร์ชันนี้หมายเลขไอพีจะมีขนาด 32 บิต ซึ่งเป็นตัวเลขที่ยาวและยากต่อการจดจำ ดังนั้นเพื่อเป็นการง่ายหมายเลขไอพีจึงนิยมเขียนให้อยู่ในรูปแบบด็อตเดซิมอล (Dotted Decimal Notation) การเขียนให้อยู่ในรูปแบบนี้จะทำได้โดยการจัดกลุ่มเลขฐานสองเป็น 4 กลุ่ม กลุ่มละ 8 บิต หลังจากนั้นให้แปลงเลขฐานสองแต่ละกลุ่มให้เป็นเลขฐานสิบแล้วเชื่อมเลขทั้งสี่กลุ่มด้วยจุด เนื่องจากหมายเลขไอพีที่เป็นฐานสิบนี้เป็นการแปลงมาจากเลขฐานสอง 8 บิต ดังนั้นเลขฐานสิบแต่ละส่วนจะต้องอยู่ระหว่าง 0 ถึง 255 เพราะฉะนั้นหมายเลขไอพีที่ถูกต้องจะอยู่ระหว่าง 0.0.0.0 ถึง 255.255.255.255



รูปที่ 2.7 การแบ่งส่วนของหมายเลขไอพีและด็อตเดซิมอลโนเตชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 ประเภทของหมายเลขไอพี

IPv4 ที่ใช้อยู่ในปัจจุบันจะแบ่งหมายเลขไอพีออกเป็น 5 ประเภท (Class) คือ A, B, C, D และ E โดยหมายเลขไอพีทั้ง 32 บิต จะถูกจัดให้เป็น 2 กลุ่มดังนี้คือ กลุ่มแรกจะเป็นตัวเลขที่ใช้บอกหมายเลขเครือข่าย (Network ID) และกลุ่มที่สองจะเป็นตัวเลขที่ใช้บอกหมายเลขโฮสต์ (Host ID) ที่อยู่ในเครือข่าย ข้อกำหนดที่ใช้ในการแบ่งประเภทของหมายเลขไอพีมีดังนี้

- Class A : บิตแรกจะเป็นเลข 0 เท่านั้นและส่วนที่บอกหมายเลขเครือข่าย คือ 8 บิตแรก ดังนั้นจะมีได้ทั้งหมด 126 เครือข่าย(หมายเลขเครือข่าย 0 จะไม่ใช้) ส่วนอีก 24 บิตที่เหลือจะเป็นเลขที่ใช้บอกหมายเลขโฮสต์ ดังนั้นในแต่ละเครือข่ายจะมีโฮสต์ได้ทั้งหมด 16,777,124 เครื่อง (หมายเลข 0.0.0 และ 255.255.255 จะไม่ใช้) เนื่องจากเครือข่ายมีจำนวนน้อยมากเมื่อเทียบกับจำนวนโฮสต์ฉะนั้นหมายเลขไอพีประเภทนี้จึงไม่เหมาะสำหรับเครือข่ายขนาดใหญ่ ซึ่งประกอบด้วยหลายเครือข่ายเชื่อมต่อกัน เพราะในการส่งข้อมูลระหว่างเครือข่ายนั้นเราท์เตอร์จะใช้เฉพาะหมายเลขเครือข่ายเท่านั้น
- Class B : สองบิตแรกจะเป็น 10 เท่านั้น ส่วนหมายเลขเครือข่ายจะใช้ 16 บิตแรก ดังนั้นจะมีจำนวนเครือข่ายได้ทั้งหมด 16,382 เครือข่าย ส่วนอีก 16 บิตที่เหลือจะเป็นหมายเลขโฮสต์ ซึ่งจะทำให้แต่ละเครือข่ายมีโฮสต์ได้ทั้งหมด 65,534 เครื่อง
- Class C : มีบิตเริ่มต้นเป็น 110 ส่วนหมายเลขเครือข่ายจะใช้ 24 บิตแรก ซึ่งจะได้ทั้งหมด 2,097,152 เครือข่าย ส่วน 8 บิตสุดท้ายเป็นหมายเลขโฮสต์ ซึ่งมีทั้งหมด 254 เครื่อง
- Class D : เลขไอพีเริ่มต้นด้วย 1110 ซึ่งจะเป็นเลขไอพีที่ใช้สำหรับการมัลติคาสต์ หรือสำหรับการส่งข้อมูลแบบมีโฮสต์ปลายทางหลายเครื่อง แต่อาจจะอยู่คนละเครือข่ายกัน
- ประเภทสุดท้ายคือเลขไอพีที่เริ่มต้นด้วย 11110 เป็นหมายเลขไอพีที่สงวนไว้ใช้ในอนาคต หมายเลขเหล่านี้จะถูกกำหนดให้โดยศูนย์ข้อมูลเครือข่าย หรือ InterNIC (Internet Network Information Center)

จะเห็นได้ว่าในหมายเลขไอพีแต่ละประเภทจะมีหลายหมายเลขที่สงวนไว้ใช้สำหรับกรณีพิเศษ เช่น หมายเลข 0.0.0.0 จะใช้โดยโฮสต์ในขณะที่เริ่มเปิดเครื่อง เลขไอพีที่มีหมายเลขเครือข่ายเป็น 0 ทั้งหมดจะใช้อ้างอิงถึงเครือข่ายที่โฮสต์นั้นอยู่ เช่น ถ้าโฮสต์หนึ่งมีเลขไอพีเป็น 172.20.1.24 ซึ่งจะจัดอยู่ในประเภท B โดยมีหมายเลขเครือข่ายเป็น 172.20 เมื่อโฮสต์นี้อ้างถึงหมายเลขไอพี 0.0.1.32 จะมีความหมายเช่นเดียวกับหมายเลข 172.20.1.32 เป็นต้น ส่วนหมายเลขไอพีที่ประกอบด้วยเลข 1 ทั้ง 32 บิตนั้นจะถูกใช้สำหรับการส่งข้อมูลแบบแพร่กระจาย (Broadcast) ในเครือข่ายนั้นๆ ส่วนเลขไอพีที่มีหมายเลขโฮสต์เป็นหนึ่งทั้งหมดนั้น จะใช้สำหรับการส่งข้อมูลแบบแพร่กระจายไปยังเครือข่ายนั้นๆ เช่นเลขไอพี 172.20.255.255 เป็นเลขที่ใช้สำหรับการส่งข้อมูลแบบแพร่กระจายไปยังโฮสต์ทุกเครื่องที่อยู่ในเครือข่าย 172.20 เป็นต้น ส่วนเลขหมาย 127.xxx.yyy.zzz โดยเลข xxx, yyy และ zzz จะเป็นเลขอะไรก็ได้ (แต่โดยทั่วไปจะนิยมใช้เป็นตัวเลข 127.0.0.1) นั้น จะใช้สำหรับการส่งข้อมูลไปยังตัวเอง (Loopback) ซึ่งข้อมูลจะไม่ถูกส่งออกไปในเครือข่าย แต่โฮสต์นั้นจะถือเสมือนว่าข้อมูลนั้นได้รับผ่านเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 อินเตอร์เน็ตส่วนบุคคล/อินเตอร์เน็ตสาธารณะ (Private/Public Internet)

การที่จะเชื่อมต่อคอมพิวเตอร์เข้ากับอินเตอร์เน็ตนั้นจำเป็นที่จะต้องร้องขอหมายเลขไอพีจาก InterNIC ซึ่งเป็นองค์กรที่รับผิดชอบเกี่ยวกับการแจกจ่ายหมายเลขไอพี และเป็นที่ทราบกันดีแล้วว่าหมายเลขไอพีของแต่ละเครื่องที่อยู่ในเครือข่ายเดียวกันจะต้องไม่ซ้ำกัน ดังนั้นถ้าเป็นเครือข่ายส่วนบุคคลที่ไม่มีการเชื่อมต่อเข้ากับอินเตอร์เน็ต ก็สามารถใช้หมายเลขไอพีอะไรก็ได้ และไม่ต้องร้องขอหมายเลขไอพีจาก InterNIC เพียงแค่กำหนดให้หมายเลขไอพีของโฮสต์ในเครือข่ายไม่ซ้ำกันก็พอ

อย่างไรก็ตาม เมื่อมีการเชื่อมต่อเครือข่ายส่วนบุคคลเข้ากับอินเตอร์เน็ต อาจจะทำให้หมายเลขไอพีที่ใช้ในเครือข่ายส่วนบุคคลไปซ้ำกับโฮสต์ที่อยู่ในอินเตอร์เน็ตแล้วก็ได้ ดังนั้นเพื่อป้องกันปัญหาดังกล่าว องค์กร IETF (Internet Engineering Task Force) ได้กำหนดหมายเลขไอพีบางกลุ่มให้เป็นหมายเลขไอพีส่วนบุคคล ดังแสดงในตารางที่ 2.2

ตารางที่ 2.2 หมายเลขไอพีเครือข่ายส่วนบุคคล

ประเภท	ไอพีต่ำสุด	ไอพีสูงสุด
A	10.0.0.0	10.255.255.255
B	172.16.0.0	172.31.255.255
C	192.168.0.0	192.168.255.255

หมายเลขที่แสดงในตารางที่ 2.2 จะเป็นหมายเลขที่ไม่ใช้ในเครือข่ายอินเตอร์เน็ต กล่าวคือแพ็กเก็ตที่มีหมายเลขไอพีนี้จะไม่ถูกส่งต่อโดยเราท์เตอร์ของระบบอินเตอร์เน็ต ส่วนหมายเลขไอพีที่แจกจ่ายไปแล้ว สามารถดูรายละเอียดได้ที่ http://www.iana.org/assignments/ipv4-address-space

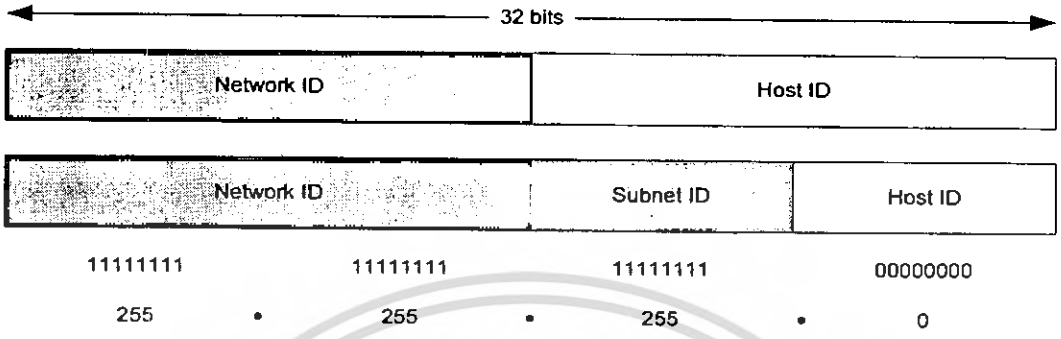
2.4.3 ซับเน็ตและซับเน็ตมาส์ก (Subnet and Subnet Mask)

เนื่องจากอินเตอร์เน็ตเติบโตเร็วมาก ทำให้มีปัญหาเกี่ยวกับการใช้หมายเลขไอพี เพราะในไม่ช้าหมายเลขไอพีที่มีทั้งหมด 32 บิต อาจถูกใช้หมดได้ นั่นคือต้องมีวิธีการใหม่ที่จะกำหนดหมายเลขไอพีให้กับโฮสต์ วิธีหนึ่งที่ใช้แก้ปัญหาก็คือ เพิ่มจำนวนบิตให้มีมากกว่า 32 บิต ซึ่งในขณะนี้กำลังมีการพัฒนาโปรโตคอลไอพีเวอร์ชันใหม่คือ IPv6 ซึ่งในเวอร์ชันนี้จะกำหนดให้หมายเลขไอพีมีขนาด 128 บิต และเมื่อมีการประกาศใช้แล้วจำนวนหมายเลขไอพีจะมีเพียงพออย่างแน่นอน แต่การที่จะอัพเกรดอินเตอร์เน็ตให้ใช้โปรโตคอลเวอร์ชันนี้ได้นั้นคงต้องใช้เวลาอีกนานพอสมควร เพราะเป็นการเปลี่ยนแปลงที่ใหญ่มาก

การแบ่งซับเน็ต จะเป็นอีกวิธีหนึ่งที่ช่วยให้การใช้หมายเลขไอพีมีความคุ้มค่ามากยิ่งขึ้น เนื่องจากหมายเลขไอพีบางประเภท เช่น คลาส B จะมีโฮสต์ได้ทั้งหมด 65,534 เครื่องซึ่งโดยส่วนใหญ่แล้วจะเป็นเครือข่ายที่ใหญ่เกินไปสำหรับองค์กรทั่วไป ดังนั้นจึงทำให้หมายเลขไอพีบางส่วนไม่ถูกใช้งาน การทำซับเน็ตเป็นการแก้ปัญหานี้ได้ โดยการแบ่งเป็นเครือข่ายย่อย เราท์เตอร์จะใช้เฉพาะส่วนที่เป็นหมายเลขเครือข่ายเท่านั้นในการจัดเส้นทางหรือเราท์ดิ้ง สำหรับบริษัทใหญ่ๆที่ต้องการมีเครือข่ายย่อยจะมีวิธีที่ใช้แบ่งเครือข่ายใหญ่ให้เป็นเครือข่ายย่อยโดยวิธีที่เรียกว่า "ซับเน็ต (Subnet)" การแบ่งเป็นเครือข่ายย่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือซับเน็ตนั้นจะถูกจัดการโดยเราท์เตอร์หรือเกตเวย์ของเครือข่ายนั้นๆเท่านั้น โดยที่โฮสต์อื่นๆที่อยู่นอกเครือข่ายนี้จะมองเห็นเครือข่ายนี้เป็นเครือข่ายเดียวเท่านั้น



รูปที่ 2.8 การทำซับเน็ต

การแบ่งเครือข่ายที่ใหญ่ให้เป็นเครือข่ายย่อย นั้นทำได้โดยการแบ่งเลขไอพีส่วนที่เป็นหมายเลขของโฮสต์มาเป็นหมายเลขของเครือข่าย ดังแสดงในรูปที่ 2.7 ส่วนซับเน็ตมาส์คคือตัวเลขที่บ่งชี้ว่าส่วนไหนของเลขไอพีเป็นหมายเลขเครือข่าย และส่วนไหนเป็นหมายเลขโฮสต์ ซับเน็ตมาส์คจะมีความยาวเท่ากับหมายเลขไอพีคือ 32 บิต ซึ่งจะเริ่มต้นด้วยแถวของเลข 1 เรียงกันและตามด้วยแถวของเลข 0 การคำนวณหมายเลขเครือข่ายย่อยจะทำได้โดยการแอนด์ (AND) ระหว่างเลขซับเน็ตมาส์คและหมายเลขไอพีโดยดีฟอลต์ (Default) แล้วเน็ตมาส์คของหมายเลขไอพีแต่ละคลาสแสดงในตารางที่ 2.3

ตารางที่ 2.3 ดีฟอลต์เน็ตมาส์คของไอพีแต่ละประเภท

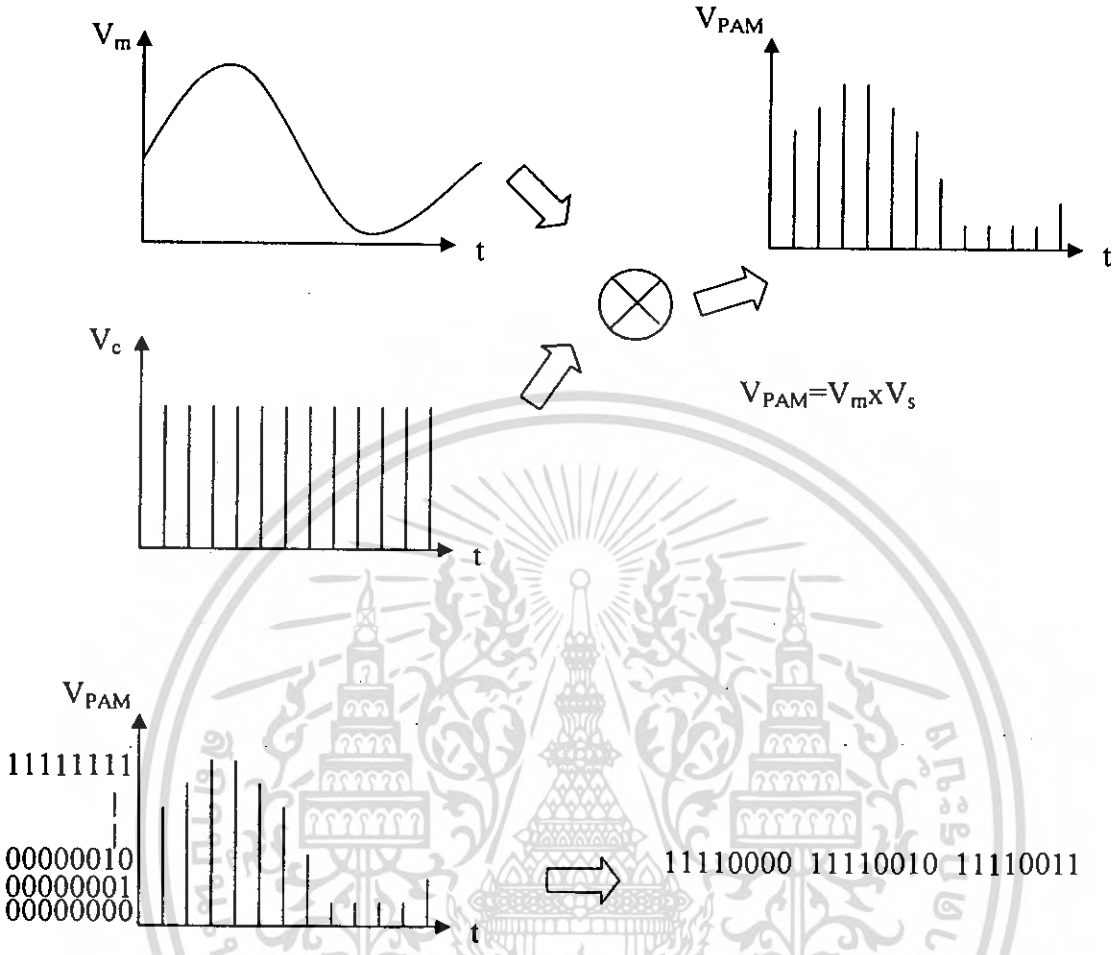
คลาส	เน็ตมาส์ค	ด็อตเดซิมอล
A	11111111 00000000 00000000 00000000	255.0.0.0
B	11111111 11111111 00000000 00000000	255.255.0.0
C	11111111 11111111 11111111 00000000	255.255.255.0

2.5 การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

- หลักการของการแปลงสัญญาณอนาลอกเป็นดิจิตอล

วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล มีหลักการทำงานคือ จะมีส่วนของการมอดูเลชั่น (Modulation) แบบพัลส์แอมปลิจูดมอดูเลชั่น (Pulse Amplitude Modulation : PAM) เป็นตัวเริ่มต้น และมีค่าของเอาท์พุทที่เป็น PAM แล้วนั้น ไปทำการจัดระดับเป็นความหมายทางดิจิตอลควอนไทเซชั่น (Quantization) เป็นรหัสไบนารี ซึ่งความละเอียดจะขึ้นอยู่กับจำนวนบิตที่ Quantization ออกมา โดยเรียกวิธีการนี้ว่าพัลส์โคดมอดูเลชั่น (Pulse Code Modulation : PCM) โดยสามารถอธิบายเป็นดังรูป2.9ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



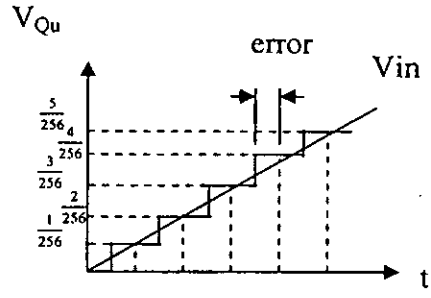
รูปที่ 2.9 การเข้ารหัสแบบ PCM

ในการ Quantization จะมีวิธีการพูดจะสรุปได้คือ

1. จัดระดับของแรงดันเอาพุตที่ได้จากการสุ่มตัวอย่าง (Sampling) ให้อยู่ในรูปของระดับจำนวน 2^n ระดับดังรูปที่ 2.9
2. นำแรงดันระดับต่างๆมาแปลงเป็นสัญญาณดิจิตอลจำนวน n บิตดังรูปที่ 2.9

การ Quantization มีข้อเสียอยู่เหมือนกันกล่าวคือ เมื่อสัญญาณอินพุตปกติเข้ามา เมื่อผ่านการ PAM จะทำให้สัญญาณเอาท์พุตมีลักษณะไม่ต่อเนื่อง (Discrete Signal) ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริงจะดูเหมือนกับขั้นบันไดดังรูปที่ 2.10 ซึ่งขั้นบันไดเหล่านี้กว้างมากหรือน้อยขึ้นอยู่กับจำนวนบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การเกิด Quantization Error

ในขั้นตอนการสุ่มตัวอย่าง สัญญาณเสียงซึ่งอยู่ในรูปของสัญญาณอนาลอกจะถูกสุ่มค่าอย่างต่อเนื่องสม่ำเสมอด้วยอัตราคงที่ค่าหนึ่ง ซึ่งตามทฤษฎีบทของไนควิสต์แล้วจะต้องทำการสุ่มด้วยอัตราที่มีค่าอย่างน้อยเป็น 2 เท่าของความถี่สูงสุดของสัญญาณเสียงเพื่อหลีกเลี่ยงปัญหาของเอเลียสซิง(aliasing) คือ การซ้อนทับกันของสเปคตรัมของสัญญาณ เนื่องจากเสียงมีองค์ประกอบความถี่อยู่ระหว่าง 300-3400 เฮิร์ตซ์ ฉะนั้นอัตราการสุ่มมาตรฐานที่เลือกใช้กันทั่วไปมีค่าเท่ากับ 8 กิโลเฮิร์ตซ์

ขั้นตอนการทำควอนไทเซชั่นส่งผลกระทบต่อคุณภาพของสัญญาณเสียงที่ได้ กล่าวคือ สัญญาณเสียงหลังจากผ่านกระบวนการควอนไทเซชั่นมีโอกาสผิดเพี้ยนไปจากสัญญาณเสียงต้นฉบับได้ เพราะสัญญาณที่มีค่าตกอยู่ช่วงใดช่วงหนึ่งได้ถูกแทนด้วยระดับของสัญญาณที่มีค่าแรงดันค่ากลางของช่วงดังกล่าว และแน่นอนว่าความผิดเพี้ยนจะทวีความรุนแรงมากขึ้นถ้าการแบ่งระดับของการควอนไทซ์มีจำนวนไม่มากนัก

2.6 การแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

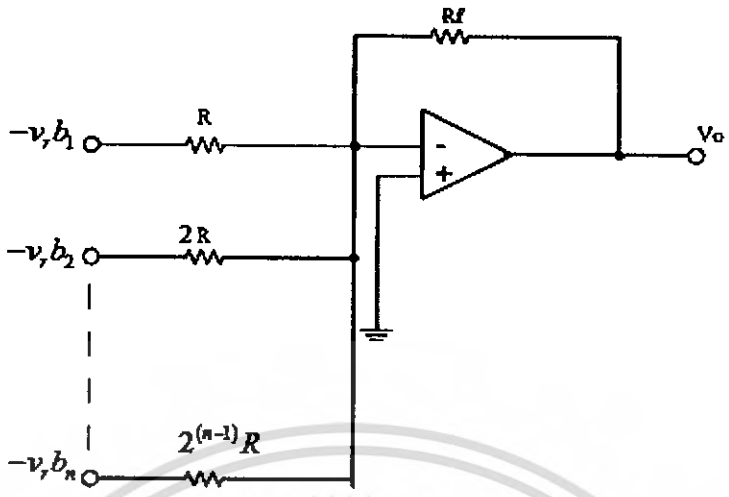
- หลักการของการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกเป็นวงจรที่ทำงานควบคู่กันกับวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล ซึ่งมีหลักการง่ายๆ อยู่ 2 หลักการคือลิเนียร์ คอมบิเนชั่น (Linear Combination) และอาร์-ทูอาร์ แลดเดอร์ (R-2R Ladder) โดยมีรายละเอียดดังต่อไปนี้

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ Linear Combination

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ Linear Combination นี้จะใช้หลักการของวงจรซัมมิ่งแอมปลิไฟเออร์ (Summing Amplifier) โดยการแบ่งค่าความต้านทานอินพุททั้งหลายให้มีค่าต่างกันแบบเชิงเส้นดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ Linear Combination

