การพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VoIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน (ปีที่ 2)

Development of IP (Internet Protocol) Telephone System (VoIP) based on SIP (Session Initiation Protocol) (Year II)

โดย

วาทิต เบญจพลกุล

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ทุนวิจัยร่วมภาครัฐกับภาคเอกชน ปี 2552

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

กรุงเทพฯ

กุมภาพันธ์ 2553

การพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VoIP) บนพื้นฐาน ของโพรโทคอลเริ่มต้นเซสชัน (ปีที่ 2)

Development of IP (Internet Protocol) Telephone System (VoIP) based on SIP (Session Initiation Protocol) (Year II)

โดย

วาทิต เบญจพลกุล D. Eng. (University of Tokyo)

ทุนวิจัยร่วมภาครัฐกับภาคเอกชน ปี 2552

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

คณะวิศวกรรมศาสตร์

จุฬาลงกรณ์มหาวิทยาลัย

กรุงเทพฯ

กุมภาพันธ์ 2553

สารบัญ

หน้า

กิตติกรรมประกาศ **ก**

บทคัดย่อ **ข**

Abstract **ค**

บทที่ 1 บทนำ **1**

ความสำคัญและที่มาของปัญหาที่ทำการวิจัย 1

วัตถุประสงค์ของโครงการ 2

เป้าหมายของโครงการ 3

ประโยชน์ที่คาดว่าจะได้รับ 3

ผลกระทบต่อสิ่งแวดล้อม 3

หน่วยงานที่จะนำผลวิจัยไปใช้ประโยชน์ 3

ผลงานวิจัยที่เกี่ยวข้อง 4

ระเบียบวิธีวิจัย 4

ขอบเขตของการวิจัย 5

ระยะเวลาการวิจัย 6

แผนการดำเนินงานตลอดโครงการ 6

สถานที่ทำการวิจัย 9

อุปกรณ์การวิจัย 9

บทที่ 2 การศึกษาระบบไมโครคอนโทรลเลอร์และการเลือกอุปกรณ์สำหรับโทรศัพท์ผ่านอินเทอร์เน็ต (VOIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน **11**

ไมโครคอนโทรลเลอร์ MCS-51 11

การเลือกอุปกรณ์สำหรับการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตต้นแบบ 25

บทที่ 3 การออกแบบส่วนประมวลผลสำหรับระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VOIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน **27**

โครงสร้างเครื่องโทรศัพท์อินเทอร์เน็ตต้นแบบ 27

ส่วนติดต่อรับส่งข้อมูลผ่านชิพอีเทอร์เน็ต RTL8019as 28

ส่วนรองรับโพรโทคอล 37

ส่วนควบคุมข้อมูลเสียงและติดต่อกับชิพเข้ารหัสเสียง PCM 48

การควบคุมการทำงานโดยรวม 57

บทที่ 4 การออกแบบส่วนติดต่อกับผู้ใช้งานสำหรับระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VOIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน 61

ส่วนแสดงผลผ่าน LCD 61

ส่วนรับข้อมูลจากผู้ใช้งานผ่านคีย์แพด 65

บทที่ 5 การทดสอบ 70

ข้อกำหนดในการทดสอบ 70

การทดสอบการทำงานในโมดติดต่อโดยตรง 72

การทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย 74

การทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง 76

บทที่ 6 ผลการทดสอบและการวิเคราะห์ผลการทดสอบ 77

ผลการทดสอบการทำงานในโมดติดต่อโดยตรง 80

ผลการทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย 91

ผลการทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง 108

อภิปรายผลการทดสอบการทำงานในโมดติดต่อโดยตรง 108

อภิปรายผลการทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย 109

อภิปรายผลการทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง 111

บทที่ 7 ต้นทุนการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตต้นแบบและแนวทางในการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตในอนาคต 112

ต้นทุนการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตต้นแบบ 112

แนวทางในการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตในอนาคต 113

บทที่ 8 สรุปและข้อเสนอแนะของการวิจัย 114

สรุปผลการพัฒนาและทดสอบ 114

ปัญหาในการพัฒนาและการแก้ไข 114

ข้อเสนอแนะ 116

ภาคผนวก โทรศัพท์ผ่านอินเทอร์เน็ต (IP Phone) ที่พัฒนาและออกแบบโดยคนไทยที่สามารถใช้งานได้จริงและนำไปใช้งานในเชิงพาณิชย์ 118

รายการอ้างอิง 121

กิตติกรรมประกาศ

โครงการวิจัยการพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VoIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน (ปีที่ 2) นี้ สำเร็จลุล่วงไปได้ด้วยดีก็ด้วยความสนับสนุนจากทุนวิจัยร่วมภาครัฐกับภาคเอกชน ปี 2552 ผู้วิจัยจึงขอแสดงความขอบคุณเป็นอย่างสูงมา ณ ที่นี้ นอกจากนี้ ผู้วิจัยขอขอบคุณ คุณณัฐวร ปานจินดา และคุณเกริกศักดิ์ จันทร์ด้ง ผู้ช่วยวิจัย ซึ่งได้มีส่วนช่วยเป็นอย่างมากในการดำเนินการสร้างและการทดสอบระบบ จนโครงการวิจัยนี้เสร็จสิ้นสมบูรณ์ สุดท้าย ผู้วิจัยขอขอบคุณภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ผู้วิจัยได้ใช้เป็นสถานที่ดำเนินการวิจัย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อ

ปัจจุบันระบบโทรศัพท์ผ่านอินเทอร์เน็ตมีแนวโน้มเข้ามาแทนที่ระบบโทรศัพท์แบบ PSTN เพิ่มมากขึ้น ทั้งนี้เนื่องจากข้อได้เปรียบทางด้านค่าใช้จ่าย ประสิทธิภาพในการใช้ช่องสัญญาณ ความสะดวกในการใช้งาน และการบำรุงรักษา อย่างไรก็ตาม เครื่องโทรศัพท์ผ่านอินเทอร์เน็ตดังกล่าวต้องนำเข้าจากต่างประเทศ โครงการวิจัยนี้จึงได้พัฒนาเครื่องโทรศัพท์ผ่านอินเทอร์เน็ตต้นแบบ เพื่อเป็นแนวทางในการผลิตเครื่องโทรศัพท์ผ่านอินเทอร์เน็ตต้นทุนต่ำขึ้นใช้เองภายในประเทศ โดยในการพัฒนาส่วนของฮาร์ดแวร์ได้ใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 รุ่น AT89C51ED2 ชิพควบคุมอีเทอร์เน็ต RTL8019as และชิพเข้ารหัสเสียง MC145480 เป็นส่วนประกอบหลัก ในส่วนของซอฟต์แวร์ได้พัฒนาส่วนติดต่อกับผู้ใช้ ส่วนติดต่อกับชิพอีเทอร์เน็ต ส่วนควบคุมข้อมูลเสียง และส่วนรองรับโพรโทคอล ICMP, ARP, IP, UDP, RTP, และ SIP ซึ่งเป็นโพรโทคอลหลักสำหรับเครื่องโทรศัพท์ผ่านอินเทอร์เน็ตพื้นฐาน

เครื่องโทรศัพท์ผ่านอินเทอร์เน็ตต้นแบบที่ได้พัฒนาขึ้นได้รับการตรวจสอบความถูกต้องของการทำงานโดยการทดสอบความถูกต้องของสัญญาณ Signaling ต่างๆ ที่เกิดขึ้นระหว่างการทำงานของเครื่องโทรศัพท์โดยอาศัยโพรโทคอลเริ่มต้นเซสชัน และจากผลการทดสอบพบว่าเครื่องโทรศัพท์ที่ได้รับการพัฒนาขึ้นนี้สามารถทำงานฟังก์ชันพื้นฐานได้แก่ การสร้างเซสชัน การยกเลิกเซสชัน การปฏิเสธเซสชันและการสิ้นสุดเซสชันได้อย่างถูกต้องตามมาตรฐานของโพรโทคอลเริ่มต้นเซสชัน ทั้งการต่อถึงกันโดยตรงและการต่อใช้งานผ่านเครื่องแม่ข่าย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

Abstract

At present, there is a strong trend that IP (Internet Protocol) telephone system will more and more replace PSTN (Public Switching Telephone Network) telephone system because of the former system's cost advantage, channel utilization efficiency, convenience to use and convenience in maintenance when compared with the latter system's. However, Thailand still has to import IP telephones from foreign countries. As a matter of fact, this research project aimed at developing an IP telephone prototype to find a model in the production of low-cost IP telephones for domestic use. In the development of IP telephone hardware, 8051-family microcontroller (model AT89C51ED2), Ethernet controller (model RTL8019as), and PCM Codec (model MC145480) were used as the main components. In addition, in the development of IP telephone software, user interface module, Ethernet interface module, voice data control module, and ICMP, ARP, IP, UDP, RTP, and SIP protocols (which are main protocols for basic IP telephone) supporting module were developed.

The developed IP telephone prototype was tested for its accuracy in operation by checking the accuracy of various SIP signaling, functioning during IP telephone operation. The test results reveal that the developed IP telephone prototype can perform basic functions of SIP protocol, such as session establishment, session cancellation, session rejection, and session termination, correctly according to the standard SIP protocol, in both direct connection and connection via a server.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1
บทนำ

1.1. ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

เนื่องจากระบบโทรศัพท์ PSTN (Public Switching Telephone Network) ที่ใช้กันแต่เดิมมีอัตราค่าบริการโทรศัพท์ทางไกลทั้งโทรศัพท์ทางไกลภายในประเทศและโทรศัพท์ทางไกลระหว่างประเทศสูงมาก จึงได้มีการพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ต (IP Telephone System หรือ VoIP (Voice over IP)) ขึ้นมาโดยระบบโทรศัพท์ผ่านอินเทอร์เน็ตนี้เป็นการส่งข้อมูลเสียงโดยเข้ารหัสเป็นสัญญาณดิจิทัลผ่านโครงข่ายอินเทอร์เน็ตและมีการแบ่งข้อมูลเสียงออกเป็นแพ็กเกตย่อย ๆ และส่งออกไป โดยใช้อินเทอร์เน็ตโพรโทคอล (Internet Protocol) ซึ่งทำให้ค่าบริการโทรศัพท์ทางไกลในลักษณะนี้เท่ากับค่าบริการใช้อินเทอร์เน็ตของผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider หรือ ISP) ซึ่งมีราคาถูกลงเรื่อยๆ จากผลของการแข่งขันกันในการให้บริการกับผู้ใช้บริการ อย่างไรก็ตามระบบโทรศัพท์ผ่านอินเทอร์เน็ตดั้งเดิม มีปัญหาเกี่ยวกับคุณภาพของการบริการ (Quality of Service หรือ QoS) ด้านการประวิงเวลา (Time Delay) ซึ่งเกิดขึ้นเนื่องจาก

1. อัตราข้อมูลที่ผู้ใช้เข้าถึงโครงข่ายอินเทอร์เน็ตไม่สูงพอที่จะทำให้การรับส่งข้อมูลระหว่างผู้ใช้กับโครงข่ายอินเทอร์เน็ตเป็นไปได้ทันตามปริมาณข้อมูลที่รับส่งกันอยู่ในขณะนั้น โดยเฉพาะอย่างยิ่งผู้ใช้ที่อยู่ตามบ้านซึ่งส่วนใหญ่ยังคงรับส่งข้อมูลได้ด้วยความเร็วไม่เกินประมาณ 56 kbps อย่างไรก็ตาม ในอนาคตอันใกล้นี้ปัญหานี้น่าจะได้รับการแก้ไขให้ดีขึ้นโดยเทคโนโลยี xDSL (x Digital Subscriber Line) ที่จะทำให้ผู้ใช้สามารถรับส่งข้อมูลด้วยความเร็วสูงขึ้นได้

2. หลักการทำงานของโครงข่ายอินเทอร์เน็ตเป็นการแบ่งปันทรัพยากรที่มีอยู่ให้ผู้ใช้บริการทุกคนในโครงข่ายขณะนั้นได้ใช้ร่วมกัน ดังนั้นในขณะที่มีผู้ใช้จำนวนมากเข้ามาใช้โครงข่ายอินเทอร์เน็ตพร้อมๆ กัน ก็จะทำให้ผู้ใช้แต่ละคนต้องรอคอยจังหวะในการรับส่งข้อมูลที่ยาวนานขึ้น อย่างไรก็ตามผู้ให้บริการอินเทอร์เน็ตก็มีการขยายทรัพยากรของโครงข่ายอินเทอร์เน็ตเพื่อรองรับกับจำนวนผู้ใช้ที่มีจำนวนมากขึ้นเรื่อย ๆ อยู่ตลอดเวลา ดังนั้นปัญหานี้ก็น่าที่จะได้รับการแก้ไขให้ดีขึ้นตามการขยายทรัพยากรของโครงข่ายอินเทอร์เน็ตของผู้ให้บริการอินเทอร์เน็ตเช่นกัน

3. ในตอนเริ่มต้นการต่อการสนทนาทางระบบโทรศัพท์ผ่านอินเทอร์เน็ตดั้งเดิมนั้นมีการใช้โพรโทคอล H.323 ซึ่งซับซ้อนและมีปริมาณการส่งการสัญญาณ (Signalling) ค่อนข้างมาก ทำให้ระบบโทรศัพท์ผ่านอินเทอร์เน็ตดั้งเดิมต้องเสียเวลาในขั้นตอนนี้ค่อนข้างมากทำให้การประวิงเวลาในระบบสูงขึ้นด้วย อย่างไรก็ตามในระยะหลังได้มีผู้เสนอโพรโทคอลเริ่มต้นเซสชัน (Session Initiation Protocol

หรือ SIP) ซึ่งมีความยุ่งยากและความซับซ้อนน้อยกว่าโพรโทคอล H.323 มาก ซึ่งก็น่าที่จะสามารถแก้ไขปัญหาของการประวิงเวลาในระบบโทรศัพท์ผ่านอินเทอร์เน็ตดั้งเดิมได้

เหตุผลดังกล่าวข้างต้นทำให้ผู้ใช้มีความต้องการใช้งานระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่มีคุณภาพของการบริการที่ตนพอใจมากขึ้นเรื่อยๆ แนวโน้มเช่นนี้เป็นแนวโน้มที่เหมือนกันในประเทศต่างๆ ทั่วโลกและเป็นสาเหตุที่ทำให้ผู้ผลิตอุปกรณ์การสื่อสารข้อมูลหันมาให้ความสนใจในการวิจัยและพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่มีคุณภาพของการบริการที่ผู้ใช้บริการพอใจมากขึ้นเรื่อย ๆ

ประเทศไทยซึ่งเป็นประเทศผู้ใช้เทคโนโลยีอินเทอร์เน็ตประเทศหนึ่งก็มีแนวโน้มในการใช้ระบบโทรศัพท์ผ่านอินเทอร์เน็ตมากขึ้นเรื่อยๆ เช่นกัน แต่เนื่องจากประเทศไทยเป็นผู้นำเข้าเทคโนโลยีอินเทอร์เน็ตเกือบ 100 % ดังนั้นผลกระทบต่อการสูญเสียเงินตราต่างประเทศจึงคาดได้ว่าน่าจะมีแนวโน้มสูงมากขึ้นอย่างต่อเนื่องเช่นกัน

รัฐบาลไทยได้เล็งเห็นความสำคัญของเทคโนโลยีอินเทอร์เน็ตในแง่ที่เป็นเครื่องมือที่จะช่วยให้ประเทศไทยมีขีดความสามารถด้านธุรกิจในตลาดโลกที่สูงขึ้น และในขณะเดียวกันก็ได้เล็งเห็นความสำคัญของการสร้างและพัฒนาบุคลากรด้านเทคโนโลยีสารสนเทศทั้งในภาครัฐและภาคเอกชนควบคู่ไปด้วย

จากเหตุผลและความเป็นมาข้างต้น คณะผู้วิจัยชุดนี้จึงเสนอทำโครงการวิจัยร่วมภาครัฐและเอกชนในหัวข้อ “การพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน” เพื่อเป็นการสร้างรากฐานการวิจัยและพัฒนาเทคโนโลยีด้านนี้และเป็นการสนองตอบต่อนโยบายของรัฐบาลด้านการเพิ่มศักยภาพในการแข่งขันของประเทศ การเสริมสร้างการพัฒนาที่ยั่งยืนของประเทศ และการสร้างและพัฒนาบุคลากรด้านเทคโนโลยีสารสนเทศทั้งในภาครัฐบาลและภาคเอกชน รวมทั้งสอดคล้องกับนโยบายที่ 1 ของกรอบนโยบายเทคโนโลยีสารสนเทศ ระยะ พ.ศ. 2544-2553 ของประเทศไทยอีกด้วย [1]

1.2 วัตถุประสงค์ของโครงการ

1) เพื่อวิจัย พัฒนา ออกแบบและสร้างอุปกรณ์ต้นแบบระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชันที่สามารถติดต่อระหว่างกันได้ที่มีราคาถูกกว่าระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่นำเข้ามาจากต่างประเทศ โดยคำนึงถึงต้นทุนในการผลิตและคุณลักษณะของเครื่องต้นแบบเพื่อนำผลงานวิจัยไปใช้ในเชิงพาณิชย์และสามารถพัฒนาหน้าที่การทำงานใหม่ ๆ ที่จะเกิดขึ้นต่อไปในอนาคตได้

2) เพื่อสร้างองค์ความรู้ด้านระบบโทรศัพท์ผ่านอินเทอร์เน็ตให้กับภาครัฐบาลและภาคเอกชน

3) เพื่อเป็นการสร้างความแข็งแกร่งให้กับรากฐานการวิจัยและพัฒนาเทคโนโลยีระบบโทรศัพท์ผ่านอินเทอร์เน็ตในภาครัฐบาลและภาคเอกชน

4) เพื่อสนองตอบต่อนโยบายของรัฐบาลด้านการเพิ่มศักยภาพในการแข่งขันของประเทศและการเสริมสร้างการพัฒนาที่ยั่งยืนของประเทศ

5) เพื่อสร้างและพัฒนาบุคลากรด้านเทคโนโลยีสารสนเทศทั้งในภาครัฐบาลและภาคเอกชน

1.3 เป้าหมายของโครงการ

เครื่องต้นแบบระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชันที่สามารถติดต่อระหว่างกันได้ที่มีราคาถูกกว่าและมีประสิทธิภาพการทำงานที่ดีกว่าระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่ใช้โพรโทคอลแบบเดิมคือ H.323 ซึ่งซับซ้อนกว่า ที่นำเข้ามาจากต่างประเทศ และสามารถพัฒนาหน้าที่การทำงานใหม่ๆ ที่จะเกิดขึ้นต่อไปในอนาคตได้ด้วยตนเอง เช่น ผู้พัฒนาซอฟต์แวร์ (Software Developer) สามารถเพิ่มเติมความสามารถในการประยุกต์ใช้งานอื่นๆ เช่น ไปรษณีย์อิเล็กทรอนิกส์ (E-mail) Web Browser ฯลฯ เข้าไปในเครื่องโทรศัพท์ชนิดนี้ได้ ผลลัพธ์ที่ได้คือองค์ความรู้ที่สามารถนำไปจดสิทธิบัตรได้

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1) หากสามารถพัฒนาเครื่องต้นแบบจนนำไปสู่การผลิตในเชิงพาณิชย์ นอกจากจะสามารถประหยัดเงินตราต่างประเทศในการลดการนำเข้าแล้ว ยังสามารถผลิตเพื่อเป็นการส่งออกสินค้าไปต่างประเทศได้อีกด้วย

2) ทำให้เกิดความร่วมมือด้านการวิจัยและพัฒนาระหว่างสถาบันการศึกษาภาครัฐบาลกับภาคเอกชน

3) บุคลากรในสถาบันการศึกษาภาครัฐบาลมีโอกาสเพิ่มพูนประสบการณ์ในการทำงานจริงกับภาคเอกชน

4) เกิดการจ้างงานบัณฑิตในรูปแบบผู้ช่วยวิจัย

15. ผลกระทบต่อสิ่งแวดล้อม

ไม่มี

1.6 หน่วยงานที่จะนำผลวิจัยไปใช้ประโยชน์

หน่วยงานที่สามารถนำผลวิจัยไปใช้ประโยชน์ได้มีทั้งหน่วยงานภาครัฐและภาคเอกชนที่มีความต้องการใช้งานระบบโทรศัพท์ผ่านอินเทอร์เน็ต และผู้ใช้ทั่วไปที่อยู่ตามบ้านหรือตามสำนักงานทั้งขนาดเล็กและขนาดใหญ่

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.7 ผลงานวิจัยที่เกี่ยวข้อง

จากที่ได้กล่าวมาข้างต้น ผู้ผลิตอุปกรณ์การสื่อสารข้อมูลจำนวนมากได้หันมาสนใจการวิจัยและพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตด้วยแรงผลักดันด้านความต้องการของผู้ใช้ในตลาดซึ่งมีแนวโน้มสูงขึ้นอย่างต่อเนื่อง (ดูเอกสารอ้างอิง [2-9]) ระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่ใช้โพรโทคอลแบบเดิมคือ H.323 ซึ่งซับซ้อนกว่า ที่ต้องนำเข้ามาจากต่างประเทศนั้นน่าจะมีสิทธิบัตรคุ้มครองอยู่โดยปกติทั่วไปอยู่แล้ว หากผู้ใดทำการลอกเลียนแบบจากอุปกรณ์นั้นๆ โดยตรงก็จะมีความผิดฐานละเมิดสิทธิบัตรดังกล่าว อย่างไรก็ตามโครงการวิจัยร่วมนี้ไม่ได้ทำการลอกเลียนแบบ แต่เป็นการออกแบบวงจรและเขียนซอฟต์แวร์ด้วยตนเองโดยให้วงจรที่ออกแบบเองและซอฟต์แวร์ที่เขียนขึ้นเองนี้สามารถทำหน้าที่ได้ตามความต้องการระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่ตั้งไว้ ซึ่งวงจรที่ได้ออกแบบเองและซอฟต์แวร์ที่เขียนขึ้นเองนั้นมีทางเป็นไปได้มากมาย จึงมีความเป็นไปได้ที่น้อยมากที่จะไปเหมือนกับของระบบโทรศัพท์ผ่านอินเทอร์เน็ตซึ่งต้องนำเข้ามาจากต่างประเทศ นอกจากนี้โครงการวิจัยนี้ต้องการพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VoIP) บนพื้นฐานของโพรโทคอลชนิดใหม่คือโพรโทคอลเริ่มต้นเซสชัน ดังนั้น การออกแบบและสร้างอุปกรณ์ต้นแบบระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่สามารถติดต่อระหว่างกันได้ในโครงการวิจัยร่วมนี้จึงเป็นแนวทางการแก้ปัญหาที่ถูกต้องและเหมาะสม

1.8 ระเบียบวิธีวิจัย

1. ศึกษาระบบโทรศัพท์ผ่านอินเทอร์เน็ต โดยศึกษาจากเอกสารอ้างอิงต่างๆ ที่เกี่ยวข้อง
2. ศึกษามาตรฐานโพรโทคอล H.323 โดยศึกษาจากเอกสารอ้างอิงต่างๆ ที่เกี่ยวข้อง
3. ศึกษามาตรฐานโพรโทคอล SIP โดยศึกษาจากเอกสารอ้างอิงต่างๆ ที่เกี่ยวข้อง
4. ศึกษาการประยุกต์ใช้โปรแกรมภาษาต่างๆ กับการพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตโดยใช้ซอฟต์แวร์ โดยศึกษาจากเอกสารอ้างอิงต่างๆ ที่เกี่ยวข้อง
5. พัฒนาโปรแกรมในส่วนสร้างและแยก SIP Message ตาม RFC 2543 โดยบริษัทที่ร่วมโครงการจะทำหน้าที่ชี้แนะแนวทางและช่วยเหลือข้อมูลด้านเทคนิค
6. เขียน Flow Chart กำหนดขั้นตอนการทำงานของโปรแกรมแต่ละส่วน
7. พัฒนาโปรแกรมตามขั้นตอนการทำงานของโปรแกรมที่ได้กำหนดไว้
8. ทดสอบการทำงานของโปรแกรมและแก้ไขข้อผิดพลาด
9. จัดแยกโปรแกรมเป็นมอดูล (Library Module) เพื่อให้ง่ายต่อการพัฒนาเป็นเครื่องต้นแบบที่เป็นฮาร์ดแวร์ต่อไป
10. ศึกษาวิธีการสื่อสารระหว่างเครื่องโทรศัพท์ผ่านอินเทอร์เน็ต โดยในขั้นตอนนี้ผู้วิจัยจะศึกษาข้อมูลเพิ่มเติมภายใต้คำแนะนำของหัวหน้าโครงการ ส่วนทางบริษัทที่ร่วมโครงการจะคอยแนะนำและสาธิตอุปกรณ์ต่างๆ ที่เกี่ยวข้อง ศึกษาวิธีการทางฮาร์ดแวร์ในการต่อการ

สื่อสารระหว่างเครื่องโทรศัพท์ผ่านอินเทอร์เน็ต ศึกษาการเขียนโปรแกรมของไมโครโปรเซสเซอร์ (Digital Signal Processor) เพื่อเรียนรู้วิธีการต่อถึงกัน ศึกษาวิธีการทางฮาร์ดแวร์และซอฟต์แวร์ในการต่อการสื่อสารระหว่างกัน

11. ออกแบบและสร้างเครื่องต้นแบบเครื่องโทรศัพท์ผ่านอินเทอร์เน็ตที่สามารถติดต่อระหว่างกัน โดยทางหัวหน้าโครงการและบริษัทที่ร่วมโครงการจะร่วมกันดูแลและช่วยเหลือในการแก้ไขปัญหาต่างๆ ที่เกิดขึ้น ในขั้นตอนนี้บริษัทที่ร่วมโครงการจะได้ให้ความสะดวกในเรื่องสถานที่ปฏิบัติงานของโครงการวิจัยร่วมอีกด้วย
12. ทดสอบการทำงานและวิเคราะห์ประสิทธิภาพของเครื่องต้นแบบเครื่องโทรศัพท์ผ่านอินเทอร์เน็ตที่ได้ออกแบบและสร้างขึ้นมา โดยในขั้นตอนนี้จะต้องมีอุปกรณ์เพิ่มเติมในการทดสอบและเก็บข้อมูล หัวหน้าโครงการจะคอยเป็นที่ปรึกษาถึงวิธีในการทดสอบ รวมทั้งการประเมินผลการทดสอบและทางบริษัทที่ร่วมโครงการจะคอยช่วยเหลือในด้านอุปกรณ์ทดสอบต่างๆ ที่จำเป็น
13. สรุปผลการดำเนินการ แนวทางการพัฒนาต่อไปในอนาคต ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข ขั้นตอนสุดท้ายนี้ผู้ร่วมโครงการจะต้องอภิปรายและสรุปปัญหาต่างๆ ที่เกิดขึ้นระหว่างการทำโครงการรวมทั้งระดับความสำเร็จของโครงการร่วมกับหัวหน้าโครงการและบริษัทที่ร่วมโครงการ เพื่อเป็นแนวทางในการพัฒนาต่อไป

1.9 ขอบเขตของการวิจัย

เพื่อวิจัย พัฒนา ออกแบบและสร้างอุปกรณ์ต้นแบบระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลชนิดใหม่คือโพรโทคอลเริ่มต้นเซสชันที่สามารถติดต่อระหว่างกันได้ที่มีราคาถูกกว่าระบบโทรศัพท์ผ่านอินเทอร์เน็ตที่ใช้โพรโทคอลแบบเดิมคือ H.323 ซึ่งซับซ้อนกว่า ที่ต้องนำเข้ามาจากต่างประเทศ โดยคำนึงถึงต้นทุนในการผลิตและคุณลักษณะของเครื่องต้นแบบเพื่อนำผลงานวิจัยไปใช้ในเชิงพาณิชย์และสามารถพัฒนาหน้าที่การทำงานใหม่ ๆ ที่จะเกิดขึ้นต่อไปในอนาคตได้ โดยแบ่งออกเป็น 2 ขั้นตอนดังนี้

ปีที่ 1 การพัฒนาต้นแบบระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชันโดยใช้ซอฟต์แวร์

ปีที่ 2 การพัฒนาเครื่องต้นแบบระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชันโดยใช้ฮาร์ดแวร์

1.10 ระยะเวลาการวิจัย

2 ปี

1.11 แผนการดำเนินงานตลอดโครงการ

โปรดดูตารางที่แนบมา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โครงการวิจัยร่วม: การพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน ปีที่ 1 (การพัฒนาโดยใช้ซอฟต์แวร์)

ขั้นตอน	แนวทางการทำโครงการ/ แผนการดำเนินโครงการ	ต.ค. 50	พ.ย.	ธ.ค.	ม.ค. 51	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
1	ศึกษาระบบโทรศัพท์ผ่านอินเทอร์เน็ต												
2	ศึกษามาตรฐานโพรโทคอล H.323	↔											
3	ศึกษามาตรฐานโพรโทคอล SIP		↔										
4	ศึกษาการประยุกต์ใช้โปรแกรมภาษา ต่าง ๆ กับการพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตโดยใช้ซอฟต์แวร์		↔										
5	พัฒนาโปรแกรมในส่วนสร้างและแยก SIP Message ตาม RFC 2543			↔		↔							
6	เขียน Flow Chart กำหนดขั้นตอนการทำงานของโปรแกรมแต่ละส่วน						↔						
7	พัฒนาโปรแกรมตามขั้นตอนการทำงานของโปรแกรมที่ได้กำหนดไว้						↔				↔		
8	ทดสอบการทำงานของโปรแกรมและแก้ไขข้อผิดพลาด										↔	↔	
9	จัดแยกโปรแกรมเป็นมอดูล (Library Module) เพื่อให้ง่ายต่อการพัฒนาเป็นเครื่องต้นแบบที่เป็นฮาร์ดแวร์ต่อไป												↔

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

โครงการวิจัยร่วม: การพัฒนาระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน ปีที่ 2 (การพัฒนาโดยใช้ฮาร์ดแวร์)

ขั้นตอน	แนวทางการทำโครงการ/ แผนการดำเนินโครงการ	ต.ค. 51	พ.ย.	ธ.ค.	ม.ค. 52	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
10	ศึกษาวิธีการสื่อสารระหว่างเครื่องโทรศัพท์ผ่านอินเทอร์เน็ต												
	- ศึกษาวิธีการทางฮาร์ดแวร์ในการต่อการสื่อสารระหว่างเครื่องโทรศัพท์ผ่านอินเทอร์เน็ต - ศึกษาการเขียนโปรแกรมของไมโครโปรเซสเซอร์ - ศึกษาวิธีการทางฮาร์ดแวร์และซอฟต์แวร์ในการต่อการสื่อสารระหว่างกัน		←	←									
11	ออกแบบและสร้างเครื่องต้นแบบ				←	—	—	—	—	→			
12	ทดสอบการทำงานและวิเคราะห์ประสิทธิภาพของเครื่องต้นแบบ								←	—	—	→	
13	สรุปผลการดำเนินการ แนวทางการพัฒนาต่อไปในอนาคต ปัญหาที่เกิดขึ้นและแนวทางการแก้ไข												↔

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1.12 สถานที่ทำการวิจัย

ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

1.13 อุปกรณ์การวิจัย

<u>ปีที่ 1</u>

1. คอมพิวเตอร์ส่วนบุคคลแบบ Desktop 1 เครื่อง

   สำหรับพัฒนาซอฟต์แวร์ระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน, จำลองการทำงานของระบบโทรศัพท์ผ่านอินเทอร์เน็ตเพื่อให้ได้การออกแบบระบบที่ดีที่สุดก่อนการลงมือสร้างฮาร์ดแวร์ และเพื่อตรวจสอบการทำงานของฮาร์ดแวร์และช่วยวิเคราะห์การสัญญาณ (Signalling) ของระบบ

2. คอมพิวเตอร์ส่วนบุคคลแบบ Notebook 1 เครื่อง

   สำหรับพัฒนาซอฟต์แวร์ระบบโทรศัพท์ผ่านอินเทอร์เน็ตบนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน, จำลองการทำงานของระบบโทรศัพท์ผ่านอินเทอร์เน็ตเพื่อให้ได้การออกแบบระบบที่ดีที่สุดก่อนการลงมือสร้างฮาร์ดแวร์ และเพื่อตรวจสอบการทำงานของฮาร์ดแวร์และช่วยวิเคราะห์การสัญญาณ (Signalling) ของระบบ เช่นเดียวกับคอมพิวเตอร์ส่วนบุคคลแบบ Desktop ใช้ในกรณีทำงานร่วมกับภาคเอกชนนอกสถานที่

<u>ปีที่ 2</u>

-

หมายเหตุ

เครื่องคอมพิวเตอร์ส่วนบุคคลชนิด Desktop จำนวน 1 เครื่อง มีคุณลักษณะเฉพาะดังนี้

- หน่วยประมวลผลกลางมีความเร็วไม่น้อยกว่า Core 2 Duo 2.33 GHz
- หน่วยความจำหลัก 2 GB หรือมากกว่า
- ฮาร์ดดิสก์ขนาดไม่ต่ำกว่า 250 GB 7,200 รอบต่อนาที x 2
- จอภาพ LCD ขนาด 19 นิ้วหรือใหญ่กว่า แสดงผลที่ความละเอียดไม่น้อยกว่า 1280x1024 จุดภาพ
- การ์ดจอแบบ Geforce หรือดีกว่า ขนาดไม่น้อยกว่า 256 MB PCI Express
- Case with Power Supply
- Reader All in one
- Drive DVD-RW SATA
- Speaker

- Mouse Optical
- Keyboard

เครื่องคอมพิวเตอร์ส่วนบุคคลชนิด Notebook จำนวน 1 เครื่อง มีคุณลักษณะเฉพาะดังนี้

- หน่วยประมวลผลกลางมีความเร็วไม่น้อยกว่า Core 2 Duo 1.8 GHz
- หน่วยความจำหลัก DDR2 SDRAM 512 MB หรือมากกว่า
- ฮาร์ดดิสก์ขนาดไม่ต่ำกว่า 120 GB SATA
- จอภาพ WXGA ขนาด 14.1 นิ้วหรือใหญ่กว่า
- Drive DVD SuperMulti Double Layer
- 5-in-1 Card Reader
- Built in Camera 1.3M
- Microsoft Windows Vista Home Basic
- รับประกันแบบ International อย่างน้อย 1 ปี

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2
การศึกษาระบบไมโครคอนโทรลเลอร์และการเลือกอุปกรณ์สำหรับระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VOIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน

2.1 ไมโครคอนโทรลเลอร์ MCS-51

2.1.1 โครงสร้างพื้นฐาน และคุณสมบัติของ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ได้มีการนำไปใช้งานกันอย่างแพร่หลาย ปัจจุบันบริษัทผู้ผลิตได้พัฒนาความสามารถของไมโครคอนโทรลเลอร์ MCS-51 รุ่นใหม่ ๆ ให้สามารถทำงานได้อย่างมีประสิทธิภาพ และมีความเร็วเพิ่มขึ้นมาก แต่ยังคงโครงสร้างพื้นฐานที่สำคัญไว้ดังเดิมมีรายละเอียดดังนี้

1. เป็นไมโครคอนโทรลเลอร์ที่มีหน่วยประมวลผลกลางแบบ 8 บิต
2. มีคำสั่งคำนวณทางคณิตศาสตร์ และตรรกศาสตร์ (Boolean processor)
3. มีแอดเดรสบัสขนาด 16 บิตทำให้สามารถอ้างตำแหน่งหน่วยความจำโปรแกรม และหน่วยความจำข้อมูลได้ 64 กิโลไบต์
4. มีหน่วยความจำ (RAM) ภายในขนาด 128 ไบต์ (8051/8031) หรือ 256 ไบต์ (8052/8032)
5. มีพอร์ตอนุกรมทำงานแบบดูเพล็กซ์เต็ม (Full Duplex) 1 พอร์ต
6. มีพอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 บิต
7. มีไทเมอร์ 2 ตัว (8051/8031) หรือ 3 ตัว (8052/8032)
8. มีวงจรควบคุมการเกิดอินเตอร์รัปต์ 5 ประเภท (8051/8031) หรือ 6 ประเภท (8052/8032)
9. มีวงจรออสซิลเลเตอร์ภายในตัว

ไมโครคอนโทรลเลอร์ MCS-51 มีวงจรออสซิลเลเตอร์อยู่ภายใน ดังนั้นในการใช้งานจึงสามารถต่อคริสตอล และตัวเก็บประจุเข้ากับขาสัญญาณ XTAL1 และ XTAL2 ได้โดยตรงดังแสดงในรูปที่ 2.1 โดยความถี่ของคริสตอลที่ต่อเข้ากับไมโครคอนโทรลเลอร์จะเป็นตัวระบุความเร็วในการทำงานโดยตรง ในไมโครคอนโทรลเลอร์ MCS-51 ปกติ 1 แมชชีนไซเคิล (machine cycle) จะใช้สัญญาณนาฬิกาจำนวน 12 ลูก และในการทำงานแต่ละคำสั่งไมโครคอนโทรลเลอร์จะใช้เวลาในการทำงาน 1-4 แมชชีนไซเคิล ขึ้นอยู่กับความซับซ้อนของคำสั่งนั้น ทำให้ความสัมพันธ์ของเวลาที่ใช้ในการทำงาน กับความถี่คริสตอลเป็นดังสมการที่ (2.1)

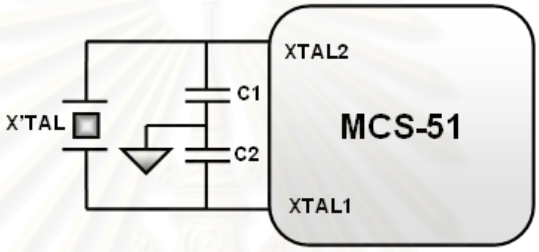
$$T = \frac{C \times 12}{f_{XTAL}} \qquad (2.1)$$

โดย T = ช่วงเวลาที่ใช้ในการทำงาน

C = จำนวนแมชชีนไซเคิล

f_{XTAL} = ความถี่ของคริสตอล

ในปัจจุบันผู้ผลิตได้พัฒนาให้ไมโครคอนโทรลเลอร์สามารถทำงานได้เร็วขึ้นโดยเพิ่มความสามารถในการรองรับคริสตอลความถี่สูงขึ้น รวมไปถึงการปรับปรุงการทำงานภายในให้ไมโครคอนโทรลเลอร์ใช้จำนวนสัญญาณนาฬิกาในการสร้างแมชชีนไซเคิลน้อยลง โดยในบางรุ่น 1 แมชชีนไซเคิลใช้สัญญาณนาฬิกาเพียงแค่ 1 ลูกเท่านั้น



รูปที่ 2.1 การต่อคริสตอลเข้ากับไมโครคอนโทรลเลอร์ MCS-51

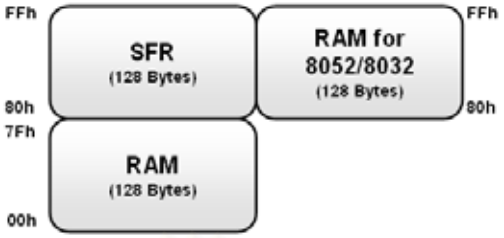
2.1.2 ระบบหน่วยความจำของ MCS-51

ระบบหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 นั้นสามารถแบ่งออกเป็น 3 ส่วน คือ หน่วยความจำโปรแกรม (Program Memory) หน่วยความจำข้อมูลภายใน (Internal RAM) และหน่วยความจำข้อมูลภายนอก (External RAM)

หน่วยความจำโปรแกรมใช้สำหรับเก็บคำสั่งที่ควบคุมการทำงานไมโครคอนโทรลเลอร์ สามารถเก็บข้อมูลได้โดยไม่ต้องอาศัยไฟเลี้ยง ไมโครคอนโทรลเลอร์ส่วนใหญ่จะมีหน่วยความจำโปรแกรมอยู่ภายในตัว โดยโครงสร้างของหน่วยความจำโปรแกรมนั้นมีหลายชนิดเช่น ROM (Read Only Memory), EEPROM (Erasable Programmable Read Only Memory), Flash เป็นต้น ในการใช้งานปกติไมโครคอนโทรลเลอร์จะไม่เขียนข้อมูล แต่อ่านคำสั่งจากหน่วยความจำโปรแกรมนี้เท่านั้น ส่วนการลงโปรแกรมให้กับไมโครคอนโทรลเลอร์นั้นมีวิธีหลากหลาย และขึ้นอยู่กับชนิดของหน่วยความจำที่ใช้

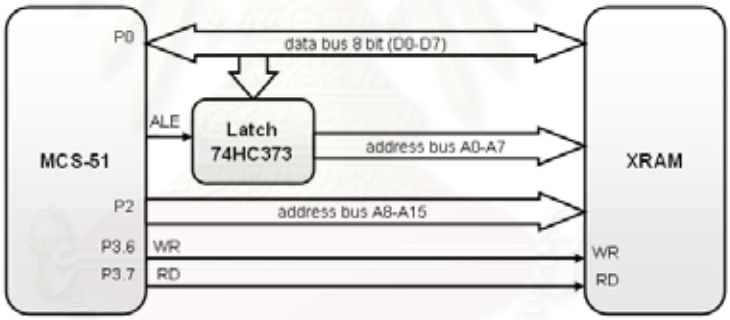
หน่วยความจำข้อมูลภายในของไมโครคอนโทรลเลอร์ MCS-51 นั้นมีขนาด 128-256 ไบต์ ขึ้นอยู่กับรุ่นของไมโครคอนโทรลเลอร์ แบ่งออกเป็น 2 ส่วนคือหน่วยความจำส่วนล่าง (00h-7Fh) และหน่วยความจำส่วนบน (80h-FFh) หน่วยความจำส่วนล่างนั้นใช้สำหรับเก็บข้อมูลทั่วไป ส่วนหน่วยความจำส่วนบนนั้นเป็นหน่วยความจำพิเศษที่ใช้สำหรับกำหนดการทำงานของไมโครคอนโทรลเลอร์ หรือเรียกว่า SFR (Special Function Register) ชิพรุ่นใหม่ ๆ ที่มีโครงสร้าง

เป็นไปตามตระกูล 8052 นั้นมีหน่วยความจำทั่วไปที่ต้องเข้าถึงด้วยวิธี Indirect อยู่ในตำแหน่งหน่วยความจำส่วนบนซ้อนทับอยู่กับ SFR ดังแสดงในรูปที่ 2.2



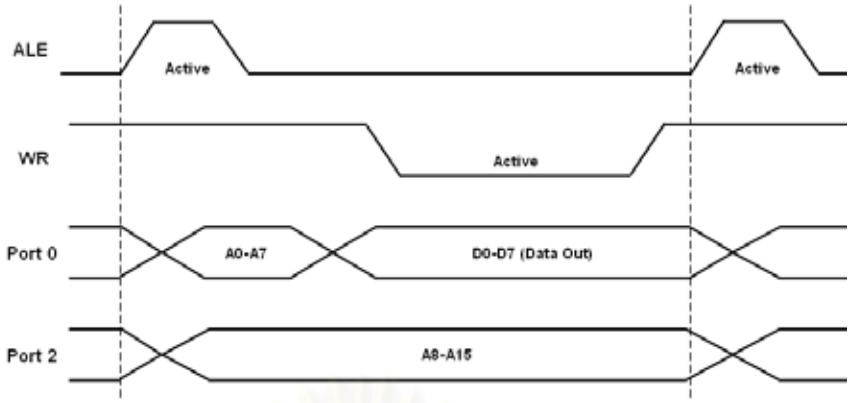
รูปที่ 2.2 โครงสร้างหน่วยความจำข้อมูลภายใน MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 สามารถต่อหน่วยความจำภายนอกแบบ Memory Map เพิ่มได้ถึง 64 กิโลไบต์ โดยใช้พอร์ต 0 เป็นสายสัญญาณข้อมูลขนาด 8 บิต (Data Bus D0-D7) และสายสัญญาณกำหนดตำแหน่งหน่วยความจำไบต์ล่าง (Address Bus A0-A7) ในการใช้งานนั้นต้องใช้ไอซี Latch ทำหน้าที่แยกสัญญาณทั้งสองออกจากกันด้วยสัญญาณควบคุม ALE จากไมโครคอนโทรลเลอร์ ส่วนสายสัญญาณกำหนดตำแหน่งหน่วยความจำไบต์บน (A8-A15) และสายสัญญาณควบคุม WR, RD นั้นใช้พอร์ต 2 และพอร์ต 3.6, 3.7 ได้โดยตรงตามลำดับ ดังรูปที่ 2.3



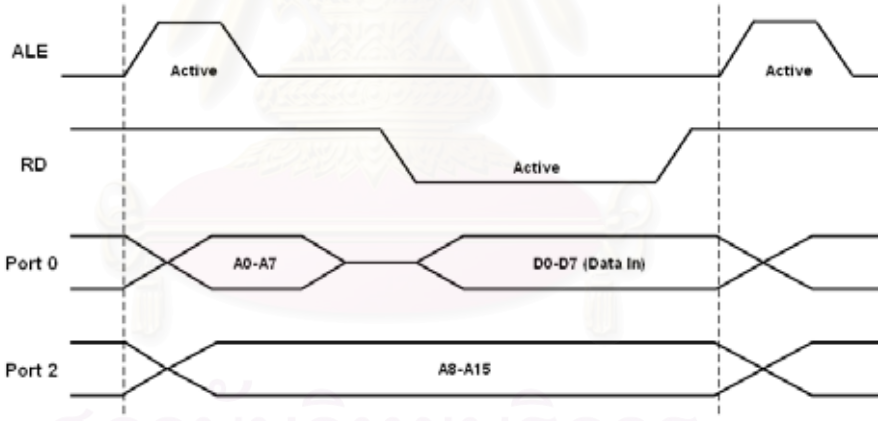
รูปที่ 2.3 การต่อหน่วยความจำข้อมูลภายนอกเพิ่ม

ในการเขียนข้อมูลลงหน่วยความจำภายนอก ในช่วงขอบขาขึ้นของสัญญาณ ALE พอร์ต 2 จะส่งตำแหน่งไบต์บน (A8-A15) ให้กับหน่วยความจำภายนอก ส่วนพอร์ต 0 จะส่งตำแหน่งไบต์ล่าง (A0-A7) เพื่อให้ไอซีแลชค้างค่าชี้ตำแหน่งแอดเดรสไบต์ล่างนี้ไว้ เมื่อสัญญาณ ALE เปลี่ยนสถานะจากลอจิก ‘1’ เป็น ‘0’ พอร์ต 0 จะเปลี่ยนไปส่งค่าข้อมูลขนาด 8 บิตให้กับหน่วยความจำภายนอก และสัญญาณ WR จะเปลี่ยนสถานะจาก ลอจิก ‘1’ เป็น ‘0’ เพื่อให้หน่วยความจำภายนอกรับข้อมูลดังแสดงในรูปที่ 2.4



รูปที่ 2.4 การเขียนข้อมูลลงหน่วยความจำข้อมูลภายนอก

ในการอ่านข้อมูลจากหน่วยความจำภายนอก ในช่วงขอบขาขึ้นของสัญญาณ ALE พอร์ต 2 จะส่งตำแหน่งไบต์ล่าง (A8-A15) ให้กับหน่วยความจำภายนอก ส่วนพอร์ต 0 จะส่งตำแหน่งไบต์ล่าง (A0-A7) เพื่อให้ไอซีแลชค้างค่าชี้ตำแหน่งแอดเดรสไบต์ล่างนี้ไว้ เมื่อสัญญาณ ALE เปลี่ยนสถานะจากลอจิก '1' เป็น '0' พอร์ต 0 จะเปลี่ยนโมดการทำงานเป็นพอร์ตอินพุตเพื่อรอรับข้อมูลจากหน่วยความจำภายนอก และเมื่อสัญญาณ RD เปลี่ยนสถานะจาก ลอจิก '1' เป็น '0' ไมโครคอนโทรลเลอร์จะอ่านข้อมูลจากหน่วยความจำภายนอกทางพอร์ต 0 ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอก

2.1.3 ไทเมอร์/เคาน์เตอร์ภายใน MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 มีวงจรไทเมอร์/เคาน์เตอร์ (Timer/Counter) ทำหน้าที่นับความถี่ของสัญญาณนาฬิกา หรือนับการเปลี่ยนแปลงลอจิกอยู่ในตัว โดยสามารถทำงานได้หลายโมดขึ้นอยู่กับความต้องการของผู้ใช้ ดังนั้นในการใช้งานจึงต้องกำหนดค่าให้กับรีจิสเตอร์ควบคุมที่เกี่ยวข้องต่าง ๆ เสียก่อน โดยมีรายละเอียดดังนี้

รีจิสเตอร์ TH0, TL0, TH1, TL1, TH2, TL2: เป็นรีจิสเตอร์ที่ใช้ในการนับของวงจรไทเมอร์/เคาน์เตอร์ 0, 1, และ 2 ตามลำดับ

รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง 0x89 ในพื้นที่ของรีจิสเตอร์ SFR ไม่สามารถเข้าถึงได้ในระดับบิต ใช้ในการกำหนดรูปแบบการทำงานของวงจรไทเมอร์/เคาน์เตอร์ 1 และ 2 มีโครงสร้างดังแสดงในรูปที่ 2.6

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0
Timer/Counter 1				Timer/Counter 0			

รูปที่ 2.6 โครงสร้างรีจิสเตอร์ TMOD

GATE: ใช้เลือกลักษณะการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ โดยถ้าบิต GATE มีค่าเป็น '0' ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TR ในรีจิสเตอร์ TCON เป็น '1' ซึ่งเป็นการควบคุมทางซอฟต์แวร์ แต่ถ้าบิต GATE มีค่าเป็น '1' ไทเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TR ในรีจิสเตอร์ TCON เป็น '1' และสถานะลอจิกที่อินพุตอินเตอร์รัปต์ INT0 และ INT1 เป็น '1' ซึ่งเป็นการควบคุมทางฮาร์ดแวร์

C/T (Timer or Counter selector): ใช้เลือกการทำงานของไทเมอร์/เคาน์เตอร์ โดยถ้าบิต C/T นี้เป็น '0' วงจรนับจะใช้สัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์เป็นอินพุต และทำงานเป็นไทเมอร์ แต่ถ้าเป็น '1' วงจรนับจะใช้สัญญาณภายนอกที่เข้ามาทางพอร์ต T0 หรือ T1 เป็นอินพุต และทำงานเป็นเคาน์เตอร์

M1, M0 (Mode selection bit): ใช้เลือกโมดการทำงานของไทเมอร์เคาน์เตอร์ โดยถ้ามีค่าเป็น "00", "01", "10", "11" วงจรนับจะทำงานในโมด 0, 1, 2, และ 3 ตามลำดับ ในการทำงานในโมด 0 เป็นการนับแบบ 13 บิต โดยใช้ 5 บิตล่างของรีจิสเตอร์ TL กับทั้ง 8 บิตของรีจิสเตอร์ TH เมื่อมีการนับไมโครคอนโทรลเลอร์จะเพิ่มค่ารีจิสเตอร์ TL ครั้งละ 1 เรื่อย ๆ จนรีจิสเตอร์ TL มีค่าเป็น "00011111_2" และเมื่อเกิดการนับเพิ่มอีกครั้ง รีจิสเตอร์ TL จะส่งบิต 1 ให้ TH นับต่อพร้อมกับรีเซตตัวเองเป็น "00000000_2" เมื่อมีการนับต่อจนกว่ารีจิสเตอร์ TL และ TH มีค่าเป็น "00011111_2" และ "11111111_2" ตามลำดับ จะเกิดการโอเวอร์โฟลว์กับบิต TF ซึ่งจะทำให้ไมโครคอนโทรลเลอร์สร้างสัญญาณอินเตอร์รัปต์สำหรับไทเมอร์/เคาน์เตอร์ และรีเซตรีจิสเตอร์ TH และ TF เป็น 0 อีกครั้ง ส่วนในโมด 1 เป็นการนับแบบ 16 บิตโดยใช้รีจิสเตอร์ TL เก็บข้อมูลไบต์ล่าง และ TH เก็บข้อมูลไบต์บน ในการทำงานจะเริ่มนับค่าจาก 0x0000 หรือจากค่าที่ตั้งเอาไว้จนถึง 0xFFFF และเกิดการโอเวอร์โฟลว์กับบิต TF ซึ่งจะทำให้ไมโครคอนโทรลเลอร์สร้างสัญญาณอินเตอร์รัปต์สำหรับไทเมอร์/เคาน์เตอร์ขึ้น และรีเซตรีจิสเตอร์ TH และ TL ให้มีค่าเป็น 0x0000 เพื่อเริ่มต้นการนับต่อไป ส่วนในโมด 2 เป็นการนับแบบตั้งค่าอัตโนมัติ 8 บิตโดยใช้รีจิสเตอร์ TL เป็นตัวนับ และใช้รีจิสเตอร์ TH เก็บค่าเริ่มต้น เมื่อรีจิสเตอร์ TL ได้นับจนมีค่าเป็น 0xFF จะเกิดการโอเวอร์โฟลว์กับ

บิต TF ไมโครคอนโทรลเลอร์จะสร้างสัญญาณอินเตอร์รัปต์สำหรับไทเมอร์/เคาน์เตอร์ และเปลี่ยนค่าของ รีจิสเตอร์ TL ให้เป็นค่าที่เก็บเอาไว้ใน TH เพื่อเริ่มต้นนับต่อไป ส่วนโมด 3 นั้นเป็นการนับ 8 บิตสำหรับวงจรไทเมอร์/เคาน์เตอร์ 0 โดยใช้รีจิสเตอร์ TH0 และ TL0 เป็นตัวนับทั้งคู่ โดยใช้ TF1 เก็บการเกิดโอเวอร์โฟลว์ของ TH0 และใช้ TF0 เก็บการเกิดโอเวอร์โฟลว์ของ TL0 การทำงานในโมดนี้จะทำให้ไทเมอร์/เคาน์เตอร์ 1 ไม่สามารถสร้างสัญญาณอินเตอร์รัปต์ได้

รีจิสเตอร์ TCON: มีขนาด 8 บิต สามารถเข้าถึงในระดับบิตได้โดยตรง ทำหน้าที่ควบคุมการทำงานของวงจรไทเมอร์/เคาน์เตอร์ และกำหนดรูปแบบการเกิดอินเตอร์รัปต์จากภายนอกมีโครงสร้างดังแสดงในรูปที่ 2.7

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

รูปที่ 2.7 โครงสร้างรีจิสเตอร์ TCON

IT0: ใช้เลือกลักษณะของสัญญาณอินเตอร์รัปต์จากภายนอกตัวที่ 0 (INT0) โดยถ้าเป็น ‘0’ ไมโครคอนโทรลเลอร์จะใช้ขอบขาลงเป็นตัวกำเนิดสัญญาณอินเตอร์รัปต์ แต่ถ้าเป็น ‘1’ ไมโครคอนโทรลเลอร์จะใช้ช่วงสัญญาณลอจิกต่ำเป็นตัวกำเนิดสัญญาณอินเตอร์รัปต์

IE0: ใช้ในการแสดงการเกิดอินเตอร์รัปต์จากภายนอกโดยจะมีค่าเป็น ‘1’ เมื่อเกิดอินเตอร์รัปต์จากภายนอกขึ้น และจะกลับมามีค่าเป็น ‘0’ เมื่อไมโครคอนโทรลเลอร์ได้ทำงานรองรับการเกิดอินเตอร์รัปต์จากภายนอกนั้นเรียบร้อยแล้ว