จากรูปที่ 2.11 สามารถพิจารณาวงจรเป็นแบบอินเวอร์ทติง ซัมมิ่งแอมปลิไฟเออร์ (Inverting Summing Amplifier) ได้โดยแยกพิจารณาในกรณีแต่ละบิทใดๆ (b_i) เป็น "1" อัตราขยายของวงจรจะเท่ากับ $-R_f/R$ ตั้งแต่บิตแรก (b_1) จนถึงสุดท้าย (b_n)

$$v_o = \frac{2R_f v_r}{R} \sum_{i=1}^{n} b_i 2^{-i}$$

ค่าของแรงดันเต็มสเกล (Voltage Full Scale : V_{fs}) จากสมการทำได้โดยกำหนดให้ทุกบิตมีค่าเป็น "1" ดังนั้น ค่า V_{fs} จะมีค่าเท่ากับ

$$v_{fs} = \frac{2R_f v_r}{R}$$

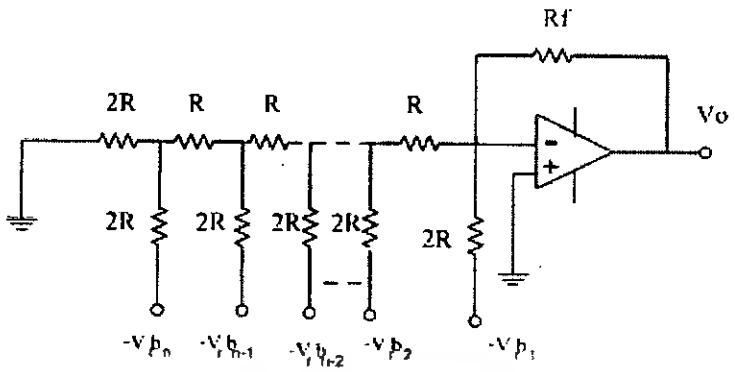
โดยที่ b_1 คือบิตที่นัยสำคัญสูงสุด (Most Significant Bit : MSB)

โดยที่ b_n คือบิตที่นัยสำคัญต่ำสุด (Least Significant Bit : LSB)

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ R-2R Ladder

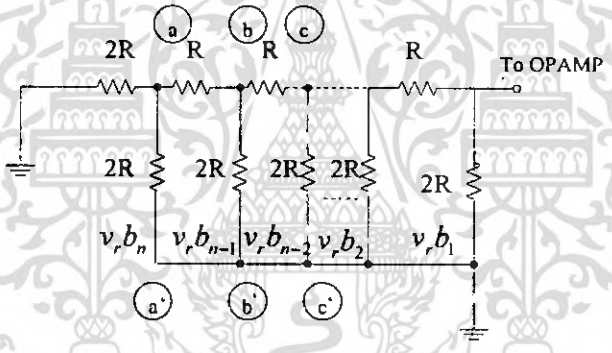
วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ R-2R นี้จะใช้หลักการของวงจร Summing Amplifier โดยการแบ่งค่าความต้านทานอินพุตทั้งหลายเป็น Ladder ให้มีค่าเป็น R และ 2R ไปจนครบจำนวนบิตที่ต้องการดังรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 วงจรแปลงสัญญาณ Digital เป็น Analog แบบ R-2R Ladder

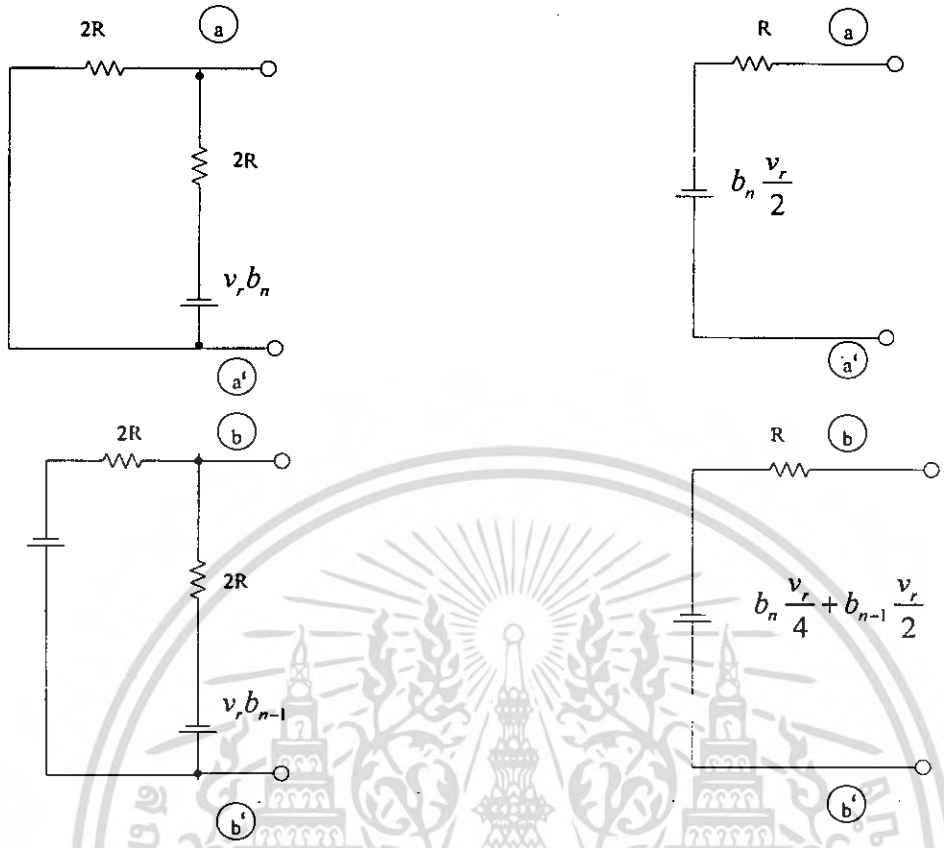
จากรูปที่ 2.12 สามารถวิเคราะห์หาค่าแรงดันเอาท์พุตของวงจรได้โดยแปลงวงจรในส่วน R-2R Ladder ให้อยู่ในรูปวงจรเสมือนได้เป็น



รูปที่ 2.13 วงจรเสมือนของรูปที่ 2.12

จากรูปที่ 2.13 สามารถวิเคราะห์โดยการทำการแยกโนดเริ่มตั้งแต่โนด a,a' ไปจนถึง c,c' โดยใช้ทฤษฎีของเทวินิน (Thevinin) ตามรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 การวิเคราะห์โดยใช้วงจรเสมือนเทวินิน

จากรูปที่ 2.14 สมการแรงดันเอาท์พุตสามารถสรุปได้โดยมีค่าเท่ากับ

$$v_o = \frac{R_f v_r}{R} \sum_{i=1}^{n} b_i 2^{-i}$$

โดยที่ V_r คือแรงดันอินพุต (ประมาณ 5 โวลท์) ค่าของแรงดันเต็มสเกล (Voltage Full Scale:V_{fs}) จากสมการที่ ทำได้โดยกำหนดให้ทุกบิตมีค่าเป็น "1" ดังนั้นค่า V_{fs} จะมีค่าเท่ากับ

$$v_{fs} = \frac{2R_f v_r}{R}$$

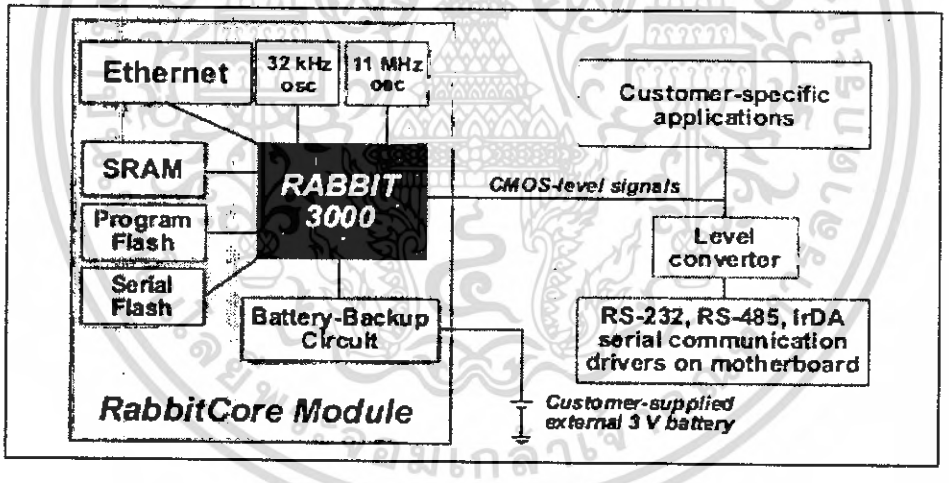
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ชุดพัฒนา RCM 3720

ชุดพัฒนา Rabbit 3720 ประกอบไปด้วย แรบบิทคอร์โมดูล (Rabbit Core Module) และ โปรโตไทป์ปิ้ง บอร์ด (Prototyping Board) ซึ่งเราสามารถเขียนโปรแกรมทดสอบ Rabbit 3720 รวมไปถึงการออกแบบแอพพลิเคชั่น

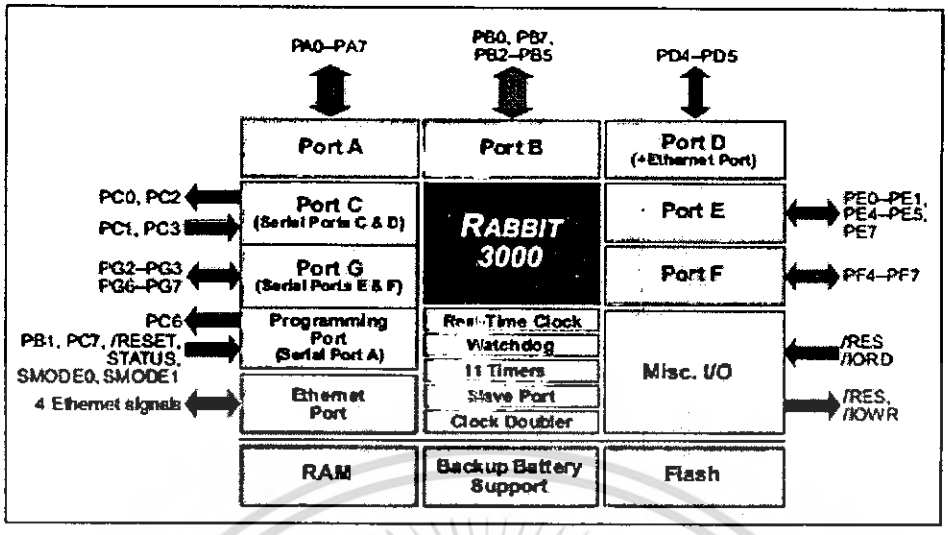
ชุดพัฒนานี้เป็นชุดพัฒนาสำหรับ ทีซีพี/ไอพี (TCP/IP) เป็นของบริษัท Rabbit Semiconductor ซึ่งชุดพัฒนา RCM 3720 (แรบบิทคอร์โมดูล 3720:Rabbit Core Module 3720) ใช้โปรเซสเซอร์ของบริษัทตัวเองโดยรุ่น Rabbit 3000 มีคุณสมบัติคร่าวๆดังนี้

- ใช้โปรเซสเซอร์ Rabbit 3000 ทำงานที่ 22.1 MHz ใช้ไฟเลี้ยง 3.3 V
- ใช้ พอร์ต10/100 base T
- หน่วยความจำแฟลช 512 KB/SDRAM 512 KB
- 56 พอร์ตขนาน I/O
- มีพอร์ตอนุกรม 6 พอร์ต
- มี 52 ดิจิตอล I/O พอร์ต
- มีโปรแกรมสำหรับประยุกต์สำหรับพัฒนา คือ ไดนามิค ซี (Dynamic C) ซึ่งเป็นภาษาพื้นฐาน



รูปที่ 2.15 RCM 3720

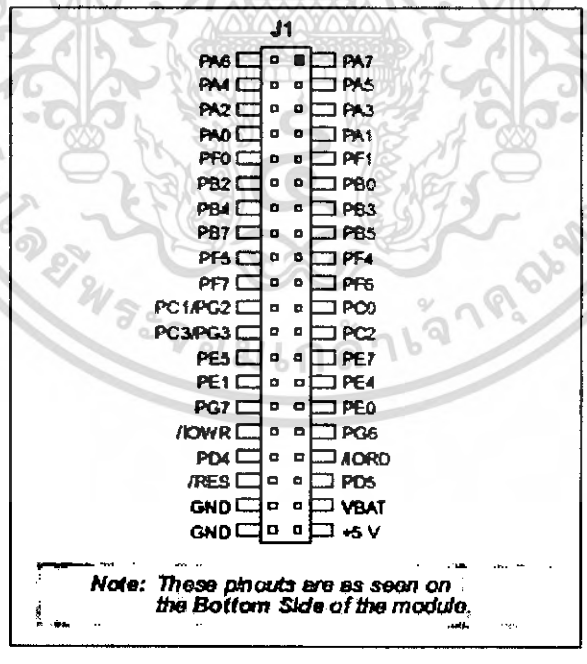
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 การใช้พอร์ตต่างๆ ของ Rabbit 3000

2.7.1 RCM 3720 ดิจิตอลอินพุตและเอาท์พุต

RCM 3720 มี 52 อินพุต/เอาท์พุต แบ่งได้เป็น 7 พอร์ต พอร์ตละ 8 บิต บน J1 J2 โดยจะมี 44 ขาสัญญาณเป็นได้ทั้ง เอาท์พุตและอินพุต คือPA0-PA7 ,PB2-PB7,PD2-PD7,PE0-PE1,PF0-PF7,PG2-PG7



รูปที่ 2.17 แสดงรายละเอียดขาของ J1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงขาต่างๆของ RCM 3720

Pin		Pin Name	Default Use	Alternate Use	Notes
Header J1	1–8	PA[7:0]	Parallel I/O	External data bus (ID0–ID7) Slave port data bus (SD0–SD7)	External Data Bus
	9	PF1	Input/Output	QD1A CLKC	
	10	PF0	Input/Output	QD1B CLKD	
	11	PB0	Input/Output	CLKB	
	12	PB2	Input/Output	IA0 /SWR	External Address 0 Slave port write
	13	PB3	Input/Output	IA1 /SRD	External Address 1 Slave port read
	14	PB4	Input/Output	IA2 SA0	External Address 2 Slave Port Address 0
	15	PB5	Input/Output	IA3 SA1	External Address 3 Slave Port Address 1
	16	PB7	Input/Output	IA5 /SLAVEATTN	External Address 5 Slave Port Attention
	17	PF4	Input/Output	AQD1B PWM0	
	18	PF5	Input/Output	AQD1A PWM1	
	19	PF6	Input/Output	AQD2B PWM2	
	20	PF7	Input/Output	AQD2A PWM3	
	21	PC0	Output	TXD	Serial Port D
	22	PC1/PG2	Input/Output	RXD/TXF	Serial Port D Serial Port F
	23	PC2	Output	TXC	Serial Port C
	24	PC3/PG3	Input/Output	RXC/RXF	Serial Port C Serial Port F
	25	PE7	Input/Output	I7 /SCS	External Address 7 Slave Port Chip Select

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงขาต่างๆของ RCM 3720 (ต่อ)

	Pin	Pin Name	Default Use	Alternate Use	Notes
Header J1	26	PE5	Input/Output	I5 INT1B	
	27	PE4	Input/Output	I4 INT0B	
	28	PE3	Input/Output	I1 INT1A	I/O Strobe 1 Interrupt 1A
	29	PE0	Input/Output	I0 INT0A	I/O Strobe 0 Interrupt 0A
	30	PG7	Input/Output	RXE	Serial Port E
	31	PG6	Input/Output	TXE	
	32	/IOWR	Output		External write strobe
	33	/IORD	Input		External read strobe
	34	PD4	Input/Output	ATXB	Alternate Serial Port B
	35	PD5	Input/Output	ARXB	
	36	/RES	Reset output	Reset input	Reset output from Reset Generator
	37	VBAT			
	38	GND			
	39	+5 V			
	40	GND			

2.7.2 การเชื่อมต่อหน่วยความจำของไอโอ (I/O)

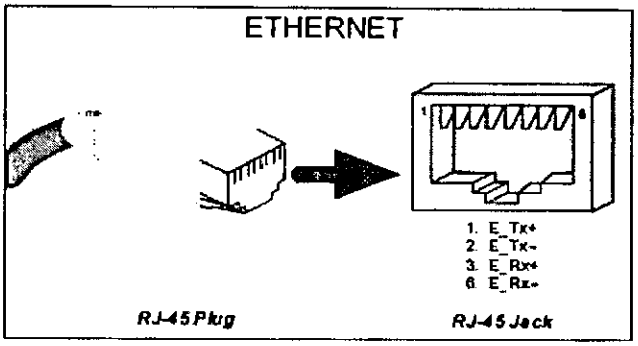
Rabbit 3000 มีบัสแอดเดรส (A0-A19) และบัสดาต้า (D0-D7) ไปยังชิพหน่วยความจำแบบแฟลช และ SDRAM มีอินพุต/เอาท์พุตสำหรับเขียน (/IOWR) และอินพุต/เอาท์พุต สำหรับอ่าน (/IORD) เพื่อใช้ติดต่อกับอุปกรณ์ภายนอก

พอร์ตขนาน A สามารถถูกใช้เป็นดาต้าบัส ไอ/โอจากภายนอกได้โดยตรง (external I/O data bus) โดยแยกอินพุตและเอาท์พุตออกจากดาต้าบัสหลัก พอร์ตขนาน B (PB3-PB7) ก็สามารถใช้เป็นดาต้าบัสข้อมูลภายนอก ได้เช่นเดียวกัน

2.7.3 พอร์ตของอีเธอร์เน็ต (Ethernet Port)

Programmable I/O Interface เป็นส่วนของการเชื่อมต่อกับอุปกรณ์ภายนอกเพื่อควบคุมหรืออ่านสถานะต่างๆของอุปกรณ์ โดยใช้สามารถโปรแกรมเองได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 พอร์ต Ethernet RJ-45

2.7.4 รูปแบบการออกแบบ Rabbit 3000

Rabbit 3000 ได้พัฒนาการออกแบบใหม่ขึ้นมา แต่ตัวประมวลผลและชุดคำสั่งยังคงมีคุณสมบัติใกล้เคียงกันเมื่อเปรียบเทียบกับตัวประมวลผลตัวอื่นๆ โดยที่มีรูปแบบและโครงสร้างของบอร์ดและแผนผังของรีจิสเตอร์คล้ายกับ Z80 และ Z180 ซึ่งมีความล้าหลังและมีความฟุ่มเฟือยในการใช้ทรัพยากรจึงได้เพิ่มประสิทธิภาพโดยการเพิ่ม 1 ไบต์ต่อ 1 ออปโค๊ต สำหรับชุดคำสั่งใหม่ซึ่งเป็นข้อได้เปรียบของการพัฒนาโดยที่มีความคล้ายกับ Z80 และ Z180 ซึ่งง่ายต่อความเข้าใจ ในภาษาแอสแซมบลีของ Rabbit โดยที่ชุดคำสั่งของ Z80 และ Z180 นั้นสามารถนำมาใช้ในการเขียนแอสแซมบลีและประมวลผลบน Rabbit

มีการเปลี่ยนเทคโนโลยีที่ใช้บน Z80 และ Z180 และลักษณะพิเศษของ Rabbit จะถูกนำมาแทนที่ เช่น Rabbit ไม่สนับสนุนการใช้ไดนามิคแรม (Dynamic Ram) แต่สนับสนุนการใช้สแตติคแรม (Static Ram) เพราะมีราคาถูกลง Rabbit ไม่ใช้ DMA (Direct Memory Access) เพราะ DMA ไม่สามารถนำไปใช้กับระบบอื่นได้

ผู้ที่เคยเขียนภาษาซี บนชุดคำสั่งของ Z80 มักจะประสบปัญหาเรื่องการทำงานของภาษาซี เพราะส่วนใหญ่ของปัญหาคือการขาดแคลนคำสั่งที่ใช้ ในการควบคุม 16-บิตเวิร์ค และสำหรับการควบคุมการรับส่งข้อมูลไปยังคอมพุตแอดเดรส (computed address) โดยเฉพาะในขณะที่สแต็กเก็บข้อมูลอยู่

Rabbit สนับสนุน 4 ระดับ สำหรับการจัดอันดับ ค่าความสำคัญ (Priorities) ซึ่งเป็นลักษณะที่เพิ่มประสิทธิภาพในการจัดการในเรื่องความเร็วของ เส้นทางในการอินเตอร์รัพ

2.7.5 หน่วยประมวลผล Rabbit 8 บิตกับหน่วยประมวลผลอื่น

Rabbit 3000 โปรเซสเซอร์ได้ถูกออกแบบโดยใช้ระบบวัตถุประสงค์ของการฝึกหัด (Object of creating practical system) ซึ่งในการแก้ปัญหาที่เกิดขึ้นในปัจจุบันโดยใช้รูปแบบเศรษฐศาสตร์

- Rabbit 3000 ถูกออกแบบให้มีการทำงานโดยที่ใช้ ระบบ EMI น้อย และมีสัญญาณนาฬิกามากกว่า 40เมกะเฮิร์ตซ์
- Rabbit 3000 มีการประมวลโดยมีการกำหนดรูปแบบ การจัดวางและมีความกะทัดรัดในเรื่องชุดคำสั่ง ที่มีส่วนช่วยเหลือ คือ ตัวคอมไพเลอร์ (Complier) และไลเบอร์รี่ (Library) ที่มีกฎจำนวนมาก มีการเตรียม Rabbit สำหรับไมโครคอนโทรลเลอร์ 186,386,8051 และ z80
- หน่วยความจำของ Rabbit ได้การยอมรับเรื่องประสิทธิภาพและง่ายต่อการออกแบบโดยไม่ต้องอาศัยอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Rabbit 3000 มีบัสภายนอกใช้ 2 ช่วงสัญญาณนาฬิกา สำหรับในการอ่าน และ 3 ช่วงสัญญาณนาฬิกา สำหรับการเขียน ซึ่งเป็นข้อได้เปรียบ เมื่อเปรียบเทียบกับการออกแบบสัญญาณนาฬิกาเดี่ยว โดยกล่าวถึงข้อได้เปรียบว่าง่ายต่อการออกแบบเพื่อหลีกเลี่ยง การแย่งบัสเขียนด้วยโซลิด ดาต้า (Solid data) และการถือครองของข้อมูล การเปลี่ยนแปลงของผลลัพธ์ของหน่วยความจำที่สามารถใช้เวลามากกว่าครึ่งหนึ่งของรอบการทำงานบัสความสามารถใช้สัญญาณนาฬิกาแบบไม่สมมาตร (Asymmetric Clock) ที่ถูกควบคุมโดยสัญญาณนาฬิกาแบบคู่ซึ่งข้อได้เปรียบถ้าเปรียบเทียบกับระบบสัญญาณนาฬิกาเดี่ยว คือมีการทำงานโดยใช้มีการเพิ่มความเร็วบัสเป็น 2 เท่า ซึ่งไม่สามารถเป็นความจริงได้ ในความเป็นจริงของหน่วยความจำ

- Rabbit 3000 สามารถทำงานได้ที่ 3.6 โวลต์ หรือน้อยกว่านี้ แต่สามารถทำงานได้ที่ 5 โวลต์ ขาเข้าและมี เส้นทางที่ 2 ของบัส (Second complete Bus) สำหรับการทำงานของ I/O สามารถทำงานได้โดยแอพพลิเคชั่นเหมือนกับการออกแบบคำสั่ง ลักษณะพิเศษของระบบอีกอย่างคือการที่สามารถใช้งานได้โดยการออกแบบการใช้ 3 โวลต์ และ 5 โวลต์ ผสมกันและสามารถหลีกเลี่ยงปัญหาที่เกี่ยวกับการโหลดปัญหาระบบ EMI ซึ่งมีผลมาจากหน่วยความจำของบัส ไปเชื่อมต่ออุปกรณ์ I/O ที่เป็นจำนวนมาก

- Rabbit 3000 อาจจะโปรแกรมจากระยะไกลรวมไปถึงการโคลด์บู๊ต (cold-boot) cold-boot เส้นทางการเชื่อมต่อแบบอนุกรม นอกเหนือจากเส้นทางนี้ มีการเชื่อมต่อกับระบบเครือข่ายมีการใช้งานผ่านอินเตอร์เน็ตสามารถสร้าง และ ติดต่อกับ Rabbit โดยผ่านอินเตอร์เน็ต

- Rabbit 3000 สามารถติดต่อกับอุปกรณ์ภายนอกได้จำนวนมาก เพื่อสำหรับการแข่งขันกับตัวประมวลผลตัวอื่นๆ

- Rabbit 3000 มีขนาด 8 บิต ในการประมวลผล โดยมี 8 บิต ดาต้าบัสภายนอก และ 8 บิตดาต้าบัสภายใน เพราะการทำงานส่วนใหญ่ จะทำงาน 8 บิตบัสภายนอก และเพื่อใช้ชุดคำสั่งให้มีความกระทัดรัดการทำงานหลายอย่างมีประสิทธิภาพพอกับ 16 บิต

- ส่วนประกอบภายนอกของชิพอุปกรณ์ที่มีประโยชน์และมีขนาดเล็ก จะประกอบด้วยพอร์ตอนุกรม ระบบสัญญาณนาฬิกา เครื่องกำเนิดเวลา/วัน พอร์ตขนาน พอร์ทลูก (slave port) การถอดรหัสไทเมอร์ เป็นต้น

2.7.6 โทเลอแรนท์ (Tolerant) 5V อินพุต

Rabbit 3000 มีการทำงานโดยมีโวลล์อยู่ระหว่าง 1.8V-3.6V แต่ Rabbit 3000 ส่วนใหญ่สามารถรับ 5V อินพุตท์ ยกเว้น ขาตัวจ่ายไฟ กับขาบัฟเฟอร์ออสซิเลเตอร์ ในขณะที่สัญญาณมีขนาด 5V ส่งไปยังขา Tolerant ถ้ามีความต่างศักย์ มากกว่านั้น Rabbit จะปิดการทำงานทันที ข้อได้เปรียบของ 5V tolerant คือการให้อุปกรณ์ที่มีขนาด 5V สามารถทำงานได้โดยผ่านระบบสวิตซ์ ก่อนที่จะไปติดต่อโดยตรงกับ rabbit โดย มีการทำงานที่ 5V และอินพุตที่สามารถเข้าได้ต้องมีความต่างศักย์อยู่ระหว่าง 0.8 V-2 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.7 พอร์ตอนุกรม

มีพอร์ตอนุกรม 6 พอร์ต คือ A,B,C,D,E และ F พอร์ตอนุกรมทั้ง 6 สามารถทำงานอะซิงโครนัส เพื่อให้อัตราการส่งข้อมูลของระบบสัญญาณนาฬิกาแบ่งจาก 16 พอร์ตอะซิงโครนัสสามารถจัดการที่ 7-8 บิตข้อมูล บิตที่ 9 เป็นบิตแอดเดรส ซึ่งถูกส่งไปเป็นไบต์แรกของข้อความ (message) พอร์ตอนุกรม A,B,C และ D จะทำงานในโหมดสัญญาณนาฬิกาแบบอนุกรม ในโหมดนี้จะรับสัญญาณนาฬิกาเข้ามาเพื่อซิงโครนัสข้อมูลเข้าและออกตามจังหวะของสัญญาณนาฬิกาในการสื่อสารของอุปกรณ์ 2 ตัว ที่สนับสนุนการใช้สัญญาณนาฬิกา เมื่อ Rabbit 3000 ใช้สัญญาณนาฬิกาอัตราการส่งข้อมูล จะสูงถึง 80 % ของระบบสัญญาณนาฬิกาที่ถูกแบ่งจาก 128 หรือ 183,750 bps ที่ความเร็วของสัญญาณนาฬิกาที่ 22.1 MHz

พอร์ตอนุกรม E และ F สามารถตั้งค่าให้เป็น SDLC/HDLC ซึ่งรูปแบบของ SDLC จะสนับสนุนโปรโตคอลอินฟราเรด ทั้ง 2 พอร์ต

ทั้งหมด 6 พอร์ตสามารถทำงานได้บนโหมดอะซิงโครนัสซึ่งมีขนาดอัตราในการส่งเท่ากับสัญญาณนาฬิกา หาร 8 โดยที่พอร์ตของอะซิงโครนัสใช้ 7 บิตหรือ 8 บิต โดยที่มีพาริตี้บิต หรือไม่มีก็ได้ โดยให้บิตที่ 9 เป็นแอดเดรส โดยมีการตั้งค่าหรือลบเพื่อแสดงที่ไบต์แรกของ ข้อความ เมื่อต้องการความช่วยเหลือ

โปรแกรมพอร์ตอนุกรม สามารถบอกข้อมูลจากเอาท์พุตชิพรีจิสเตอร์ ขณะที่ไบต์สุดท้ายได้มาถึงเป็นสิ่งที่สำคัญมากสำหรับ RS-489 เพราะว่าการสื่อสารแบบทางเดียว (Half-duplex) ไม่สามารถติดต่อได้โดยตรงจนกระทั่งบิตสุดท้ายได้ถูกส่งออกไป เราจะเรียกว่า แอดเดรสบิต (address bit) หลังจากดาต้าบิตสุดท้ายถ้าแอดเดรสบิต มักจะถูกติดตามด้วยบิตหยุด (stop bit) ปกติเราใช้สำหรับการส่ง 2 บิตหยุด หรือพาริตี้บิต ถ้ามีความต้องการส่งโดยตรง

พอร์ตอนุกรม A,B,C,D สามารถทำงานที่สัญญาณนาฬิกาในโหมดอนุกรมได้ในการทำงานโหมดนี้ สัญญาณอะซิงโครนัสจะควบคุมข้อมูลที่เข้าออกโดยที่ พอร์ตอนุกรมของ Rabbit หรือ อุปกรณ์ทางไกลที่ช่วยเหลือการหาสัญญาณนาฬิกา ในขณะที่ Rabbit จัดเตรียมสัญญาณนาฬิกานั้นมีอัตราในการส่งอาจจะสูงขึ้นเป็นครึ่งหนึ่งของระบบความถี่สัญญาณนาฬิกาในขณะที่สัญญาณนาฬิกาจัดการโดยอุปกรณ์ชนิดอื่นๆ มีค่ามากที่สุดของข้อมูล จะถูกแบ่งระบบออกเป็น 6 ส่วนจำเป็นที่จะต้อง จัดหาสัญญาณนาฬิกาแบบภายนอกด้วยสัญญาณนาฬิกาแบบภายในบนโหมดอนุกรม

พอร์ตอนุกรม A มีความสามารถพิเศษ คือสามารถทำให้ระบบคูลบู๊ต (cool-boot) ได้หลังจากการตั้งค่าพอร์ตอนุกรม A เป็น พอร์ตปกติที่ใช้สำหรับพัฒนาสำหรับการพัฒนาโปรแกรมโดย Dynamic C ทุกๆพอร์ตอนุกรม มีไทม์เมอร์โหมดพิเศษ ไว้เพื่อการติดต่อแบบมาตรฐานการสื่อสารข้อมูล

2.7.8 ระบบสัญญาณนาฬิกา

ออสซิเลเตอร์ ตามปกติใช้คริสตัลภายนอก ชนิดของความถี่ซึ่งอยู่ระหว่าง 1.8-26 MHz การประมวลผลสัญญาณนาฬิกา ถูกกำหนดจากเอาท์พุตของ ออสซิเลเตอร์ โดยมีความถี่ 2 เท่าของความถี่การใช้ความถี่โดยตรง หรือการหารด้วย 2, 4, 6, 8 การประมวลผลสัญญาณจะมีค่าเท่ากับ 32.768 KHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

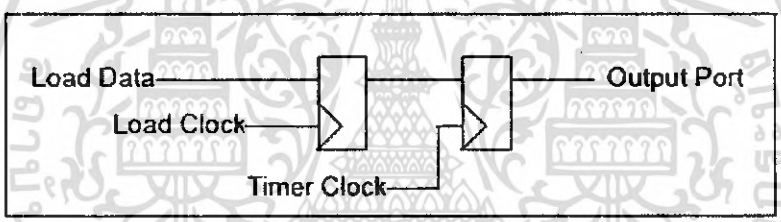
สัญญาณนาฬิกาแบบต่อเนื่อง (real-time clock) ออสซิเลเตอร์ สำหรับการทำงานที่ต้องการกำลังต่ำ โดยปกติ ออสซิเลเตอร์ สามารถปิดภายใต้การควบคุมของซอฟแวร์

2.7.9 ออสซิเลเตอร์ 32.768 KHz อินพุต

32.768 KHz ออสซิเลเตอร์อินพุต ได้ถูกออกแบบเพื่อที่จะรองรับ 32.768 KHz โดยแนะนำให้ใช้สัญญาณกำลัง (Tinny Logic) ซึ่งเป็นส่วนที่มีการสนับสนุนทางด้านเอกสารและมีราคาที่ถูก 32.768 KHz ถูกใช้สำหรับ แบตเตอรี่สำรอง 48 บิตเคาท์เตอร์ ซึ่งมีหน้าที่รักษาแบบ real-time clock (RTC) ซึ่งสามารถอ่านและตั้งค่าได้ด้วยโปรแกรม

2.7.10 พอร์ตขนาน I/O

มีทั้งหมด 56 พอร์ตขนาน I/O โดยแบ่งออกเป็น 7 เส้น แต่ละเส้นมีขนาด 8 บิต โดยมีจุดมุ่งหมาย A-G ส่วนใหญ่ของพอร์ตมีหลายฟังก์ชั่นในแต่ละคาต้าอนุกรม ส่วนขนาน D,E,F มีความสามารถในการทำไทม์เมอร์อะซิงโครนัส เอาท์พุต

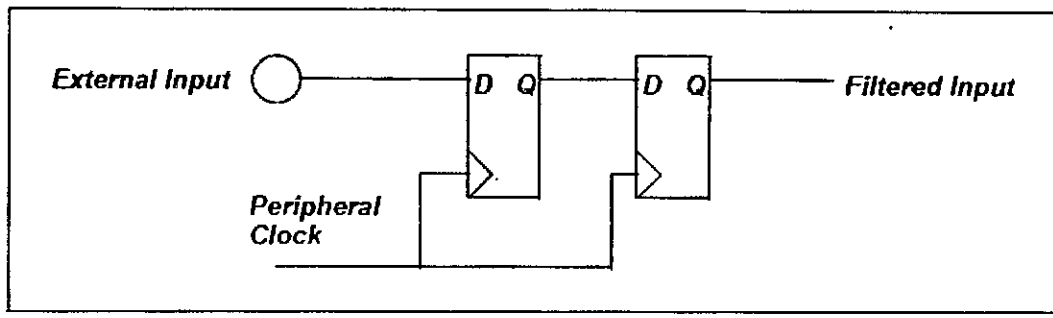


รูปที่ 2.19 เอาท์พุตรีจิสเตอร์สำหรับ พอร์ตขนาน D และ E

โดยที่สโตรบ (strobe) ไปยังพอร์ตซึ่งถูกโหลดเก็บไว้ในรีจิสเตอร์เลเวลแรก แล้วรีจิสเตอร์จะถูกส่งไปยังเอาท์พุตรีจิสเตอร์แล้วเลือกเวลาของสัญญาณนาฬิกา โดยสัญญาณนาฬิกาถูกเลือกโดยที่เอาท์พุตของไทม์เมอร์ A1,B1,B2 หรือ สัญญาณนาฬิการอบข้าง(หารด้วย2) สัญญาณไทม์เมอร์เป็นข้อได้เปรียบในการสร้างพัลส์ที่สามารถควบคุมได้ ซึ่งขอบเขตได้ถูกวางตำแหน่งเรียบร้อยแล้วและที่ความถูกต้องสูง ในเวลานั้น แอฟพลิเคชั่น (Application) นี้จะประกอบไปด้วยสัญญาณการติดต่อสื่อสาร ขนาดความกว้างของพัลล์มอดูเลชั่นและการควบคุมสเต็ปมอเตอร์ (step motor)

ขาอินพุตส่งไปยังพอร์ตขนานจะถูกกรองด้วยแคสเคส ดีฟลิป-ฟล็อป ซึ่งจะป้องกันพัลส์ขนาดเล็กที่เข้ามาแล้ว สัญญาณนาฬิกาที่อยู่รอบมาจาก อะซิงโครนัส พัลส์ภายนอก ไปยังพัลส์ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การกรองสัญญาณดิจิตอลของขาอินพุต

พอร์ตลูก (Slave Port)

พอร์ตลูกได้ถูกออกแบบให้มาใช้กับโปรเซสเซอร์ตัวอื่นๆหรือไม่ก็ใช้กับ Rabbit ตัวอื่นๆ มีพอร์ตสามารถแบ่งได้ด้วย พอร์ตขนาน A และ พอร์ตข้อมูล โดยที่ตัวแม่ สามารถอ่านรีจิสเตอร์ทั้ง 3 รีจิสเตอร์ก่อนแล้วค่อยเลือกเส้นทางได้มาจาก รีจิสเตอร์แอดเดรสและอ่านได้มาจาก strobe เป็นสาเหตุ ที่ทำให้รีจิสเตอร์ สามารถมีเอาท์พุตด้วยพอร์ตได้โดยที่รีจิสเตอร์ตัวเดิมสามารถเขียนเหมือนกับ รีจิสเตอร์ I/O โดย Rabbit ตัวลูก ที่เพิ่มอีก 3 เส้นทางจะเอามาทำเป็นเส้นทางย้อนกลับ

Rabbit ลูก สามารถอ่านรีจิสเตอร์ได้ด้วยรีจิสเตอร์ที่เหมือนกัน เหมือนกันกับ I/O รีจิสเตอร์ในขณะที่ข้อมูลที่เข้ามาเขียนลงรีจิสเตอร์ แล้วส่งไปยังรีจิสเตอร์อีกตัวหนึ่ง ในขณะที่ตัวแม่สามารถให้ตัวลูกอ่านข้อมูลที่เข้าไป ส่วนสายอื่นๆ จะบอกให้ตัวแม่ จะบอกการไหลของข้อมูลและยังไม่ได้อ่านจาก ตัวแม่ พอร์ตลูกสามารถส่งย้อนสัญญาณไปยังตัวแม่เพื่อที่จะจัดรูปแบบ เพื่อใช้ในการติดต่อสื่อสารบนพอร์ตลูก

2.7.11 Auxiliary I/O Bus

Rabbit 3000 มีชุดคำสั่งที่ช่วยเหลือการเข้าถึงหน่วยความจำและการเข้าถึง I/O การเข้าถึงหน่วยความจำจะมีพื้นที่ประมาณ 1 MB ส่วน I/O จะมีพื้นที่ว่างประมาณ 64 kB I/O ทั่วไปการออกแบบตัวประมวลผลโดยใช้แอดเดรส และ แถวข้อมูล ชุดเดียวกัน โดยใช้สำหรับหน่วยความจำ และที่ว่างของ I/O การแชร์ข้อมูลและแอดเดรส บ่อยครั้งมีการแย่งชิงกันหรือออกแบบโดยมีเงื่อนไขที่ซับซ้อน บัสของหน่วยความจำส่วนใหญ่ จะมีเวลาวิกฤต (Critical Time) และ ความสามารถในการโหลดถูกกำหนดโดยแชร์ข้อมูลด้วยบัส I/O

Rabbit 3000 มีคำสั่งที่ใช้ในการแบ่งบัส สำหรับ I/O และหน่วยความจำของ Auxiliary I/O บัสถูกใช้ขาเดียวกันโดยพอร์ตลูก พอร์ตขนาน A ถูกเตรียมไว้สำหรับ 8 แถวข้อมูล พอร์ตขนาน B ขนาด 2- 7 บิต เตรียมไว้ 6 แถวแอดเดรส ต้องใช้ในการส่งสัญญาณอย่างน้อย 6 เส้นจาก 16 เส้น เป็นการใช้พื้นที่ I/O เต็มที่ Auxiliary บัส โดยปกติจะทำงานบนวนรอบการทำงาน I/O ชิพ เลือก เช่นเดียวกับ กรอ่านและเขียน strobe ซึ่งมีหลายค่าที่ขาต่างๆดังนั้นการรวมแอดเดรสสามารถเป็นบัฟเฟอร์ได้

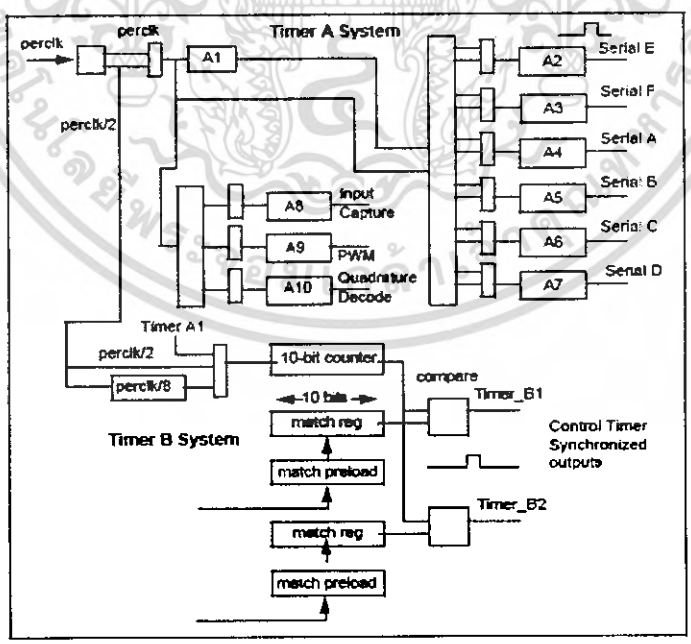
การติดต่ออุปกรณ์ I/O กับ auxiliary bus จะช่วยเพิ่มความสามารถในการโหลดสำหรับ RCM 3000 มีประมาณ 2-3 ขา ที่ต้องการการติดต่อจาก core module ตั้งแต่พอร์ตลูกและ I/O บัสสามารถแชร์ ขาได้ และบัสหน่วยความจำ ไม่มีความต้องการมากเพื่อเตรียมพร้อมไว้สำหรับความสามารถ I/O เพราะ I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีความสามารถน้อยและมีความเร็วน้อยกว่า บัสหน่วยความจำในอนาคตสามารถทำงานโดยปราศจาก EMI และปัญหาการสะท้อนของสัญญาณ 5V ซึ่งสามารถไปบน I/O บัส ตั้งแต่ อินพุต Rabbit 3000 สามารถรับ 5V ซึ่งสัญญาณนี้ทำได้ง่าย

2.7.12 ไทม์เมอร์ (Timers)

Rabbit มีระบบ timer หลายระบบโดยมีสัญญาณอินเตอร์รัพคาบ ซึ่งมาจาก 32.768 KHz ออสซิเลเตอร์ หารด้วย 6 จะมีอินเตอร์รัพทุก 448 ไมโครวินาที ถ้าเป็นไปได้ มีจุดมุ่งหมายตามความต้องการของสัญญาณอินเตอร์รัพ Timer A ประกอบด้วย 10 เคาท์ดาวน์ (countdown) ขนาด 8 บิตและสามารถรีโหลดรีจิสเตอร์ซึ่งสามารถแคสเคส ขึ้น 2 ขั้น แต่ละครั้ง รีจิสเตอร์ countdown สามารถตั้งค่าโดยแบ่งตามจำนวนเลขซึ่งอยู่ระหว่าง 1-256 เอาท์พุตจาก 6 ค่า ไทม์เมอร์ซึ่งเตรียมค่าไว้ อัตราการส่งสัญญาณนาฬิกาสำหรับรีจิสเตอร์พอร์ตอนุกรมจำนวนหนึ่งจาก รีจิสเตอร์ทั้งหมด เป็นสาเหตุที่ทำให้เกิดอินเตอร์รัพและสัญญาณนาฬิกาไทม์เมอร์ซิงโครไนส์ พอร์ตเอาท์พุตขนาน Timer B ประกอบด้วย 10 บิตเคาท์เตอร์ ซึ่งไม่สามารถเขียนได้แต่อ่านได้ มีรีจิสเตอร์โดยที่ขนาด 10 บิตและเปรียบเทียบ ถ้ารีจิสเตอร์สามารถเข้าได้กับเคาท์เตอร์และพัลส์ เป็นค่าเอาท์พุตที่ออกมา ดังนั้นไทม์เมอร์สามารถถูกโปรแกรมไปยังเอาท์พุตที่เป็นพัลส์เพื่อจะตัดสินใจนับในอนาคต พัลส์ถูกสร้างมาจากสัญญาณนาฬิกาของไทม์เมอร์ซิงโครไนส์รีจิสตอร์เอาท์พุตพอร์ตขนาน เหมือนกับสาเหตุที่มาจากอินเตอร์รัพซึ่ง Time B สะดวกในการสร้างและมีความเที่ยงตรงในเรื่องของเวลา เพราะว่าได้ถูกโปรแกรมส่วนควบคุมเอาไว้



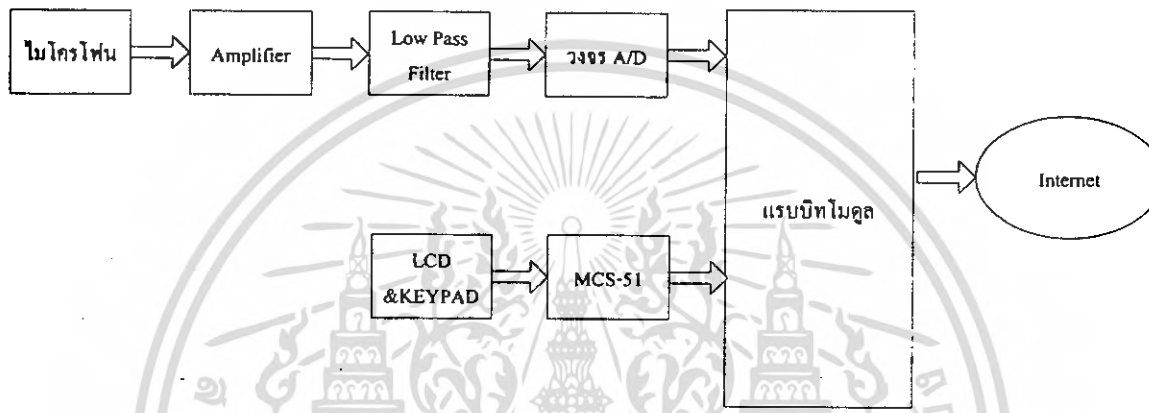
รูปที่ 2.21 ไทม์เมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

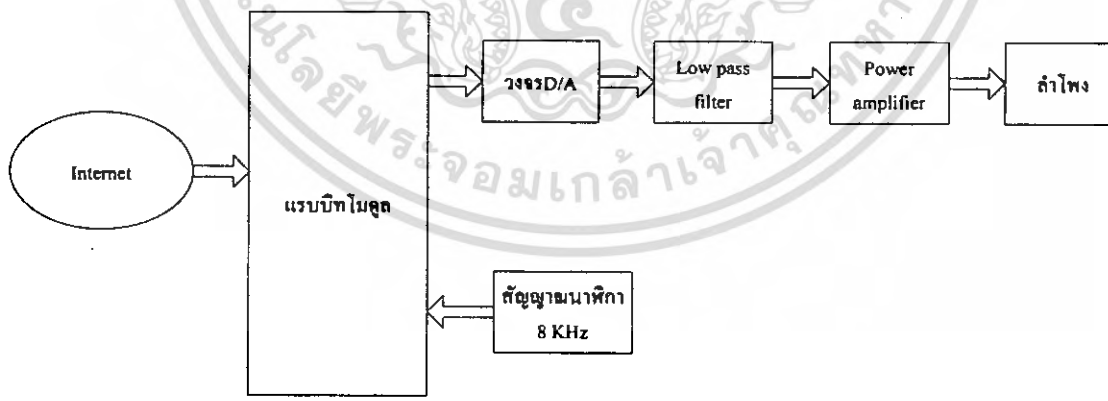
บทที่ 3
การคำนวณและการสร้าง

3.1 บล็อกไดอะแกรมรวมของโครงงานนี้

ด้านส่ง



ด้านรับ

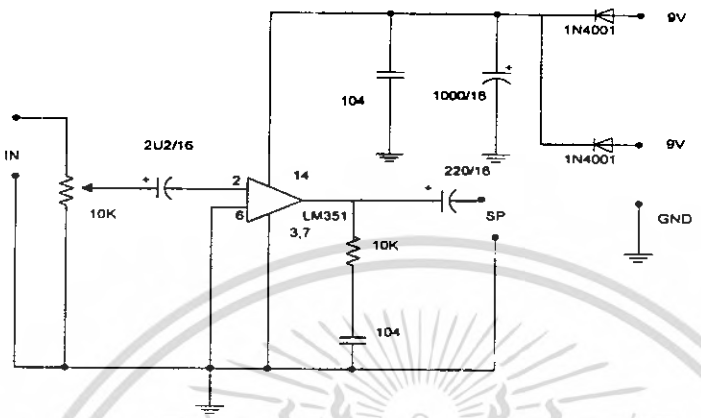


รูปที่ 3.1 บล็อคไดอะแกรมรวมของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 วงจรเพาเวอร์แอมป์

ในส่วนนี้จะประกอบด้วยเพาเวอร์แอมป์ 0.5 วัตต์ เป็นวงจรเครื่องขยายขนาดเล็กแต่ใช้งานได้อย่างกว้างขวางโดยใช้ไอซีเบอร์ LM380N ทำงานด้วยอัตราขยาย 36 เดซิเบลแสดงดังรูปที่ 3.2

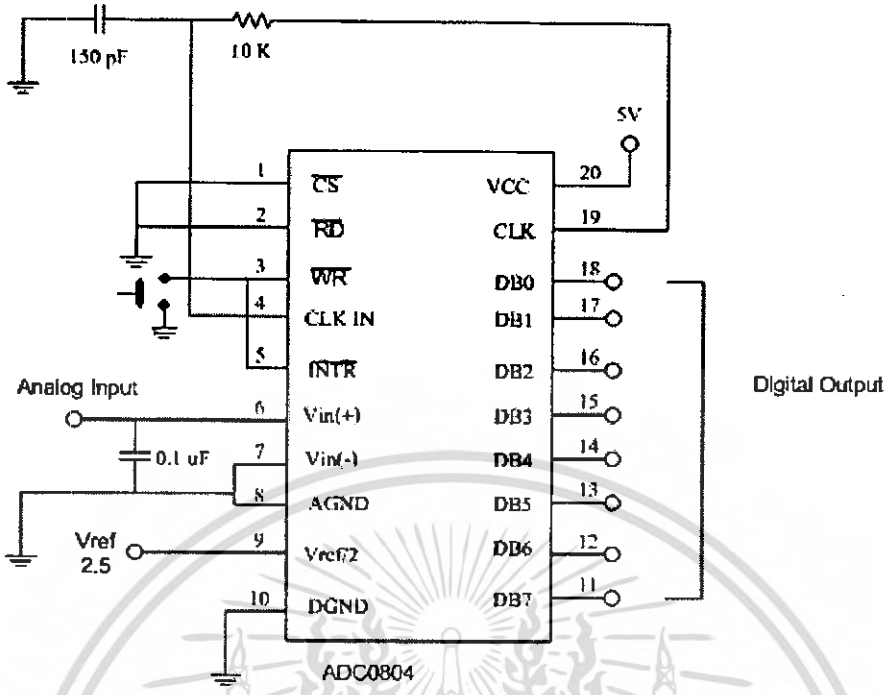


รูปที่ 3.2 วงจรเพาเวอร์แอมป์

3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

ที่ด้านส่ง อินพุตของวงจรก็คือสัญญาณเสียงที่มาจากวงจรปรีไมค์ ซึ่งเป็นสัญญาณอนาลอก เมื่อนำไปผ่านวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลใช้ ไอซีเบอร์ ADC0804 และนำเข้าสู่แรบบิทโมดูลเพื่อทำการส่งข้อมูลที่แปลงได้ไปสู่อินเตอร์เน็ต โดยไอซี ADC0804 จะมีตัวกำเนิดสัญญาณนาฬิกาที่ขา CLK R และ CLK IN 2 ขานี้จะต้องต่อเข้ากับ R และ C เพื่อทำให้เกิดการออสซิเลทภายในค่าความถี่คำนวณจาก $f_{clk} = 1/(1.1RC)$ ถ้าใช้ R = 10 กิโลโอห์ม และ C = 150 พิโคฟารัด จะออสซิเลทที่ความถี่ 606.06 กิโลเฮิรตซ์ จำนวนคล็อก (clock) สูงสุดที่สามารถทำการแปลงค่าสัญญาณอนาลอกเป็นดิจิตอลจะใช้ประมาณ 73 clocks ทำให้ค่าการแปลงสัญญาณ (Conversion Time) ได้เท่ากับ 73/606.06 kHz = 120 ไมโครวินาที คือประมาณ 8.3 kHz