IT1: เหมือนกับ IT0 แต่เป็นการกำหนดสำหรับการเกิดอินเตอร์รัปต์จากภายนอกตัวที่ 1

IE1: เหมือนกับ IE0 แต่เป็นการกำหนดสำหรับการเกิดอินเตอร์รัปต์จากภายนอกตัวที่ 1

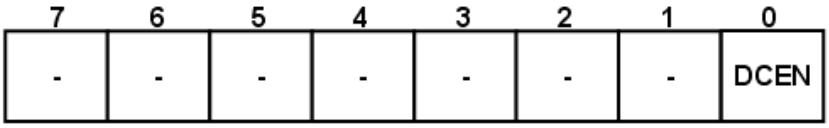
TR0: ใช้ควบคุมการเปิดปิดการทำงานของไทเมอร์/เคาน์เตอร์ 0โดยถ้าเป็น ‘1’ วงจรไทเมอร์เคาน์เตอร์ 0 จะทำงาน แต่ถ้าเป็น ‘0’ วงจรไทเมอร์เคาน์เตอร์ 0 จะหยุดทำงาน

TF0: เป็นบิตแสดงการเกิดโอเวอร์โฟลว์ของไทเมอร์เคาน์เตอร์ 0

TR1: เหมือนกับ TR0 แต่ใช้กับไทเมอร์/เคาน์เตอร์ 1

TF1: เหมือนกับ TF0 แต่ใช้กับไทเมอร์/เคาน์เตอร์ 1

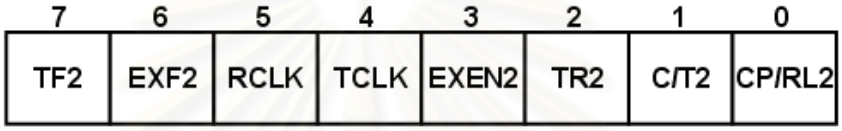
รีจิสเตอร์ T2MOD: มีขนาด 8 บิตไม่สามารถเข้าถึงได้ในระดับบิต ใช้เลือกโมดการทำงานของวงจรไทเมอร์/เคาน์เตอร์ 2 ซึ่งมีในไมโครคอนโทรลเลอร์ MCS-51 ตระกูล 8052 เท่านั้น มีโครงสร้างดังแสดงในรูปที่ 2.8



รูปที่ 2.8 โครงสร้างรีจิสเตอร์ T2MOD

DCEN: ใช้ในการกำหนดรูปแบบการนับของไทเมอร์/เคาน์เตอร์ 2 โดยถ้าเป็น ‘0’ จะเป็นการทำงานแบบนับขึ้นเท่านั้น แต่ถ้าเป็น ‘1’ จะทำงานแบบนับขึ้นหรือลงก็ได้โดยขึ้นอยู่กับสัญญาณอินพุตที่ขา T2EX ถ้าเป็นลอจิก ‘1’ จะเป็นการนับขึ้น และถ้าเป็นลอจิก ‘0’ จะเป็นการนับลง

รีจิสเตอร์ T2CON: มีขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต ใช้ควบคุมการทำงานไทเมอร์/เคาน์เตอร์ 2 มีโครงสร้างดังแสดงในรูปที่ 2.9



รูปที่ 2.9 โครงสร้างรีจิสเตอร์ T2CON

CP/RL2: ใช้เลือกโมดการทำงานของไทเมอร์/เคาน์เตอร์ 2 โดยถ้าเป็น ‘1’ จะทำงานในโมดตรวจจับสัญญาณ (Capture) และถ้าเป็น ‘0’ จะทำงานในโมดตั้งค่านับอัตโนมัติ (Auto-Reload)

C/T2: ใช้เลือกการทำงานของไทเมอร์/เคาน์เตอร์ 2 โดยถ้าบิต C/T2 นี้เป็น ‘0’ วงจรนับจะใช้สัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์เป็นอินพุต และทำงานเป็นไทเมอร์ แต่ถ้าเป็น ‘1’ วงจรนับจะใช้สัญญาณภายนอกที่เข้ามาทางพอร์ต T0 หรือ T1 เป็นอินพุต และทำงานเป็นเคาน์เตอร์

TR2: ใช้ควบคุมการเปิดปิดการทำงานของไทเมอร์/เคาน์เตอร์ 2 โดยถ้าเป็น ‘1’ วงจรไทเมอร์เคาน์เตอร์ 2 จะทำงาน แต่ถ้าเป็น ‘0’ วงจรไทเมอร์เคาน์เตอร์ 2 จะหยุดทำงาน

EXEN2: ใช้ควบคุมการเปิดปิดวงจรตรวจจับการเปลี่ยนแปลงระดับลอจิกจาก ‘1’ เป็น ‘0’ ของสัญญาณที่ขา T2EX โดยถ้ามีค่าเป็น ‘1’ จะเป็นการเปิดวงจรตรวจจับ

TCLK: ใช้เลือกไทเมอร์/เคาน์เตอร์ในการทำงานเป็นตัวกำหนดอัตราบอดสำหรับการส่งข้อมูลแบบอนุกรม โดยถ้ามีค่าเป็น ‘1’ คือให้ไทเมอร์/เคาน์เตอร์ 2 เป็นตัวกำหนดอัตราบอดสำหรับการส่งข้อมูล แต่ถ้าเป็น ‘0’ คือให้ไทเมอร์/เคาน์เตอร์ 1 เป็นตัวกำหนดอัตราบอดสำหรับการส่งข้อมูลแบบอนุกรม

RCLK: ใช้เลือกไทเมอร์/เคาน์เตอร์ในการทำงานเป็นตัวกำหนดอัตราบอดสำหรับการรับข้อมูลแบบอนุกรม โดยถ้ามีค่าเป็น ‘1’ คือให้ไทเมอร์/เคาน์เตอร์ 2 เป็นตัวกำหนดอัตราบอดสำหรับการรับข้อมูล แต่ถ้าเป็น ‘0’ คือให้ไทเมอร์/เคาน์เตอร์ 1 เป็นตัวกำหนดอัตราบอดสำหรับการรับข้อมูลแบบอนุกรม

EXF2: เป็นบิตแสดงการตั้งค่าใหม่ หรือการตรวจจับสัญญาณ ซึ่งเกิดจากสัญญาณที่ขา T2EX เปลี่ยนระดับจากลอจิก ‘1’ เป็น ‘0’ ในกรณีที่บิต EXEN2 ถูกตั้งให้เป็น ‘1’

TF2: เป็นบิตแสดงการเกิดโอเวอร์โฟลว์ของไทเมอร์เคาน์เตอร์ 2

จะเห็นได้ว่าโมดการทำงานของไทเมอร์/เคาน์เตอร์ 2 นั้นขึ้นอยู่กับรีจิสเตอร์ RCLK, TCLK, และ CP/RL2 และสามารถแบ่งการทำงานออกเป็น 3 โมดคือโมดตรวจจับสัญญาณ, โมดตั้งค่าการนับอัตโนมัติ, และโมดกำเนิดอัตราบอดในการสื่อสารข้อมูลแบบอนุกรม ในโมดตรวจจับสัญญาณ การใช้งานต้องกำหนดบิต EXEN2 ให้เป็น ‘1’ ก่อนเพื่อให้วงจรตรวจจับขอบขาลงของสัญญาณที่ขาอินพุต T2EX ซึ่งตรงกับขา P1.1 ทำงาน เมื่อเปิดวงจรไทเมอร์/เคาน์เตอร์ 2 แล้ว รีจิสเตอร์ TL2 และ TH2 จะนับไปเรื่อย ๆ จนกว่าขา T2EX จะตรวจจับขอบขาลงของสัญญาณได้ เมื่อตรวจจับได้ ค่าของรีจิสเตอร์ TL2 และ TH2 จะถูกส่งต่อไปยังรีจิสเตอร์ RCAP2L และ RCAP2H ตามลำดับ พร้อมกับเซตบิต EXF2 ให้เป็น ‘1’ เพื่อสร้างสัญญาณอินเตอร์รัปต์จากไทเมอร์/เคาน์เตอร์ขึ้น ส่วนในโมดตั้งค่าการนับอัตโนมัตินั้นสามารถทำงานได้ 2 ลักษณะขึ้นอยู่กับสถานะของบิต EXEN2 ในกรณีที่บิต EXEN2 เป็น ‘0’ เมื่อไทเมอร์/เคาน์เตอร์ 2 นับครบรอบ จะเกิดอินเตอร์รัปต์จากการโอเวอร์โฟลว์ของบิต TF2 จากนั้นไมโครคอนโทรลเลอร์จะโหลดค่าในรีจิสเตอร์ RCAP2L และ RCAP2H เข้ามาในรีจิสเตอร์ตัวนับ TL2 และ TH2 โดยอัตโนมัติ เพื่อเริ่มนับค่าใหม่ แต่ในกรณีที่บิต EXEN2 เป็น ‘1’ เมื่อไทเมอร์/เคาน์เตอร์ 2 นับจนครบรอบ จะเกิดอินเตอร์รัปต์ และโหลดค่าจากรีจิสเตอร์ RCAP2L และ RCAP2H เข้ามาในรีจิสเตอร์ TL2 และ TH2 เมื่อบิต EXF2 มีค่าเป็น ‘1’ ซึ่งเกิดจากสัญญาณที่ขา T2EX เปลี่ยนแปลงจากระดับลอจิก ‘1’ เป็น ‘0’

2.1.4 ระบบอินเตอร์รัปต์ของ MCS-51

การอินเตอร์รัปต์ เป็นการขัดจังหวะการทำงานโดยปกติของไมโครคอนโทรลเลอร์ ในไมโครคอนโทรลเลอร์ MCS-51 สามารถตอบสนองการอินเตอร์รัปต์ได้จาก 6 แหล่งกำเนิดคือ อินเตอร์รัปต์จากสัญญาณภายนอกที่ขา INT0 และ INT1, อินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 0, 1, และ 2 และอินเตอร์รัปต์จากพอร์ตสื่อสารอนุกรม ในการเลือกใช้งานอินเตอร์รัปต์นั้นสามารถทำได้โดยการตั้งค่าภายในรีจิสเตอร์ IE (Interrupt Enable) ซึ่งเป็นรีจิสเตอร์ควบคุมการเปิดปิดอินเตอร์รัปต์ต่าง ๆ มีแอดเดรสอยู่ที่ 0xA8 ใน SFR และมีโครงสร้างดังแสดงในรูปที่ 2.10

7	6	5	4	3	2	1	0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

รูปที่ 2.10 โครงสร้างรีจิสเตอร์ IE

EA: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์ทั้งหมดโดยถ้ามีค่าเป็น ‘0’ ไมโครคอนโทรลเลอร์จะไม่ตอบสนองการอินเตอร์รัปต์ทั้งหมด แต่ถ้าเป็น ‘1’ ไมโครคอนโทรเลอร์จะสามารถตอบสนองอินเตอร์รัปต์ต่าง ๆ ได้

ET2: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 2 โดยถ้าเป็น ‘1’ จะเป็นการเปิด และถ้าเป็น ‘0’ จะเป็นการปิด

ES: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์จากพอร์ตสื่อสารอนุกรม โดยถ้าเป็น ‘1’ จะเป็นการเปิด และถ้าเป็น ‘0’ จะเป็นการปิด

ET1: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 1 โดยถ้าเป็น ‘1’ จะเป็นการเปิด และถ้าเป็น ‘0’ จะเป็นการปิด

EX1: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์จากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT1 โดยถ้าเป็น ‘1’ จะเป็นการเปิด และถ้าเป็น ‘0’ จะเป็นการปิด

ET0: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 0 โดยถ้าเป็น ‘1’ จะเป็นการเปิด และถ้าเป็น ‘0’ จะเป็นการปิด

EX0: ใช้ควบคุมการตอบสนองอินเตอร์รัปต์จากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT0 โดยถ้าเป็น ‘1’ จะเป็นการเปิด และถ้าเป็น ‘0’ จะเป็นการปิด

การกำหนดระดับความสำคัญให้กับสัญญาณอินเตอร์รัปต์ในไมโครคอนโทรลเลอร์ MCS-51 นั้น สามารถทำได้ 2 ระดับคือสูงและต่ำ ในการใช้งานถ้าต้องการตั้งให้อินเตอร์รัปต์หนึ่งมีความสำคัญสูงสามารถทำได้โดยการตั้งค่ารีจิสเตอร์ IP (Interrupt Priority) ซึ่งอยู่ที่ตำแหน่ง 0xB8 ใน SFR และมีโครงสร้างดังแสดงในรูปที่ 2.11 ให้ตำแหน่งอินเตอร์รัปต์นั้นมีค่าเป็น ‘1’

7	6	5	4	3	2	1	0
-	-	PT2	PS	PT1	PX1	PT0	PX0

รูปที่ 2.11 โครงสร้างรีจิสเตอร์ IP

PT2: อินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 2

PS: อินเตอร์รัปต์จากพอร์ตสื่อสารอนุกรม

PT1: อินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 1

PX1: อินเตอร์รัปต์จากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT1

PT0: อินเตอร์รัปต์จากวงจรไทเมอร์/เคาน์เตอร์ 0

PX0: อินเตอร์รัปต์จากสัญญาณภายนอกที่ป้อนเข้ามายังขา INT0

2.1.5 การใช้งาน EEPROM ภายใน AT89C51ED2

ไมโครคอนโทรลเลอร์ AT89C51ED2 เป็นไมโครคอนโทรลเลอร์ตระกูล 8051 รุ่นพิเศษ ซึ่งมี EEPROM ภายในขนาด 2048 ไบต์ สามารถใช้เก็บข้อมูลโดยไม่ต้องอาศัยไฟเลี้ยงได้ ในการใช้งาน EEPROM ภายในนั้น ต้องควบคุมผ่านรีจิสเตอร์พิเศษ EECON ที่ตำแหน่ง 0xD2 มีโครงสร้างดังแสดงในรูปที่ 2.12

7	6	5	4	3	2	1	0
-	-	-	-	-	-	EEE	EEBUSY

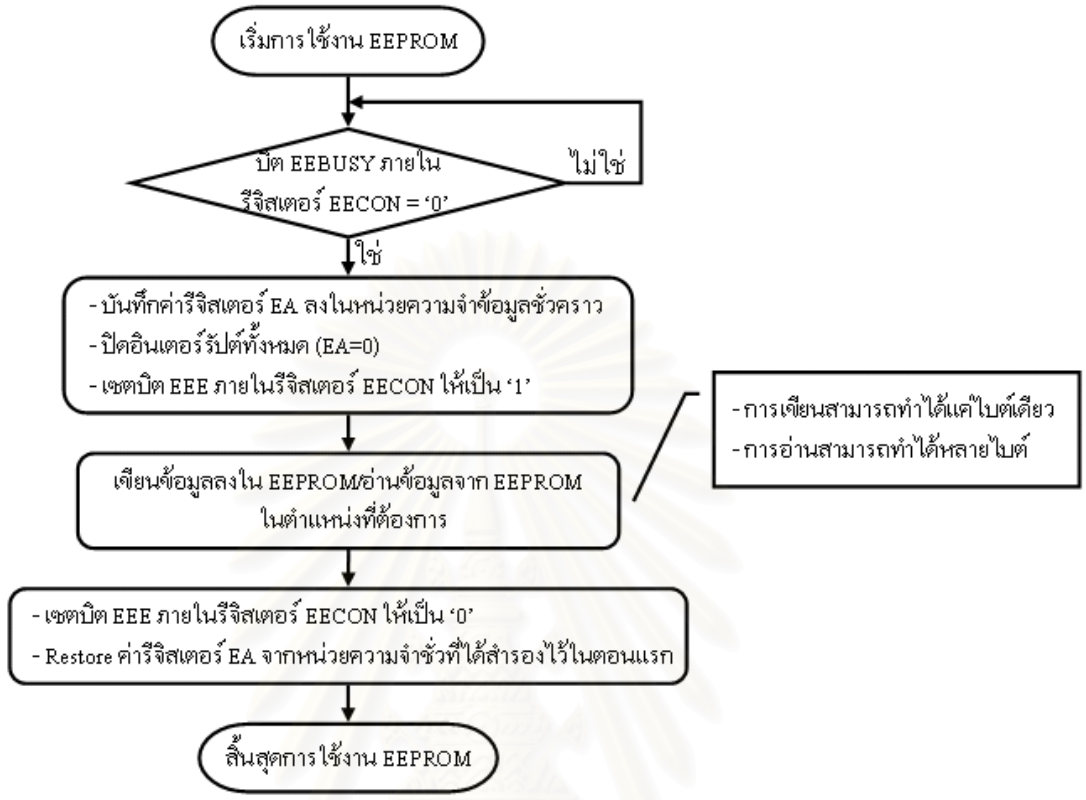
รูปที่ **2.12** โครงสร้างรีจิสเตอร์ EECON

EEE (Enable EEPROM Space bit): ใช้แบ่งแยกการอ่าน และเขียนข้อมูลของไมโครคอนโทรลเลอร์โดยถ้า EEE เป็น ‘1’ ไมโครคอนโทรลเลอร์จะอ่าน และเขียนข้อมูลลง EEPROM แต่ถ้าเป็น ‘0’ ไมโครคอนโทรลเลอร์จะอ่าน และเขียนข้อมูลลงหน่วยความจำภายนอก

EEBUSY (Programming Busy flag): ใช้บ่งบอกสถานะ การอ่าน และเขียนข้อมูลของไมโครคอนโทรลเลอร์ กับ EEPROM โดยถ้า EEBUSY เป็น ‘1’ แสดงว่าไมโครคอนโทรลเลอร์กำลังอ่าน หรือเขียนข้อมูลลงใน EEPROM อยู่ แต่ถ้าเป็น ‘0’ แสดงว่าไมโครคอนโทรลเลอร์ได้อ่าน หรือเขียนข้อมูลลง EEPROM เรียบร้อยแล้ว

การใช้งาน EEPROM ภายในไมโครคอนโทรเลอร์ AT89C51ED2 สามารถทำได้ดังแสดงในรูปที่ 2.13 เมื่อต้องการอ่าน หรือเขียนข้อมูลลง EEPROM ต้องตรวจสอบบิต EEBUSY ภายในรีจิสเตอร์ EECON ก่อน และรอจนกว่าบิต EEBUSY เป็น ‘0’ ซึ่งแสดงว่าไมโครคอนโทรลเลอร์ไม่ได้ทำการอ่าน หรือเขียนข้อมูลลง EEPROM เมื่อบิต EEBUSY เป็น ‘0’ ให้ทำการบันทึกสถานะของบิต EA ซึ่งเป็นบิตควบคุมการเกิดอินเตอร์รัปต์ทั้งหมดภายในไมโครคอนโทรลเลอร์ลงในหน่วยความจำข้อมูลชั่วคราวก่อน และตั้งให้รีจิสเตอร์ EA เป็น ‘0’ เพื่อปิดอินเตอร์รัปต์ทั้งหมด ทั้งนี้เนื่องจากหน่วยความจำข้อมูลภายนอก กับ EEPROM มีการใช้สายสัญญาณกำหนดตำแหน่งร่วมกัน ดังนั้นในขณะที่ทำการอ่าน หรือเขียนข้อมูลลง EEPROM ถ้าเกิดอินเตอร์รัปต์ และการทำงานรองรับอินเตอร์รัปต์นั้นมีการใช้งานหน่วยความจำภายนอก จะทำให้การอ่านและเขียนข้อมูลลง EEPROM ผิดพลาดได้ เมื่อปิดอินเตอร์รัปต์ทั้งหมดแล้วจึงค่อยเซตบิต EEE ภายในรีจิสเตอร์ EECON ให้เป็น ‘1’ เพื่อเลือกอ่าน หรือเขียนข้อมูลลง EEPROM จากนั้นจึงสามารถอ่าน หรือเขียนข้อมูลลง EEPROM ได้ การเขียนข้อมูลลง EEPROM นั้นสามารถทำได้ทีละไบต์ ดังนั้นเมื่อเขียนข้อมูลลง EEPROM แล้ว ในการเขียนข้อมูลไบต์ต่อไปต้องเซตบิต EEE ให้เป็น ‘0’ ก่อน แล้วค่อยเริ่มขั้นตอนแรกใหม่ ส่วนการอ่านข้อมูลจาก EEPROM นั้นสามารถทำได้ทีละหลายไบต์ เหมือนกับการอ่านข้อมูลจากหน่วยความจำภายนอกทุกอย่าง เมื่อทำการอ่าน หรือเขียนเสร็จแล้ว สามารถหยุด

การใช้งาน EEPROM ได้โดยเซตบิต EEE ให้เป็น ‘0’ และนำค่าภายในบิต EA ที่ได้สำรองไว้ในหน่วยความจำ กลับมาเขียนลงในบิต EA เป็นอันสิ้นสุดการใช้งาน EEPROM



รูปที่ 2.13 การใช้งาน EEPROM ภายในไมโครคอนโทรลเลอร์ AT89C51ED2

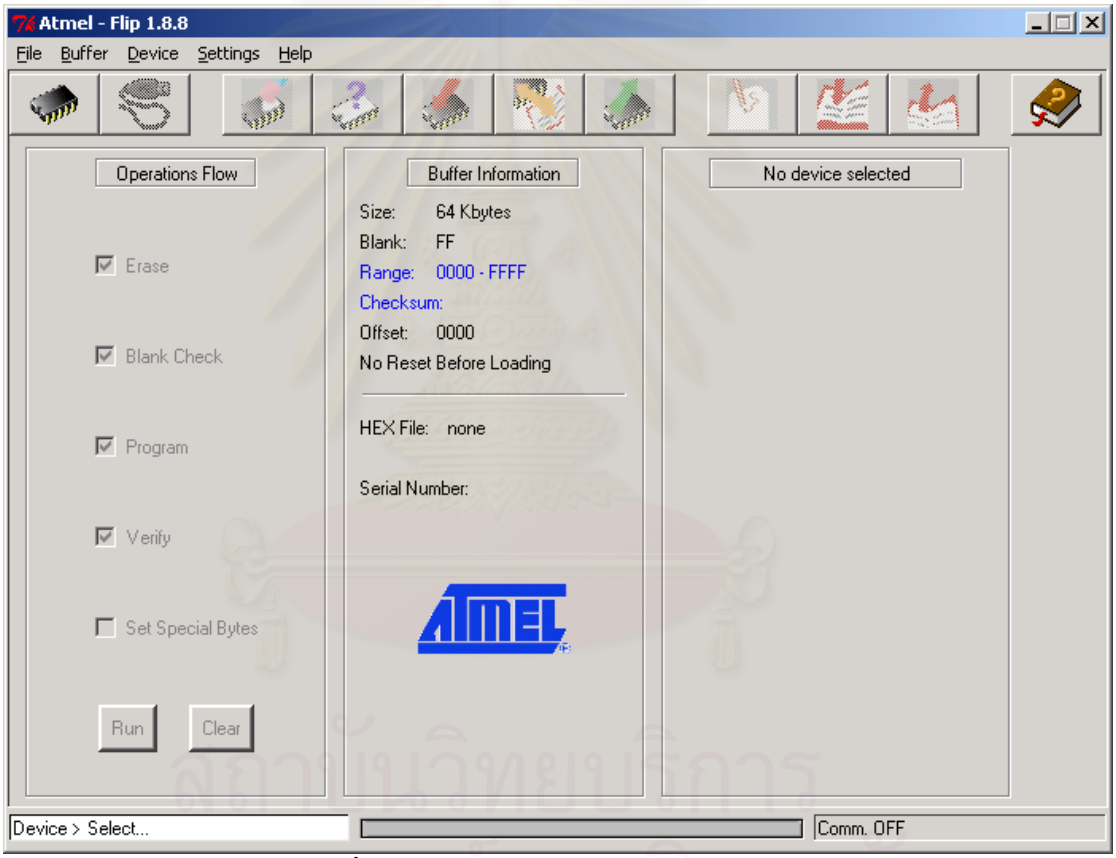
ในการพัฒนาเครื่องโทรศัพท์อินเทอร์เนตนี้ได้ใช้ EEPROM เก็บพารามิเตอร์ต่าง ๆ ที่สำคัญต่อการปรับตั้งเครื่องโทรศัพท์ได้แก่ ไอพีแอดเดรสเครื่องโทรศัพท์, ซับเนตมาสค์เครื่องโทรศัพท์, ไอพีแอดเดรสเกตเวย์, ไอพีแอดเดรสเครื่องแม่ข่าย, SIP URL, และโมดการทำงานของเครื่องโทรศัพท์ โดยมีแอดเดรสของข้อมูลดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 การใช้ EEPROM เก็บข้อมูลในเครื่องโทรศัพท์อินเทอร์เนต

ข้อมูล	ขนาด (ไบต์)	ตำแหน่ง
ไอพีแอดเดรสของเครื่องโทรศัพท์	4	0-3
ซับเนตมาสค์เครื่องโทรศัพท์	4	4-7
ไอพีแอดเดรสเกตเวย์	4	8-11
ไอพีแอดเดรสเซอร์เวอร์	4	12-15
SIP URL	30	16-45
โมดการทำงานของเครื่องโทรศัพท์	1	46

2.1.6 การโหลดโปรแกรมลงในหน่วยความจำแบบ Flash ของ AT89C51ED2 โดยใช้โปรแกรม FLIP

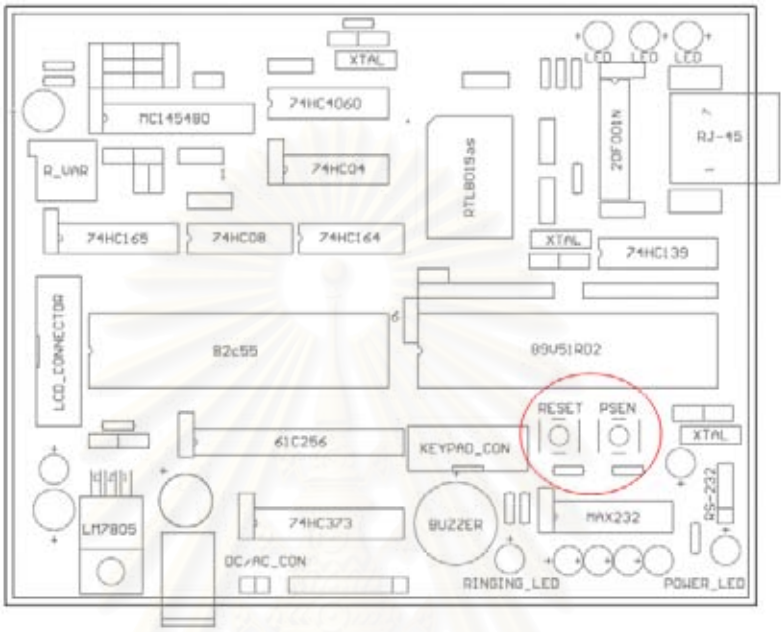
โปรแกรม FLIP (Flexible In-system Programmer) เป็นโปรแกรมของบริษัท ATMELใช้สำหรับโหลดโปรแกรมลงไมโครคอนโทรลเลอร์ สามารถใช้กับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท ATMEL ที่รองรับการดาวโหลดข้อมูลแบบ ISP (In-System Programming) รวมถึงไมโครคอนโทรลเลอร์เบอร์ AT89C51ED2 ที่ใช้ในการพัฒนาเครื่องโทรศัพท์อินเทอร์เนตในงานวิจัยนี้ด้วย FLIP สามารถทำงานภายใต้ระบบปฏิบัติการ Windows9x/ME/NT/2000/XP โดยสนับสนุนการเชื่อมต่อแบบ RS-232, CAN, และ USB แต่สำหรับไมโครคอนโทรลเลอร์เบอร์ AT89C51ED2 นั้นสามารถเชื่อมต่อผ่านทางพอร์ตอนุกรม RS-232 ได้เท่านั้น