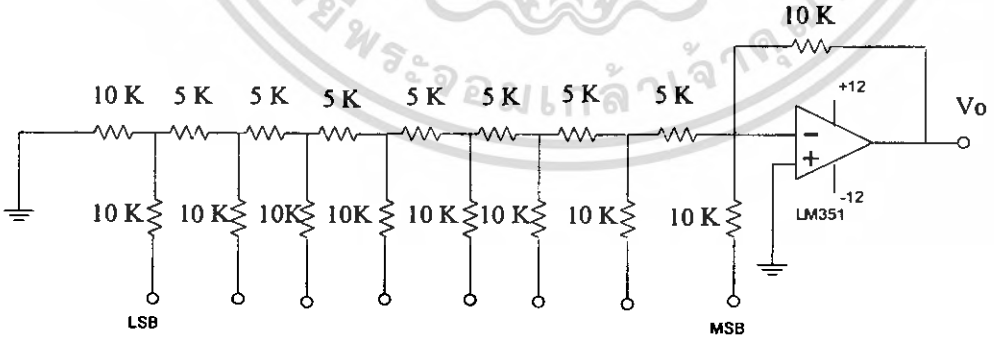
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

3.4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

ในด้านรับจะใช้วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก เพื่อแปลงข้อมูลที่รับมาให้กลับเป็นสัญญาณอนาลอกโดยจะใช้หลักการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแบบ R-2R Ladder

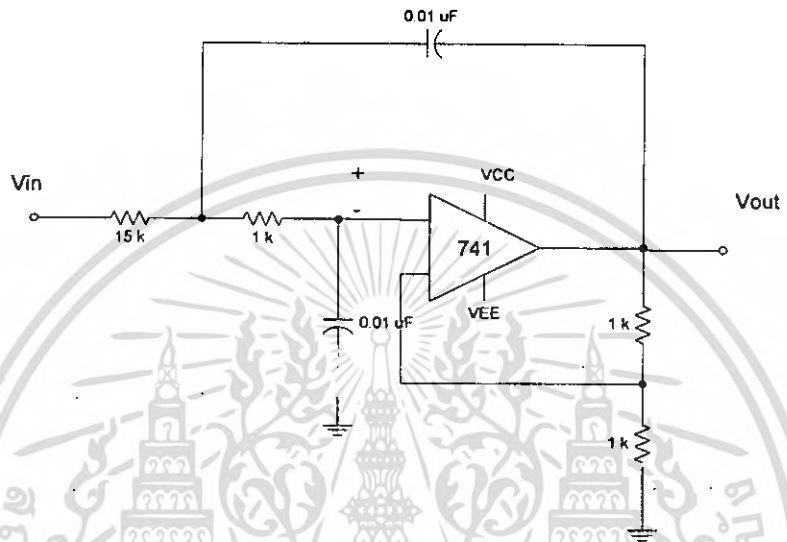


รูปที่ 3.4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วงจรกรองความถี่ต่ำผ่าน (Low Pass Filter)

เมื่อทำการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกแล้วจะต้องนำเข้าสู่วงจร Low Pass Filter เพื่อทำการกรองสัญญาณความถี่ต่ำไว้และกำจัดความถี่สูงทิ้งไป โดยทำการใช้ Op-Amp เบอร์ 741 มาสร้างเป็นวงจร Low Pass Filter ซึ่งเป็นการต่อวงจรแบบแอคทีฟ ฟิลเตอร์ (Active Filter) ในอันดับที่ 2 (second order)



รูปที่ 3.5 วงจร Low Pass Filter

เราสามารถกำหนดความถี่คัตออฟซึ่งได้จากการคำนวณดังสมการ

$$fc = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}}$$

จากรูปที่ 3.5 ออกแบบให้

$R_1 = 15\ k\Omega$

$R_2 = 1\ k\Omega$

$C_1 = 0.01\ \mu F$

$C_2 = 0.01\ \mu F$

$$fc = \frac{1}{2\pi\sqrt{15\times10^{3}\times1\times10^{3}\times0.01\times10^{-6}\times0.01\times10^{-6}}}$$

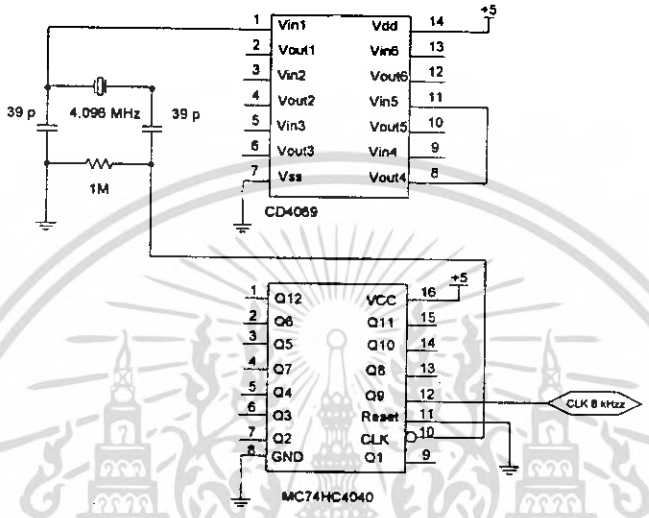
$$fc = 4.1kHz$$

เหตุผลที่จะเลือกความถี่คัตออฟประมาณ 4 kHz เพราะความถี่เสียงมีช่วงความถี่ประมาณ 4 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 วงจรสร้างสัญญาณนาฬิกา

ในที่นี้เลือกใช้ออสซิลเลเตอร์ขนาด 4.096 เมกะเฮิรตซ์ โดยต่อเข้ากับวงจรอินเวอร์เตอร์ ในที่นี้เลือก ใช้ไอซีเบอร์ CD4069 จะได้สัญญาณนาฬิกาขนาด 4.096 เมกะเฮิรตซ์จากนั้นนำไปเข้าวงจรหารความถี่ซึ่งใช้ไอซีเบอร์ 74HC4040 โดยหารเพื่อให้เหลือความถี่สัญญาณนาฬิกาที่ใช้ในวงจรซึ่งใช้ความถี่ 8 กิโลเฮิรตซ์ในการป้อนให้กับวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลและแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

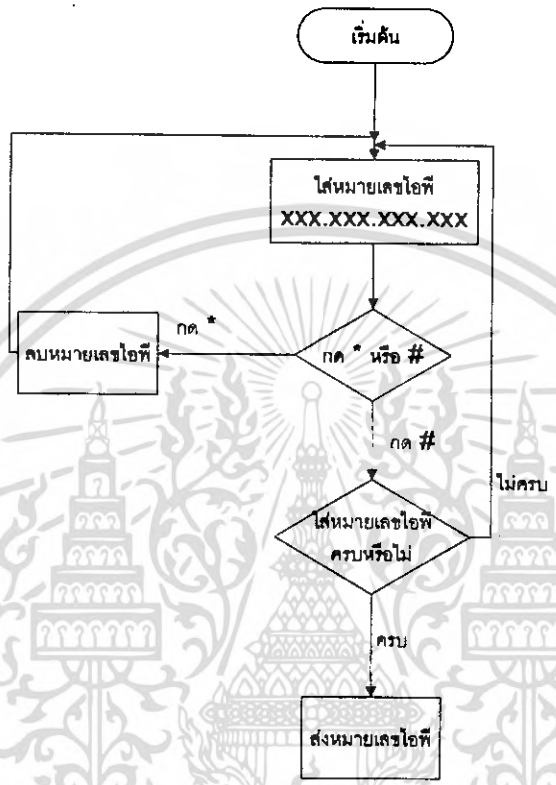


รูปที่ 3.6 วงจรสร้างสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 แอลซีดีและคีย์แพด

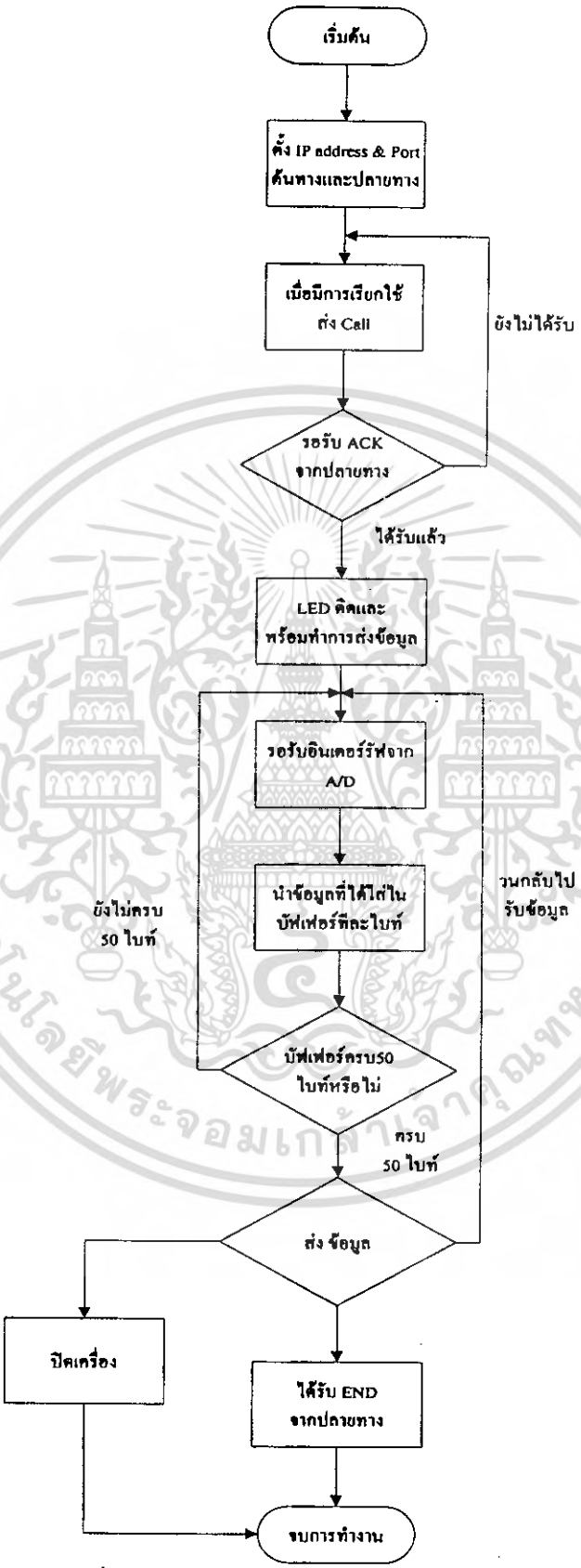
ในการติดต่อจะมีการกำหนดปลายทางโดยการกดเลขหมายไอพี แอดเดรสปลายทางโดยการกดเลขหมาย 12 ตัวเช่น 161.246.003.014 เลขหมายจะปรากฏที่หน้าจอแอลซีดี สามารถทำการแก้ไขเลขหมายได้หากกดเลขหมายผิด โดยการกดที่แป้น " * " และเมื่อกดเลขหมายครบทั้งหมดแล้วจึงจะสามารถส่งเลขหมาย



รูปที่ 3.7 โฟร์ลชาร์ตแสดงการทำงานของแอลซีดีและคีย์แพด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

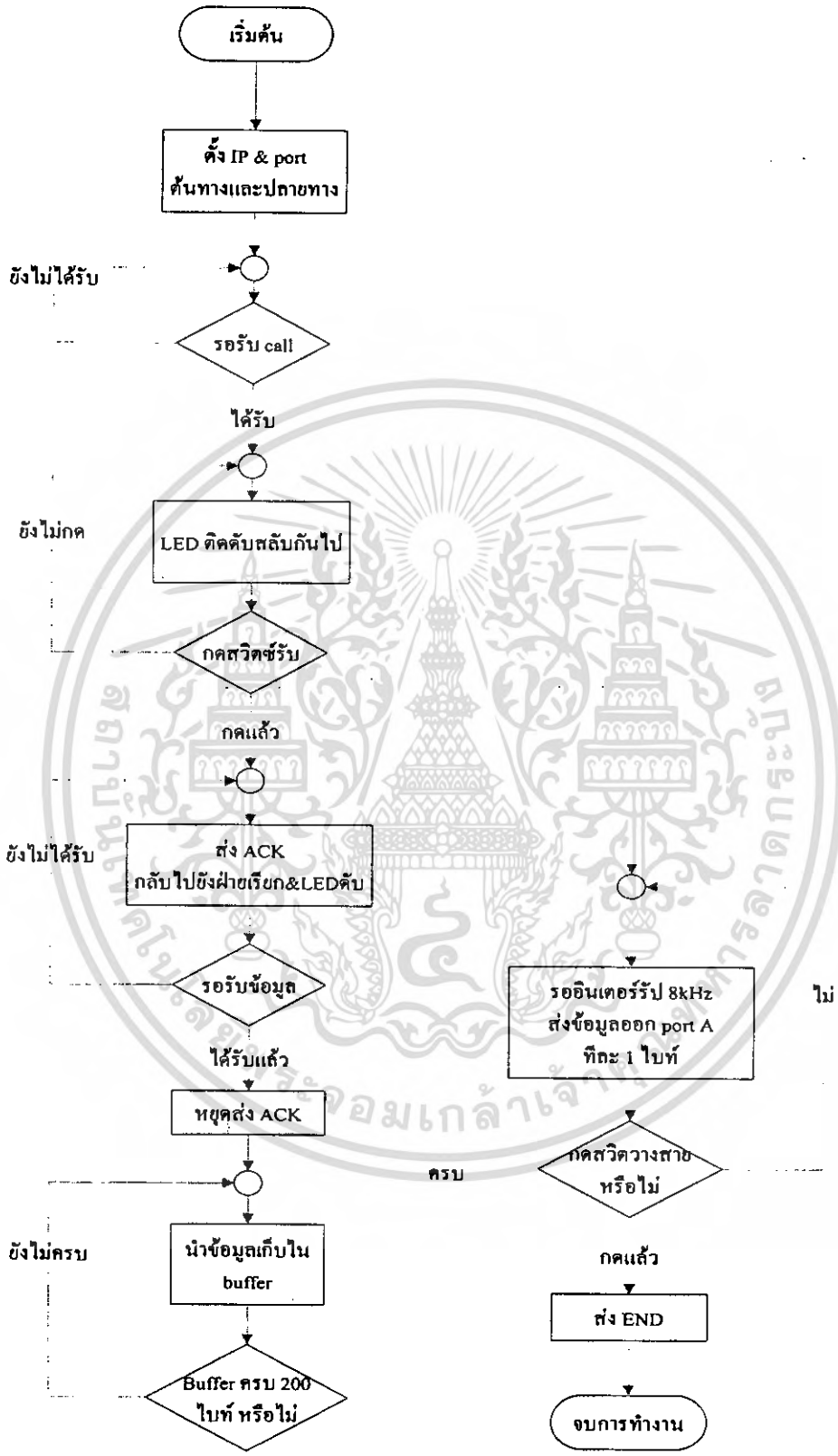
3.8 การส่งข้อมูลในแรบบิทโมดูล



รูปที่ 3.8 โฟร์ลชาร์ตแสดงการทำงานทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 การรับข้อมูลในแรบบิทโมดูล



รูปที่ 3.9 โฟร์ลชาร์ตแสดงการทำงานทางด้านรับ

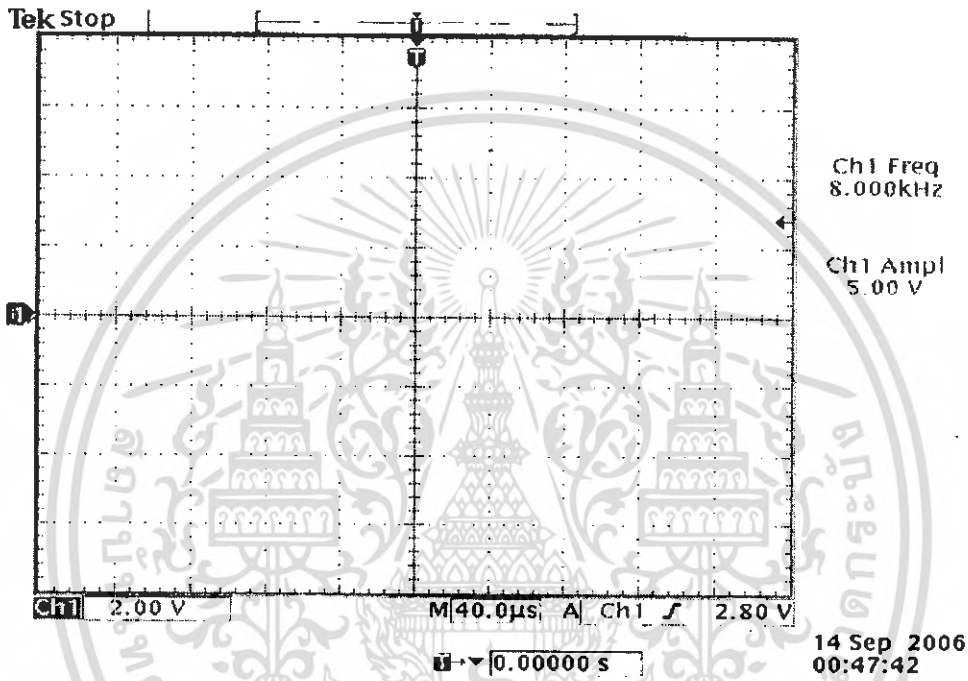
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 วงจรสร้างสัญญาณนาฬิกา

เมื่อทำการต่อรูปวงจรตามรูปที่ 3.6 ในบทที่ 3 จะได้สัญญาณนาฬิกาที่มีความถี่ 8 กิโลเฮิรตซ์ แสดงดังรูปที่ 4.1



รูปที่ 4.1 สัญญาณเอาท์พุตจากวงจรสร้างสัญญาณนาฬิกา

4.2 วงจรกรองสัญญาณความถี่ต่ำผ่าน (Low Pass Filter)

เมื่อทำการต่อวงจรกรองสัญญาณความถี่ต่ำผ่านตามที่ได้ออกแบบไว้แล้วทำการป้อนสัญญาณไซน์ขนาด 1 Vp-p เข้าไปโดยค่อย ๆ ทำการเปลี่ยนความถี่ไปเรื่อย ๆ จะได้ผลดังตารางที่

ตารางที่ 4.1 ผลการทดลองวงจร Low Pass Filter (เมื่อป้อนอินพุต 1 Vp-p)

ความถี่ (กิโลเฮิรตซ์)	แรงดันเอาท์พุท (โวลต์)	อัตราขยาย	อัตราขยาย(dB)
100	0.94	0.94	-0.5374
200	0.91	0.91	-0.8191
300	0.92	0.92	-0.7242
400	0.93	0.93	-0.6303

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่ (กิโลเฮิรตซ์)	แรงดันเอาท์พุท (โวลต์)	อัตราขยาย	อัตราขยาย(dB)
500	0.93	0.93	-0.6303
800	0.94	0.94	-0.5374
1k	0.93	0.93	-0.6303
2k	0.72	0.72	-0.8533
3k	0.47	0.47	-6.5580
4k	0.32	0.32	-9.897
5k	0.26	0.26	-11.7005
6k	0.23	0.23	-12.7644
7k	0.21	0.21	-13.5556
8k	0.20	0.20	-13.9794

4. 3 วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (ด้านส่ง)

จากการออกแบบวงจรในบทที่ 3 จะได้วงจรที่แปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลเมื่อทำการทดสอบด้วยการป้อนโวลเตจเข้าไปที่อินพุตของวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลจะได้ผลการทดลองดังตารางที่ 4.2

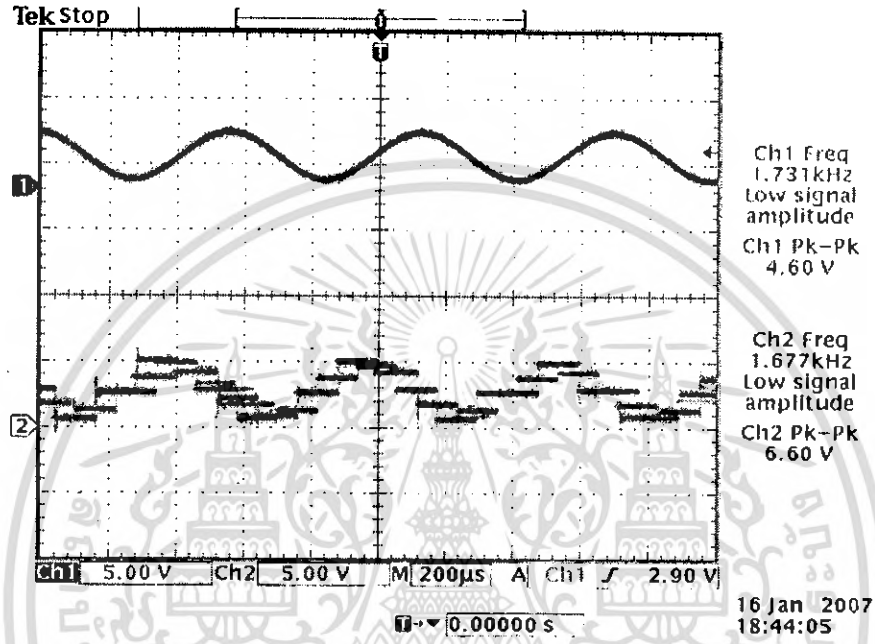
ตารางที่ 4.2 ผลการทดลองวงจร แปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

แรงดันอนาลอกอินพุต (โวลต์)	โลจิกของเอาท์พุตที่ดูจากแอลอีดี	แรงดันดิจิตอลเอาท์พุตที่วัดได้ (ฐานสิบ)
0	00000000	0
0.5	00011001	25
1	00110010	50
1.5	01001011	75
2	01100101	101
2.5	01111110	126
3	10010111	151
3.5	10101111	175
4	11001000	200
4.5	11100010	226
5	11111111	255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. 4 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก (ด้านรับ)

เมื่อทำการต่อรูปวงจรตามรูปที่ 3.4 และทำการใส่ค่าเอาท์พุตที่ได้จากวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลที่ทำการแปลงสัญญาณรูปไซน์ เข้าไปที่อินพุตของวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก จะได้ผลการทดลองที่ 4.2



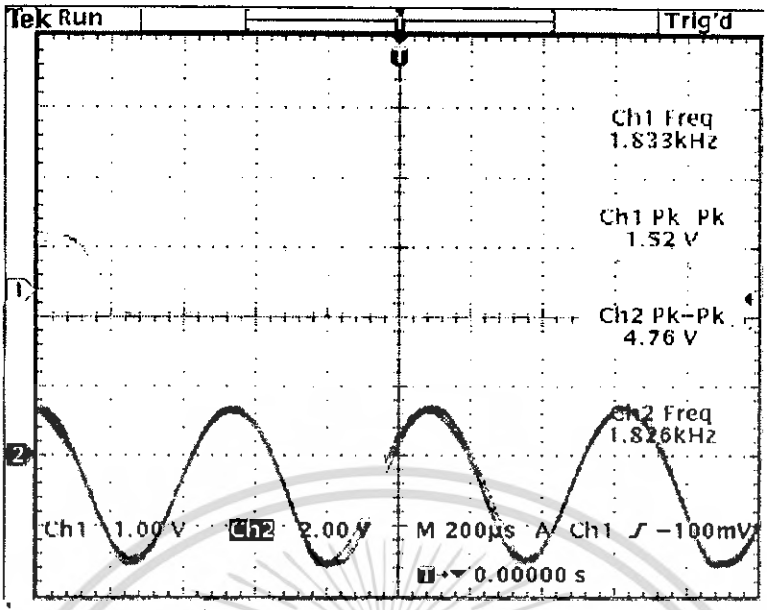
รูปที่ 4.2 สัญญาณเอาท์พุตจากวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก

โดย Ch1 คือสัญญาณอินพุตรูปไซน์

Ch2 คือสัญญาณเอาท์พุตที่ได้จากวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล