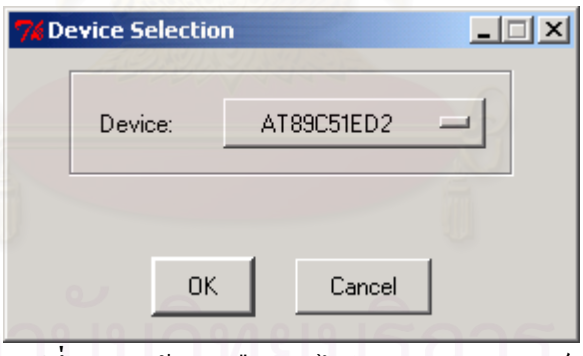
รูปที่ 2.14 โปรแกรม FLIP ของบริษัท Atmel

เมื่อเริ่มเปิดโปรแกรม FLIP ขึ้นมาโปรแกรมจะมีหน้าตาดังแสดงในรูปที่ 2.14 การใช้โปรแกรม FLIP จัดการกับหน่วยความจำแบบ Flash บนไมโครคอนโทรลเลอร์ AT89C51ED2 นั้นทำได้ในขณะที่ไมโครคอนโทรลเลอร์อยู่ในโมดมอนิเตอร์เท่านั้น การทำให้ไมโครคอนโทรลเลอร์เข้าสู่โมดมอนิเตอร์ทำได้โดยรีเซตไมโครคอนโทรลเลอร์ในขณะที่สัญญาณที่ขา PSEN ของไมโครคอนโทรลเลอร์มีสถานะลอจิกเป็น ‘0’ ดังนั้นในการทำให้ไมโครคอนโทรลเลอร์บนเครื่องโทรศัพท์ดังแสดงในรูปที่ 2.15 เข้าสู่โมดมอนิเตอร์ต้องกดสวิตซ์ PSEN ค้างไว้ แล้วกดสวิตซ์

RESET และปล่อยในขณะที่ยังคงกดปุ่ม PSEN ค้างไว้อยู่ เมื่อปล่อยสวิตซ์ RESET แล้วจึงค่อยปล่อยสวิตซ์ PSEN ตามลำดับ เมื่อไมโครคอนโทรลเลอร์อยู่ในโมดมอนิตอร์แล้ว จึงสามารถใช้โปรแกรม FLIP จัดการกับหน่วยความจำโปรแกรมแบบ Flash บนไมโครคอนโทรลเลอร์ได้ตามต้องการ ไม่ว่าจะเป็นการลบข้อมูลในหน่วยความจำ หรือดาวโหลดข้อมูลลงในหน่วยความจำ



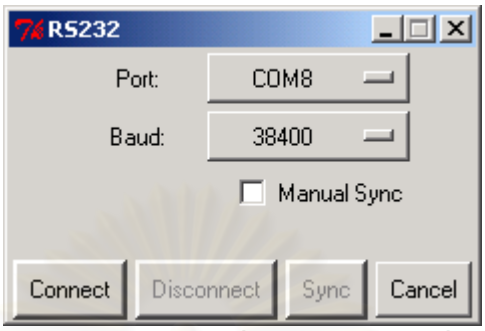
รูปที่ 2.15 ตำแหน่งปุ่ม RESET และ PSEN บนเครื่องโทรศัพท์อินเทอร์เนต



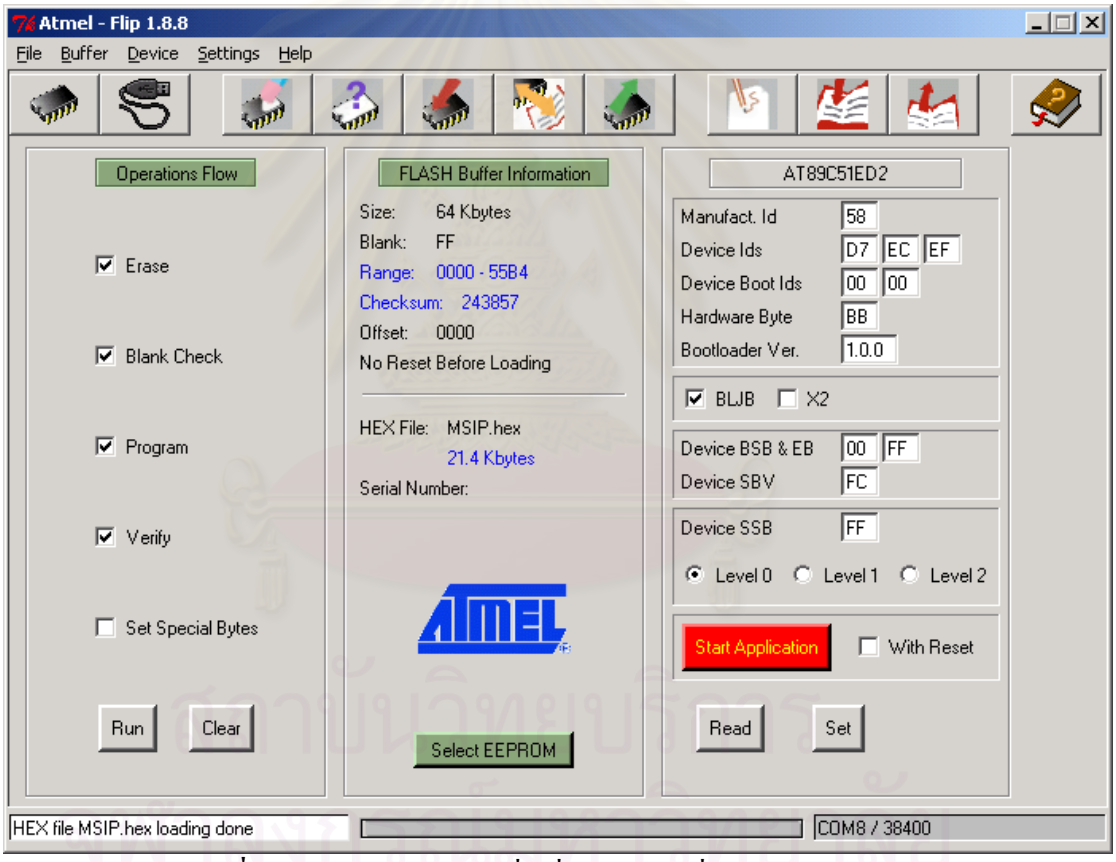
รูปที่ 2.16 หน้าต่างเลือกรุ่นไมโครคอนโทรลเลอร์

ในการเริ่มใช้งานโปรแกรม FLIP เริ่มแรกต้องเลือกรุ่นของไมโครคอนโทรลเลอร์ที่ใช้ก่อน โดยเลือกเมนู Device -> Select จะมีหน้าต่างเลือกรุ่นไมโครคอนโทรลเลอร์ขึ้นมาดังแสดงในรูปที่ 2.16 เปลี่ยนให้เป็นรุ่น AT89C51ED2 แล้วกด OK หน้าต่างของโปรแกรม FLIP จะขึ้นรายละเอียดของไมโครคอนโทรลเลอร์ AT89C51ED2 ให้ผู้ใช้งานเห็น จากนั้นจึงค่อยเปิดพอร์ตสื่อสารแบบอนุกรม RS-232 โดยเลือกเมนู Setting -> Communication Port -> RS-232 จะมีหน้าต่างควบคุมการสื่อสารแบบอนุกรม RS-232 ขึ้นมาดังแสดงในรูปที่ 2.17 เลือกพอร์ตสื่อสารแบบอนุกรมให้ตรงกับที่ต่อกับเครื่องโทรศัพท์ และเลือกอัตรา Baudrate ตามที่ต้องการ จากนั้นจึงกดปุ่ม

Connect เพื่อเปิดพอร์ตและเริ่มการเชื่อมต่อระหว่างโปรแกรม FLIP กับเครื่องโทรศัพท์ เมื่อโปรแกรม FLIP เชื่อมต่อกับเครื่องโทรศัพท์ได้แล้วจะมีหน้าตาดังแสดงในรูปที่ 2.18 ซึ่งจะเห็นได้ว่าผู้ใช้งานสามารถเข้าไปจัดการหน่วยความจำแบบ Flash ภายในไมโครคอนโทรลเลอร์ได้แล้ว



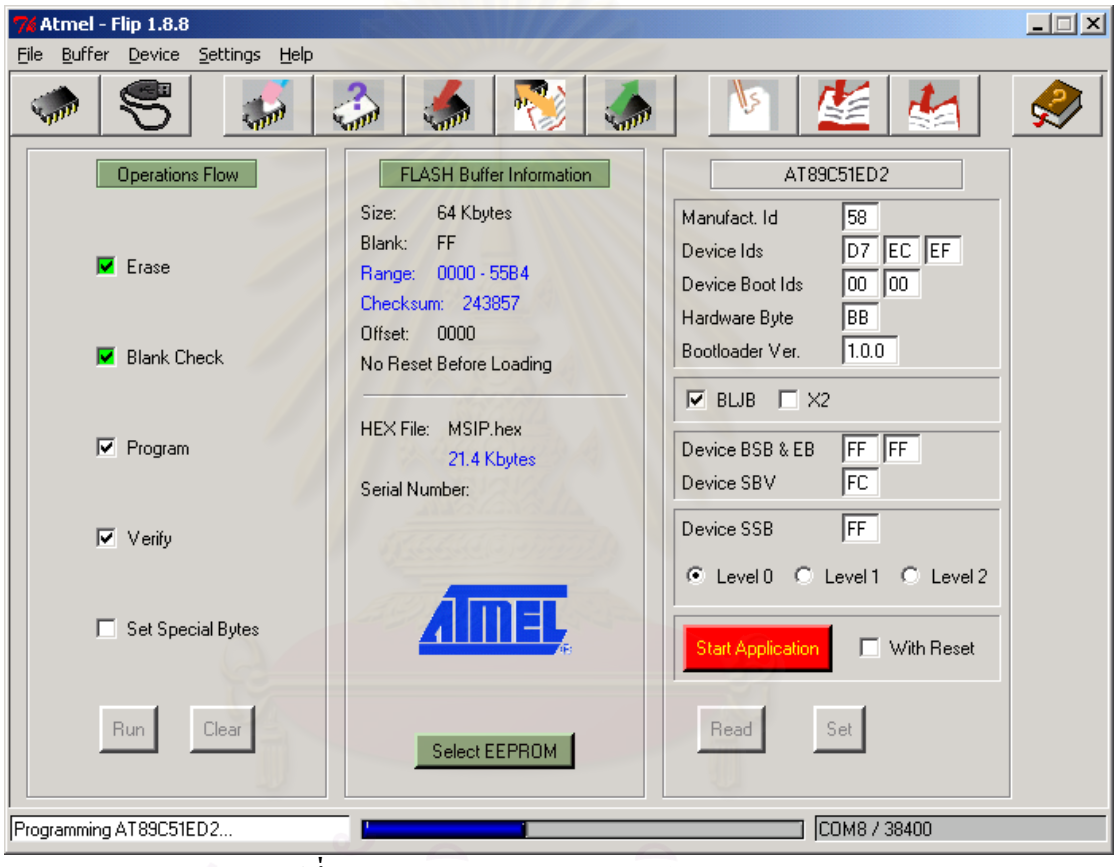
รูปที่ 2.17 หน้าต่างควบคุมการเชื่อมต่อผ่านพอร์ตสื่อสารอนุกรม



รูปที่ 2.18 โปรแกรม FLIP เมื่อเชื่อมต่อกับเครื่องโทรศัพท์ได้แล้ว

ในการโหลดไฟล์ HEX ลงในหน่วยความจำแบบ Flash ของไมโครคอนโทรลเลอร์ เริ่มแรกต้องโหลดไฟล์ HEX มาไว้ในบัฟเฟอร์ของโปรแกรม FLIP ก่อน โดยเลือกเมนู File -> Load HEX และเลือกไฟล์ HEX ที่ต้องการ เมื่อโหลดไฟล์ HEX ที่ต้องการลงในบัฟเฟอร์ของโปรแกรม FLIP เรียบร้อยแล้ว สามารถโหลดไฟล์ HEX นั้นลงในหน่วยความจำแบบ Flash ของไมโครคอนโทรลเลอร์ได้โดยเลือกการกระทำ Erase, Blank Check, Program, และ Verify ใน

Operation Flow เพื่อปรับตั้งให้โปรแกรม FLIP ลบข้อมูลในหน่วยความจำแบบ Flash ภายในไมโครคอนโทรลเลอร์, ตรวจสอบหน่วยความจำ Flash ว่าข้อมูลถูกลบหมดเรียบร้อย, โหลดข้อมูลไฟล์ HEX ลงในหน่วยความจำ Flash, และตรวจสอบความถูกต้องของข้อมูลที่ได้โหลดลงไปตามลำดับ จากนั้นจึงค่อยกดปุ่ม RUN เพื่อให้โปรแกรม FLIP เริ่มทำงานตามกระบวนการที่ได้เลือกไว้ ในขณะที่โปรแกรม FLIP กำลังทำงานนั้น โปรแกรม FLIP จะแสดงสถานะการทำงานตั้งแต่ต้นจนโหลดโปรแกรม และตรวจสอบข้อมูลภายในหน่วยความจำ Flash จนเสร็จเรียบร้อยดังแสดงในรูปที่ 2.19



รูปที่ 2.19 การแสดงการทำงานของโปรแกรม FLIP

เมื่อโหลดไฟล์ HEX ลงในหน่วยความจำแบบ Flash ของไมโครคอนโทรลเลอร์เรียบร้อยแล้ว ให้ปิดพอร์ตสื่อสารอนุกรม โดยกดปุ่ม Disconnect ที่หน้าต่างควบคุมการสื่อสารแบบอนุกรมของโปรแกรม FLIP และกดปุ่ม RESET ที่เครื่องโทรศัพท์เพื่อให้เครื่องโทรศัพท์กลับเข้าสู่โมดการทำงานปกติ และพร้อมใช้งานต่อไป

2.2 การเลือกอุปกรณ์สำหรับการพัฒนาเครื่องโทรศัพท์อินเทอร์เนตต้นแบบ

โดยทั่วไปเครื่องโทรศัพท์อินเทอร์เนตจะประกอบไปด้วยส่วนประมวลผล, หน่วยความจำ, ส่วนเข้ารหัสเสียง, ส่วนติดต่อกับอีเทอร์เนต, และส่วนติดต่อกับผู้ใช้งาน ซึ่งในการพัฒนาเครื่องโทรศัพท์อินเทอร์เนตต้นแบบในวิทยานิพนธ์นี้ได้ใช้เกณฑ์การเลือกอุปกรณ์ต่างๆ จากทั้งใน

เรื่องราคา, ความสามารถในการทำงาน, ความง่ายในการหาซื้อ และความสะดวกต่อการนำมาพัฒนา

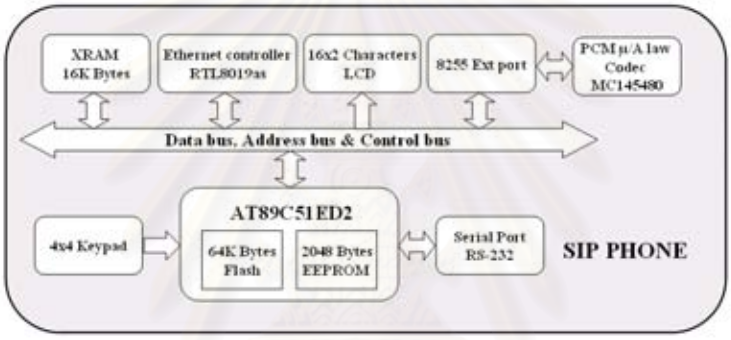
ในส่วนประมวลผลนั้นต้องสามารถติดต่อกับอุปกรณ์อื่นได้โดยง่าย, ง่ายต่อการพัฒนาทางซอฟท์แวร์, และสามารถทำงานได้อย่างรวดเร็วเนื่องจากเครื่องโทรศัพท์ต้องรับส่งข้อมูลเสียงแบบเวลาจริง ในการพัฒนาจึงได้เลือกใช้ชิพ AT89C51ED2 ซึ่งเป็นไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัท ATMEL สามารถทำงานได้เร็วกว่า MCS-51 พื้นฐาน 2 เท่า ภายในมีหน่วยความจำโปรแกรมแบบแฟลชขนาด 64 กิโลไบต์ และสามารถโหลดโปรแกรมแบบ ISP (In-System Programming) ลงตัวชิพโดยผ่านพอร์ตอนุกรมได้โดยตรง ทำให้การพัฒนาสามารถทำได้สะดวก นอกจากนั้นยังมี EEPROM ขนาด 2,048 ไบต์ และหน่วยความจำข้อมูลภายนอกเพิ่มเติมขนาด 1,792 ไบต์ แต่เนื่องจากการทำงานของเครื่องโทรศัพท์อินเทอร์เนตนั้นจำเป็นต้องใช้หน่วยความจำข้อมูลมากกว่า 1,792 ไบต์ ดังนั้นจึงต้องใช้ชิพหน่วยความจำข้อมูลภายนอกเพิ่มขนาด 32 กิโลไบต์ ในส่วนอุปกรณ์เข้ารหัสเสียงได้ใช้ชิพ MC145480 ของบริษัท MOTOROLA เนื่องจากสามารถหาซื้อได้ภายในประเทศ และรองรับการเข้ารหัสเสียง PCM ทั้ง μ-law และ A-law ซึ่งเป็นการเข้ารหัสเสียงพื้นฐานที่เครื่องโทรศัพท์อินเทอร์เนตต้องสามารถรองรับได้ แต่เนื่องจากชิพ MC145480 นั้นรับส่งข้อมูล PCM เป็นแบบอนุกรม ดังนั้นเพื่อให้ง่ายต่อการรับส่งข้อมูลกับไมโครคอนโทรลเลอร์จึงได้ใช้ชิพ S/P, P/S และชิพ 82c55 แปลงข้อมูลให้เป็นแบบขนาน และสามารถทำ Memory Map เพื่อรับส่งข้อมูลกับไมโครคอนโทรลเลอร์ได้โดยง่าย ซึ่งจะอธิบายรายละเอียดในบทที่ 3 ต่อไป ในส่วนติดต่อกับอีเทอร์เนตนั้นได้ใช้ชิพควบคุมอีเทอร์เนต RTL8019as ของบริษัท REALTEK เนื่องจากชิพ RTL8019as มีรีจิสเตอร์ควบคุมตามมาตรฐาน NE2000 และสามารถรับส่งติดต่อกับไมโครคอนโทรลเลอร์ MCS-51 ผ่านระบบบัสข้อมูลขนาด 8 บิตได้โดยตรง ในส่วนติดต่อกับผู้ใช้งานนั้น ส่วนรับข้อมูลได้ใช้คีย์แพด 4x4 เนื่องจากเครื่องโทรศัพท์ต้องรับข้อมูลตัวเลข, ตัวอักขระ และการควบคุมเซสชันจากผู้ใช้งาน ในส่วนแสดงผลข้อมูลได้ใช้ LCD 16x2 ซึ่งสามารถแสดงได้ทั้งตัวอักขระ และตัวเลขเพื่อแสดงผลการทำงานให้กับผู้ใช้งานได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3
การออกแบบส่วนประมวลผลหลักสำหรับระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VOIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน

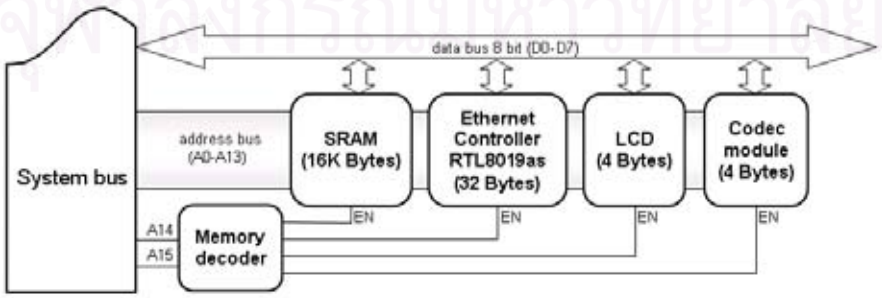
3.1 โครงสร้างเครื่องโทรศัพท์อินเทอร์เนตต้นแบบ

เครื่องโทรศัพท์อินเทอร์เนตต้นแบบประกอบไปด้วย AT89C51ED2 เป็นส่วนประมวลผลหลัก, หน่วยความจำภายนอก, ส่วนควบคุมอีเทอร์เนต, ส่วนเข้ารหัสเสียง และส่วนติดต่อกับผู้ใช้งานดังแสดงในรูปที่ 3.1



รูปที่ 3.1 โครงสร้างโทรศัพท์อินเทอร์เนตที่ได้พัฒนาขึ้น

ในการติดต่อสื่อสารข้อมูลระหว่างไมโครคอนโทรลเลอร์กับ ส่วนควบคุมอีเทอร์เนต, ส่วนแสดงผล LCD, และส่วนเข้ารหัสสัญญาณเสียงนั้นได้ต่อแบบ Memory Map โดยใช้สายสัญญาณข้อมูล, สายสัญญาณกำหนดตำแหน่งหน่วยความจำ, และสายสัญญาณควบคุม เช่นเดียวกับการติดต่อกับหน่วยความจำภายนอกตามที่ได้อธิบายข้างต้น และใช้ชิพเข้ารหัส 74HC135 เข้ารหัสตำแหน่งหน่วยความจำบิต A14, A15 เพื่อแยกตำแหน่งของอุปกรณ์แต่ละส่วนออกจากกันดังแสดงในรูปที่ 3.2 เมื่อทำ Memory Map แล้วอุปกรณ์แต่ละส่วนจะมีตำแหน่งดังแสดงในตารางที่ 3.1



รูปที่ 3.2 การทำ Memory Map ภายในเครื่องโทรศัพท์อินเทอร์เนต

ตารางที่ 3.1 ตำแหน่งการทำ Memory Map ภายในเครื่องโทรศัพท์อินเทอร์เนต

อุปกรณ์ภายนอก	จำนวนสายสัญญาณกำหนดตำแหน่งหน่วยความจำ (บิต)	ขนาด (ไบต์)	ตำแหน่ง
หน่วยความจำภายนอก SRAM	14	16,384	0x0000-0x3FFF
ส่วนควบคุมอีเทอร์เนต RTL8019as	5	32	0x4000-0x4031
ส่วนแสดงผล LCD	2	4	0x8000-0x8003
ส่วนเข้ารหัสเสียง PCM	2	4	0xC000-0xC003

ในการพัฒนาแบ่งออกได้เป็น 4 ส่วนหลัก ๆ คือ ส่วนติดต่อกับผู้ใช้งาน, ส่วนควบคุมการรับส่งข้อมูลผ่านชิพอีเทอร์เนต, ส่วนรองรับโพรโทคอล และส่วนควบคุมข้อมูลเสียง

3.2 ส่วนติดต่อรับส่งข้อมูลผ่านชิพอีเทอร์เนต RTL8019as

3.2.1 โครงสร้าง และคุณลักษณะของชิพอีเทอร์เนต RTL8019as

ชิพอีเทอร์เนต RTL8019as เป็นชิพที่รองรับการรับส่งข้อมูลตามมาตรฐาน 10BaseT มีรีจิสเตอร์ควบคุมการทำงานตามมาตรฐาน NE2000 ซึ่งเป็นมาตรฐานสำหรับอุตสาหกรรมการออกแบบชิพอีเทอร์เนต ดังแสดงในตารางที่ 3.2 ตัวชิพประกอบไปด้วยรีจิสเตอร์ควบคุม 4 หน้า, หน่วยความจำ SRAM สำหรับบัฟเฟอร์ข้อมูลขนาด 16K ไบต์ ฟังก์ชันการคำนวณ CRC (Cyclic Redundancy Check), วงจรเข้ารหัสข้อมูลแบบแมนเชสเตอร์ (Manchester Coding), และวงจรรับส่งข้อมูลผ่านอีเทอร์เนต

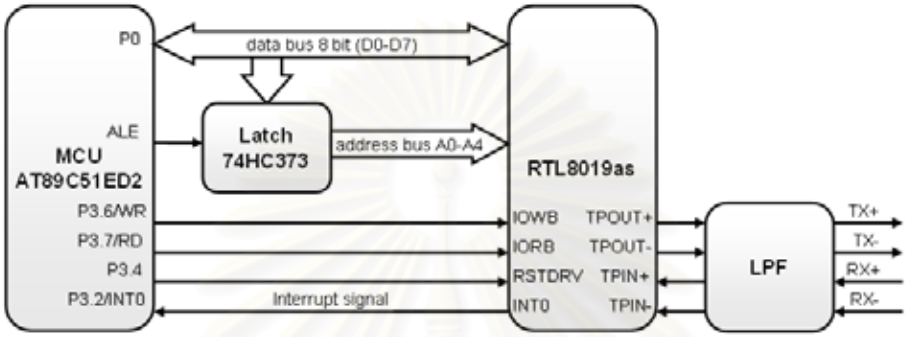
ตารางที่ 3.2 โครงสร้างรีจิสเตอร์ควบคุมตามมาตรฐาน NE2000 ภายในชิพ RTL8019as

No (Hex)	Page 1		Page 2	Page 3	Page 4	
	[R]	[W]	[R/W]	[R]	[R]	[W]
00	CR	CR	CR	CR	CR	CR
01	CLDA0	PSTART	PAR0	PSTART	***9346CR***	***9346CR***
02	CLDA1	PSTOP	PAR1	PSTOP	***BPAGE***	***BPAGE1***
03	BNRY	BNRY	PAR2	-	***CONFIG0***	-
04	TSR	TPSR	PAR3	TPSR	***CONFIG1***	***CONFIG1***
05	NCR	TBSR0	PAR4	-	***CONFIG2***	***CONFIG2***
06	FIFO	TBCR1	PAR5	-	***CONFIG3***	***CONFIG3***
07	ISR	ISR	CURR	-	-	***TEST***
08	CRDA0	RSAR0	MAR0	-	***CSNSAV***	-
09	CRDA1	RSAR1	MAR1	-	-	***HLTCLK***
0A	***8019ID0***	RBCR0	MAR2	-	-	-
0B	***8019ID1***	RBCR1	MAR3	-	***INTR***	-
0C	RSR	RCR	MAR4	RCR	-	***FMWP***
0D	CNTR0	TCR	MAR5	TCR	***CONFIG4***	-
0E	CNTR1	DCR	MAR6	DCR	-	-
0F	CNTR2	IMR	MAR7	IMR	-	-
10-17	Remote DMA Port					
18-1F	Reset Port					

- รีจิสเตอร์ที่เป็นตัวเอียงทึบ เป็นรีจิสเตอร์ควบคุมเพิ่มเติมของชิพ RTL8019as นอกเหนือจากมาตรฐาน NE2000

ในการพัฒนา ต่อไมโครคอนโทรลเลอร์กับชิพ RTL8019as แบบ Memory Map โดยให้ชิพ RTL8019as เป็นหน่วยความจำอยู่ในตำแหน่ง 0x4000 ถึง 0x4031 ซึ่งในการต่อนั้นนอกจากจะใช้

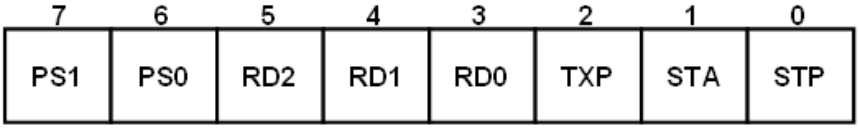
สายสัญญาณข้อมูล D0-D7, สายสัญญาณกำหนดตำแหน่งหน่วยความจำ A0-A4, และสายสัญญาณควบคุม WR, RD แล้ว ยังต่อพอร์ต 3.4 เข้ากับขา RSTDRV เพื่อควบคุมการรีเซตการทำงานของชิพ RTL8019as และต่อพอร์ต 3.2 เข้ากับขา INT0 เพื่อรับสัญญาณอินเตอร์รัปต์จากชิพ RTL8019as ด้วยดังแสดงในรูปที่ 3.3