จะเห็นได้ว่ารูปสัญญาณเอาท์พุตที่ได้จะมีลักษณะเป็นขั้น ๆ ไม่ต่อเนื่องกันเมื่อนำสัญญาณเอาท์พุตที่ได้จากการแปลงสัญญาณนี้มาผ่านวงจรกรองความถี่ต่ำผ่านจะได้ผลตามรูปที่ 4.3 ช่องสัญญาณที่ 2 ซึ่งจะเห็นว่ารูปสัญญาณที่ได้มีความต่อเนื่องเหมือนกับสัญญาณดั้งเดิมมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 สัญญาณเอาท์พุตจากวงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก เมื่อทำการผ่านวงจรกรองความถี่ต่ำผ่าน

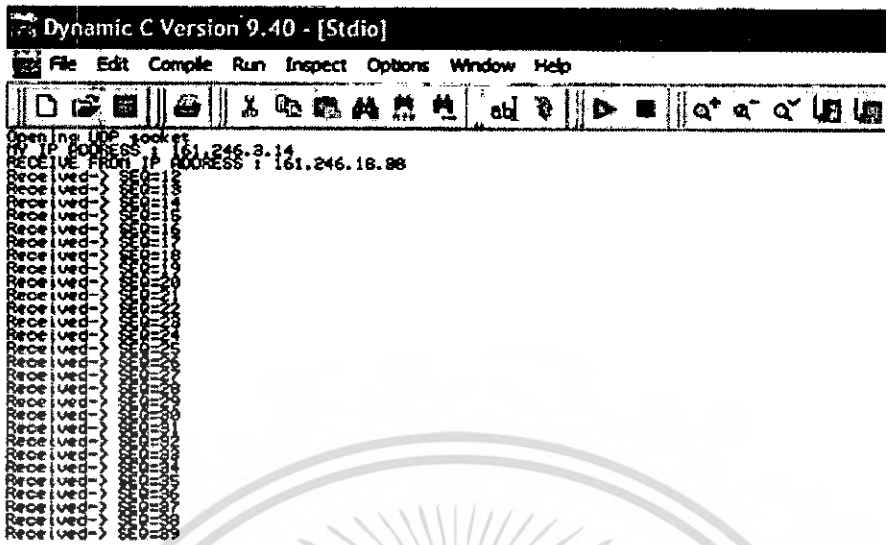
โดย Ch1 คือสัญญาณอินพุตรูปไซน์

Ch2 คือสัญญาณเอาท์พุตที่ได้จากวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล ที่ผ่านวงจรกรองความถี่ต่ำผ่าน

4.5 การรับส่งข้อมูลผ่านแพ็กเก็ตยูดีพีข้ามเน็ตเวิร์ก

เมื่อทำการเขียนโปรแกรมด้วยโปรแกรมไดนามิค ซี ให้แรบบิทโมดูลทางด้านส่งและทางด้านรับแล้วทำการส่งข้อมูลใส่ลงในแพ็กเก็ตยูดีพีข้ามวงแลน โดยแรบบิททางด้านส่งอยู่ที่ฝั่งของเน็ตเวิร์ค 161.246.18.0 โดยใช้ไอพีเบอร์ 161.246.18.38 ส่วนทางด้านรับอยู่ที่ฝั่งเน็ตเวิร์ค 161.246.3.0 โดยใช้ไอพีเบอร์ 161.246.3.14 โดยทางด้านรับจะทำการแสดงเบอร์ไอพีทางด้านส่งและไอพีทางด้านรับ และแสดงข้อมูลที่ใส่ลงไปในยูดีพีดาตาแกรม ซึ่งแสดงเป็นค่าตัวเลขของแพ็กเก็ตที่ทำการส่งไปและทางด้านรับสามารถรับได้ ซึ่งแสดงผลได้ดังรูปที่ 4.4

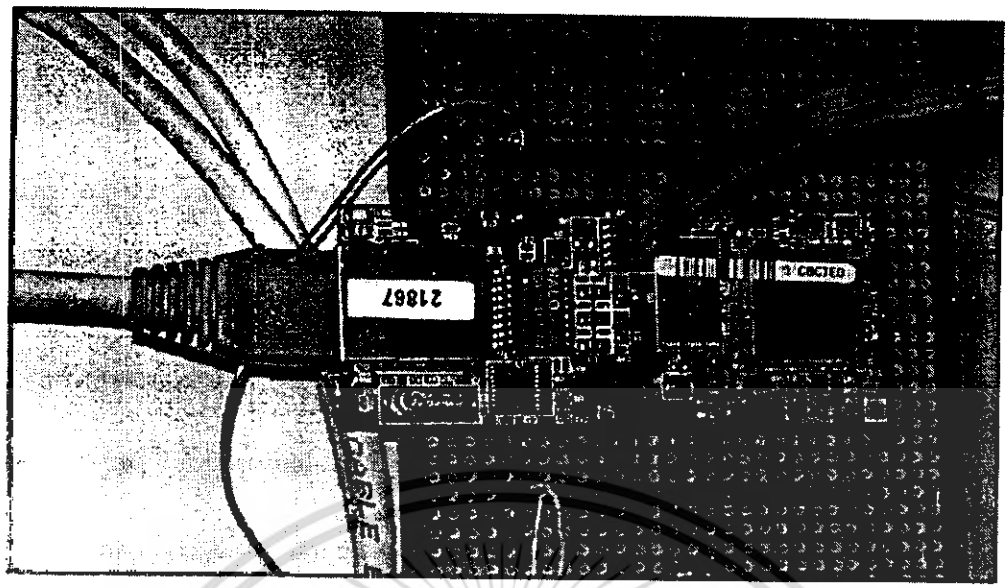
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงผลที่ได้ทางด้านรับเมื่อทำการส่งข้อมูลผ่านแพ็กเก็ตยูดีพี

รูปที่ 4.5 แสดงผลที่ได้ทางด้านส่งเมื่อไม่สามารถทำการส่งข้อมูลผ่านแพ็กเก็ตยูดีพีได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



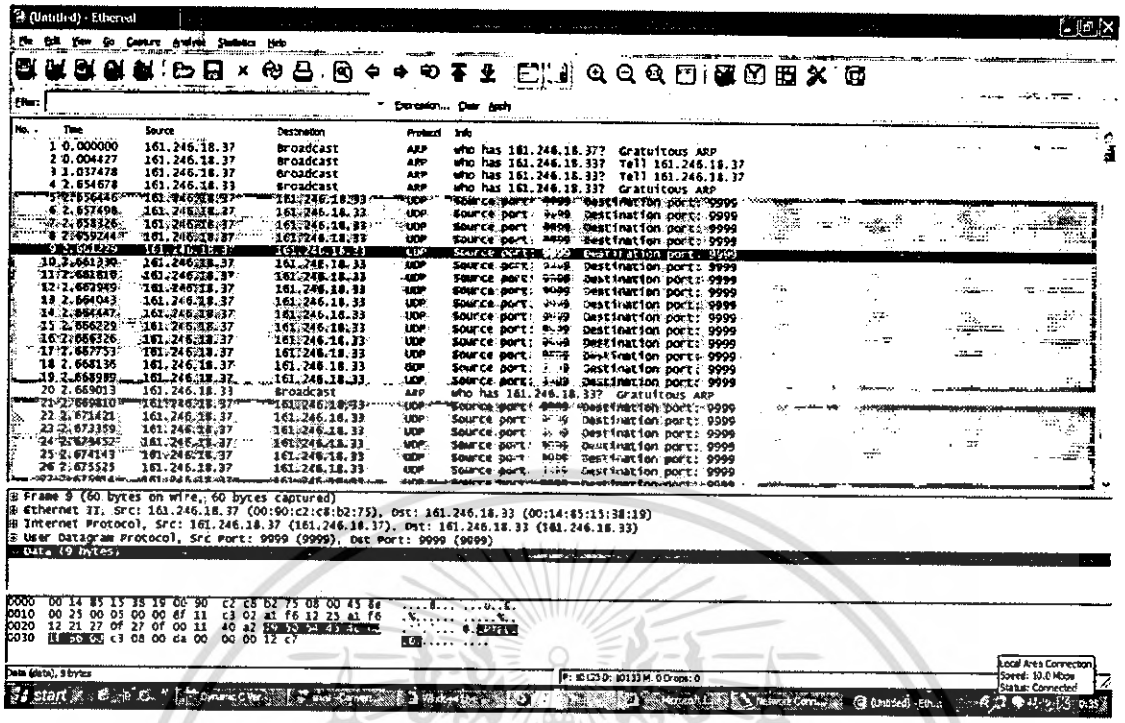
รูปที่ 4.6 แสดงภาพของแรบบิทโมดูลขณะทำการส่งข้อมูล

4.6 การส่งข้อมูลต่าง ๆ ผ่านแพ็กเก็ตยูดีพีด้วยแรบบิทโมดูล

เมื่อทำการเขียนโปรแกรมด้วยโปรแกรมไดนามิก ซี ให้แรบบิทโมดูลทางด้านส่งแล้วทำการส่งข้อมูลใส่ลงในแพ็กเก็ตยูดีพีและตรวจจับแพ็กเก็ตข้อมูลโดยใช้โปรแกรมอีเธอร์เรียล (Ethereal) โดยกำหนดให้แรบบิททางด้านส่งใช้ไอพีแอดเดรสเบอร์ 161.246.18.37 ทำการส่งให้ด้านรับที่ไอพีแอดเดรส 161.246.18.33 จะได้ผลการทดลองตามรูป

รูปที่ 4.7 เมื่อทำการเชื่อมต่อแรบบิทโมดูลเข้ากับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

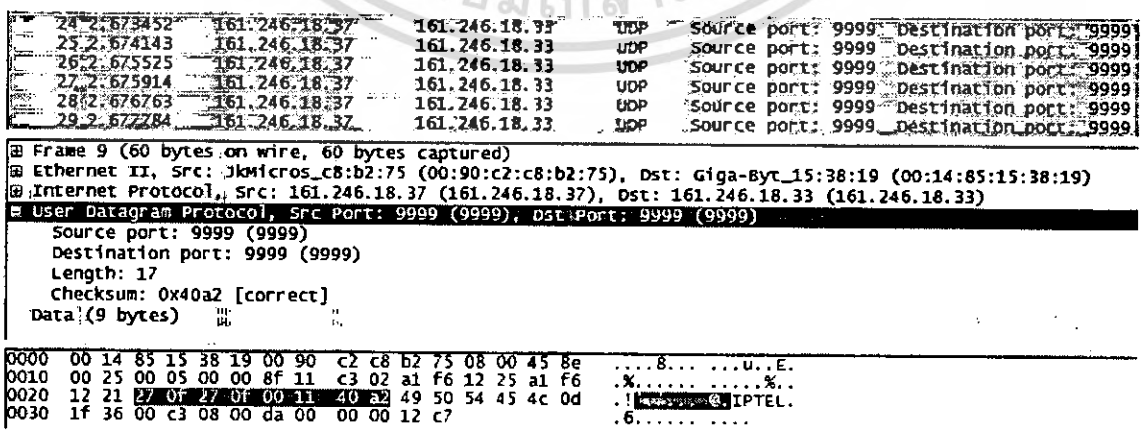


รูปที่ 4.8 ผลที่ได้จากโปรแกรมอีเธอร์เรียลเมื่อทำการส่งแพ็กเก็ตยูดีพีจากด้านส่งไปด้านรับ

จากรูปโปรแกรมอีเธอร์เรียลจะบอกถึงประเภทของโปรโตคอลที่ใช้ทำการติดต่อกันและยังบอกถึงไอพีแอดเดรสต้นทางที่ทำการส่งข้อมูล ไอพีแอดเดรสปลายทางและหมายเลขพอร์ตทั้งทางด้านส่งและด้านรับซึ่งในที่นี้ในโปรแกรมได้กำหนดเป็นพอร์ต 9999 นอกจากนี้โปรแกรมอีเธอร์เรียลยังบอกถึงความยาวของข้อมูลและข้อมูลต่าง ๆ ที่อยู่ในแพ็กเก็ตได้อีกด้วย

4.7 ด้านส่งส่งแพ็กเก็ต CALL ไปทางด้านรับ

เมื่อมีการเรียกใช้ทางด้านส่ง ด้านส่งจะรับรู้และทำการส่งแพ็กเก็ต CALL ไปทางด้านรับเพื่อบอกให้ด้านรับทราบว่ามีการติดต่อเข้ามา



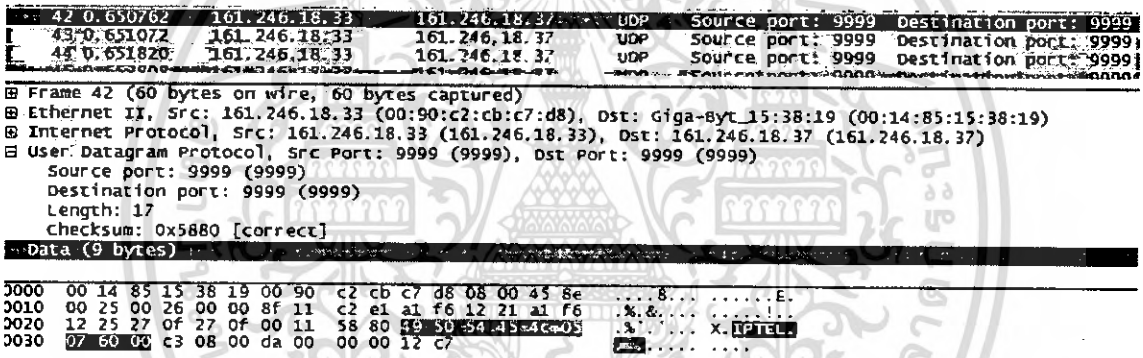
รูปที่ 4.9 เมื่อด้านส่งทำการส่งแพ็กเก็ต CALL ไปทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงถึงข้อมูลต่าง ๆ ของแพ็กเก็ต CALL ซึ่งเป็นแพ็กเก็ตประเภทยูดีพี ที่มีเฮดเดอร์ทั้งหมด 8 ไบต์ บอกพอร์ตต้นทาง 2 ไบต์ พอร์ตปลายทาง 2 ไบต์ บอกขนาดความยาวของแพ็กเก็ต 2 ไบต์ และส่วนของเช็คซัม (Checksum) อีก 2 ไบต์ ส่วนต่อมาคือข้อมูลที่อยู่ในแพ็กเก็ต CALL ซึ่งในที่นี้ได้เขียนโปรแกรมให้เขียนขึ้นต้นเฟรมว่า "IPTEL" โดย "I" มีรหัสแอสกีเป็น 49 ,"P" มีรหัสแอสกีเป็น 50, "T" มีรหัสแอสกีเป็น 54, "E" มีรหัสแอสกีเป็น 45, " L" มีรหัสแอสกีเป็น 4c ตามด้วยชนิดของเฟรมยูดีพีที่กำหนดในโปรแกรมในที่นี้คือ CALL ซึ่งกำหนดในโปรแกรมเป็นเลข 13 เลขฐาน 16 คือ 0d นั่นเอง

4.8 ด้านรับตอบกลับแพ็กเก็ต ACK ไปทางด้านส่ง

เมื่อด้านรับได้รับแพ็กเก็ต CALL จะรับรู้ด้วยการที่ ไฟ LED จะติดและดับสลับกันครั้งละ 3 วินาที เมื่อด้านรับทำการกดสวิตช์เพื่อรับรู้การติดต่อไฟ LED จะดับ และทางด้านรับจะทำการส่งแพ็กเก็ต ACK ไปยังด้านส่งเพื่อบอกให้ด้านส่งทราบว่าพร้อมที่จะทำการรับข้อมูลแล้ว



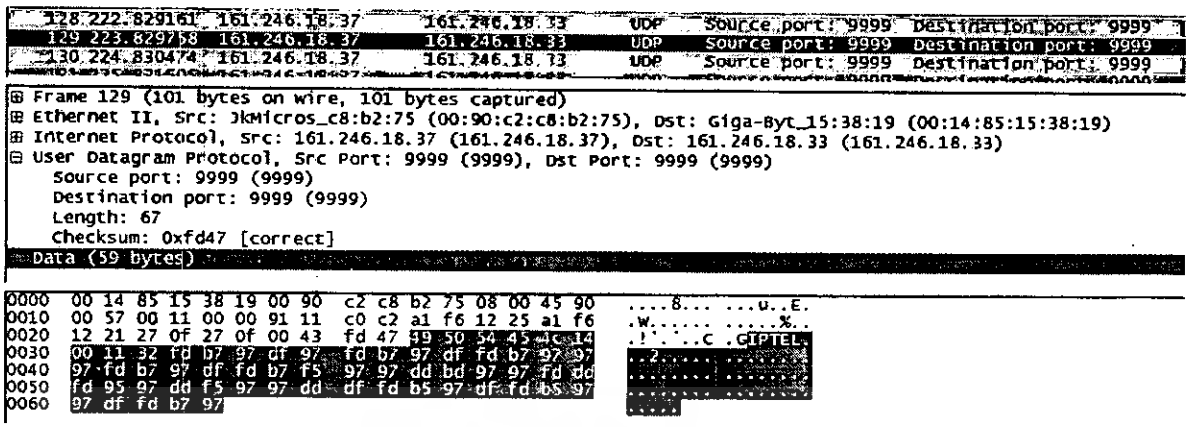
รูปที่ 4.10 เมื่อด้านรับทำการตอบกลับแพ็กเก็ต ACK ไปทางด้านส่ง

จากรูปแสดงถึงข้อมูลต่าง ๆ ของแพ็กเกต ACK ซึ่งเป็นแพ็กเก็ตประเภทยูดีพีมีลักษณะของเฮดเดอร์เหมือนกับแพ็กเก็ต CALL และส่วนของข้อมูลขึ้นต้นเฟรมด้วย "IPTEL" และ 05 บ่งบอกถึง ชนิดของแพ็กเก็ตนี้ว่าเป็น ACK

4.9 ด้านส่งส่งแพ็กเก็ต DATA ไปทางด้านรับ

เมื่อด้านรับตอบกลับมาด้วยแพ็กเก็ต ACK และด้านส่งได้รับแล้วจะมีไฟ LED ติดเพื่อบอกว่าด้านรับพร้อมที่จะรับข้อมูลให้ด้านส่งสามารถส่งข้อมูลได้

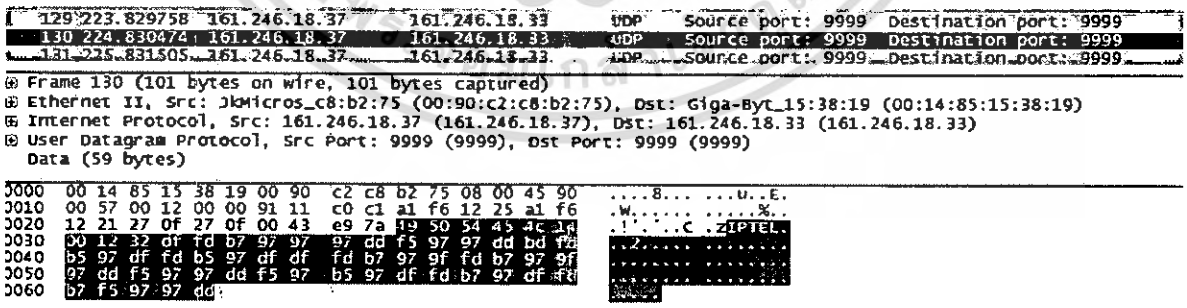
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 เมื่อด้านส่งทำการส่งแพ็กเก็ต DATA ไปทางด้านรับ

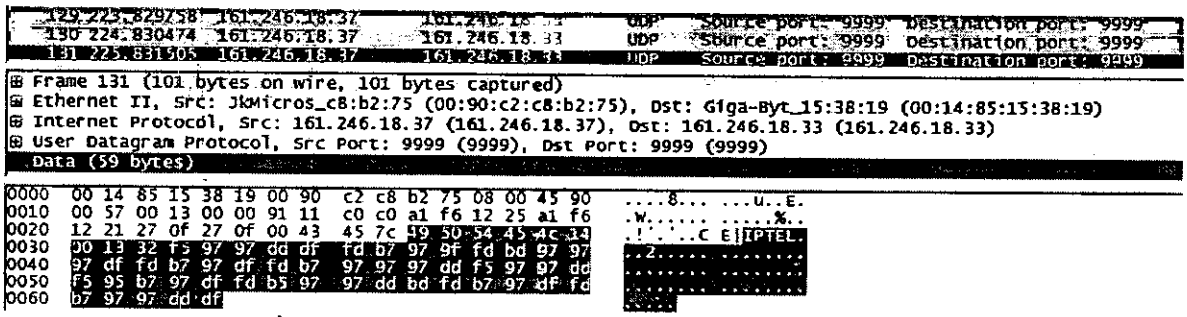
จากรูปแสดงถึงข้อมูลต่าง ๆ ของแพ็กเก็ต DATA ซึ่งเป็นแพ็กเก็ตประเภทยูดีพีในส่วนของข้อมูลขึ้นต้นเฟรมด้วย "IPTEL" และตามด้วยชนิดของแพ็กเก็ตว่าเป็น DATA ในที่นี้เลขฐาน 16 แสดงเป็น 14 แปลงเป็นเลขฐาน 10 คือ 20 ตามที่ได้กำหนดไว้ในโปรแกรม ลำดับหมายเลขของแพ็กเก็ตคือ 00 11 หมายถึงว่าแพ็กเก็ตนี้เป็นแพ็กเก็ตลำดับที่ 17 ส่วนความยาวของข้อมูลคือ 32 แปลงเป็นเลขฐาน 10 ได้เป็น 50 ไบต์ ซึ่งก็คือข้อมูลเสียงที่รับจากวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลจำนวน 50 ไบต์นั่นเอง แต่ในโปรแกรมอีเธอร์เรียลจะบอกความยาวของข้อมูลทั้งหมดที่อยู่ในแพ็กเก็ตยูดีพีนี้คือ 59 ไบต์ เป็นไบต์ที่กำหนดค่าต่าง ๆ ต้นเฟรมข้อมูลได้แก่ "IPTEL",ชนิดแพ็กเก็ต,ลำดับหมายเลขของแพ็กเก็ต ,ความยาวของข้อมูลในเฟรมข้อมูลรวม 9 ไบต์ ข้อมูลเสียงอีก 50 ไบต์รวมทั้งหมด 59 ไบต์

ในการทดลองได้สังเกตที่ลำดับหมายเลขของแพ็กเก็ตจะมีการเรียงกันตามลำดับและมีการสูญหายของข้อมูลน้อย จนแทบไม่มีผลกับการนำข้อมูลมาแปลงกับเป็นสัญญาณอนาลอกดั้งเดิมเลยเมื่อกำหนดให้ทางด้านส่งส่งข้อมูลครั้งละ 50 ไบต์



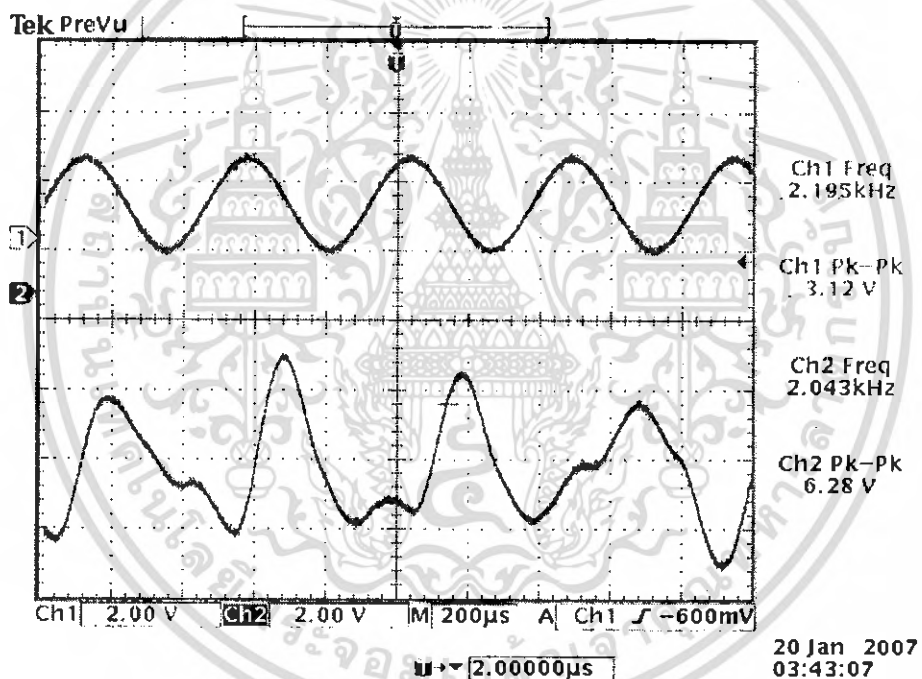
รูปที่ 4.12 แสดงแพ็กเก็ต DATA ลำดับที่ 00 12 คือลำดับที่ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 แสดงแพ็กเก็ต DATA ลำดับที่ 00 13 คือลำดับที่ 19

เมื่อทำการจับสัญญาณที่ทำการแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอกของข้อมูลสัญญาณรูปไซน์ที่ทำการแปลงเป็นสัญญาณดิจิตอลส่งผ่านแรบบิทโมดูลทางด้านส่งแล้วทำการส่งผ่านเครือข่ายอีเธอร์เน็ตมาที่ด้านรับแล้วผ่านวงจรกรองความถี่ต่ำผ่านจะได้ผลดังรูป