รูปที่ 3.4 การต่อไมโครคอนโทรลเลอร์ กับชิพอีเทอร์เนต RTL8019as

ในการควบคุมชิพ RTL8019as เพื่อให้รับส่งข้อมูลนั้นสามารถทำได้โดยการอ่าน และเขียนข้อมูลลงในรีจิสเตอร์ควบคุมภายในชิพ RTL8019as ซึ่งมี 4 หน้า และมีรีจิสเตอร์ที่สำคัญดังนี้

รีจิสเตอร์ CR (Command Register) อยู่ที่ตำแหน่ง 0x01 ของทุกหน้า สามารถกระทำการอ่านและเขียนได้เป็นรีจิสเตอร์ที่ใช้ในการเลือกหน้ารีจิสเตอร์ และเป็นรีจิสเตอร์ที่ใช้ควบคุมพอร์ต DMA ซึ่งเป็นพอร์ตที่ใช้ในการรับส่งข้อมูลกับอุปกรณ์ภายนอก มีโครงสร้างดังแสดงในรูปที่ 3.5 บิต PS1, PS0 เป็นบิตที่ใช้ในการเลือกหน้ารีจิสเตอร์ โดยถ้า PS1, PS0 = “00” เป็นการใช้งานหน้า 0, “01” เป็นการใช้งานหน้า 1, “10” เป็นการใช้งานหน้า 2 และ “11” เป็นการใช้งานหน้า 3 ตามลำดับ RD2-0 เป็นบิตที่ใช้ในการควบคุมการรับส่งข้อมูลผ่านพอร์ต DMA โดยถ้า RD2-0 = “001” เป็นการควบคุมให้พอร์ต DMA ทำงานในโมดส่งข้อมูล, “010” ทำงานในโมดรับข้อมูล, “011” เป็นการควบคุมให้ชิพอีเทอร์เนตทำงานส่งแพ็กเกตออกไป และ “1**” เป็นตัวแสดงว่าการรับส่งข้อมูลผ่านพอร์ต DMA นั้นได้เสร็จสิ้นแล้ว บิต TXP เป็นบิตควบคุมการส่งข้อมูลออก โดยก่อนส่งต้องปรับให้บิต TXP นี้เป็น ‘1’ ก่อน เมื่อข้อมูลได้ส่งออกไปสำเร็จแล้วบิต TXP นี้จะเป็น ‘0’ โดยอัตโนมัติ ส่วนบิต STA, STP ใช้ควบคุมการเปิดปิดการรับส่งข้อมูลโดยถ้า STA, STP = “10” ชิพจะสามารถรับส่งข้อมูลได้ แต่ถ้า STA, STP = “01” ชิพจะไม่สามารถรับส่งข้อมูลได้



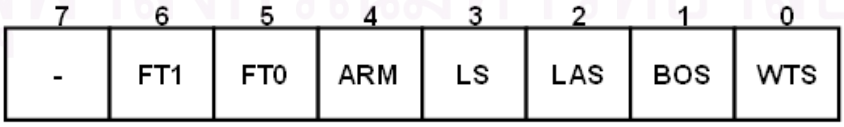
รูปที่ 3.5โครงสร้างรีจิสเตอร์ CR

รีจิสเตอร์ ISR (Interrupt Status Register) อยู่ที่ตำแหน่ง 0x07 ของหน้า 0 สามารถกระทำการอ่าน และเขียนได้ ใช้ในการบ่งบอกสาเหตุการเกิดสัญญาณอินเตอร์รัปต์ภายในชิพอีเทอร์เนต มีโครงสร้างดังแสดงในรูปที่ 3.6 บิต RST จะเป็น ‘1’ เมื่อชิพถูกรีเซต สามารถเคลียร์เองได้เมื่อมีการเขียนค่ารีจิสเตอร์ CR ให้อยู่ในโมดสามารถรับส่งข้อมูลได้ (STA, STP = “10”) บิต RDC จะเป็น ‘1’ เมื่อมีการรับส่งค่าผ่านพอร์ต DMA สำเร็จ บิต OVW จะเป็น ‘1’ เมื่อบัฟเฟอร์ภาครับล้น บิต TXE เป็น ‘1’ เมื่อส่งข้อมูลแล้วเกิดการชนขึ้น บิต RXE เป็น ‘1’ เมื่อข้อมูลที่ได้รับมามีความผิดพลาด เช่นตรวจสอบ CRC ไม่ผ่าน หรือซิงโครไนซ์ข้อมูลไม่ได้ เป็นต้น PTX เป็น ‘1’ เมื่อชิพอีเทอร์เนตได้ส่งข้อมูลออกไปโดยไม่มีความผิดพลาดเกิดขึ้น และ PRX เป็น ‘1’ เมื่อชิพอีเทอร์เนตได้รับข้อมูลซึ่งไม่มีความผิดพลาดเข้ามา

รูปที่ 3.6โครงสร้างรีจิสเตอร์ ISR

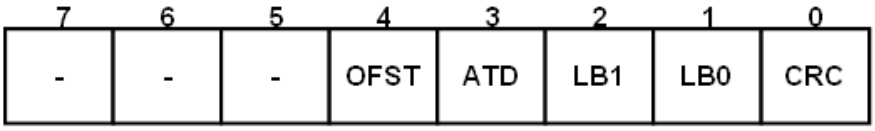
รีจิสเตอร์ IMR (Interrupt Mask Register) อยู่ที่ตำแหน่ง 0xFH ในการเขียนอยู่ในหน้า 0 ส่วนในการอ่านอยู่ในหน้า 2 ใช้ในการเลือกใช้งานอินเตอร์รัปต์ โดยมีตำแหน่งสอดคล้องกับรีจิสเตอร์ ISR ซึ่งในการเลือกใช้งานอินเตอร์รัปต์นั้น ๆ สามารถทำได้โดยการเซตบิตตำแหน่งนั้นให้เป็น ‘1’

รีจิสเตอร์ DCR (Data Configuration Register) อยู่ที่ตำแหน่ง 0x0E ในการเขียนอยู่ในหน้า 0 ส่วนในการอ่านอยู่ในหน้า 2 ใช้สำหรับระบุโมดการทำงานของชิพอีเทอร์เนต มีโครงสร้างดังแสดงในรูปที่ 3.7



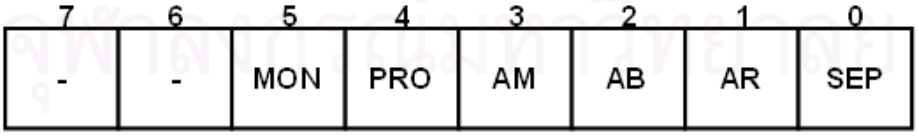
รูปที่ 3.7โครงสร้างรีจิสเตอร์ DCR

รีจิสเตอร์ TCR (Transmit Configuration Register) อยู่ที่ตำแหน่ง 0x0D ในการเขียนอยู่ในหน้า 0 ส่วนการอ่านอยู่ในหน้า 2 มีโครงสร้างดังแสดงในรูปที่ 3.8 ใช้กำหนดโมดการส่งข้อมูล



รูปที่ 3.8 โครงสร้างรีจิสเตอร์ TCR

รีจิสเตอร์ RCR (Receive Configuration Register) อยู่ที่ตำแหน่ง 0x0C ในการเขียนอยู่ในหน้า 0 ส่วนการอ่านอยู่ในหน้า 2 มีโครงสร้างดังแสดงในรูปที่ 3.9 ใช้สำหรับระบุโมดการรับข้อมูล โดยถ้าบิต MON เป็น '1' ชิพอีเทอร์เนตจะทำงานในโมดมอนิเตอร์ โดยจะตรวจเช็ค ฮาร์ดแวร์แอดเดรส ปลายทาง, CRC, การซิงโครไนซ์เฟรม แต่ไม่เก็บเฟรมที่ได้รับมานั้นเข้ามาในบัฟเฟอร์ แต่ถ้าเป็น '0' ชิพอีเทอร์เนตจะเก็บเฟรมนั้นไว้ในบัฟเฟอร์เพื่อรอการเรียกใช้งานต่อไป บิต PRO เป็นบิตที่ใช้ตั้งการเลือกรับข้อมูลโดยถ้าเป็น '1' ชิพอีเทอร์เนตจะรับเฟรมที่เข้ามาหมดโดยไม่สนใจฮาร์ดแวร์แอดเดรสปลายทาง แต่ถ้าเป็น '0' ชิพอีเทอร์เนตจะเลือกรับเฉพาะเฟรมที่มีฮาร์ดแวร์แอดเดรสตรงกับตัวเองเท่านั้น บิต AM เป็นบิตที่ใช้ตั้งการเลือกรับเฟรมแบบมัลติคาสต์ (multicast)โดยถ้า AM เป็น '1' ชิพอีเทอร์เนตจะรับเฟรมที่มีฮาร์ดแวร์แอดเดรสมัลติคาสต์ แต่ถ้าเป็น '0' ชิพอีเทอร์เนตจะไม่รับเฟรมที่มีฮาร์ดแวร์แอดเดรสแบบมัลติคาสต์ บิต AB เป็นบิตที่ใช้ตั้งการเลือกรับข้อมูลแบบแพร่สัญญาณโดยถ้า AB เป็น '1' ชิพอีเทอร์เนตจะรับเฟรมที่มีฮาร์ดแวร์แอดเดรสแบบมัลติคาส์ท แต่ถ้าเป็น '0' ชิพอีเทอร์เนตจะไม่รับเฟรมที่มีฮาร์ดแวร์แอดเดรสแบบแพร่สัญญาณ บิต AR เป็นบิตที่ใช้ตั้งการเลือกรับตามขนาดของเฟรมโดยถ้า AR เป็น '1' ชิพอีเทอร์เนตจะรับเฟรมที่มีขนาดเล็กกว่า 64 ไบต์ แต่ถ้าเป็น '0' ชิพอีเทอร์เนตจะไม่รับเฟรมที่มีขนาดเล็กกว่า 64 ไบต์ และบิต SEP เป็นบิตที่ใช้ตั้งการเลือกรับแพ็กเกตที่มีความผิดพลาดโดยถ้า SEP เป็น '1' ชิพอีเทอร์เนตจะรับเฟรมที่มีความผิดพลาด แต่ถ้าเป็น '0' ชิพอีเทอร์เนตจะไม่รับเฟรมที่มีความผิดพลาด



รูปที่ 3.9 โครงสร้างรีจิสเตอร์ RCR

รีจิสเตอร์ PAR0-5 (Physical Address Register) อยู่ที่ตำแหน่งที่ 0x01 ถึง 0x06 ของหน้าที่ 1 สามารถกระทำได้ทั้งการอ่าน และเขียน ใช้เก็บฮาร์ดแวร์แอดเดรสของโฮสต์

รีจิสเตอร์ PSTART (Page Start Register) อยู่ที่ตำแหน่ง 0x01 ในการเขียนอยู่ในหน้า 0 ส่วนในการอ่านอยู่ในหน้า 2 เป็นรีจิสเตอร์ที่ใช้กำหนดตำแหน่งจุดเริ่มต้นของวงบัฟเฟอร์ (Buffer Ring) ข้อมูลภาครับ

รีจิสเตอร์ PSTOP (Page Stop Register) อยู่ที่ตำแหน่ง 0x02 ในการเขียนอยู่ในหน้า 0 ส่วนในการอ่านอยู่ในหน้า 2 ใช้กำหนดตำแหน่งจุดสิ้นสุดของวงบัฟเฟอร์ข้อมูลภาครับ

รีจิสเตอร์ BNRY (Boundary Register) อยู่ที่ตำแหน่ง 0x03 ของหน้า 0 เป็นรีจิสเตอร์ที่ใช้ป้องกันการซ้อนทับของข้อมูล โดยจะเป็นตัวชี้ตำแหน่งหน้าของบัฟเฟอร์ข้อมูลที่ได้อ่านออกไป

รีจิสเตอร์ TPSR (Transmit Page Start Register) อยู่ที่ตำแหน่ง 0x03 ในการเขียนอยู่ในหน้า 0 ส่วนในการอ่านอยู่ในหน้า 2

รีจิสเตอร์ CURR (Current Page Register) อยู่ที่ตำแหน่ง 0x07 ของหน้าที่ 1 เป็นรีจิสเตอร์ชี้ตำแหน่งหน้าของบัฟเฟอร์ข้อมูลที่ได้รับเข้ามา

รีจิสเตอร์ TBCR0, 1 (Transmit Byte Count Register) อยู่ที่ตำแหน่ง 0x05 และ 0x06 ของหน้าที่ 0 ใช้กำหนดขนาดข้อมูลที่จะส่งออกไป

รีจิสเตอร์ RBCR0, 1 (Remote Byte Count Register) อยู่ที่ตำแหน่ง 0x0A และ 0x0B ของหน้าที่ 0 ใช้กำหนดขนาดข้อมูลที่จะอ่าน หรือเขียนผ่านพอร์ต DMA

บัฟเฟอร์ภายในชิพ RTL8019as นั้นมีโครงสร้างเป็นแบบวงแบ่งออกเป็นหน้า ๆ แต่ละหน้ามีขนาด 256 ไบต์ ในกรณีที่เฟรมมีขนาดเล็กกว่า 256 ไบต์ หรือเป็นเศษไม่เต็ม ชิพอีเทอร์เนตจะเพิ่มข้อมูลแพดจนเต็ม 256 ไบต์และเก็บข้อมูลเฟรมใหม่ไว้ในหน้าต่อไป ในการใช้งานสามารถเข้าถึงได้โดยการเขียน และอ่านข้อมูลผ่านพอร์ต DMA ที่ตำแหน่ง 10H-17H ของตัวชิพ โดยการทำงานในภาครับนั้นถูกควบคุมด้วยรีจิสเตอร์ PSTART, PSTOP, BNRY, และ CURR ตามลำดับ

3.2.2 การเริ่มต้นใช้งานชิพอีเทอร์เนต RTL8019as

ในการใช้งานชิพอีเทอร์เนตดังแสดงในรูปที่ 3.10 เริ่มต้นต้องรีเซต และตั้งค่ารีจิสเตอร์ควบคุมต่าง ๆ ภายในตัวชิพอีเทอร์เนตเพื่อกำหนดรูปแบบการทำงานเสียก่อน การรีเซตตัวชิพอีเทอร์เนตควรรีเซตทั้งในระดับฮาร์ดแวร์ และซอฟต์แวร์ โดยการรีเซตในระดับฮาร์ดแวร์สามารถทำได้โดยป้อนพัลส์สัญญาณที่มีความกว้างมากกว่า 800 ns เข้าที่ขา RSTDRV ส่วนการรีเซตในระดับซอฟต์แวร์สามารถทำได้โดยการเขียนค่าที่อ่านได้จากพอร์ตรีเซตนั้นกลับไปที่พอร์ตรีเซตของตัวชิพเอง เมื่อชิพเข้าไปอยู่ในสถานะรีเซตทางซอฟต์แวร์ ไมโครคอนโทรลเลอร์จะตรวจสอบบิต RST ภายในรีจิสเตอร์ ISR โดยถ้ามีค่าเป็น ‘1’ แสดงว่าชิพเข้าสู่สถานะรีเซต และพร้อมที่จะรับการตั้งค่าควบคุมต่าง ๆ แล้ว แต่ถ้าบิต RST เป็น ‘0’ แสดงว่าชิพยังไม่ได้เข้าสู่โมดรีเซต และอาจมีความผิดพลาดทางฮาร์ดแวร์เกิดขึ้น เมื่อได้รีเซต

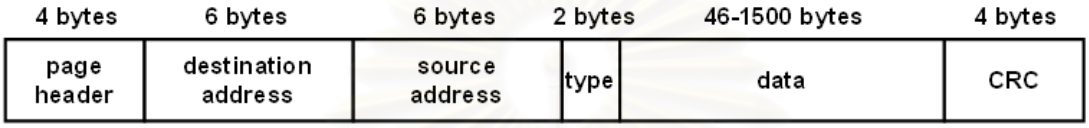
ทางซอฟต์แวร์แล้ว ในการเซตค่ารีจิสเตอร์ควบคุมการทำงานต่าง ๆ ควรเขียนตั้งค่าบิต STA, STP ภายในรีจิสเตอร์ CR เป็น '0' และ '1' ตามลำดับก่อนเพื่อให้ชิพอยู่ในโมดหยุดการทำงาน จากนั้นจึงค่อยเขียนค่าลงรีจิสเตอร์ควบคุมต่าง ๆ เพื่อกำหนดการทำงานของชิพอีเทอร์เนต ได้แก่รูปแบบการรับส่งข้อมูลผ่านพอร์ต DMA, การเลือกรับเฟรมเข้ามาเก็บในบัฟเฟอร์, ตำแหน่งเริ่มต้น และสิ้นสุดของบัฟเฟอร์ภาครับ และภาคส่ง, การทำ CRC บนตัวชิพ, ฮาร์ดแวร์แอดเดรสของอุปกรณ์, และการควบคุมอินเตอร์รัปต์ ดังแสดงในรูปที่ 2.28 เมื่อตั้งโมดการทำงานเรียบร้อยแล้ว จึงค่อยตั้งค่าบิต STA, STP ให้เป็น '1' และ '0' ตามลำดับเพื่อให้ชิพเริ่มทำงาน และรีเซตรีจิสเตอร์ ISR เพื่อเคลียร์บิตแสดงอินเตอร์รัปต์จากการรีเซตทางซอฟต์แวร์ให้ชิพพร้อมทำงานต่อไป

รูปที่ 3.10 การ Initialize ชิพอีเทอร์เนต

3.2.3 การรับข้อมูลผ่านชิพอีเทอร์เนต RTL8019as

จากการตั้งโมดการทำงานของชิพข้างต้น เมื่อมีข้อมูลเข้ามาชิพอีเทอร์เนตจะตรวจสอบการซิงโครไนซ์เฟรม, CRC และฮาร์ดแวร์แอดเดรสปลายทาง ภายในเฮดเดอร์อีเทอร์เนต โดยถ้าข้อมูลที่ได้รับมานั้นมีการซิงโครไนซ์เฟรมที่ถูกต้อง, ไม่มีข้อมูลผิดพลาด, และมีฮาร์ดแวร์แอดเดรสปลายทางตรงกับค่าที่เก็บในรีจิสเตอร์ PAR0-PAR5 หรือมีฮาร์ดแวร์แอดเดรสเป็นแอดเดรสแพร่สัญญาณ ชิพอีเทอร์เนตจะเก็บข้อมูลที่ได้รับมาไว้ในบัฟเฟอร์ พร้อมกับส่งสัญญาณอินเตอร์รัปต์ไปหา

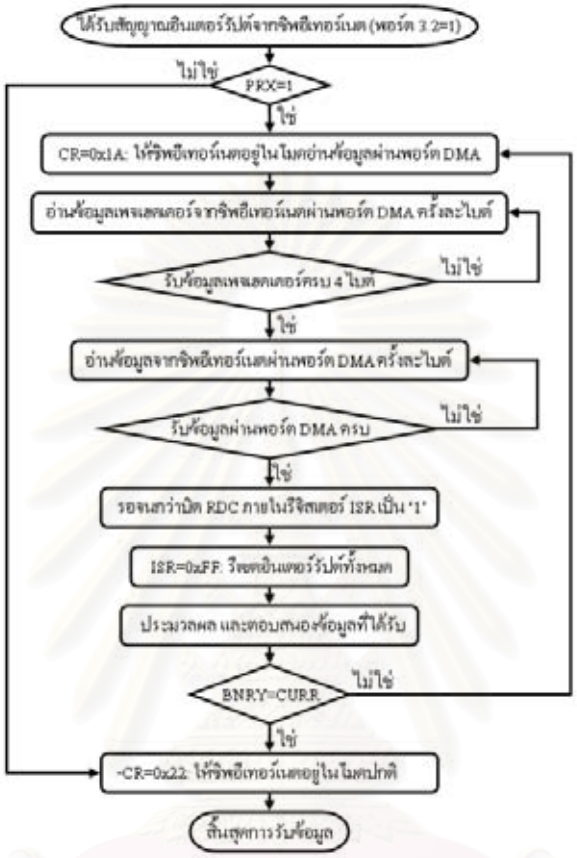
ไมโครคอนโทรลเลอร์ผ่านทางพอร์ต INT0 และปรับเปลี่ยนบิต PRX ภายในรีจิสเตอร์ ISR ให้เป็น ‘1’ เมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณอินเตอร์รัปต์จากชิพอีเทอร์เนต ไมโครคอนโทรลเลอร์จะอ่านค่ารีจิสเตอร์ ISR เพื่อตรวจสอบชนิดของอินเตอร์รัปต์ที่เกิดขึ้น ถ้าตรวจพบว่าบิต PRX เป็น ‘1’ ไมโครคอนโทรลเลอร์จะเซตบิต RD2, RD1, RD0 ภายในรีจิสเตอร์ CR ให้เป็น “001” ตามลำดับ เพื่อให้สามารถรับข้อมูลจากพอร์ต DMA ของชิพอีเทอร์เนตได้ โดยข้อมูลที่รับได้จากชิพอีเทอร์เนตจะมีโครงสร้างดังแสดงในรูปที่ 3.11



รูปที่ 3.11 โครงสร้างของข้อมูลที่ได้รับจากชิพอีเทอร์เนตผ่าน DMA พอร์ต

จะเห็นได้ว่าในส่วนต้นของโครงสร้างข้อมูลที่ได้รับจากชิพอีเทอร์เนตผ่าน DMA พอร์ตนั้นไม่เหมือนกับโครงสร้างเฟรมอีเทอร์เนตมาตรฐานทั่วไป ทั้งนี้เนื่องจากข้อมูลส่วน Preamble นั้นใช้ในการซิงโครไนซ์เฟรมเท่านั้น ชิพอีเทอร์เนตจึงไม่นำส่วน Preamble นี้เข้ามาเก็บในบัฟเฟอร์ภาครับด้วย แต่จะเพิ่มข้อมูล Page Header ซึ่งเป็นข้อมูลขนาด 4 ไบต์ใช้บ่งบอกพารามิเตอร์ภายในบัฟเฟอร์ภาครับไว้ เพื่อให้ไมโครคอนโทรลเลอร์สามารถควบคุม และรับข้อมูลได้อย่างถูกต้องแทน ข้อมูลภายใน Page Header ไบต์แรกบ่งบอกถึงสถานะของเฟรมข้อมูลที่รับมาได้ ไบต์ที่สองบ่งบอกตำแหน่งหน้าบัฟเฟอร์ภาครับถัดไป และไบต์ที่สามและสี่บ่งบอกขนาดข้อมูลในเฟรม ดังนั้นในการรับข้อมูลจากชิพอีเทอร์เนตผ่านพอร์ต DMA ไมโครคอนโทรลเลอร์จะอ่านข้อมูล Page Header และตรวจสอบขนาดทั้งหมดของเฟรมข้อมูลภายในบัฟเฟอร์ก่อน แล้วจึงค่อยอ่านค่าข้อมูลจากชิพอีเทอร์เนตนั้นมาเก็บไว้ในหน่วยความจำภายนอกครั้งละไบต์จนครบตามที่ระบุไว้ใน Page Header จากนั้นไมโครคอนโทรลเลอร์จะตรวจสอบบิต RDC ภายในรีจิสเตอร์ ISR โดยจะรอจนกว่าบิต RDC นี้มีค่าเป็น ‘1’ เพื่อยืนยันว่าข้อมูลเฟรมนั้นได้ถูกอ่านออกจากพอร์ต DMA ครบและสำเร็จแล้ว เมื่อบิต RDC นี้มีค่าเป็น ‘1’ แล้วไมโครคอนโทรลเลอร์จะรีเซตอินเตอร์รัปต์ทั้งหมด โดยการเขียนค่า ‘1’ ลงในรีจิสเตอร์ ISR ทุกบิต จากนั้นจะเปรียบเทียบรีจิสเตอร์ CURR และ BNRY ซึ่งเป็นรีจิสเตอร์ชี้ตำแหน่งหน้าการอ่าน และรับข้อมูลของบัฟเฟอร์ภาครับตามลำดับ โดยถ้ารีจิสเตอร์ CURR และBNRY มีค่าเท่ากันแสดงว่าไม่มีข้อมูลภายในบัฟเฟอร์ภาครับแล้วไมโครคอนโทรลเลอร์จะรีเซตอินเตอร์รัปต์ และปรับบิต RD2 ภายในรีจิสเตอร์ CR ให้มีค่าเป็น ‘1’ เพื่อให้ชิพออกจากโมดการอ่านข้อมูลจากพอร์ต DMA และเป็นการสิ้นสุดกระบวนการรับข้อมูลจากชิพอีเทอร์เนต แต่ถ้ารีจิสเตอร์ CURR และ BNRY มีค่าไม่เท่ากันแสดงว่ายังมีเฟรมข้อมูลหลงเหลืออยู่ในบัฟเฟอร์ภาครับอยู่ ไมโครคอนโทรลเลอร์จะรับข้อมูลในบัฟเฟอร์นั้น

โดยใช้กระบวนการดังที่กล่าวไว้ข้างต้นรับข้อมูลจากบัฟเฟอร์ภาครับจนกว่าจะหมด ดังแสดงในรูปที่ 3.12

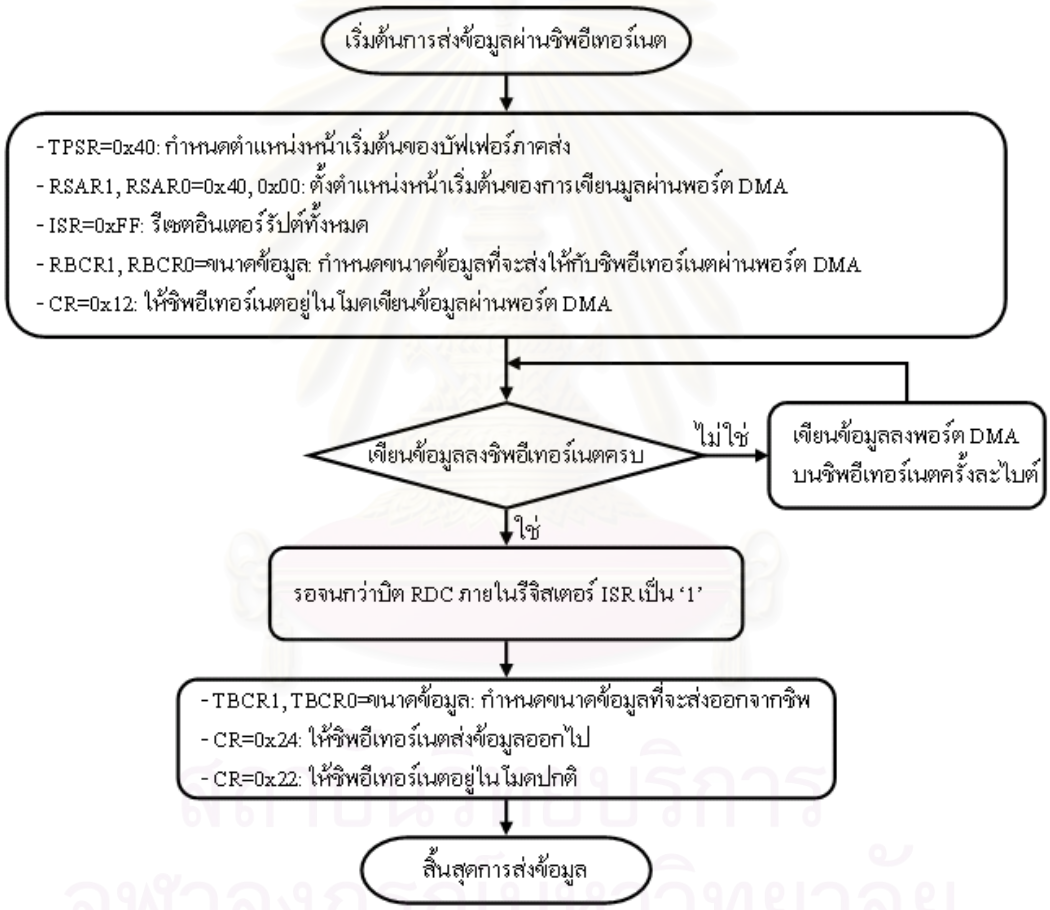


รูปที่ 3.12 การควบคุมเพื่อรับข้อมูลจากชิพอีเทอร์เนต

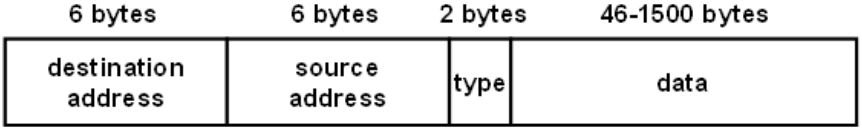
3.2.4 การส่งข้อมูลผ่านชิพอีเทอร์เนต RTL8019as