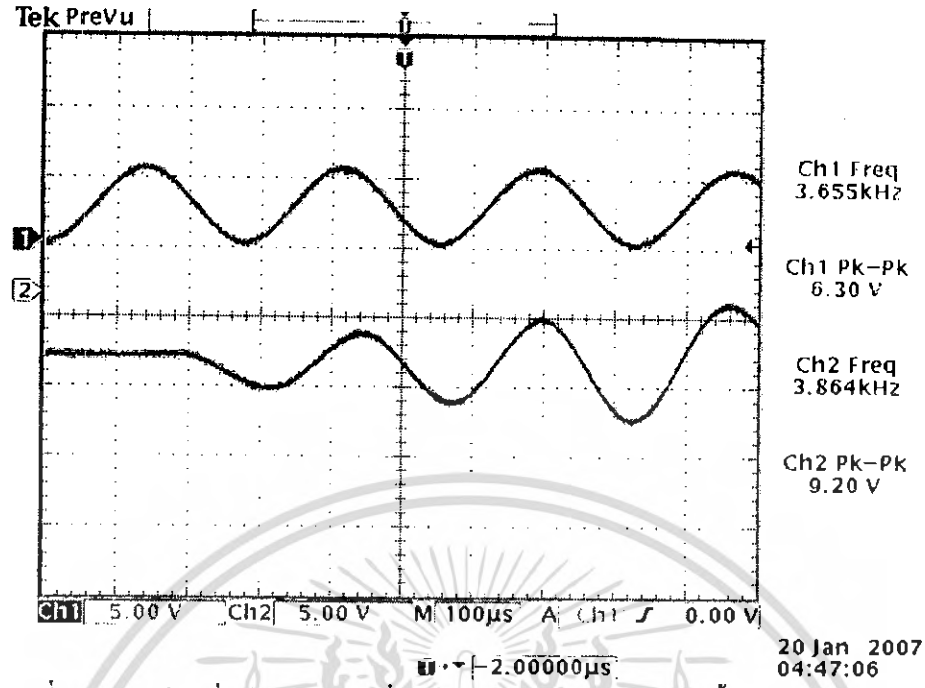
รูปที่4.14 เอาท์พุตที่ได้จากด้านรับเมื่อกำหนดให้ด้านส่งส่งข้อมูลครั้งละ 200 ไบต์

โดย

CH 1 : อินพุตรูปไซน์ที่ด้านส่ง

CH 2 : อินพุตรูปไซน์ที่ด้านรับ

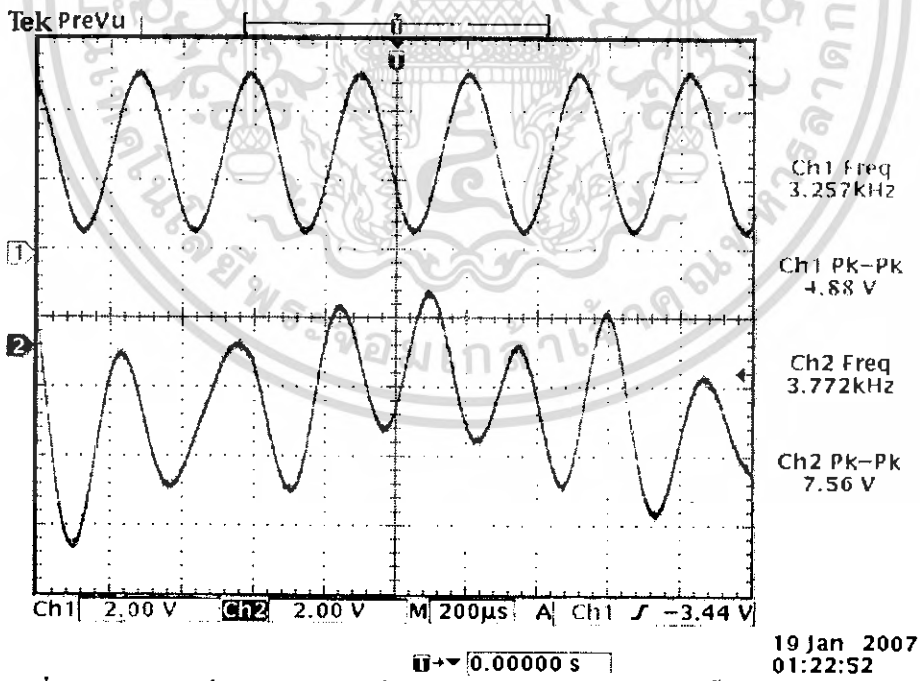
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 เอาท์พุตที่ได้จากด้านรับเมื่อกำหนดให้ด้านส่งส่งข้อมูลครั้งละ 50 ไบต์ โดย

CH 1 : อินพุตรูปไซน์ที่ด้านส่ง

CH 2 : อินพุตรูปไซน์ที่ด้านรับ



รูปที่ 4.16 เอาท์พุตที่ได้จากด้านรับเมื่อกำหนดให้ด้านส่งส่งข้อมูลครั้งละ 10 ไบต์ โดย

CH 1 : อินพุตรูปไซน์ที่ด้านส่ง

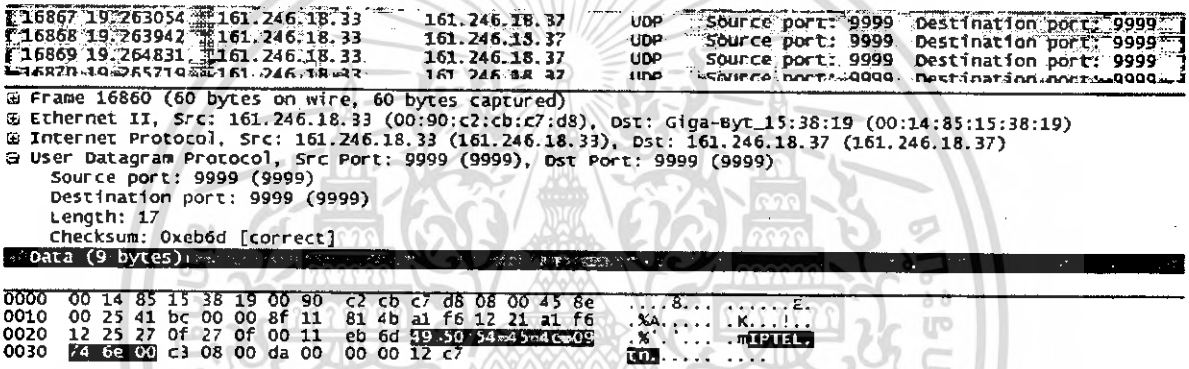
CH 2 : อินพุตรูปไซน์ที่ด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองได้ทำการทดลองเปลี่ยนจำนวนข้อมูลที่เก็บในบัฟเฟอร์ทางด้านส่งให้ทำการส่งข้อมูลครั้งละจำนวนต่าง ๆ กันโดยจะเห็นได้ว่าเมื่อส่งครั้งละ 200 ไบต์รูปร่างของสัญญาณจะมีความผิดเพี้ยนไปจากเดิมมากพอสมควรแต่มีความถี่ของสัญญาณผิดเพี้ยนจากต้นฉบับไม่มากนัก เมื่อเปลี่ยนให้ส่งข้อมูลครั้งละ 10 ไบต์รูปร่างสัญญาณไม่ค่อยผิดเพี้ยน แต่ความถี่เปลี่ยนมาก และเมื่อเปลี่ยนมาส่งครั้งละ 50 ไบต์ จะเห็นว่ารูปร่างสัญญาณและความถี่ไม่ผิดเพี้ยนไปจากเดิมมากนักจึงเลือกให้ด้านส่ง ทำการส่งข้อมูลครั้งละ 50 ไบต์

4.10 ด้านรับส่งแพ็กเก็ต END ไปทางด้านส่ง

เมื่อด้านรับทำการยกเลิกการติดต่อจะมีการส่งแพ็กเก็ต END เพื่อบอกด้านส่งว่าให้ยกเลิกการติดต่อ



รูปที่ 4.17 เมื่อด้านรับทำการส่งแพ็กเกต END ไปทางด้านส่ง

จากรูปแสดงถึงข้อมูลต่าง ๆ ของแพ็กเก็ต END ซึ่งเป็นแพ็กเก็ตประเภทยูดีพีมีลักษณะของข้อมูลขึ้นต้นเฟรมด้วย "IPTEL" และ 09 บ่งบอกถึง ชนิดของแพ็กเก็ตนี้ว่าเป็น END

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
วิจารณ์และสรุปผล

จากการศึกษาข้อมูลต่างๆและได้ลงมือทำโครงงานนี้ทำให้ทราบว่าในการส่งสัญญาณเสียงผ่านโครงข่ายอินเตอร์เน็ตประกอบด้วย วงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล เพื่อแปลงสัญญาณเสียงเป็นสัญญาณดิจิตอลให้สามารถส่งข้อมูลผ่านโครงข่ายอินเตอร์เน็ต แต่เนื่องจากสัญญาณเสียงพูดที่ทำการส่งนั้นต้องมีการต่อเนื่องตามเวลาจริง และมีการส่งข้อมูลผ่านไปในรูปแบบของแพ็กเก็ตยูดีพี จึงอาจทำให้สัญญาณมีความผิดเพี้ยนไปบ้าง และเนื่องจากต้องมีการผ่านวงจรหลาย ๆ วงจรอาจจะทำให้เกิดการติดขัดของสัญญาณขึ้นได้ส่วนสัญญาณที่ได้ทางด้านรับต้องนำข้อมูลที่ได้นั้นไปแปลงเป็นสัญญาณอนาลอกแล้วจึงนำเข้าวงจรกรองความถี่ต่ำผ่านเพื่อกรองเอาส่วนที่เป็นความถี่สูงออกเพื่อให้สัญญาณมีความเรียบและนำเข้าวงจรขยายสัญญาณเนื่องจากข้อมูลที่ได้มีขนาดลดลงแต่ในโครงงานนี้พบปัญหาเรื่องของสัญญาณรบกวนที่เกิดจากตัวอุปกรณ์ต่าง ๆ สายต่อพ่วงภายในวงจร ส่วนทางการส่งข้อมูลผ่านแรบบิทโมดูลได้ทดลองส่งข้อมูลผ่านระหว่างโครงข่ายย่อยภายในสถาบัน โดยกำหนดพอร์ตที่ใช้ขึ้นมาเองไม่ให้ตรงกับพอร์ตที่เป็นที่รู้จัก(well known port) ซึ่งจากการทดลองสามารถทำการส่งข้อมูลข้ามไปได้ แต่ไม่สามารถทำการใช้คำสั่งปิง (ping) เพื่อตรวจสอบการเชื่อมต่อข้ามโครงข่ายได้ อาจเพราะมีการห้ามใช้พอร์ตสำหรับคำสั่งนี้

ปัญหาที่พบในการทดลอง

1. ในการส่งข้อมูลจังหวะในการแปลงข้อมูลทางด้านรับและด้านส่งอาจไม่ตรงกันพอดีทำให้การรับข้อมูลอาจมีความผิดเพี้ยนได้
2. ในการส่งข้อมูลต้องผ่านวงจรหลายวงจรทำให้อาจเกิดการดีเลย์
3. ถ้ามีการส่งข้อมูลเข้าไปในแต่ละแพ็กเก็ตเป็นจำนวนมากเกินไปจะทำให้รูปสัญญาณมีความผิดเพี้ยนสูง
4. เกิดสัญญาณรบกวนจากภายนอกและสายสัญญาณต่าง ๆ ภายในวงจร
5. เนื่องจากไมโครโปรเซสเซอร์แรบบิท 3000 ยังไม่มีการใช้งานอย่างแพร่หลาย ทำให้ขาดตัวอย่างในการศึกษาและเป็นไมโครโปรเซสเซอร์ที่ยังไม่เคยใช้งานมาก่อนจึงใช้งานได้ไม่เต็มความสามารถของไมโครโปรเซสเซอร์
6. ไม่คุ้นเคยกับชุดคำสั่งภาษาไดนามิค ซี ทำให้แอพลิเคชันที่พัฒนาออกมาขาดความสมบูรณ์
7. เนื่องจากบอร์ดและสายลงโปรแกรมแรบบิทโมดูลมีราคาแพงจึงต้องทำการศึกษาและสร้างขึ้นมาใช้งานเอง

แนวทางการพัฒนาต่อไป

สามารถนำวงจรทางด้านรับและด้านส่งมารวมกันและทำการพัฒนาโปรแกรมเพื่อให้สามารถทำการส่งและรับสัญญาณได้แบบฟูลดูเพลกซ์ (Full-Duplex)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. วิสันต์ อาชาเดโชพล,ระบบโทรศัพท์ดิจิตอล,หจก.สำนักพิมพ์ฟิสิกส์เซ็นเตอร์
2. สมยศ จุฌณะปิยะ, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์, สถาบันเทคโนโลยีพระจอมเกล้า-เจ้าคุณทหารลาดกระบัง พิมพ์ครั้งที่ 5 พ.ศ. 2546
3. Kamal Hyder and Bob Perrin,Embedded Systems Design Using the Rabbit 3000 Microprocessor
4. Jan Axelson ,Embedded Ethernet and Internet complete
5. http://www.rabbitsemiconductor.com/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทางด้านส่ง

```
/*

pkt format
"IPTEL" ,type,sequence,length,data

*/

#class auto
#define MAX_UDP_SOCKET_BUFFERS 10

/*********************** IP Address & PORT ***************************/

#define TCPCONFIG 1     /* #define _PRIMARY_STATIC_IP   "161.246.18.33"
                           #define _PRIMARY_NETMASK     "255.255.255.0"
                         #ifndef MY_NAMESERVER
                           #define MY_NAMESERVER        "161.246.18.4"
                         #endif
                         #ifndef MY_GATEWAY
                           #define MY_GATEWAY           "161.246.18.1"
                         #endif                    */

#define UDP_PORT 9999
#define LOCAL_PORT 9999
#define DEFAULT_REMOTE_UDP "161.246.18.33"      //destination IP

#memmap xmem
#use "dcrtcp.lib"                                 //UDP Library

/*********************** Constant ***********************************/

#define pktsize 1500
#define rxsize 1000

/************************* Tags ***********************************/

#define DATA 20
#define CALL 13
#define ACK 5
#define END 9

/************************* variable ***********************************/

unsigned char seqH;                     //seq num higher byte
unsigned char seqL;                     //seq num lower byte
char rx_buf[50];                        //receive buffer
int rx_pt;                              //pointer for receive buffer
int pass;
int ready;

udp_Socket udpSocket;

/************************** subs ***********************************/

void Lchannel();

/************************ send packets ******************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int send_packet(char *msg,int len)
      {
      int retval;
      retval = udp_send(&udpSocket,msg,len);
      if(retval<0){ return 0; }
      tcp_tick(NULL);
      return 1;
      }

int pack_packet(char *msg,int len,int type,int mode)
      {
      char pkt[pktsize];
      int i,j;
   #GLOBAL_INIT
      {     seqL=0;
            seqH=0;
      }
   if(mode==1){
            seqL++;
         if(seqL==0){ seqH++;}
               }
      i=0;
      pkt[i++]='I';
      pkt[i++]='P';
      pkt[i++]='T';
      pkt[i++]='E';
      pkt[i++]='L';
      pkt[i++]=type;
      pkt[i++]=seqH;
      pkt[i++]=seqL;
      pkt[i++]=len;

      for(j=0;j<len;j++){ pkt[i+j]=msg[j]; }
      return send_packet(pkt,i+len);
      }

/********************** receive packets ****************************/

int unpack(char *msg,int len,int mode)
      {
      char pkt[pktsize];
      int i;

 if(msg[0]=='I'&&msg[1]=='P'&&msg[2]=='T'&&msg[3]=='E'&&msg[4]=='L')
{
      seqH=msg[6];
      seqL=msg[7];

      if(msg[5]==ACK)
      {
      BitWrPortI(PBDR,&PBDRShadow,1,7);    //receive ACK LED ON
      ready=1;
      }

      else if(msg[5]==END)
      {
       BitWrPortI(PBDR,&PBDRShadow,0,7); //receive ACK LED OFF
      }

  }
       }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int receive_packet(void)
      {
      static char buf[128];
      int rx_len,i;

   tcp_tick(NULL);
   rx_len = udp_recv(&udpSocket,buf,sizeof(buf));
   if(-1==rx_len){ return 0; }

   return unpack(buf,rx_len,0);
      }

/********************** initial interrupt ***********************/

void intr(void)
      {
WrPortI(PEDDR,&PEDDRShadow,0x00);     //set port E as all inputs
SetVectExtern3000(0,Lchannel);        //mapping the ISR vector
WrPortI(I0CR,&I0CRShadow,0x00);       //enable external INT0
      }

/*************************** main *********************************/

void main(void)
{
      char *msg;
      rx_pt=0;

      WrPortI(SPCR,NULL,0x80);      //set port A as input
      sock_init();
      intr();

pass = udp_open(&udpSocket,LOCAL_PORT,resolve(DEFAULT_REMOTE_UDP),
UDP_PORT, NULL);

      if(!pass)
       {
            printf("udp open error\n");
            exit(1);
       }

      while(BitRdPortI(PBDR,0)==0)
       {
            ready=0;
         while(ready==0)
            {
            pack_packet(msg,0,CALL,1);
            receive_packet();
            }

      if((BitRdPortI(PBDR,7)==1)&&(ready==1))
      {
         while(1)
            {
            tcp_tick(&udpSocket);

//INT0 on PE0,falling edge
            WrPortI(I0CR,&I0CRShadow,0x05);

//send DATA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(rx_pt>=50)
              {
              pack_packet(rx_buf,rx_pt,DATA,1);
              rx_pt=0;
               }

              }
             }
        }
}

/************ External trigger Interrupt Routine *****************/

nodebug root interrupt void Lchannel()
{

    WrPortI(SPCR,NULL,0x80);
    rx_buf[rx_pt++]=RdPortI(PADR);

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมทางด้านรับ

```
/*

pkt format
"IPTEL" ,type,sequence,length,data

*/

#class auto
#define MAX_UDP_SOCKET_BUFFERS 10

/*********************** IP Address & PORT ***************************/

#define TCPCONFIG 1     /* #define _PRIMARY_STATIC_IP  "161.246.18.33"
                           #define _PRIMARY_NETMASK    "255.255.255.0"
                         #ifndef MY_NAMESERVER
                           #define MY_NAMESERVER       "161.246.18.4"
                         #endif
                         #ifndef MY_GATEWAY
                           #define MY_GATEWAY          "161.246.18.1"
                         #endif                    */

#define LOCAL_PORT 9999
#define UDP_PORT 9999
#define DEFAULT_REMOTE_UDP   "161.246.18.37"    //destination IP

#memmap xmem
#use "dcrtcp.lib"                               //UDP Library

/************************ Constant ***********************************/

#define pktsize 1500
#define txsize 8000

/************************* Tags ***************************************/

#define DATA 20
#define CALL 13
#define ACK 5
#define END 9

/************************* variable ***********************************/

unsigned char seqH;                     //seq num higher byte
unsigned char seqL;                     //seq num lower byte
char tx_buf[txsize];                    //transmit buffer
int tx_pt1,tx_pt2;                      //pointer for transmit buffer
int tx_start,bufsize;
int pass,ack,data;

udp_Socket udpSocket;

/************************** subs **************************************/

void codec_isr();

/************************ send packets ********************************/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int send_packet(char *msg,int len)
      {
      int retval;
      retval = udp_send(&udpSocket,msg,len);
      if(retval<0){ return 0; }
      tcp_tick(NULL);
      return 1;
       }

int pack_packet(char *msg,int len,int type,int mode)
      {
   char pkt[pktsize+10];
   int i,j;
   #GLOBAL_INIT
      {     seqL=0;
            seqH=0;
      }
   if(mode==1){
            seqL++;
         if(seqL==0){ seqH++;}
              }
      i=0;
      pkt[i++]='I';
      pkt[i++]='P';
      pkt[i++]='T';
      pkt[i++]='E';
      pkt[i++]='L';
      pkt[i++]=type;
      pkt[i++]=seqH;
      pkt[i++]=seqL;
      pkt[i++]=len;

      for(j=0;j<len;j++){ pkt[i+j]=msg[j]; }
      return send_packet(pkt,i+len);
      }

/********************** receive packets ****************************/

int unpack(char *msg,int len,int mode)
      {
      char pkt[pktsize];
      int i,count;

if(msg[0]=='I'&&msg[1]=='P'&&msg[2]=='T'&&msg[3]=='E'&&msg[4]=='L')
{
      seqH=msg[6];
      seqL=msg[7];

      if(msg[5]==CALL)
      {
          Costate                          //receive CALL LED ON-OFF
            {
            BitWrPortI(PBDR,&PBDRShadow,1,7);
            waitfor(DelaySec(3));
            BitWrPortI(PBDR,&PBDRShadow,0,7);
            waitfor(DelaySec(3));
            }
       }

      else if(msg[5]==DATA)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            data=1;
            for(i=0;i<msg[8];i++)
            {
            tx_buf[tx_pt1++]=msg[i+9];    //keep data in buffer
            bufsize++;
            if(tx_pt1>=txsize){ tx_pt1=0;}
            if(tx_pt2>=txsize){ tx_pt2=0;}
            }
            return 1;
        }
}
      }

int receive_packet(void)
      {
      static char buf[128];
      int rx_len,i;
      tcp_tick(NULL);
      rx_len = udp_recv(&udpSocket,buf,sizeof(buf));
      if(-1==rx_len){ return 0; }
      return unpack(buf,rx_len,0);
      }

/********************** initial interrupt ************************/

void intr(void)
      {
WrPortI(PEDDR,&PEDDRShadow,0x00);     //set port E as all inputs
SetVectExtern3000(0,codec_isr);       //mapping the ISR vector
WrPortI(I0CR,&I0CRShadow,0x00);       //enable external INT0
      }

/*************************** main ********************************/

void main(void)
{
      char *msg;
      WrPortI(SPCR,NULL,0x84);        //set port A as output
      sock_init();
      intr();
      tx_pt1=0;
      tx_pt2=0;

pass = udp_open(&udpSocket,LOCAL_PORT,resolve(DEFAULT_REMOTE_UDP),
UDP_PORT, NULL);

      if(!pass)
      {
            printf("udp open error\n");
            exit(1);
      }

      while(1)
      {
      data=0;
      ack=0;
      tcp_tick(&udpSocket);
      receive_packet();

      if((BitRdPortI(PBDR,0)==0)&&(data==0))
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            BitWrPortI(PBDR,&PBDRShadow,0,7);
            pack_packet(msg,0,ACK,1);
            }

      else if(BitRdPortI(PBDR,0)==1)
            {
            pack_packet(msg,0,END,1);
            }
      while((BitRdPortI(PBDR,0)==0)&&(data==1))
            {
            tcp_tick(&udpSocket);
            receive_packet();
            if(bufsize>200){ tx_start=1; }

//INT0 on PE0,rising edge
            WrPortI(I0CR,&I0CRShadow,0x09);
            }
        }
}

/************ External trigger Interrupt Routine *****************/

nodebug root interrupt void codec_isr()
{
      WrPortI(SPCR,NULL,0x84);
      if(tx_start==1&&bufsize>0)
      {
      if(tx_buf[tx_pt2]!=0)
            {
            WrPortI(PADR,NULL,tx_buf[tx_pt2++]);
            }
            else
            {
            WrPortI(PADR,NULL,tx_buf[tx_pt2-1]);
            tx_pt2++;
            }
           bufsize--;
                                          }
      else{ WrPortI(PADR,NULL,tx_buf[tx_pt2]); }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

November 1999

ADC0801/ADC0802/ADC0803/ADC0804/ADC0805 8-Bit µP Compatible A/D Converters

General Description

The ADC0801, ADC0802, ADC0803, ADC0804 and ADC0805 are CMOS 8-bit successive approximation A/D converters that use a differential potentiometric ladder—similar to the 256R products. These converters are designed to allow operation with the NSC800 and INS8080A derivative control bus with TRI-STATE output latches directly driving the data bus. These A/Ds appear like memory locations or I/O ports to the microprocessor and no interfacing logic is needed.

Differential analog voltage inputs allow increasing the common-mode rejection and offsetting the analog zero input voltage value. In addition, the voltage reference input can be adjusted to allow encoding any smaller analog voltage span to the full 8 bits of resolution.

Features

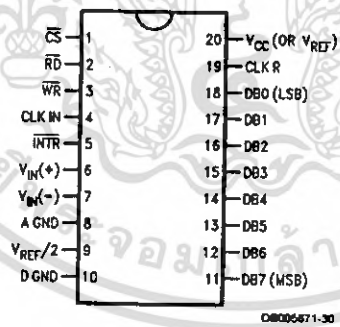
- Compatible with 8080 µP derivatives—no interfacing logic needed - access time - 135 ns
- Easy interface to all microprocessors, or operates "stand alone"
- Differential analog voltage inputs
- Logic inputs and outputs meet both MOS and TTL voltage level specifications
- Works with 2.5V (LM336) voltage reference
- On-chip clock generator
- 0V to 5V analog input voltage range with single 5V supply
- No zero adjust required
- 0.3" standard width 20-pin DIP package
- 20-pin molded chip carrier or small outline package
- Operates ratiometrically or with 5 V_{DC}, 2.5 V_{DC}, or analog span adjusted voltage reference

Key Specifications

- Resolution: 8 bits
- Total error: ±¼ LSB, ±½ LSB and ±1 LSB
- Conversion time: 100 µs

Connection Diagram

ADC080X
Dual-In-Line and Small Outline (SO) Packages



DB005671-30

See Ordering Information

Ordering Information

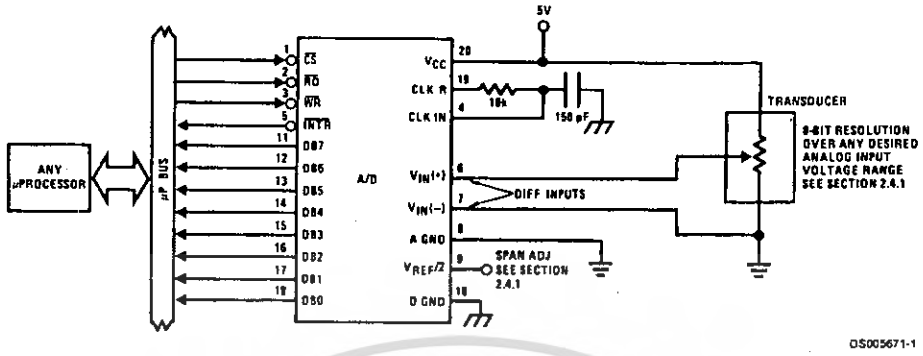
TEMP RANGE		0°C TO 70°C	0°C TO 70°C	−40°C TO +85°C
ERROR	±¼ Bit Adjusted			ADC0801LCN
	±½ Bit Unadjusted	ADC0802LCWM		ADC0802LCN
	±½ Bit Adjusted			ADC0803LCN
	±1Bit Unadjusted	ADC0804LCWM	ADC0804LCN	ADC0805LCN/ADC0804LCJ
PACKAGE OUTLINE		M20B—Small Outline	N20A—Molded DIP	

Z-80® is a registered trademark of Zilog Corp.

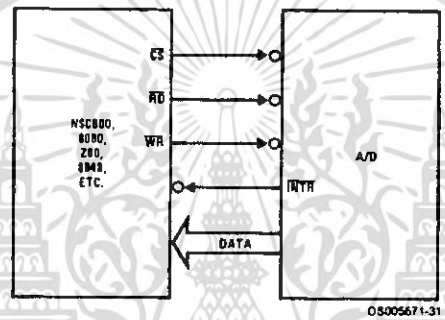
© 2001 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications



8080 Interface



Error Specification (Includes Full-Scale, Zero Error, and Non-Linearity)			
Part Number	Full-Scale Adjusted	$V_{REF}/2$=2.500 V_{DC} (No Adjustments)	$V_{REF}/2$=No Connection (No Adjustments)
ADC0801	±¼ LSB		
ADC0802		±½ LSB	
ADC0803	±½ LSB		
ADC0804		±1 LSB	
ADC0805			±1 LSB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Notes 1, 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (V_{CC}) (Note 3)	6.5V
Voltage	
Logic Control Inputs	−0.3V to +18V
At Other Input and Outputs	−0.3V to (V_{CC}+0.3V)
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (plastic)	260°C
Dual-In-Line Package (ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C
Storage Temperature Range	−65°C to +150°C
Package Dissipation at T_A=25°C	875 mW
ESD Susceptibility (Note 10)	800V

Operating Ratings (Notes 1, 2)

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
ADC0804LCJ	−40°C≤T_A≤+85°C
ADC0801/02/03/05LCN	−40°C≤T_A≤+85°C
ADC0804LCN	0°C≤T_A≤+70°C
ADC0802/04LCWM	0°C≤T_A≤+70°C
Range of V_{CC}	4.5 V_{DC} to 6.3 V_{DC}

Electrical Characteristics

The following specifications apply for V_{CC}=5 V_{DC}, $T_{MIN} \le T_A \le T_{MAX}$ and f_{CLK}=640 kHz unless otherwise specified.

Parameter	Conditions	Min	Typ	Max	Units
ADC0801: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±¼	LSB
ADC0802: Total Unadjusted Error (Note 8)	V_{REF}/2=2.500 V_{DC}			±½	LSB
ADC0803: Total Adjusted Error (Note 8)	With Full-Scale Adj. (See Section 2.5.2)			±½	LSB
ADC0804: Total Unadjusted Error (Note 8)	V_{REF}/2=2.500 V_{DC}			±1	LSB
ADC0805: Total Unadjusted Error (Note 8)	V_{REF}/2-No Connection			±1	LSB
V_{REF}/2 Input Resistance (Pin 9)	ADC0801/02/03/05	2.5	8.0		kΩ
	ADC0804 (Note 9)	0.75	1.1		kΩ
Analog Input Voltage Range	(Note 4) V(+) or V(−)	Gnd−0.05		V_{CC}+0.05	V_{DC}
DC Common-Mode Error	Over Analog Input Voltage Range		±1/16	±⅛	LSB
Power Supply Sensitivity	V_{CC}=5 V_{DC} ±10% Over Allowed V_{IN}(+) and V_{IN}(−) Voltage Range (Note 4)		±1/16	±⅛	LSB

AC Electrical Characteristics

The following specifications apply for V_{CC}=5 V_{DC} and $T_{MIN} \le T_A \le T_{MAX}$ unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
T_C	Conversion Time	f_{CLK}=640 kHz (Note 6)	103		114	μs
T_C	Conversion Time	(Notes 5, 6)	66		73	1/f_{CLK}
f_{CLK}	Clock Frequency	V_{CC}=5V, (Note 5)	100	640	1460	kHz
	Clock Duty Cycle		40		60	%
CR	Conversion Rate in Free-Running Mode	$\overline{INTR}$ tied to $\overline{WR}$ with $\overline{CS}$ =0 V_{DC}, f_{CLK}=640 kHz	8770		9708	conv/s
$t_{W(\overline{WR})L}$	Width of $\overline{WR}$ Input (Start Pulse Width)	$\overline{CS}$ =0 V_{DC} (Note 7)	100			ns
t_{ACC}	Access Time (Delay from Falling Edge of $\overline{RD}$ to Output Data Valid)	C_L=100 pF		135	200	ns
t_{1H}, t_{0H}	TRI-STATE Control (Delay from Rising Edge of $\overline{RD}$ to Hi-Z State)	C_L=10 pF, R_L=10k (See TRI-STATE Test Circuits)		125	200	ns
t_{WI}, t_{RI}	Delay from Falling Edge of $\overline{WR}$ or $\overline{RD}$ to Reset of $\overline{INTR}$			300	450	ns
C_{IN}	Input Capacitance of Logic Control Inputs			5	7.5	pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics (Continued)

The following specifications apply for V_{CC}=5 V_{DC} and $T_{MIN} \le T_A \le T_{MAX}$ unless otherwise specified.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
C_{OUT}	TRI-STATE Output Capacitance (Data Buffers)			5	7.5	pF
CONTROL INPUTS [Note: CLK IN (Pin 4) is the input of a Schmitt trigger circuit and is therefore specified separately]						
V_{IN} (1)	Logical "1" Input Voltage (Except Pin 4 CLK IN)	V_{CC}=5.25 V_{DC}	2.0		15	V_{DC}
V_{IN} (0)	Logical "0" Input Voltage (Except Pin 4 CLK IN)	V_{CC}=4.75 V_{DC}			0.8	V_{DC}
I_{IN} (1)	Logical "1" Input Current (All Inputs)	V_{IN}=5 V_{DC}		0.005	1	μA_{DC}
I_{IN} (0)	Logical "0" Input Current (All Inputs)	V_{IN}=0 V_{DC}	−1	−0.005		μA_{DC}
CLOCK IN AND CLOCK R						
V_T+	CLK IN (Pin 4) Positive Going Threshold Voltage		2.7	3.1	3.5	V_{DC}
V_T−	CLK IN (Pin 4) Negative Going Threshold Voltage		1.5	1.8	2.1	V_{DC}
V_H	CLK IN (Pin 4) Hysteresis (V_T+)−(V_T−)		0.6	1.3	2.0	V_{DC}
V_{OUT} (0)	Logical "0" CLK R Output Voltage	I_O=360 µA V_{CC}=4.75 V_{DC}			0.4	V_{DC}
V_{OUT} (1)	Logical "1" CLK R Output Voltage	I_O=−360 µA V_{CC}=4.75 V_{DC}	2.4			V_{DC}
DATA OUTPUTS AND $\overline{INTR}$						
V_{OUT} (0)	Logical "0" Output Voltage Data Outputs $\overline{INTR}$ Output	 I_{OUT}=1.6 mA, V_{CC}=4.75 V_{DC} I_{OUT}=1.0 mA, V_{CC}=4.75 V_{DC}			 0.4 0.4	 V_{DC} V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O=−360 µA, V_{CC}=4.75 V_{DC}	2.4			V_{DC}
V_{OUT} (1)	Logical "1" Output Voltage	I_O=−10 µA, V_{CC}=4.75 V_{DC}	4.5			V_{DC}
I_{OUT}	TRI-STATE Disabled Output Leakage (All Data Buffers)	V_{OUT}=0 V_{DC} V_{OUT}=5 V_{DC}	−3		 3	μA_{DC} μA_{DC}
I_{SOURCE}		V_{OUT} Short to Gnd, T_A=25°C	4.5	6		mA_{DC}
I_{SINK}		V_{OUT} Short to V_{CC}, T_A=25°C	9.0	16		mA_{DC}
POWER SUPPLY						
I_{CC}	Supply Current (Includes Ladder Current) ADC0801/02/03/04LCJ/05 ADC0804LCN/LCWM	f_{CLK}=640 kHz, $V_{REF}/2$=NC, T_A=25°C and $\overline{CS}$ =5V		 1.1 1.9	 1.8 2.5	 mA mA

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All voltages are measured with respect to Gnd, unless otherwise specified. The separate A Gnd point should always be wired to the D Gnd.

Note 3: A zener diode exists, internally, from V_{CC} to Gnd and has a typical breakdown voltage of 7 V_{DC}.

Note 4: For $V_{IN}(-) \ge V_{IN}(+)$ the digital output code will be 0000 0000. Two on-chip diodes are tied to each analog input (see block diagram) which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. Be careful, during testing at low V_{CC} levels (4.5V), as high level analog inputs (5V) can cause this input diode to conduct–especially at elevated temperatures, and cause errors for analog inputs near full-scale. The spec allows 50 mV forward bias of either diode. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 50 mV, the output code will be correct. To achieve an absolute 0 V_{DC} to 5 V_{DC} input voltage range will therefore require a minimum supply voltage of 4.950 V_{DC} over temperature variations, initial tolerance and loading.

Note 5: Accuracy is guaranteed at f_{CLK} = 640 kHz. At higher clock frequencies accuracy can degrade. For lower clock frequencies, the duty cycle limits can be extended so long as the minimum clock high time interval or minimum clock low time interval is no less than 275 ns.

Note 6: With an asynchronous start pulse, up to 8 clock periods may be required before the internal clock phases are proper to start the conversion process. The start request is internally latched, see *Figure 4* and section 2.0.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics (Continued)

Note 7: The $\overline{CS}$ input is assumed to bracket the $\overline{WR}$ strobe input and therefore timing is dependent on the $\overline{WR}$ pulse width. An arbitrarily wide pulse width will hold the converter in a reset mode and the start of conversion is initiated by the low to high transition of the $\overline{WR}$ pulse (see timing diagrams).

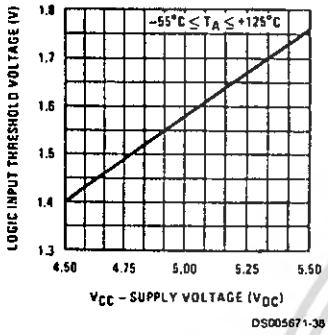
Note 8: None of these A/Ds requires a zero adjust (see section 2.5.1). To obtain zero code at other analog input voltages see section 2.5 and *Figure 7*.

Note 9: The $V_{REF}/2$ pin is the center point of a two-resistor divider connected from V_{CC} to ground. In all versions of the ADC0801, ADC0802, ADC0803, and ADC0805, and in the ADC0804LCJ, each resistor is typically 16 kΩ. In all versions of the ADC0804 except the ADC0804LCJ, each resistor is typically 2.2 kΩ.

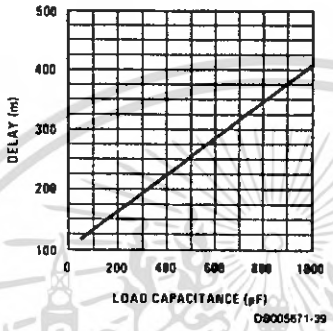
Note 10: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Typical Performance Characteristics

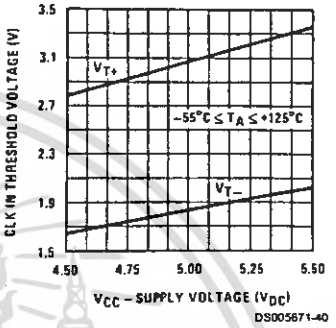
Logic Input Threshold Voltage vs. Supply Voltage



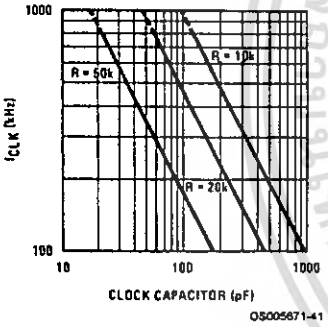
Delay From Falling Edge of RD to Output Data Valid vs. Load Capacitance



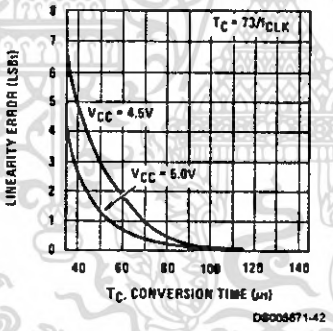
CLK IN Schmitt Trip Levels vs. Supply Voltage



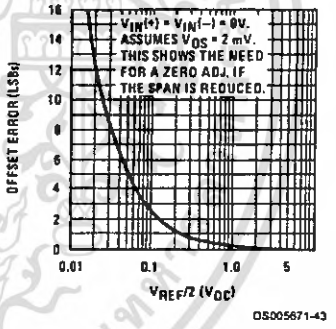
f_{CLK} vs. Clock Capacitor



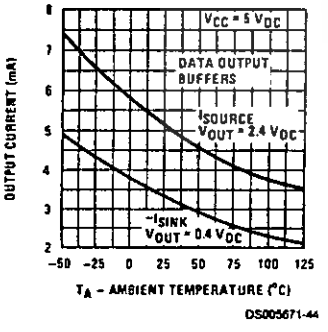
Full-Scale Error vs Conversion Time



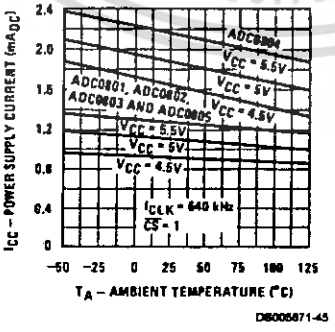
Effect of Unadjusted Offset Error vs. $V_{REF}/2$ Voltage



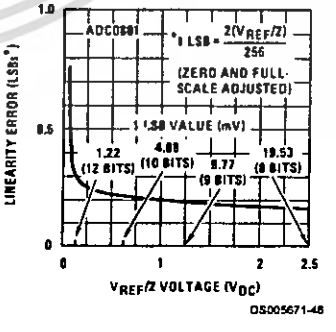
Output Current vs Temperature



Power Supply Current vs Temperature (Note 9)



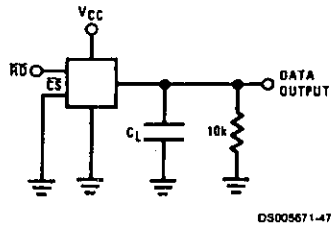
Linearity Error at Low $V_{REF}/2$ Voltages



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

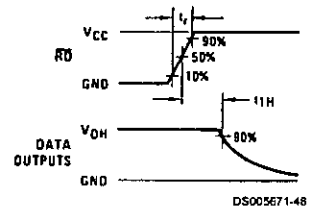
TRI-STATE Test Circuits and Waveforms

t_{1H}



DS005671-47

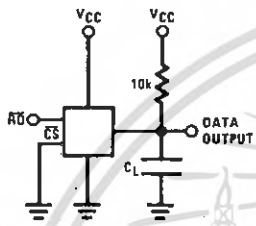
t_{1H}, C_L=10 pF



DS005671-48

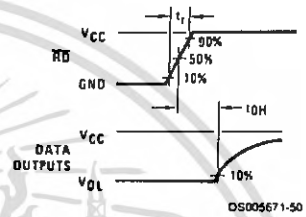
t_r=20 ns

t_{0H}



DS005671-49

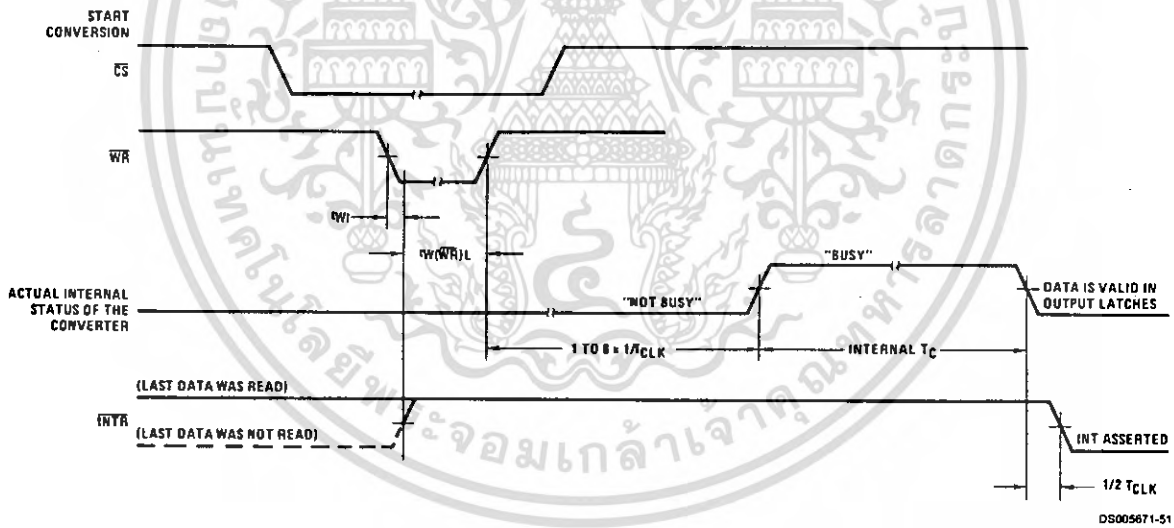
t_{0H}, C_L=10 pF



DS005671-50

t_r=20 ns

Timing Diagrams (All timing is measured from the 50% voltage points)

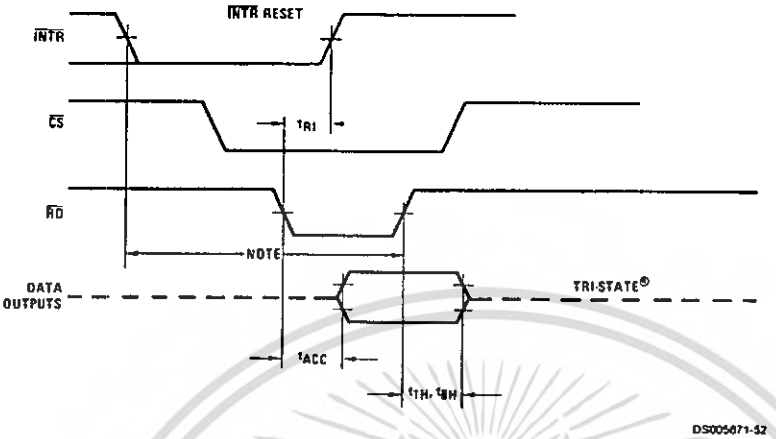


DS005671-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Timing Diagrams (All timing is measured from the 50% voltage points) (Continued)

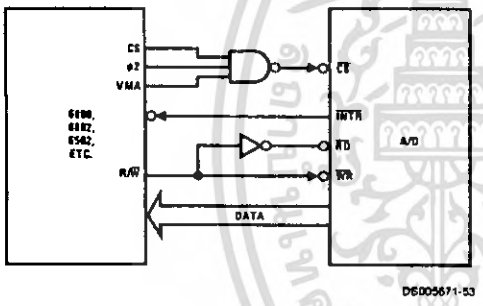
Output Enable and Reset with $\overline{INTR}$



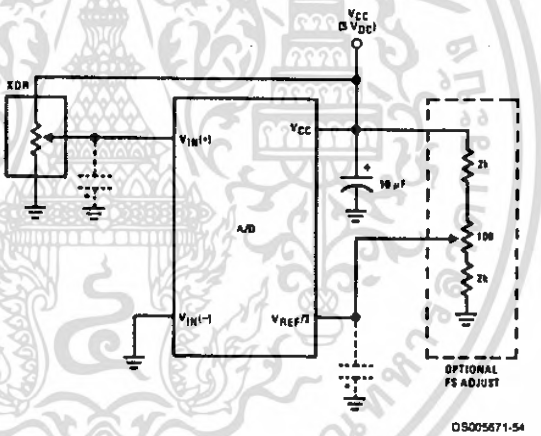
Note: Read strobe must occur 8 clock periods ($8/f_{CLK}$) after assertion of interrupt to guarantee reset of $\overline{INTR}$.

Typical Applications

6800 Interface



Ratiometeric with Full-Scale Adjust



Note: before using caps at V_{IN} or $V_{REF}/2$, see section 2.3.2 Input Bypass Capacitors.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

National Semiconductor

May 1999

DAC0808
8-Bit D/A Converter

General Description

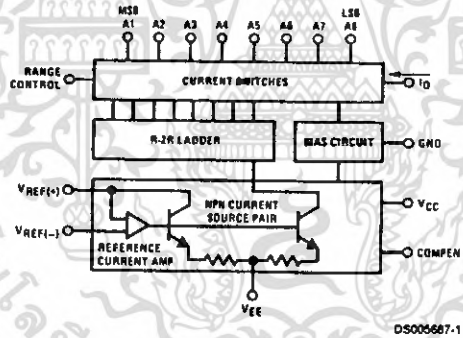
The DAC0808 is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 $I_{REF}/256$. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 µA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 is independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

Features

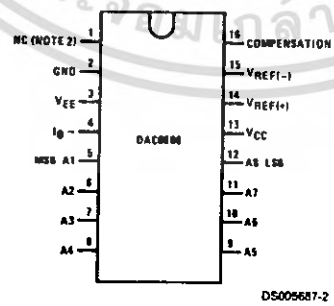
- Relative accuracy: ±0.19% error maximum
- Full scale current match: ±1 LSB typ
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/µs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

Block and Connection Diagrams



DS005687-1

Dual-In-Line Package



DS005687-2

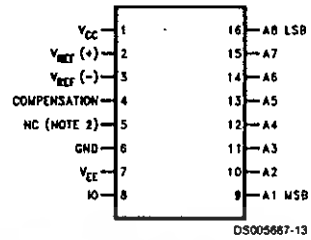
Top View
Order Number DAC0808
See NS Package M16A or N16A

© 2001 National Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block and Connection Diagrams (Continued)

Small-Outline Package



DS005687-13

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	N PACKAGE (N16A) (Note 1)		SO PACKAGE (M16A)
8-bit	0°C≤T_A≤+75°C	DAC0808LCN	MC1408P8	DAC0808LCM

Note 1: Devices may be ordered by using either order number.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (Note 2)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Power Supply Voltage	
V_{CC}	+18 V_{DC}
V_{EE}	−18 V_{DC}
Digital Input Voltage, V5–V12	−10 V_{DC} to +18 V_{DC}
Applied Output Voltage, V_O	−11 V_{DC} to +18 V_{DC}
Reference Current, I_{14}	5 mA
Reference Amplifier Inputs, V14, V15	V_{CC}, V_{EE}
Power Dissipation (Note 4)	1000 mW
ESD Susceptibility (Note 5)	TBD
Storage Temperature Range	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (Plastic)	260°C
Dual-In-Line Package (Ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C

Operating Ratings

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
DAC0808	$0 \le T_A \le +75°C$

Electrical Characteristics

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, V_{REF}/R14 = 2 mA, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
E_r	Relative Accuracy (Error Relative to Full Scale I_O)	(*Figure 4*)				%
	DAC0808LC (LM1408-8)				±0.19	%
	Settling Time to Within ½ LSB (Includes t_{PLH})	T_A=25°C (Note 7), (*Figure 5*)		150		ns
t_{PLH}, t_{PHL}	Propagation Delay Time	T_A = 25°C, (*Figure 5*)		30	100	ns
TCI_O	Output Full Scale Current Drift			±20		ppm/°C
MSB	Digital Input Logic Levels	(*Figure 3*)				
V_{IH}	High Level, Logic "1"		2			V_{DC}
V_{IL}	Low Level, Logic "0"				0.8	V_{DC}
MSB	Digital Input Current	(*Figure 3*)				
	High Level	V_{IH} = 5V		0	0.040	mA
	Low Level	V_{IL} = 0.8V		−0.003	−0.8	mA
I_{15}	Reference Input Bias Current	(*Figure 3*)		−1	−3	μA
	Output Current Range	(*Figure 3*)				
		V_{EE} = −5V	0	2.0	2.1	mA
		V_{EE} = −15V, T_A = 25°C	0	2.0	4.2	mA
I_O	Output Current	V_{REF} = 2.000V, R14 = 1000Ω, (*Figure 3*)	1.9	1.99	2.1	mA
	Output Current, All Bits Low	(*Figure 3*)		0	4	μA
	Output Voltage Compliance (Note 3)	$E_r \le 0.19\%$, T_A = 25°C				
	V_{EE}=−5V, I_{REF}=1 mA				−0.55, +0.4	V_{DC}
	V_{EE} Below −10V				−5.0, +0.4	V_{DC}
SRI_{REF}	Reference Current Slew Rate	(*Figure 6*)	4	8		mA/μs
	Output Current Power Supply Sensitivity	−5V ≤ V_{EE} ≤ −16.5V		0.05	2.7	μA/V
	Power Supply Current (All Bits Low)	(*Figure 3*)				
I_{CC}				2.3	22	mA
I_{EE}				−4.3	−13	mA
	Power Supply Voltage Range	T_A = 25°C, (*Figure 3*)				
V_{CC}			4.5	5.0	5.5	V_{DC}
V_{EE}			−4.5	−15	−16.5	V_{DC}
	Power Dissipation					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics (Continued)

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, $V_{REF}/R14$ = 2 mA, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
	All Bits Low	V_{CC} = 5V, V_{EE} = −5V		33	170	mW
		V_{CC} = 5V, V_{EE} = −15V		106	305	mW
	All Bits High	V_{CC} = 15V, V_{EE} = −5V		90		mW
		V_{CC} = 15V, V_{EE} = −15V		160		mW

Note 2: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 3: Range control is not required.