ในการส่งข้อมูลผ่านชิพอีเทอร์เนตนั้นมีความซับซ้อนน้อยกว่าการรับข้อมูลมาก ดังแสดงในรูปที่ 3.13 เริ่มแรกไมโครคอนโทรลเลอร์จะตั้งค่ารีจิสเตอร์ควบคุม TPSR เพื่อกำหนดตำแหน่งหน้าเริ่มต้นของบัฟเฟอร์ภาคส่ง และตั้งค่ารีจิสเตอร์ RSAR0 และ RSAR1 เพื่อระบุตำแหน่งการเขียนข้อมูลผ่านพอร์ต DMA ให้เริ่มเขียนที่ตำแหน่งหน้าเริ่มต้นของบัฟเฟอร์ภาคส่ง และตั้งค่า RBCR1 และ RBCR0 ซึ่งใช้ในการกำหนดขนาดของข้อมูลที่จะส่งเข้าพอร์ต DMA จากนั้นไมโครคอนโทรลเลอร์จะรีเซตรีจิสเตอร์ ISR และปรับบิต RD0, RD1, RD2 ภายในรีจิสเตอร์ CR ให้เป็น “010” ตามลำดับ เพื่อให้พอร์ต DMA ทำงานในโมดรับข้อมูล จากนั้นจึงเขียนข้อมูลลงพอร์ต DMA ครั้งละไบต์โดยข้อมูลที่เขียนนั้นต้องมีโครงสร้างดังแสดงในรูปที่ 3.14 เมื่อเขียนข้อมูลลงพอร์ต DMA ครบแล้ว ไมโครคอนโทรลเลอร์จะตรวจสอบบิต RDC ภายในรีจิสเตอร์ ISR และรอจนกว่าบิต RDC นี้มีค่าเป็น

‘1’ เพื่อรับการยืนยันจากชิพอีเทอร์เนตว่ากระบวนการส่งข้อมูลผ่านพอร์ต DMA นั้นได้สำเร็จแล้ว เมื่อถึงขั้นตอนนี้ข้อมูลจะยังคงอยู่ในบัฟเฟอร์ภาคส่งของชิพอีเทอร์เนต และยังไม่ถูกส่งออกไป ในการส่งข้อมูลออกจากชิพอีเทอร์เนตทำได้โดยการกำหนดค่ารีจิสเตอร์ควบคุม TBCR1, TBCR0 เพื่อกำหนดขนาดของข้อมูลที่ต้องการส่ง และตั้งบิต TXP ภายในรีจิสเตอร์ CR ให้เป็น ‘1’ ชิพอีเทอร์เนตจะคำนวณ CRC สำหรับข้อมูลในบัฟเฟอร์ภาคส่งนั้น และเพิ่มข้อมูล CRC ขนาด 4 ไบต์ไว้ตอนท้ายของเฟรม และส่ง preamble ขนาด 8 ไบต์ออกไป ตามด้วยข้อมูลภายในบัฟเฟอร์ภาคส่งตามลำดับ เพื่อให้ได้เฟรมข้อมูลตามมาตรฐานของอีเทอร์เนต และเป็นการสิ้นสุดการควบคุมการส่งข้อมูล



รูปที่ 3.13 การควบคุมการส่งข้อมูลผ่านชิพอีเทอร์เนต



รูปที่ 3.14 รูปแบบโครงสร้างข้อมูลที่ส่งให้ชิพอีเทอร์เนตผ่านพอร์ต DMA

3.3 ส่วนรองรับโพรโทคอล

ในส่วนรองรับโพรโทคอลทำหน้าที่ประมวลผลข้อมูล, สร้างข้อมูล, และควบคุมการทำงานให้เป็นไปตามมาตรฐานโพรโทคอล ARP (Address Resolution Protocol), IP (Internet Protocol), UDP (User Datagram Protocol), RTP (Real Time Protocol), และ SIP (Session Initiation Protocol) ตามลำดับ

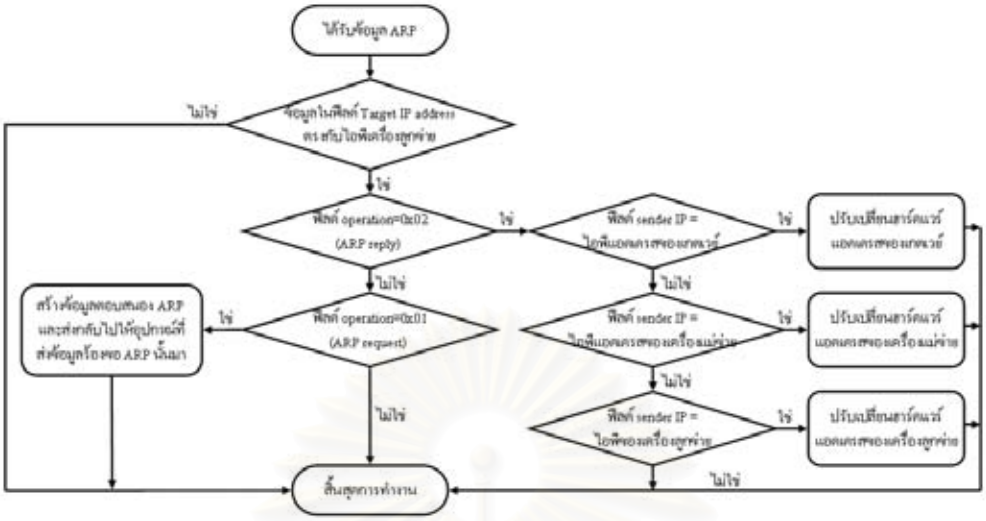
3.3.1 การทำงานรองรับข้อมูลที่ได้รับ

ในภาครับนั้น เมื่อได้รับข้อมูลจากชิพอีเทอร์เนตแล้ว ไมโครคอนโทรลเลอร์จะตรวจสอบ และแยกแยะข้อมูลที่ได้รับว่าเป็นข้อมูลประเภทใด โดยตรวจสอบจากฟิลด์ Type ในอีเทอร์เนตเฮดเดอร์, ฟิลด์ Protocol ในเฮดเดอร์ไอพี, และฟิลด์ UDP destination port number ใน UDP เฮดเดอร์ ซึ่งจะสามารถแบ่งแยกข้อมูลออกได้เป็น ข้อมูลร้องขอ ARP, ข้อมูลตอบสนอง ARP, ข้อมูล ICMP, ข้อมูล SIP, ข้อมูลเสียง, และข้อมูลอื่น ๆ ที่ไม่สนใจ ดังแสดงในตารางที่ 3.3

ตารางที่ 3.3 การจำแนกข้อมูลของไมโครคอนโทรลเลอร์ในส่วนภาครับ

Type of data	Type (Ethernet header)	Protocol (IP header)	UDP destination port
ARP	0x0806 (ARP)	-	-
ICMP	0x0800 (IP)	0x01 (ICMP)	-
SIP	0x0800 (IP)	0x11 (UDP)	5060
Voice	0x0800 (IP)	0x11 (UDP)	49170

ในกรณีที่ข้อมูลที่ได้รับมานั้นเป็นข้อมูล ARP ไมโครคอนโทรลเลอร์จะตรวจสอบฟิลด์ Target IP Address และเปรียบเทียบกับไอพีแอดเดรสของเครื่องโทรศัพท์ ถ้ามีค่าไม่ตรงกันไมโครคอนโทรลเลอร์จะไม่สนใจ และจบการทำงานรองรับข้อมูลนั้น แต่ถ้ามีค่าตรงกันไมโครคอนโทรลเลอร์จะตรวจสอบฟิลด์ Operation เพื่อแยกแยะว่าข้อมูล ARP นี้เป็นข้อมูลร้องขอ ARP หรือข้อมูลตอบสนอง ARP ถ้าตรวจสอบแล้วเป็นข้อมูลร้องขอ ARP ไมโครคอนโทรลเลอร์จะสร้างข้อมูลตอบสนอง ARP ภายในหน่วยความจำภายนอก และส่งให้ชิพอีเทอร์เนตเพื่อตอบสนองการร้องขอนั้น แต่ถ้าเป็นข้อมูล ARP ตอบสนอง ไมโครคอนโทรลเลอร์จะตรวจสอบฟิลด์ Sender IP Address ว่าเป็นไอพีแอดเดรสของเครื่องลูกข่ายปลายทาง, เครื่องแม่ข่าย หรือเกตเวย์ เมื่อสามารถจำแนกได้แล้วก็จะใช้ค่าภายในฟิลด์ Sender Hardware Address ปรับเปลี่ยนค่าฮาร์ดแวร์แอดเดรสของอุปกรณ์นั้นใหม่ดังแสดงในรูปที่ 3.15



รูปที่ 3.15 การรองรับโพรโทคอล ARP ในภาครับ

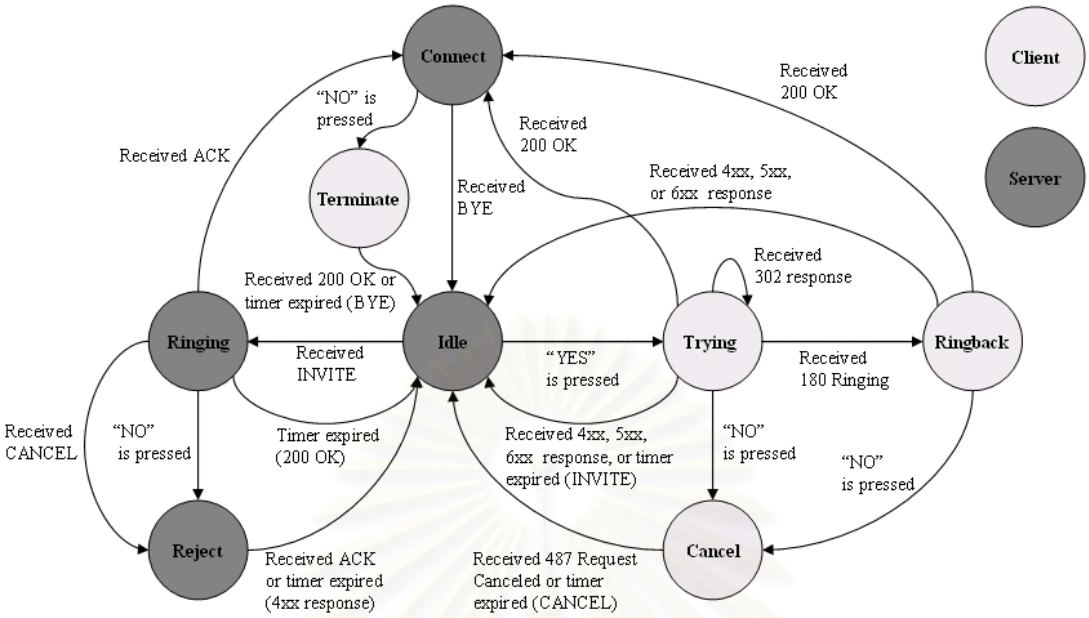
ในการส่งข้อมูลร้องขอ ARP เพื่อหาฮาร์ดแวร์แอดเดรสของเกตเวย์นั้น จะเกิดขึ้นทุกครั้งที่ไมโครคอนโทรลเลอร์ได้รับข้อมูลไอพีแอดเดรสของอุปกรณ์ภายนอกอื่นมาใหม่ และไอพีแอดเดรสนั้นอยู่ในโครงข่ายพื้นที่ท้องถิ่นเดียวกับไอพีแอดเดรสของเครื่องโทรศัพท์ ซึ่งมีอยู่ 3 ช่วงด้วยกัน คือช่วงเริ่มเปิดเครื่องโทรศัพท์, ช่วงที่ได้จัดรูปลักษณ์ (Configure) ไอพีแอดเดรสให้กับอุปกรณ์อื่นใหม่ และช่วงเริ่มสร้างเซสชันกับเครื่องลูกข่ายปลายทาง ในช่วงเริ่มเปิดเครื่องโทรศัพท์ เมื่อไมโครคอนโทรลเลอร์โหลดไอพีแอดเดรส ของอุปกรณ์ภายนอก จาก EEPROM ลงในหน่วยความจำเรียบร้อยแล้ว เครื่องโทรศัพท์จะส่งข้อมูลร้องขอ ARP เพื่อค้นหาฮาร์ดแวร์แอดเดรสของเกตเวย์ เมื่อได้รับหมายเลขฮาร์ดแวร์แอดเดรสของเกตเวย์แล้ว ในกรณีที่เครื่องโทรศัพท์ทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย เครื่องโทรศัพท์จะตรวจสอบว่าเครื่องแม่ข่ายอยู่ในโครงข่ายพื้นที่ท้องถิ่นวงเดียวกับเครื่องโทรศัพท์หรือไม่ ถ้าเครื่องแม่ข่ายอยู่ในโครงข่ายเดียวกัน เครื่องโทรศัพท์จะส่งข้อมูลร้องขอ ARP เพื่อค้นหาฮาร์ดแวร์แอดเดรสของเครื่องแม่ข่ายต่อไป ในช่วงที่ได้จัดรูปลักษณ์ไอพีแอดเดรสให้กับเกตเวย์ และเครื่องแม่ข่ายก็เช่นกัน เมื่อจัดรูปลักษณ์เรียบร้อยแล้วถ้าไอพีแอดเดรสนั้นอยู่ในโครงข่ายเดียวกับเครื่องโทรศัพท์ ไมโครคอนโทรลเลอร์จะส่งข้อมูลร้องขอ ARP เพื่อค้นหาฮาร์ดแวร์แอดเดรสของอุปกรณ์ภายนอกใหม่ ส่วนช่วงเริ่มสร้างเซสชันกับเครื่องลูกข่ายปลายทาง ในกรณีที่เครื่องโทรศัพท์ทำงานในโมดติดต่อโดยตรง เมื่อผู้ใช้งานต้องการสร้างเซสชันกับเครื่องลูกข่ายปลายทางที่อยู่ภายในโครงข่ายเดียวกัน และได้ป้อน SIP URL ปลายทางเรียบร้อยแล้ว เครื่องโทรศัพท์จะส่งข้อมูลร้องขอ ARP เพื่อค้นหาฮาร์ดแวร์แอดเดรสของเครื่องลูกข่ายปลายทางก่อนทุกครั้ง ส่วนในกรณีที่เครื่องโทรศัพท์ทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย หรือเป็นฝ่ายถูกร้องขอการสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความ SIP ที่บ่งบอกถึงไอพีแอดเดรสของเครื่องลูกข่ายปลายทางที่อยู่

ภายในโครงข่ายเดียวกัน เครื่องโทรศัพท์จะส่งข้อมูลร้องขอ ARP เพื่อค้นหาฮาร์ดแวร์แอดเดรสของเครื่องลูกข่ายทุกครั้งเช่นกัน

ในกรณีที่ข้อมูลที่ได้รับมามีในอีเทอร์เนตเฮดเดอร์มีฟิลด์ type เป็น 0x0800 แสดงว่าข้อมูลในชั้นบนเป็นโพรโทคอล IP ไมโครคอนโทรลเลอร์จะเปรียบเทียบฟิลด์ destination IP address กับไอพีแอดเดรสของตัวเครื่อง ถ้าไม่ตรงกันไมโครคอนโทรลเลอร์จะไม่สนใจข้อมูลนั้น แต่ถ้าตรงกันไมโครคอนโทรลเลอร์จะตรวจสอบข้อมูลในฟิลด์ protocol ในไอพีเฮดเดอร์เพื่อแยกแยะระหว่างข้อมูล ICMP และ UDP โดยถ้าข้อมูลที่ได้รับเป็นข้อมูล ICMP ไมโครคอนโทรลเลอร์จะสร้างข้อมูลตอบสนอง ICMP และส่งกลับไปยังปลายทาง แต่ถ้าข้อมูลที่ได้รับเป็นข้อมูล UDP ไมโครคอนโทรลเลอร์จะตรวจสอบฟิลด์ destination port ภายใน UDP เฮดเดอร์ ถ้ามีค่าเป็น 49170 ซึ่งเป็นพอร์ตที่เครื่องโทรศัพท์นั้นตกลงกับเครื่องลูกข่ายปลายทางให้เป็นพอร์ตที่ใช้ในการรับข้อมูลเสียงไว้ก่อนแล้ว ไมโครคอนโทรลเลอร์จะนำข้อมูลเสียงนั้นเขียนลงในบัฟเฟอร์ข้อมูลเสียงภาครับเพื่อรอการแสดงผล (Play Back) ออกไป แต่ถ้าข้อมูลมีพอร์ตปลายทางเป็น 5060 ซึ่งเป็นพอร์ตสำหรับใช้ในการส่งข้อความ SIP ไมโครคอนโทรลเลอร์จะตรวจสอบข้อความ SIP เพื่อตีความข้อความ SIP นั้นต่อไป

ในการสร้างและ ยกเลิกเซสชันด้วยโพรโทคอล SIP นั้น เครื่องโทรศัพท์ต้องสามารถทำงานเป็นได้ทั้งเครื่องลูกข่าย และเครื่องแม่ข่าย และต้องทำงานแบบมีสถานะ โดยมีสถานะ และการทำงานคร่าว ๆ ดังแสดงในรูปที่ 3.16 จะเห็นได้ว่าการสร้างเซสชันนั้นสามารถแบ่งออกได้เป็น 2 แบบ แบบแรกเครื่องโทรศัพท์ที่ทำหน้าที่เป็นเครื่องลูกข่ายร้องขอเพื่อขอสร้างเซสชัน และแบบที่ 2 เครื่องโทรศัพท์ที่เป็นเครื่องแม่ข่ายถูกร้องขอการสร้างเซสชัน ในการยกเลิกเซสชันก็เช่นกัน เครื่องโทรศัพท์อาจเป็นเครื่องลูกข่ายขอยกเลิกเซสชันเองก็ได้ หรือเป็นเครื่องแม่ข่ายถูกร้องขอให้ยกเลิกเซสชันก็ได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.16 แผนผังสถานะการทำงานเพื่อรองรับโพรโทคอล SIP

การเปลี่ยนสถานะของเครื่องโทรศัพท์นั้นขึ้นอยู่กับ การกดคีย์แพดบนเครื่องโทรศัพท์ และข้อมูล SIP ที่ได้รับจากภายนอก ในกรณีเครื่องโทรศัพท์ทำงานเป็นเครื่องลูกข่ายขอสร้างเซสชันกับเครื่องปลายทาง เมื่อผู้ใช้ได้ป้อนข้อมูล SIP URL ในขณะที่เครื่องโทรศัพท์อยู่ในสถานะ Idle และกดปุ่ม "YES" เพื่อเริ่มการขอสร้างเซสชัน เครื่องโทรศัพท์จะเปลี่ยนสถานะจาก Idle เป็น Trying พร้อมกับส่งข้อความ INVITE ออกไป ถ้าเครื่องโทรศัพท์ได้รับข้อมูลตอบสนอง 180 จากปลายทาง เครื่องโทรศัพท์จะเปลี่ยนสถานะจาก Trying เป็น Ringback ทันที และเมื่อได้รับข้อมูลตอบสนอง 200 OKในกรณีที่ผู้ใช้ปลายทางตกลงรับการสร้างเซสชัน เครื่องโทรศัพท์จะเปลี่ยนสถานะเป็น Connect และถือเป็นการสิ้นสุดการสร้างเซสชัน แต่ในกรณีที่ผู้ใช้งานปลายทางปฏิเสธการขอสร้างเซสชัน หรือมีปัญหาไม่สามารถสร้างเซสชันได้เครื่องโทรศัพท์จะได้รับข้อมูลตอบสนอง 4xx, 5xx, หรือ 6xx และจะกลับมาอยู่ในสถานะ Idle อีกครั้ง และในกรณีที่ผู้ใช้งานเกิดเปลี่ยนใจต้องการยกเลิกการสร้างเซสชันโดยกดปุ่ม "NO" ในขณะที่เครื่องโทรศัพท์อยู่ในสถานะ Trying หรือ Ringback เครื่องโทรศัพท์จะเปลี่ยนสถานะเป็น Cancel พร้อมกับส่งข้อความ CANCEL ไปยังปลายทาง เมื่อได้รับข้อความตอบสนอง 487 ซึ่งเป็นการตอบรับการยกเลิกเซสชัน เครื่องโทรศัพท์จะเปลี่ยนสถานะกลับเป็น Idle

ในการสร้างเซสชันในกรณีเครื่องโทรศัพท์ทำงานเป็นเครื่องแม่ข่าย เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ INVITE จะเปลี่ยนสถานะจาก Idle เป็น Ringing พร้อมกับส่งข้อความตอบสนอง 180 Ringing กลับไป เมื่อผู้ใช้กดปุ่ม "YES" เพื่อรับการร้องขอการสร้างเซสชันนั้น เครื่องโทรศัพท์จะส่ง 200 OK กลับไปและรอจนกว่าจะได้รับข้อความ ACK ซึ่งเป็นการตอบรับการสร้างเซสชัน

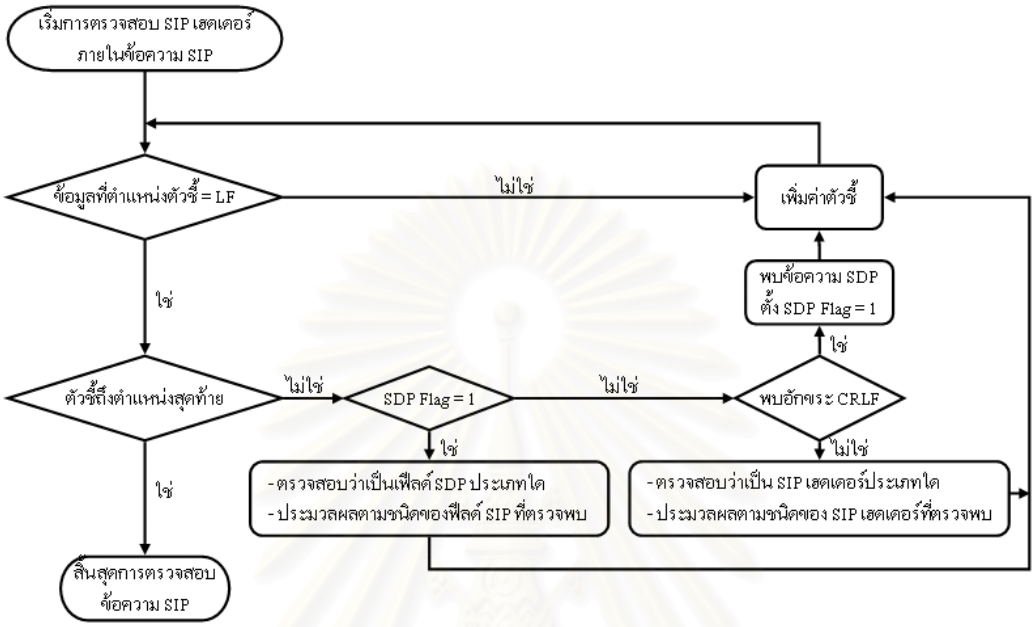
เครื่องโทรศัพท์จะเปลี่ยนสถานะเป็น Connect และเป็นการสิ้นสุดการสร้างเซสชัน แต่ในกรณีที่ผู้ใช้งานปฏิเสธการสร้างเซสชันโดยกดปุ่ม “NO” หรือปลายทางต้องการยกเลิกการสร้างเซสชันซึ่งเครื่องโทรศัพท์จะได้รับข้อความ CANCEL ในขณะที่เครื่องอยู่ในสถานะ Ringing เครื่องโทรศัพท์จะส่งข้อความตอบสนองกลับไป และเปลี่ยนสถานะกลับไปอยู่ในสถานะ Idle เมื่อเครื่องโทรศัพท์ได้รับข้อความ ACK ซึ่งเป็นการตอบรับการตอบสนองนั้น

ในการปิดเซสชัน กรณีที่เครื่องโทรศัพท์เป็นผู้ร้องขอการปิดเซสชัน เมื่อผู้ใช้งานกดปุ่ม “NO” ในขณะที่เครื่องอยู่ในสถานะ Connect เครื่องโทรศัพท์จะเปลี่ยนสถานะเป็น Terminate พร้อมกับส่งข้อความร้องขอ BYE ไปยังเครื่องปลายทาง และเมื่อได้รับข้อความตอบสนอง 200 OK เครื่องโทรศัพท์จึงจะกลับมาอยู่ในสถานะ Idle และเป็นการสิ้นสุดการขอปิดเซสชัน ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการปิดเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ BYE จากเครื่องปลายทาง เครื่องโทรศัพท์จะเปลี่ยนสถานะกลับไปยังสถานะ Idle พร้อมกับส่งข้อความตอบสนอง 200 OK กลับไป และเป็นการสิ้นสุดการขอปิดเซสชัน เช่นกัน

จะเห็นได้ว่าเครื่องโทรศัพท์ใช้ Method ภายใน Request-Line และ Status-Code ภายในStatus-Line ของข้อความ SIP ที่ได้รับมาเป็นส่วนสำคัญที่ใช้ควบคุมสถานะของเครื่องโทรศัพท์ แต่ในการสร้างเซสชันเครื่องโทรศัพท์จำเป็นต้องทราบพารามิเตอร์อื่น ๆ ที่สำคัญเช่น เซสชันไอดี, SIP URL ปลายทาง, ไอพีแอดเดรสเครื่องลูกข่ายปลายทาง, พอร์ตที่ใช้ในการรับข้อมูลเสียง, ชนิดการเข้ารหัสเสียงที่จะใช้ในเซสชัน เป็นต้น ดังนั้นในบางกรณีไมโครคอนโทรลเลอร์จะเข้าไปตรวจสอบภายในเฮดเดอร์ และรวมไปถึงข้อความ SDP (Session Description Protocol) ด้วย

การตรวจสอบ SIP เฮดเดอร์ และฟิลด์ SDP ภายในข้อความ SIP แสดงในรูปที่ 3.17 ไมโครคอนโทรลเลอร์จะไล่หาจุดเริ่มต้นของ SIP เฮดเดอร์ โดยการไล่ตรวจสอบอักขระครั้งละตัวเริ่มตั้งแต่ในส่วน Request-Line หรือ Status-Line ไปจนกว่าจะพบอักขระพิเศษ CRLF (Carriage Return/Line Feed) ซึ่งเป็นตัวระบุจุดสิ้นสุดบรรทัด เมื่อพบแล้วไมโครคอนโทรลเลอร์จะตรวจสอบอักขระในบรรทัดถัดไปเพื่อแยกแยะว่าเป็น SIP เฮดเดอร์ประเภทไหน แล้วจึงค่อยประมวลผลข้อความภายในบรรทัดนั้นตาม SIP เฮดเดอร์ที่ตรวจพบ เมื่อประมวลผลข้อความเสร็จ ไมโครคอนโทรลเลอร์จะไล่หาอักขระพิเศษ CRLF เพื่อตรวจสอบ SIP เฮดเดอร์บรรทัดต่อไป เรื่อยๆ จนกว่าจะประมวลผลครบทุกเฮดเดอร์ ในกรณีที่มีข้อความ SDP ต่อท้าย เมื่อไมโครคอนโทรลเลอร์ประมวลผล SIP เฮดเดอร์จนครบหมด และได้ไล่หาอักขระพิเศษ CRLF ที่ใช้ระบุจุดสิ้นสุดบรรทัดสุดท้ายของ SIP เฮดเดอร์แล้ว ในการตรวจสอบอักขระบรรทัดต่อไปจะพบอักขระ CRLF ซึ่งเป็นการเว้นบรรทัดระหว่าง SIP เฮดเดอร์กับข้อความ SDP ไมโครคอนโทรลเลอร์จะปรับเปลี่ยนการประมวลผลข้อความจาก SIP เฮดเดอร์ เป็น

ฟิลด์ SDP และจะไล่ตรวจสอบฟิลด์ SDP นั้นครั้งละบรรทัดโดยใช้หลักการเดียวกับที่ได้ตรวจ SIP เฮดเดอร์จนกว่าจะจบบรรทัดสุดท้าย เป็นการสิ้นสุดการประมวลผลข้อความ SIP



รูปที่ 3.17 ขั้นตอนการตรวจสอบ SIP เฮดเดอร์ และ SDP ภายในข้อความ SIP

ในการพัฒนาโพรโทคอล SIP ในโครงการวิจัยนี้ ไมโครคอนโทรลเลอร์สามารถประมวลผล SIP เฮดเดอร์ ได้ 6 ชนิดคือ Via, From, To, Cseq, Call-ID, และ Contact และสามารถประมวลผลฟิลด์ SDP ได้ 2 ชนิดคือฟิลด์ c และ m ดังแสดงในตารางที่ 3.4 จะเห็นได้ว่า วัตถุประสงค์ของการประมวลผล SIP เฮดเดอร์ และฟิลด์ SDP นั้นเพื่อเก็บข้อมูลต่าง ๆ ที่จำเป็นต่อการสร้างเซสชัน ซึ่งได้แก่พอร์ตที่ใช้ในการส่งข้อความ SIP, SIP URL ของผู้ใช้ปลายทาง, หมายเลข Cseq, หมายเลข Call-ID, ไอพีแอดเดรสปลายทาง, Tag, พอร์ตที่ใช้ในการส่งข้อมูลเสียง, การเลือกใช้วิธีการเข้ารหัสเสียง รวมไปถึงการตรวจสอบ SIP URL ที่ผู้ใช้ปลายทางต้องการสร้างเซสชัน ในกรณีที่ตัวเครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน ตัวเครื่องโทรศัพท์จะทราบข้อมูลส่วนใหญ่อยู่แล้ว มีเพียงข้อมูลที่ระบุการส่งข้อมูลเสียง และไอพีแอดเดรสของเครื่องโทรศัพท์ปลายทางในกรณีที่เครื่องทำงานในโมดทำงานร่วมกับเครื่องแม่ข่ายเท่านั้นที่เครื่องต้องตรวจสอบจาก SIP เฮดเดอร์ และฟิลด์ SDP ของข้อความตอบสนองที่ได้รับ แต่ในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสร้างเซสชัน เครื่องจะไม่มีข้อมูลเลย และต้องตรวจสอบจาก SIP เฮดเดอร์ และฟิลด์ SDP ของข้อความร้องขอ INVITE ที่ได้รับทั้งหมด

ตารางที่ 3.4 ข้อมูลที่ได้จากการประมวลผล SIP เฮดเดอร์ และฟิลด์ SDP

		INVITE	200 OK	302 moved temporarily
SIP เฮดเดอร์	Via	พอร์ต SIP ปลายทาง	-	-
	From	SIP URL ปลายทาง	-	-
	To	*Tag, ตรวจสอบ SIP URL	-	-
	Cseq	Cseq	-	-
	Call-ID	Call-ID	-	-
	Contact	-	-	*ไอพีแอดเดรส ปลายทาง
ฟิลด์ SDP	c	ไอพีแอดเดรสปลายทาง	*ไอพีแอดเดรสปลายทาง	-
	m	พารามิเตอร์ของข้อมูลเสียง	พารามิเตอร์ของข้อมูลเสียง	-

* เครื่องโทรศัพท์ทำงานในโมด ทำงานร่วมกับเครื่องแม่ข่าย

3.3.2 การสร้างข้อมูลตามมาตรฐานโพรโทคอลต่าง ๆ

ในภาคส่งนั้น เครื่องโทรศัพท์สามารถสร้างข้อความร้องขอได้ 5 ชนิดคือ INVITE, CANCEL, ACK, BYE, และ REGISTER ข้อความตอบสนองได้ 4 ชนิดคือ 180 Ringing, 200 OK, 486 Busy Here, และ 487 Request Canceled และในส่วน SIP เฮดเดอร์ นั้นเครื่องโทรศัพท์สามารถสร้างได้ 7 ชนิดคือ Via, From, To, Call-ID, Cseq, Content-Length, และ Contact และในส่วนฟิลด์ SDP สามารถสร้างได้ 6 ชนิดคือ v, o, s, c, t, และ m โดยในการสร้างข้อความ SIP นั้น ในกรณีที่มีโพรโทคอล SDP อยู่ในข้อความด้วย ไมโครคอนโทรลเลอร์จะสร้างข้อความ SDP เก็บไว้ในหน่วยความจำภายนอกก่อน เพื่อให้ทราบขนาดทั้งหมดของข้อความ SDP เพื่อนำไปใช้ในเฮดเดอร์ Content-Length จากนั้นจึงค่อยสร้างข้อความ SIP ครั้งละบรรทัดโดยแต่ละบรรทัดคั่นด้วยอักขระพิเศษ CRLF เริ่มจาก Request-line หรือ Status-line และตามด้วย SIP เฮดเดอร์ โดยในการสร้างเฮดเดอร์ Via ไมโครคอนโทรลเลอร์จะตรวจสอบสถานะของเครื่องขณะนั้น ว่าเป็นผู้ร้องขอ หรือผู้ถูกร้องขอ โดยในกรณีที่ทำงานเป็นผู้ร้องขอ ไมโครคอนโทรลเลอร์จะใช้ไอพีแอดเดรสของตัวเครื่องเอง และพอร์ต 5060 เป็นข้อมูลสำหรับการสร้างเฮดเดอร์ แต่ในกรณีที่เครื่องทำงานเป็นผู้ถูกร้องขอ ไมโครคอนโทรลเลอร์จะตรวจสอบว่าข้อความตอบสนองที่จะสร้างนี้ ตอบสนองข้อความร้องขอที่ผ่านเครื่องแม่ข่ายมาหรือไม่ ถ้าเป็นข้อความร้องขอที่ไม่ผ่านเครื่องแม่ข่าย ไมโครคอนโทรลเลอร์จะใช้ไอพีแอดเดรสของเครื่องปลายทาง และพอร์ตรับข้อความ SIP ปลายทางที่ได้จากการตรวจสอบ SIP เฮดเดอร์ในข้อความ INVITE มาเป็นข้อมูลสำหรับสร้างเฮดเดอร์ แต่ถ้าเป็นข้อความร้องขอที่ผ่านเครื่องแม่ข่ายมา ไมโครคอนโทรลเลอร์จะใช้ไอพีแอดเดรสของเครื่องแม่ข่ายแทน ในการสร้างเฮดเดอร์ From ซึ่งเป็นเฮดเดอร์ที่ระบุ SIP URL ของผู้ร้องขอ ดังนั้นในกรณีที่เครื่องโทรศัพท์อยู่ในสถานะเป็นผู้ร้องขอ ไมโครคอนโทรลเลอร์จะใช้ข้อมูล SIP URL ของตัวเครื่องเอง เป็นข้อมูลในการสร้างเฮดเดอร์ แต่ในกรณีที่เครื่องอยู่ในสถานะเป็นผู้ถูกร้องขอ

ไมโครคอนโทรลเลอร์จะใช้ข้อมูล SIP URL ของผู้ใช้ปลายทางเป็นข้อมูลในการสร้างเฮดเดอร์ ในการสร้างเฮดเดอร์ To ซึ่งเป็นเฮดเดอร์ที่ระบุ SIP URL ของผู้ใช้งานปลายทางซึ่งเป็นผู้ถูกร้องขอ ดังนั้นในกรณีที่เครื่องอยู่ในสถานะ เป็นผู้ร้องขอ ไมโครคอนโทรลเลอร์จะนำ SIP URL ปลายทางที่ได้จากการป้อนข้อมูลของผู้ใช้งานผ่านคีย์แพดมาสร้างเฮดเดอร์ และในกรณีที่เครื่องอยู่ในสถานะเป็นผู้ถูกร้องขอ ไมโครคอนโทรลเลอร์จะใช้ SIP URL ของตัวเครื่องเองเป็นข้อมูลในการสร้างเฮดเดอร์ ในการสร้างเฮดเดอร์ Call-ID ซึ่งเป็นเฮดเดอร์ที่ใช้ในการระบุเซสชัน มีรูปแบบเป็น *หมายเลขสุ่ม@ไอพีแอดเดรสของเครื่องที่ร้องขอสร้างเซสชัน* ดังนั้นในกรณีที่เครื่องทำงานเป็นผู้ร้องขอ ไมโครคอนโทรลเลอร์จะใช้หมายเลขสุ่ม ที่ได้จากการสุ่มก่อนร้องขอ และไอพีแอดเดรสของเครื่องมาเป็นข้อมูลในการสร้างเฮดเดอร์ และในกรณีที่เครื่องทำงานในสถานะเป็นผู้ถูกร้องขอ ไมโครคอนโทรลเลอร์จะนำอักขระซึ่งระบุถึง Call-ID ภายในเฮดเดอร์ Call-ID ของข้อความร้องขอมาเป็นข้อมูลในการสร้างเฮดเดอร์ ในการสร้างเฮดเดอร์ Cseq ซึ่งเป็นเฮดเดอร์ที่ระบุจำนวนครั้ง และชนิดของการร้องขอ โดยในกรณีที่เครื่องทำงานในสถานะเป็นผู้ร้องขอ ไมโครคอนโทรลเลอร์จะใช้ข้อมูลที่ทำหน้าที่นับการร้องขอภายในหน่วยความจำและชนิดการร้องขอนั้นมาเป็นข้อมูลในการสร้างเฮดเดอร์ และในกรณีที่เครื่องทำงานในสถานะเป็นผู้ถูกร้องขอ ไมโครคอนโทรลเลอร์จะนำข้อมูลที่ได้จากเฮดเดอร์ Cseq ของข้อความร้องขอมาเป็นข้อมูลในการสร้างเฮดเดอร์ ในการสร้างเฮดเดอร์ Contact ซึ่งเป็นเฮดเดอร์ที่ใช้ระบุหมายเลขไอพีแอดเดรสเครื่อง ดังนั้นไมโครคอนโทรลเลอร์จะใช้ไอพีแอดเดรสของเครื่องเป็นข้อมูลในการสร้างเฮดเดอร์ ทั้งในกรณีที่เครื่องทำงานในสถานะเป็นผู้ร้องขอ และเป็นผู้ถูกร้องขอ ในการสร้างเฮดเดอร์ Content-Type ซึ่งเป็นเฮดเดอร์ที่ระบุชนิดของข้อมูลซึ่งบรรจุอยู่ในข้อความ SIP ซึ่งในการพัฒนาโทรศัพท์ SIP นี้มีเพียงข้อมูล SDP เท่านั้น ดังนั้นในการสร้างเฮดเดอร์ Content-Type จึงเป็นข้อมูลตายตัวระบุถึงข้อมูล SDP เป็นข้อมูลภายในข้อความ SIP ในการสร้างเฮดเดอร์ Content-Length ซึ่งใช้ระบุขนาดของข้อมูลที่บรรจุอยู่ในข้อความ SIP ซึ่งก็คือขนาดของข้อมูล SDP นั่นเอง ไมโครคอนโทรลเลอร์จะใช้ข้อมูลที่ได้จากการนับขนาดของข้อมูล SDP ที่ได้สร้างไว้ก่อนมาเป็นข้อมูลในการสร้างเฮดเดอร์นี้

ในการสร้างเฮดเดอร์ RTP สำหรับรองรับข้อมูลเสียงไมโครคอนโทรลเลอร์จะสร้างฟิลด์ V ให้มีค่าเป็น 2 เพื่อเป็นการระบุการใช้โพรโทคอล RTP เวอร์ชัน 2 ซึ่งเป็นเวอร์ชันปัจจุบัน ส่วนฟิลด์ P, X, CC, และ M นั้นจะกำหนดค่าให้เป็น 0 ทั้งหมดเพื่อระบุว่าไม่มีการเพิ่มข้อมูลแพดในตอนท้ายของแพ็กเกต, ไม่มีเฮดเดอร์เพิ่มเติม, มีแหล่งกำเนิดข้อมูลเดียว, และไม่มีการกระทำพิเศษเกิดขึ้นบนแพ็กเกตนั้น, ส่วนในฟิลด์ payload type ไมโครคอนโทรลเลอร์จะกำหนดค่าเป็น 0 ในกรณีชิพเข้ารหัสเสียงทำงานในโมดเข้ารหัส PCM μ-law หรือกำหนดค่าเป็น 8 ในกรณีที่ชิพเข้ารหัสเสียงทำงานในโมดเข้ารหัส PCM A-law ตามลำดับ ในฟิลด์ sequence number และ time stamp เมื่อเริ่มเปิดเซสชัน

ไมโครคอนโทรลเลอร์จะสุ่มค่าเริ่มต้นสำหรับฟิลด์ และจะเพิ่มค่าภายในฟิลด์ทุก ๆ ครั้งที่ส่งแพ็กเก็ตข้อมูลเสียงออกไป โดยในฟิลด์ sequence number ไมโครคอนโทรลเลอร์จะเพิ่มค่าทีละ 1 เพื่อระบุลำดับแพ็กเก็ตเสียง ส่วนในฟิลด์ time stamp ไมโครคอนโทรลเลอร์จะเพิ่มค่าภายในฟิลด์ทีละ 160 เพื่อบ่งบอกจำนวนการชักตัวอย่างข้อมูลเสียงในแพ็กเก็ตนั้น ในฟิลด์ SSRCI ซึ่งใช้ระบุต้นทางผู้สร้างแพ็กเกต เมื่อเริ่มเปิดเซสชันไมโครคอนโทรลเลอร์จะสุ่มค่าภายในฟิลด์ และใช้ค่าที่สุ่มขึ้นนั้นตลอดจนกว่าจะปิดเซสชัน ส่วนในฟิลด์ CSRC ไมโครคอนโทรลเลอร์จะละทิ้งเนื่องจากมีแหล่งกำเนิดข้อมูลเสียงเพียงแหล่งเดียว

ในการสร้างเฮดเดอร์ UDP ซึ่งประกอบด้วยฟิลด์ Source Port, Destination Port, UDP length, และ Checksum ไมโครคอนโทรลเลอร์จะกำหนดฟิลด์ Source Port เป็น 50000 เสมอ ส่วนฟิลด์ Destination Port ในกรณีที่ข้อมูลเป็นข้อความร้องขอ SIP ไมโครคอนโทรลเลอร์จะกำหนดให้เป็น 5060 ซึ่งเป็นพอร์ต Default สำหรับโพรโทคอล SIP แต่ในกรณีที่ข้อมูลเป็นข้อความตอบสนอง SIP ไมโครคอนโทรลเลอร์จะใช้ค่าพอร์ตที่ได้จากเฮดเดอร์ Via ภายในข้อความร้องขอ SIP มาเป็นข้อมูลในการกำหนดพอร์ตปลายทาง และกรณีที่เป็นข้อมูลเสียงไมโครคอนโทรลเลอร์จะใช้ค่าพอร์ตที่ได้จากฟิลด์ m ภายในข้อความ SDP มาเป็นข้อมูลในการกำหนดพอร์ตปลายทางตามลำดับ ในการสร้างฟิลด์ Length ซึ่งเป็นฟิลด์ที่ระบุถึงขนาดของ UDP ดาตาแกรมนั้น ในกรณีที่ข้อมูลเป็นข้อความ SIP ไมโครคอนโทรลเลอร์จะตรวจสอบขนาดของข้อความ SIP ที่ได้สร้างขึ้น และใช้ข้อมูลจากการตรวจสอบนี้มารวมกับเฮดเดอร์ UDP ซึ่งมีขนาด 8 ไบต์เพื่อให้ได้ขนาดของ UDP ดาตาแกรม และในกรณีที่ข้อมูลเป็นข้อมูลเสียงไมโครคอนโทรลเลอร์จะกำหนดให้ฟิลด์ Length นี้มีค่าเป็น 180 เสมอ เนื่องจากข้อมูลเสียงทุกแพ็กเกตมีขนาดคงที่ 160 ไบต์ รวมกับเฮดเดอร์ RTP ขนาด 12 ไบต์และเฮดเดอร์ UDP ขนาด 8 ไบต์ตามลำดับ ในส่วนการสร้างฟิลด์ UDP Checksum นั้น ไมโครคอนโทรลเลอร์จะละไว้ ไม่คำนวณและใส่ค่าเป็น ‘0’ หมดทั้งฟิลด์

ในการสร้าง IP เฮดเดอร์ดังแสดงในตารางที่ 3.5 ไมโครคอนโทรลเลอร์จะกำหนดฟิลด์ version, header length, TOS, flag, fragment offset, และ time to live เป็น 4, 5, 0x10, 0, 0, และ 64 ตามลำดับ เพื่อให้โพรโทคอลไอพีที่ใช้เป็นเวอร์ชัน 4, ขนาดของเฮดเดอร์คงที่ 20 ไบต์, ให้เราเตอร์ส่งผ่านไอพีดาตาแกรมบนเส้นทางที่มีการประวิงเวลาต่ำที่สุด, และอนุญาตให้ แบ่ง (fragment) แพ็กเกตได้ตามลำดับ ส่วนฟิลด์ total length นั้นไมโครคอนโทรลเลอร์จะคำนวณจากขนาดของข้อความ SIP รวมกับ UDP เฮดเดอร์ 8 ไบต์ และ IP เฮดเดอร์ 20 ไบต์ตามลำดับ หรือถ้าเป็นข้อมูลเสียงจะมีค่าเป็น 200 คงที่ตลอดซึ่งได้มาจากข้อมูลเสียงขนาด 160 ไบต์รวมกับ RTP เฮดเดอร์ 12 ไบต์, UDP เฮดเดอร์ 8 ไบต์, และ IP เฮดเดอร์ 20 ไบต์ตามลำดับ ในฟิลด์ identification ไมโครคอนโทรลเลอร์จะเพิ่มค่า

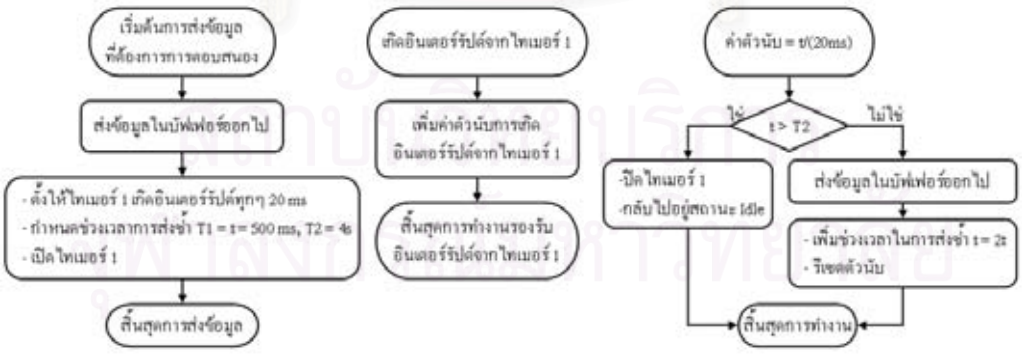
ในฟิลด์นี้ทุกครั้งที่ส่งไอพีแพ็กเกตออกไป ส่วนในฟิลด์ protocol ซึ่งใช้บ่งบอกโพรโทคอลในชั้นบนนั้น ไมโครคอนโทรลเลอร์จะกำหนดให้ฟิลด์นี้มีค่าเป็น 17 ในกรณีที่ข้อมูลที่ส่งเป็นข้อมูล SIP หรือข้อมูลเสียง ซึ่งมีโพรโทคอล UDP เป็นโพรโทคอลในชั้นเคลื่อนย้ายทั้งคู่ และกำหนดให้มีค่าเป็น 1 ในกรณีที่ข้อมูลที่ต้องการส่งเป็นข้อมูล ICMP ส่วนฟิลด์ Source IP และ Destination IP ไมโครคอนโทรลเลอร์จะนำค่าไอพีแอดเดรสของเครื่อง และไอพีแอดเดรสปลายทางเขียนลงในฟิลด์ตามลำดับ จากนั้นไมโครคอนโทรลเลอร์จะคำนวณ checksum จากIP เฮดเดอร์ที่สร้างขึ้น โดยนำข้อมูลภายในเฮดเดอร์มาบวกแบบ 1 complement 16 บิตจนครบทั้งเฮดเดอร์ แล้วค่อยนำค่าที่คำนวณได้นั้นมาใส่ในฟิลด์ checksum เป็นอันสิ้นสุดการสร้าง IP เฮดเดอร์

ตารางที่ 3.5 การสร้างฟิลด์ต่าง ๆ ใน IP เฮดเดอร์

ฟิลด์	ขนาด	ค่าที่เขียนลงในฟิลด์	หมายเหตุ
version	4 บิต	4	โพรโคคอล IP เวอร์ชัน 4
header length	4 บิต	5	IP เฮดเดอร์ขนาด 20 ไบต์
type of service	1 ไบต์	0x01	เพื่อการประวิงทางเวลาน้อยที่สุด
total length	2 ไบต์	ตามขนาดของ SIP, Voice	-
identification	2 ไบต์	เพิ่มขึ้นตามจำนวนแพ็กเกตที่ส่ง	-
flag	3 บิต	0	ให้มีการทำ Fragment ได้
fragment offset	13 บิต	0	-
time to live	1 ไบต์	64	-
protocol	1 ไบต์	ICMP=1, UDP=17	-
header checksum	2 ไบต์	ค่าที่ได้จากการคำนวณ checksum	-
source IP	4 ไบต์	ไอพีแอดเดรสของเครื่อง	-
destination IP	4 ไบต์	ไอพีแอดเดรสปลายทาง	-

การสร้างอีเทอร์เนตเฮดเดอร์ ในส่วนของฟิลด์ Destination address ไมโครคอนโทรลเลอร์จะตรวจสอบค่าไอพีแอดเดรสปลายทางซึ่งอยู่ในฟิลด์ Destination IP address ของไอพีเฮดเดอร์ว่าไอพีแอดเดรสของโฮสต์ปลายทางอยู่ในโครงข่ายพื้นที่ท้องถิ่นเดียวกับตัวเครื่องหรือไม่ ในกรณีที่เครื่องโฮสต์ปลายทางอยู่ในโครงข่ายเดียวกัน ไมโครคอนโทรลเลอร์จะใช้ฮาร์ดแวร์แอดเดรสของเครื่องปลายทางใส่ลงในฟิลด์ แต่ในกรณีที่อยู่คนละโครงข่าย ไมโครคอนโทรลเลอร์จะใช้ฮาร์ดแวร์แอดเดรสของเกตเวย์ใส่ลงในฟิลด์แทน ส่วนการสร้างฟิลด์ Source address ไมโครคอนโทรลเลอร์จะใช้ฮาร์ดแวร์แอดเดรสของตัวเครื่องใส่ลงในฟิลด์เท่านั้น และการสร้างฟิลด์ Type นั้นในกรณีที่ข้อมูลที่สร้างเป็น ARP ไมโครคอนโทรลเลอร์จะใช้ค่า 0x80 แต่กรณีที่ข้อมูลที่สร้างเป็น IP ไมโครคอนโทรลเลอร์จะใช้ค่า 0x86 ตามลำดับ

การส่งข้อมูลผ่านโครงข่ายอินเทอร์เนต ข้อมูลอาจสูญหายระหว่างทางได้ ดังนั้นเพื่อเพิ่มความเชื่อถือได้ให้กับข้อมูลที่ส่ง ในการส่งข้อมูลที่ต้องการการตอบสนอง (ข้อมูลร้องขอ ARP, ข้อความ INVITE, CANCEL, BYE, REGISTER, 200 OK, 486 Busy Here, และ 487 Request Canceled) จึงใช้อินเตอร์รัปต์จากไทเมอร์ 1 (Timer 1) จับเวลา และควบคุมการส่งซ้ำด้วยวิธี exponential backoff โดยสามารถแบ่งการควบคุมการส่งซ้ำออกเป็น 3 ส่วนได้แก่ส่วนเริ่มต้นการส่งซ้ำ, ส่วนรองรับการเกิดอินเตอร์รัปต์จากไทเมอร์ 1, และส่วนรองรับการเกิดไทเมอร์ expired ดังแสดงในรูปที่ 3.18 เมื่อมีการส่งข้อมูลที่ต้องการการตอบสนองออกไป ส่วนเริ่มต้นการส่งซ้ำจะตั้งค่าเริ่มต้นให้รีจิสเตอร์ TH1 และ TL1 เป็น 0x0F, 0xFF ตามลำดับ เพื่อกำหนดให้เกิดอินเตอร์รัปต์จากไทเมอร์ 1 ทุก ๆ 20 ms จากนั้นจึงเปิดไทเมอร์ 1 เพื่อเริ่มจับเวลา ในขณะที่ไทเมอร์ 1 ทำงาน ไมโครคอนโทรลเลอร์สามารถทำงานอื่นได้ตามปกติ แต่ทุกครั้งที่เกิดอินเตอร์รัปต์จากไทเมอร์ 1 ขึ้น ส่วนรองรับการเกิดอินเตอร์รัปต์จากไทเมอร์ 1 จะเพิ่มค่าตัวนับการเกิดอินเตอร์รัปต์ขึ้นทันที และเมื่อทำงานรองรับการเกิดอินเตอร์รัปต์เสร็จแล้ว ไมโครคอนโทรลเลอร์จะกลับไปทำงานอื่นตามปกติ รวมไปถึงการวนตรวจสอบตัวนับจำนวนการเกิดอินเตอร์รัปต์ โดยถ้าค่าตัวนับการเกิดอินเตอร์รัปต์มากกว่าค่าเปรียบเทียบตัวนับที่ตั้งไว้ซึ่งมีค่าเริ่มต้นเป็น 25 เพื่อให้ T1 = 500 ms ส่วนรองรับการเกิดไทเมอร์ expire จะส่งข้อมูลเดิมไปหาปลายทางอีกครั้งพร้อมกับรีเซตค่าตัวนับ และเพิ่มตัวเปรียบเทียบตัวนับเป็น 2 เท่าจากเดิม และถ้ายังไม่ได้รับการตอบสนองภายในช่วงเวลาที่ตั้งไว้ใหม่อีก ก็จะส่งซ้ำและเพิ่มตัวเปรียบเทียบตัวนับอย่างนี้ไปเรื่อย ๆ จนกว่าจะได้รับการตอบสนองหรือช่วงเวลาในการส่งซ้ำถูกเพิ่มจนมีค่ามากกว่า T2 = 4 s ซึ่งจะถือว่าไม่สามารถติดต่อกับปลายทางได้ ไมโครคอนโทรลเลอร์ถึงจะปิดไทเมอร์ 1 พร้อมกับยกเลิกการทำงานและกลับมาอยู่ในสถานะ Idle