Note 4: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX}, θ_{JA}, and the ambient temperature, T_A. The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_A)/\theta_{JA}$ or the number given in the Absolute Maixmum Ratings, whichever is lower. For this device, T_{JMAX} = 125°C, and the typical junction-to-ambient thermal resistance of the dual-in-line J package when the board mounted is 100°C/W. For the dual-in-line N package, this number increases to 175°C/W and for the small outline M package this number is 100°C/W.

Note 5: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 6: All current switches are tested to guarantee at least 50% of rated current.

Note 7: All bits switched.

Note 8: Pin-out numbers for the DAL080X represent the dual-in-line package. The small outline package pinout differs from the dual-in-line package.

Typical Application

$$V_O = 10V\left(\frac{A1}{2} + \frac{A2}{4} + \cdots \frac{A8}{256}\right)$$

DS005687-23

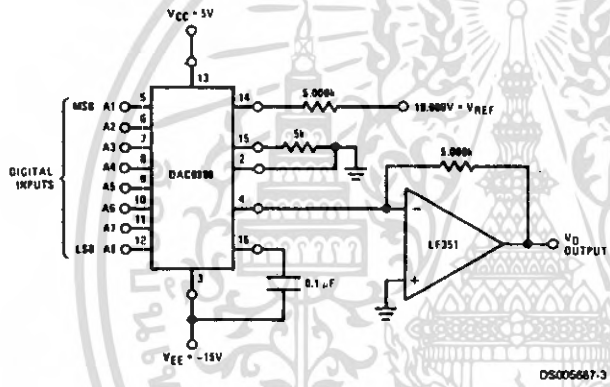
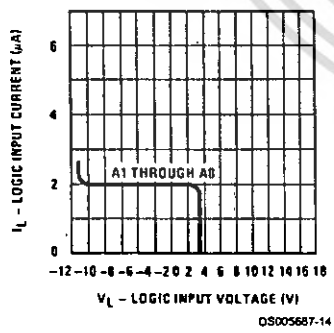


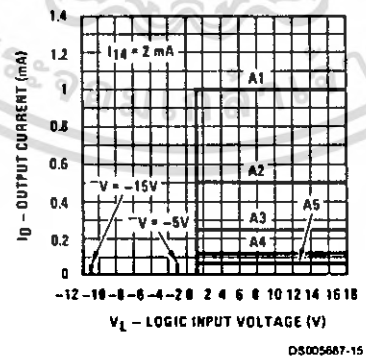
FIGURE 1. +10V Output Digital to Analog Converter (Note 8)

Typical Performance Characteristics V_{CC} = 5V, V_{EE} = −15V, T_A = 25°C, unless otherwise noted

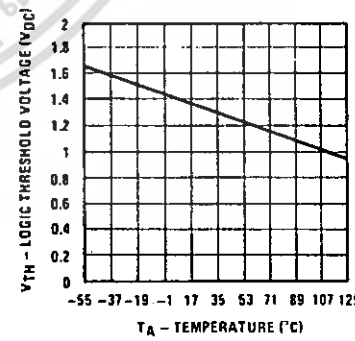
Logic Input Current vs Input Voltage



Bit Transfer Characteristics



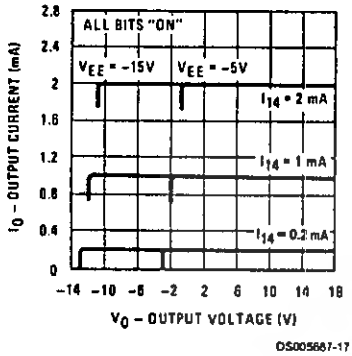
Logic Threshold Voltage vs Temperature



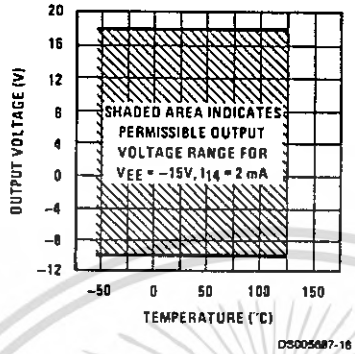
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Performance Characteristics V_{CC} = 5V, V_{EE} = −15V, T_A = 25°C, unless otherwise noted (Continued)

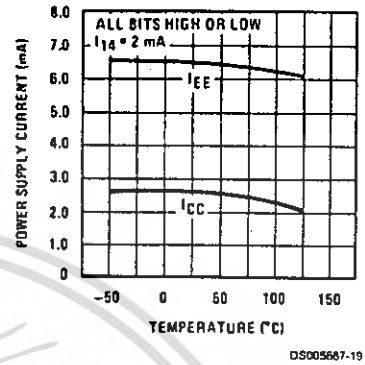
Output Current vs Output Voltage (Output Voltage Compliance)



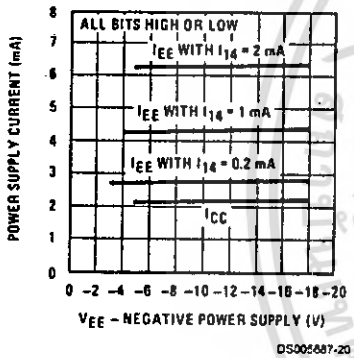
Output Voltage Compliance vs Temperature



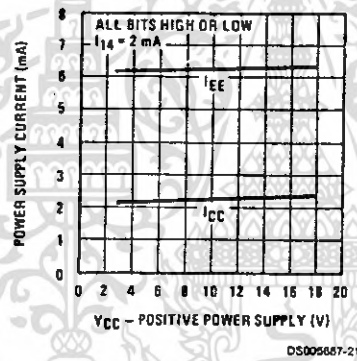
Typical Power Supply Current vs Temperature



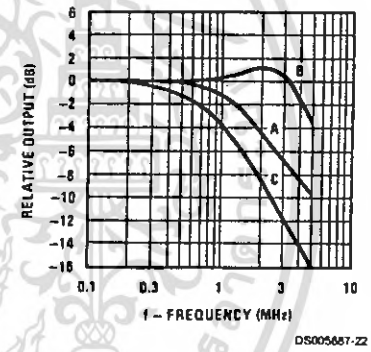
Typical Power Supply Current vs V_{EE}



Typical Power Supply Current vs V_{CC}



Reference Input Frequency Response



Unless otherwise specified: R14 = R15 = 1 kΩ, C = 15 pF, pin 16 to V_{EE}; R_L = 50Ω, pin 4 to ground.

Curve A: Large Signal Bandwidth Method of *Figure 7*, V_{REF} = 2 Vp-p offset 1V above ground.

Curve B: Small Signal Bandwidth Method of *Figure 7*, R_L = 250Ω, V_{REF} = 50 mVp-p offset 200 mV above ground.

Curve C: Large and Small Signal Bandwidth Method of *Figure 9* (no op amp, R_L = 50Ω), R_S = 50Ω, V_{REF} = 2V, V_S = 100 mVp-p centered at 0V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

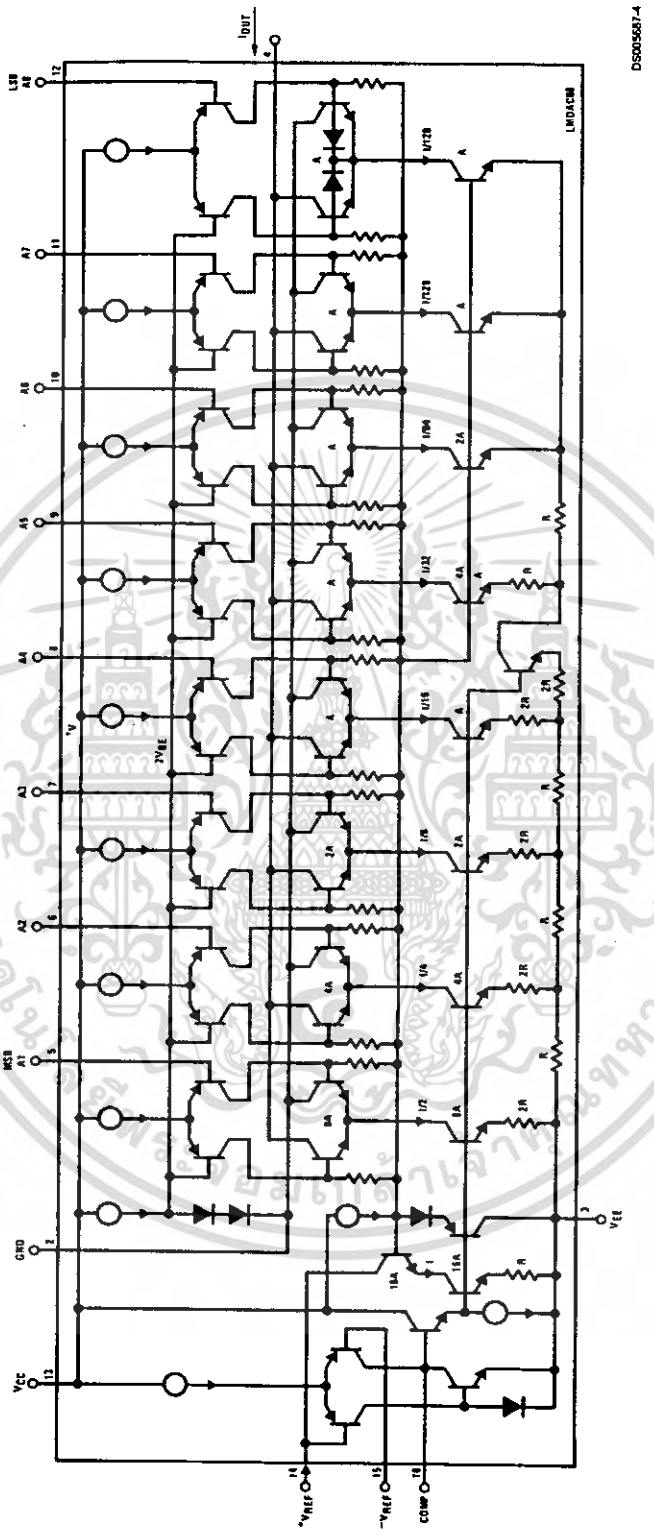
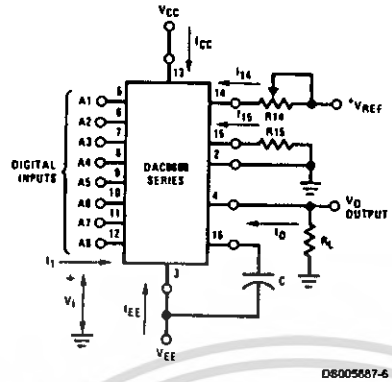


FIGURE 2. Equivalent Circuit of the DAC0808 Series (Note 8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuits



DS005687-6

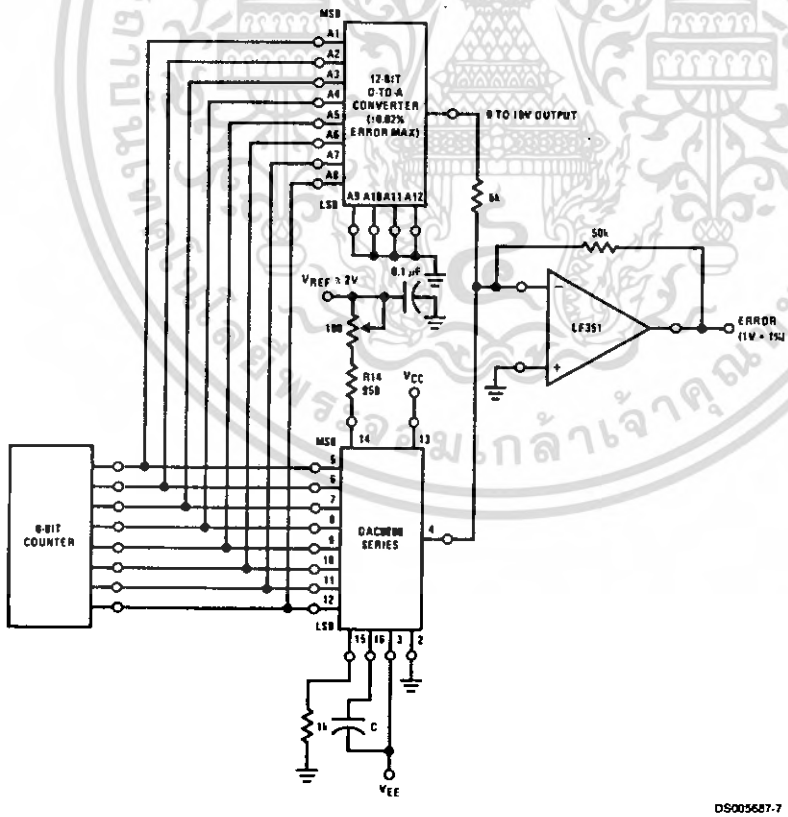
V_I and I_1 apply to inputs A1–A8.
The resistor tied to pin 15 is to temperature compensate the bias current and may not be necessary for all applications.

$$I_O = K\left(\frac{A1}{2} + \frac{A2}{4} + \frac{A3}{8} + \frac{A4}{16} + \frac{A5}{32} + \frac{A6}{64} + \frac{A7}{128} + \frac{A8}{256}\right)$$

$$\text{where } K \approx \frac{V_{REF}}{R14}$$

and A_N = "1" if A_N is at high level
A_N = "0" if A_N is at low level

FIGURE 3. Notation Definitions Test Circuit (Note 8)



DS005687-7

FIGURE 4. Relative Accuracy Test Circuit (Note 8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuits (Continued)

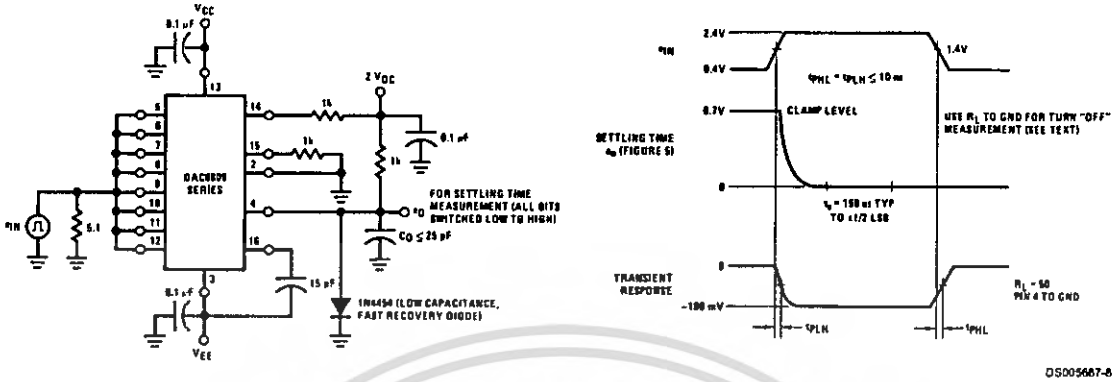


FIGURE 5. Transient Response and Settling Time (Note 8)

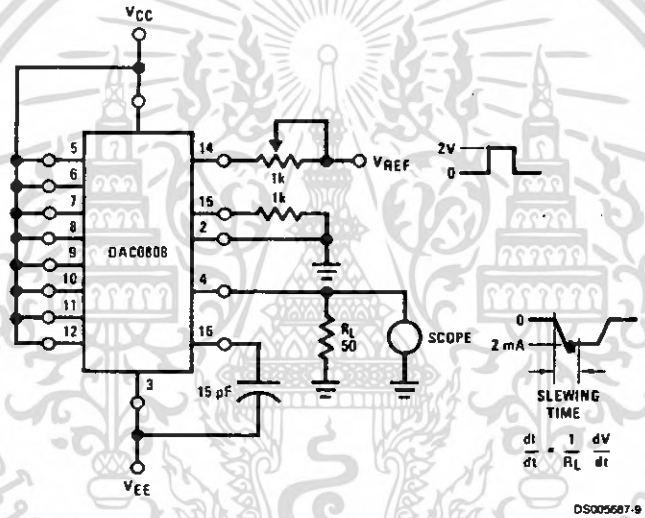


FIGURE 6. Reference Current Slew Rate Measurement (Note 8)

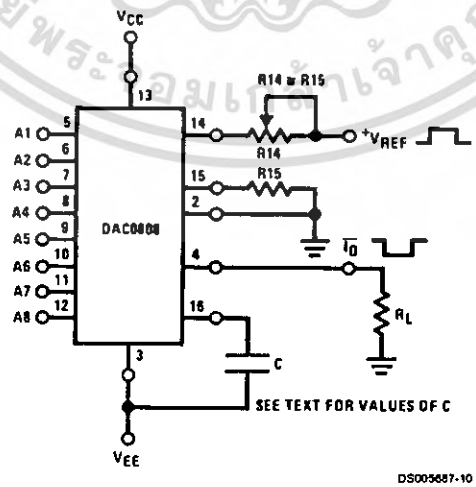


FIGURE 7. Positive V_{REF} (Note 8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Circuits (Continued)

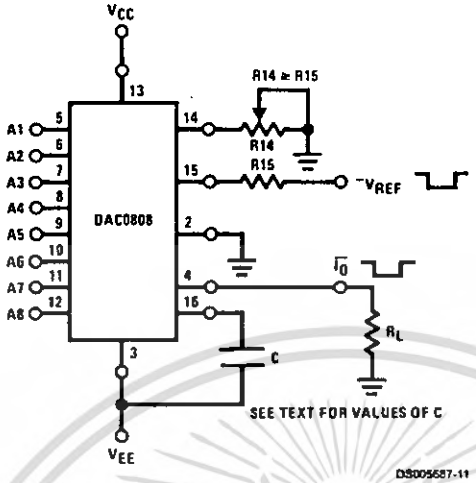


FIGURE 8. Negative V_{REF} (Note 8)

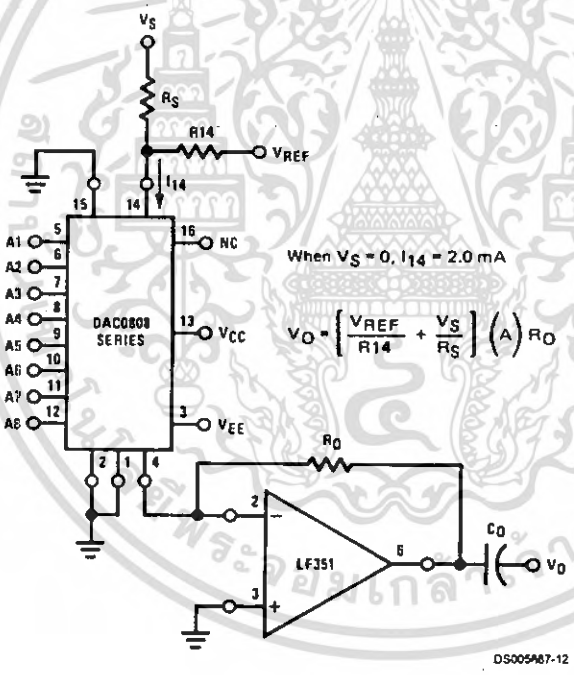


FIGURE 9. Programmable Gain Amplifier or Digital Attenuator Circuit (Note 8)

Application Hints

REFERENCE AMPLIFIER DRIVE AND COMPENSATION

The reference amplifier provides a voltage at pin 14 for converting the reference voltage to a current, and a turn-around circuit or current mirror for feeding the ladder. The reference amplifier input current, I_{14}, must always flow into pin 14, regardless of the set-up method or reference voltage polarity.

Connections for a positive voltage are shown in *Figure 7*. The reference voltage source supplies the full current I_{14}.

For bipolar reference signals, as in the multiplying mode, R15 can be tied to a negative voltage corresponding to the minimum input level. It is possible to eliminate R15 with only a small sacrifice in accuracy and temperature drift.

The compensation capacitor value must be increased with increases in R14 to maintain proper phase margin; for R14 values of 1, 2.5 and 5 kΩ, minimum capacitor values are 15, 37 and 75 pF. The capacitor may be tied to either V_{EE} or ground, but using V_{EE} increases negative supply rejection.

A negative reference voltage may be used if R14 is grounded and the reference voltage is applied to R15 as shown in *Figure 8*. A high input impedance is the main

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Application Hints (Continued)

advantage of this method. Compensation involves a capacitor to V_{EE} on pin 16, using the values of the previous paragraph. The negative reference voltage must be at least 4V above the V_{EE} supply. Bipolar input signals may be handled by connecting R14 to a positive reference voltage equal to the peak positive input level at pin 15.

When a DC reference voltage is used, capacitive bypass to ground is recommended. The 5V logic supply is not recommended as a reference voltage. If a well regulated 5V supply which drives logic is to be used as the reference, R14 should be decoupled by connecting it to 5V through another resistor and bypassing the junction of the 2 resistors with 0.1 µF to ground. For reference voltages greater than 5V, a clamp diode is recommended between pin 14 and ground.

If pin 14 is driven by a high impedance such as a transistor current source, none of the above compensation methods apply and the amplifier must be heavily compensated, decreasing the overall bandwidth.

OUTPUT VOLTAGE RANGE

The voltage on pin 4 is restricted to a range of −0.55 to 0.4V when V_{EE} = −5V due to the current switching methods employed in the DAC0808.

The negative output voltage compliance of the DAC0808 is extended to −5V where the negative supply voltage is more negative than −10V. Using a full-scale current of 1.992 mA and load resistor of 2.5 kΩ between pin 4 and ground will yield a voltage output of 256 levels between 0 and −4.980V. Floating pin 1 does not affect the converter speed or power dissipation. However, the value of the load resistor determines the switching time due to increased voltage swing. Values of R_L up to 500Ω do not significantly affect performance, but a 2.5 kΩ load increases worst-case settling time to 1.2 µs (when all bits are switched ON). Refer to the subsequent text section on Settling Time for more details on output loading.

OUTPUT CURRENT RANGE

The output current maximum rating of 4.2 mA may be used only for negative supply voltages more negative than −8V, due to the increased voltage drop across the resistors in the reference current amplifier.

ACCURACY

Absolute accuracy is the measure of each output current level with respect to its intended value, and is dependent upon relative accuracy and full-scale current drift. Relative accuracy is the measure of each output current level as a fraction of the full-scale current. The relative accuracy of the DAC0808 is essentially constant with temperature due to the excellent temperature tracking of the monolithic resistor ladder. The reference current may drift with temperature, causing a change in the absolute accuracy of output current. However, the DAC0808 has a very low full-scale current drift with temperature.

The DAC0808 series is guaranteed accurate to within ±½ LSB at a full-scale output current of 1.992 mA. This corresponds to a reference amplifier output current drive to the ladder network of 2 mA, with the loss of 1 LSB (8 µA) which is the ladder remainder shunted to ground. The input current to pin 14 has a guaranteed value of between 1.9 and 2.1 mA, allowing some mismatch in the NPN current source pair. The accuracy test circuit is shown in *Figure 4*. The 12-bit converter is calibrated for a full-scale output current of 1.992 mA. This is an optional step since the DAC0808 accuracy is essentially the same between 1.5 and 2.5 mA. Then the DAC0808 circuits' full-scale current is trimmed to the same value with R14 so that a zero value appears at the error amplifier output. The counter is activated and the error band may be displayed on an oscilloscope, detected by comparators, or stored in a peak detector.

Two 8-bit D-to-A converters may not be used to construct a 16-bit accuracy D-to-A converter. 16-bit accuracy implies a total error of ±½ of one part in 65,536 or ±0.00076%, which is much more accurate than the ±0.019% specification provided by the DAC0808.

MULTIPLYING ACCURACY

The DAC0808 may be used in the multiplying mode with 8-bit accuracy when the reference current is varied over a range of 256:1. If the reference current in the multiplying mode ranges from 16 µA to 4 mA, the additional error contributions are less than 1.6 µA. This is well within 8-bit accuracy when referred to full-scale.

A monotonic converter is one which supplies an increase in current for each increment in the binary word. Typically, the DAC0808 is monotonic for all values of reference current above 0.5 mA. The recommended range for operation with a DC reference current is 0.5 to 4 mA.

SETTLING TIME

The worst-case switching condition occurs when all bits are switched ON, which corresponds to a low-to-high transition for all bits. This time is typically 150 ns for settling to within ±½ LSB, for 8-bit accuracy, and 100 ns to ½ LSB for 7 and 6-bit accuracy. The turn OFF is typically under 100 ns. These times apply when $R_L \leq 500\Omega$ and $C_O \leq 25$ pF.

Extra care must be taken in board layout since this is usually the dominant factor in satisfactory test results when measuring settling time. Short leads, 100 µF supply bypassing for low frequencies, and minimum scope lead length are all mandatory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้