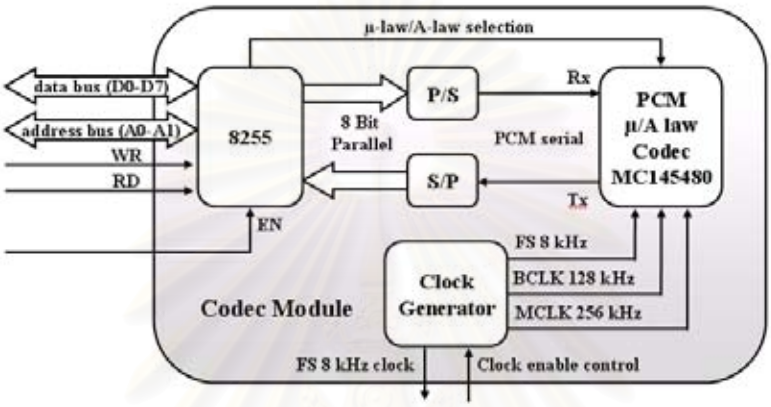


รูปที่ 3.18 การใช้ไทเมอร์ภายในไมโครคอนโทรลเลอร์ควบคุมการส่งซ้ำ

3.4 ส่วนควบคุมข้อมูลเสียง และติดต่อกับชิพเข้ารหัสเสียง PCM

3.4.1 โครงสร้างพื้นฐานของมอดูลเข้ารหัสเสียง

มอดูลเข้ารหัสสัญญาณเสียง PCM ประกอบด้วยวงจรกำเนิดสัญญาณนาฬิกา, ชิพเข้ารหัสสัญญาณเสียง PCM, วงจรแปลงการส่งข้อมูลอนุกรมเป็นขนาน (S/P), วงจรแปลงการส่งข้อมูลขนานเป็นอนุกรม (P/S), และชิพขยายพอร์ต 8255 ดังแสดงในรูปที่ 3.19



รูปที่ 3.19 โครงสร้างมอดูลเข้ารหัสสัญญาณเสียงแบบ PCM

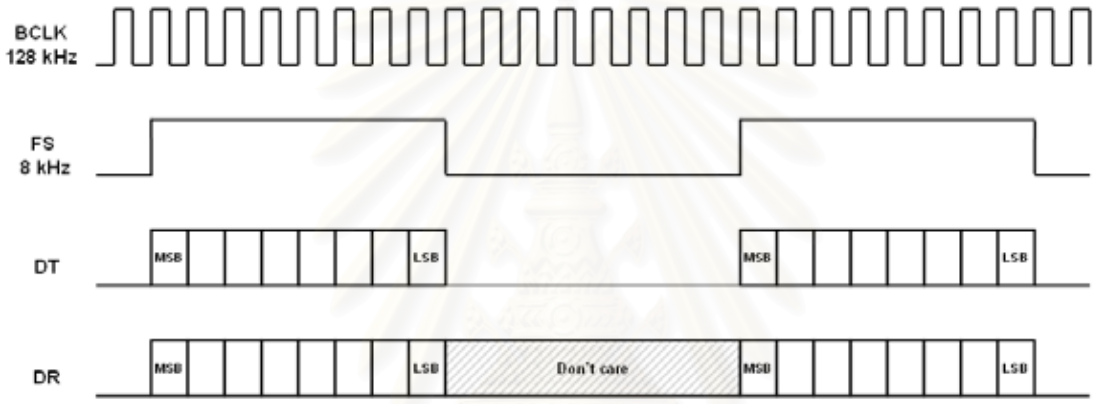
ชิพเข้ารหัสสัญญาณเสียง MC145480 ของบริษัท Motorola สามารถเข้ารหัสสัญญาณเสียง แบบ PCM ได้ทั้ง μ-law และ A-law ภายในชิพมีวงจรขยายสัญญาณแอนะลอก และฟิลเตอร์กรองสัญญาณความถี่ต่ำในตัวทำให้สามารถต่อแฮนด์เซตได้โดยตรง ในส่วนดิจิทัลตัวชิพสามารถส่ง และรับข้อมูลได้หลายรูปแบบขึ้นอยู่กับสัญญาณนาฬิกาที่ป้อนให้ มีรายละเอียดดังนี้

สัญญาณนาฬิกา Master Clock (MCLK) เป็นสัญญาณนาฬิกาหลักที่ป้อนเข้าในขา MCLK ของตัวชิพ เพื่อใช้ในกระบวนการ ADC (Analog to Digital Converter) และ DAC (Digital to Analog Converter) ในการใช้งานสามารถใช้สัญญาณนาฬิกาความถี่ 256, 512, 1536, 2048, 2560, หรือ 4096 kHz ก็ได้ ในกรณีที่ใช้สัญญาณนาฬิกาความถี่ 256 kHz และ 512 kHz นั้น สัญญาณนาฬิกา MCLK จำเป็นต้องซิงโครไนซ์กับสัญญาณนาฬิกา FS เพื่อให้ชิพทำงานได้อย่างมีประสิทธิภาพ ในการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตได้เลือกใช้สัญญาณนาฬิกาความถี่ 256 kHz ซึ่งเป็นสัญญาณความถี่ต่ำสุดที่ชิพสามารถทำงานได้ ทั้งนี้เพื่อหลีกเลี่ยงปัญหาจากสัญญาณความถี่สูง ซึ่งอาจทำให้วงจรไม่เสถียรได้

สัญญาณนาฬิกา Bit Clock (BCLK) เป็นสัญญาณนาฬิกาที่ใช้ควบคุมอัตราการรับส่งข้อมูล PCM ของตัวชิพ โดยในการใช้งานนั้นตัวชิพสามารถรับสัญญาณนาฬิกา BCLK ความถี่ตั้งแต่ 64 kHz จนถึง 4.096 MHzได้ แต่ในการพัฒนาไม่เลือกใช้สัญญาณนาฬิกา BCLK ความถี่ 64 kHz เนื่องจากในการรับข้อมูลเสียงจากไมโครคอนโทรลเลอร์มายังตัวชิพนั้น จำเป็นต้องผ่านกระบวนการแปลงข้อมูล

P/S (Parallel to Serial) ก่อน ซึ่งจากสัญญาณนาฬิกาที่มีในวงจรนั้น ในการทำ P/S ในขั้นตอนการโหลดข้อมูลแบบขนานเข้ามาในชิฟต์รีจิสเตอร์นั้น จำเป็นต้องใช้เวลาช่วงหนึ่งต่างหาก จากเวลาที่ใช้ในการส่งข้อมูลอนุกรมออกไป จึงทำให้ไม่สามารถสร้างสัญญาณ 64 kbps ที่มีอัตราข้อมูลต่อเนื่องมาให้กับชิพเข้ารหัสสัญญาณเสียงได้ ดังนั้นเพื่อให้เป็นการง่าย และไม่จำเป็นต้องเพิ่มอุปกรณ์จึงเลือกใช้สัญญาณนาฬิกา BCLK ที่ความถี่ 128 kHz แทน

สัญญาณนาฬิกา Frame Sync (FS) เป็นสัญญาณนาฬิกาความถี่ 8 kHz ใช้ในการซิงโครไนซ์เฟรมข้อมูลในแต่ละเฟรม โดยตัวชิพ MC145480 จะรับส่งเฟรมข้อมูล 8 บิตทุก ๆ คาบเมื่อสัญญาณนาฬิกา FS เป็นลอจิก ‘1’ ดังแสดงในรูปที่ 3.20

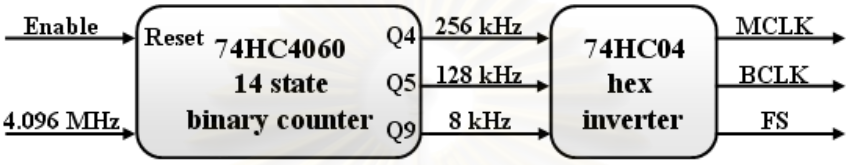


รูปที่ 3.20 ความสัมพันธ์ของการรับส่งข้อมูลกับสัญญาณนาฬิกาที่ป้อนให้ชิพ MC145480

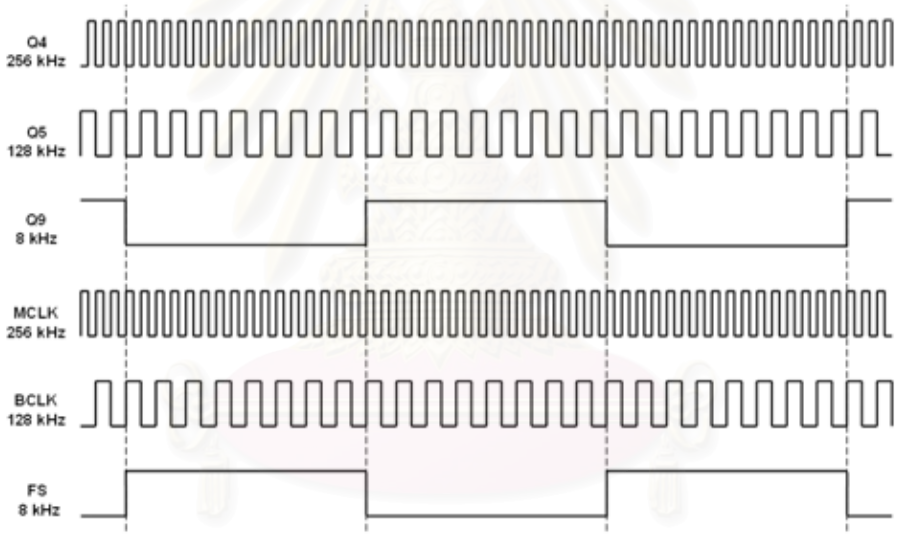
เนื่องจากข้อจำกัดทางด้านความเร็วในการทำงานของไมโครคอนโทรลเลอร์ จึงทำให้ไมโครคอนโทรลเลอร์ไม่สามารถรับส่งข้อมูลแบบอนุกรมกับชิพ MC145480 ได้โดยตรง ดังนั้นจึงต้องใช้เทคนิคการแปลง S/P และ P/S เพื่อให้ไมโครคอนโทรลเลอร์รับส่งข้อมูลแบบขนาน 8 บิตต่อทิศทางแทน ซึ่งต้องใช้พอร์ตถึง 16 พอร์ตแยกต่างหากจากสายสัญญาณข้อมูล, สายสัญญาณกำหนดตำแหน่งหน่วยความจำ, และสายสัญญาณควบคุม ซึ่งพอร์ตของไมโครคอนโทรลเลอร์นั้นไม่เพียงพอ จึงต้องใช้ชิพ 8255 ซึ่งเป็นชิพขยายพอร์ต I/O เข้ามาเพิ่มด้วย

วงจรกำเนิดสัญญาณนาฬิกาภายในมอดูลเข้ารหัสสัญญาณเสียงทำหน้าที่สร้างสัญญาณนาฬิกาความถี่ 8, 128, และ 256 kHz ใช้ในการทำงานของชิพเข้ารหัสสัญญาณเสียง, กระบวนการแปลง S/P, กระบวนการแปลง P/S, และซิงโครไนซ์เพื่อรับส่งข้อมูลกับไมโครคอนโทรลเลอร์ โดยในการสร้างสัญญาณนาฬิกาได้ใช้ชิพ 74HC4060 ซึ่งเป็นชิพวงจรนับแบบไบนารี 14 บิต (14 state binary counter) ภายในชิพมีวงจรออสซิลเลเตอร์ในตัว ซึ่งทำให้สามารถต่อกับคริสตอลได้โดยตรง โดยในการพัฒนาได้เลือกใช้คริสตอลความถี่ 4.096 MHz เพื่อให้สามารถหารความถี่ที่ต้องการได้อย่างลงตัว โดยจะได้

สัญญาณนาฬิกา 256 kHz จากเอาต์พุต Q4 (หาร 16), สัญญาณนาฬิกา 128 kHz จากเอาต์พุต Q5 (หาร 32), และสัญญาณนาฬิกา 8 kHz จากเอาต์พุต Q9 (หาร 512) ตามลำดับ แต่เนื่องจากสัญญาณนาฬิกาที่ได้มีการซิงโครไนซ์ที่ขอบขาลงของสัญญาณนาฬิกาที่มีความถี่สูงกว่า แต่ชิพเข้ารหัสเสียงต้องใช้สัญญาณนาฬิกาที่มีการซิงโครไนซ์ ช่วงขอบขาขึ้น ดังนั้นจึงต้องนำสัญญาณที่ได้มาผ่านชิพ 74HC04 ซึ่งเป็นอินเวอร์เตอร์กลับบิตสัญญาณดังแสดงในรูปที่ 3.21 เพื่อให้ได้สัญญาณนาฬิกาที่มีรูปแบบการซิงโครไนซ์ที่ถูกต้องดังแสดงในรูปที่ 3.22



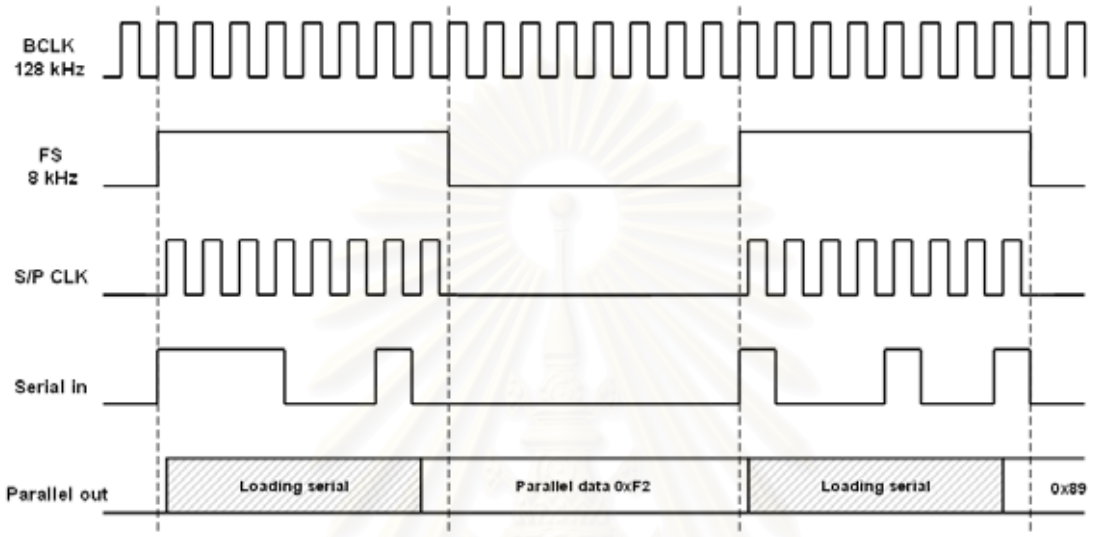
รูปที่ 3.21 การสร้างสัญญาณนาฬิกาเพื่อใช้กับชิพเข้ารหัสเสียง



รูปที่ 3.22 สัญญาณนาฬิกาที่ได้จากวงจรสร้างสัญญาณนาฬิกา

การแปลง S/P ในการพัฒนาได้ใช้ชิพ 74HC164 ตัวชิพประกอบด้วยพอร์ตอินพุตแบบอนุกรม, พอร์ตเอาต์พุตแบบขนาน 8 บิต, พอร์ตรับสัญญาณนาฬิกา, และพอร์ตรับสัญญาณควบคุม ในการใช้งานต้องป้อนสัญญาณนาฬิกาเพื่อควบคุมอัตราการรับข้อมูลแบบอนุกรม โดยตัวชิพจะเก็บข้อมูลที่ได้รับจากพอร์ตอินพุตอนุกรมเข้ามาเก็บในชิฟต์รีจิสเตอร์พร้อมกับเลื่อนข้อมูลเก่าไปยังรีจิสเตอร์ถัดไปครั้งละบิต ทุก ๆ ขอบขาขึ้นของสัญญาณนาฬิกา ดังนั้นในการแปลงข้อมูลอนุกรมที่ได้จากชิพเข้ารหัสเสียงให้อยู่ในรูปของข้อมูลขนานนั้น จึงต้องป้อนสัญญาณนาฬิกาความถี่ 128 kHz ซึ่งเท่ากับความถี่ของสัญญาณนาฬิกา BCLK เพื่อให้ชิพ 74HC164 รับข้อมูลอนุกรมเข้ามาเก็บในชิฟต์รีจิสเตอร์ครั้งละบิตได้อย่างถูกต้อง และเพื่อเป็นการสะดวกในการอ่านค่าข้อมูลแบบขนาน 8 บิตจากชิพ 74HC164 ผ่านชิพ

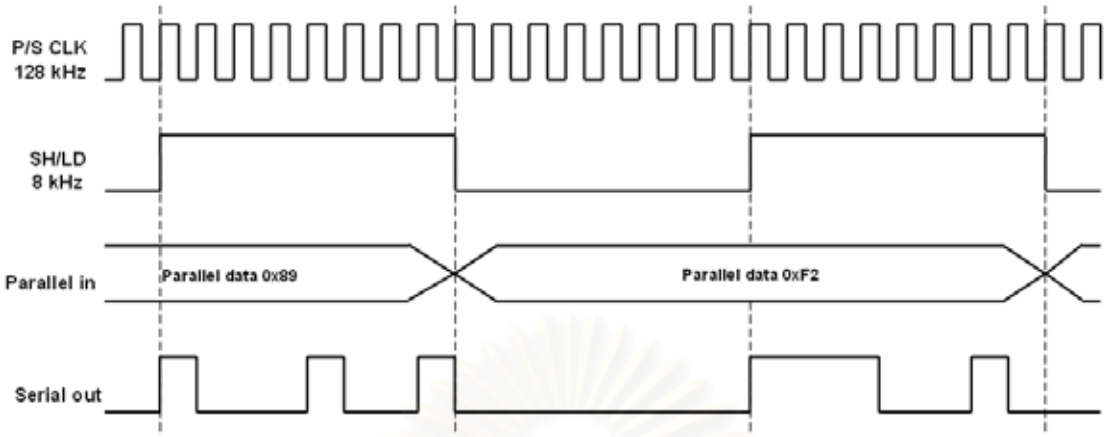
8255 จึงจ่ายสัญญาณนาฬิกาให้กับชิพ 74HC164 เฉพาะช่วงที่สัญญาณ FS มีลอจิกเป็น ‘1’ เพื่อให้ตัวชิพ 74HC164 นั้นค้างค่าข้อมูลออกทางพอร์ตขนานในช่วงเวลาที่สัญญาณ FS มีลอจิกเป็น ‘0’ ไว้ดังแสดงในรูปที่ 3.23 ซึ่งสัญญาณนาฬิกา S/P CLK สามารถสร้างได้จากการนำสัญญาณ BCLK มาแอนด์ (AND) กับสัญญาณ FS นั่นเอง



รูปที่ 3.23 สัญญาณบนชิพ S/P 74HC164

การแปลง P/S ในการพัฒนาได้ใช้ชิพ 74HC165 ตัวชิพประกอบด้วยพอร์ตอินพุตแบบขนาน 8 บิต, พอร์ตเอาต์พุตอนุกรม, พอร์ตรับสัญญาณนาฬิกา, และพอร์ตรับสัญญาณควบคุม ในการใช้เมื่อระดับสัญญาณที่พอร์ต SH/LD (Shift/Load) มีลอจิกเป็น ‘0’ ชิพจะอ่านค่าจากพอร์ตขนาน 8 บิตเข้ามาเก็บในชิฟต์รีจิสเตอร์ และเมื่อสัญญาณ SH/LD มีระดับลอจิกเป็น ‘1’ ชิพจะส่งข้อมูลในชิฟต์รีจิสเตอร์นั้นออกไปทางพอร์ตอนุกรมครั้งละบิต ทุก ๆ ขอบขาขึ้นของสัญญาณนาฬิกาที่ป้อนให้กับชิพนั้น ดังนั้นในการใช้งานเพื่อให้ชิพ 74HC165 ส่งข้อมูลให้กับชิพเข้ารหัสเสียงได้อย่างถูกต้อง จึงต้องใช้สัญญาณนาฬิกา BCLK ความถี่ 128 kHz ป้อนเป็นสัญญาณนาฬิกาให้กับตัวชิพ 74HC164 และใช้สัญญาณ FS 8 kHz เป็นสัญญาณควบคุม SH/LD ดังแสดงในรูปที่ 3.24

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.24 สัญญาณบนชิพ P/S 74HC165

ชิพ 8255 เป็นชิพที่ใช้ในการขยายพอร์ต I/O โดยตัวชิพจะมีพอร์ต I/O ขนาด 8 บิต 3 พอร์ต สามารถต่อกับไมโครคอนโทรลเลอร์แบบ Memory map ได้ ซึ่งในการพัฒนาได้ทำ Memory map ให้ชิพ 8255 อยู่ที่ตำแหน่ง C000H-C003H และมีรายละเอียดดังแสดงในตารางที่ 3.6

ตารางที่ 3.6 ตำแหน่งการทำ Memory map บนชิพ 8255 และการใช้งาน

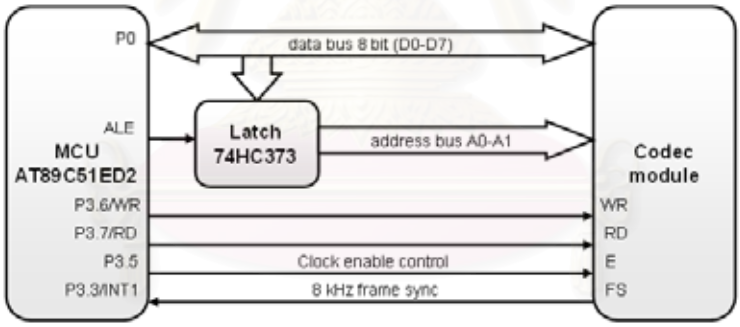
A1	A0	ตำแหน่ง (A14 = 1, A15 = 1)	การใช้งาน
0	0	C000H	พอร์ต A
0	1	C001H	พอร์ต B
1	0	C002H	พอร์ต C
1	1	C003H	รีจิสเตอร์ควบคุมการทำงานของ 8255

ชิพ 8255 นั้นสามารถทำงานได้ 3 โมด โดยสามารถเลือกโมดการทำงานได้จากการตั้งค่าในรีจิสเตอร์ควบคุมการทำงานภายในชิพ 8255 ดังแสดงในตารางที่ 3.7 ในโมด 0 ทุกพอร์ตของ 8255 เป็นอินพุตหรือเอาต์พุตโดยต้องกำหนดว่าพอร์ตไหนเป็นอินพุต หรือเอาต์พุต และไม่สามารถกำหนดให้พอร์ตหนึ่งเป็นทั้งอินพุต และเอาต์พุตในเวลาเดียวกันได้ ในโมด 2 พอร์ต A และพอร์ต B เป็นอินพุตหรือเอาต์พุตโดยใช้ไบต์ต่ำของพอร์ต C และไบต์สูงของพอร์ต C เป็น Handshaking ให้กับพอร์ต A และพอร์ต B ตามลำดับ ส่วนในโมด 2 พอร์ต A เป็นทั้งอินพุต และเอาต์พุตแบบสองทิศทาง โดยใช้พอร์ต B กับพอร์ต C ในการตรวจสอบการทำงาน ซึ่งในการพัฒนานั้นได้กำหนดค่าภายในรีจิสเตอร์ควบคุมเป็น 0x82 นั่นคือให้ชิพ 8255 ทำงานในโมด 0 เพื่อใช้งานทั้ง 3 พอร์ตโดยให้พอร์ต A เป็นเอาต์พุตเพื่อส่งข้อมูลเสียง PCM แบบขนาน 8 บิตไปยังวงจร P/S และพอร์ต B เป็นเอาต์พุตเพื่อรับข้อมูลเสียง PCM แบบขนาน 8 บิตจากวงจร S/P ตามลำดับ ส่วนพอร์ต C บิต 0 ต่อกับพอร์ตเลือก

รูปแบบการเข้ารหัสของชิพ MC145480 โดยถ้าเป็นลอจิก ‘1’ MC145480 จะเข้ารหัสแบบ μ-law และถ้าเป็นลอจิก ‘0’ จะเข้ารหัสแบบ A-law ส่วนบิต 1 ของพอร์ต C ใช้สำหรับควบคุม Buzzer เพื่อสร้างเสียงกระดิ่งในกรณีที่มีการร้องขอการเปิดเซสชันจากเครื่องลูกข่ายอื่น

ตารางที่ 3.7 การกำหนดค่ารีจิสเตอร์ควบคุมเพื่อควบคุมการทำงานของ 8255

ตำแหน่งบิต	การทำงาน
0	กำหนดการทำงานของพอร์ต C ล่างโดย ‘0’ = เอาต์พุต และ ‘1’ = อินพุต
1	กำหนดการทำงานของพอร์ต B โดย ‘0’ = เอาต์พุต และ ‘1’ = อินพุต
2	กำหนดโมดการทำงานของพอร์ต B โดย ‘0’ = โมด 0 และ ‘1’ = โมด 1
3	กำหนดการทำงานของพอร์ต C บนโดย ‘0’ = เอาต์พุต และ ‘1’ = อินพุต
4	กำหนดการทำงานของพอร์ต A โดย ‘0’ = เอาต์พุต และ ‘1’ = อินพุต
5	ใช้กำหนดโมดการทำงานของพอร์ต A โดย
6	“00” = โมด 0, “01” = โมด 1, “10”, หรือ “11” = โมด 2
7	กำหนดการทำงานของพอร์ตทุกพอร์ตโดย ‘0’ = ไม่ทำงาน และ ‘1’ = ทำงาน



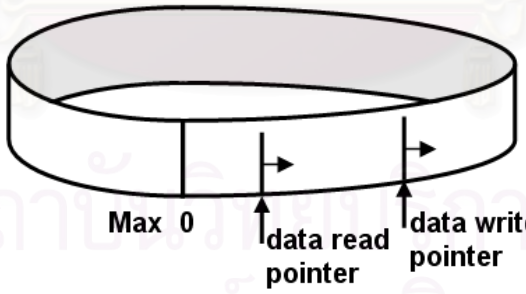
รูปที่ 3.25 การต่อไมโครคอนโทรลเลอร์เข้ากับมอดูลเข้ารหัสสัญญาณเสียง

ในการต่อกับมอดูลเข้ารหัสสัญญาณเสียงนั้น นอกจากใช้สายสัญญาณข้อมูล, สายสัญญาณกำหนดตำแหน่งหน่วยความจำ, และสายสัญญาณควบคุมแล้ว ยังใช้พอร์ต 3.5 ควบคุมการเปิดปิดวงจรสร้างสัญญาณนาฬิกา และพอร์ต 3.3 รับสัญญาณซิงโครไนซ์เฟรมจากมอดูลเข้ารหัสสัญญาณเสียงด้วย ดังแสดงในรูปที่ 3.25

3.4.2 การควบคุมข้อมูลเสียงภายในบัฟเฟอร์ภาคส่ง และภาครับ

ในการรับส่งข้อมูลเสียงภายในเครื่องโทรศัพท์นั้น มอดูลเข้ารหัสเสียงจะไม่ได้รับส่งข้อมูลกับชิพอีเทอร์เนตโดยตรง ในภาคส่งไมโครคอนโทรลเลอร์จะนำข้อมูลเสียงที่ได้จากมอดูลเข้ารหัสสัญญาณเสียงมาเก็บภายในหน่วยความจำภายนอกเพื่อรอให้ครบ 160 ไบต์ก่อน จากนั้นจึงนำไปเข้าแพ็กเกต UDP/IP แล้วจึงส่งไปให้ชิพอีเทอร์เนต และเช่นเดียวกันในภาครับ ไมโครคอนโทรลเลอร์จะนำข้อมูลเสียงที่ได้จากชิพอีเทอร์เนตมาเก็บในบัฟเฟอร์ภายในหน่วยความจำภายนอกเพื่อลดผลของ Jitter ก่อนแล้วจึงค่อยส่งข้อมูลเสียงนั้นไปให้มอดูลถอดรหัสเสียงต่อไป ดังนั้นการควบคุมการรับส่งข้อมูลเสียงภายในเครื่องโทรศัพท์สามารถแบ่งออกเป็น 2 ส่วนได้แก่ การรับส่งข้อมูลเสียงระหว่างชิพอีเทอร์เนต RTL8019as กับบัฟเฟอร์หน่วยความจำข้อมูลภายนอก และการรับส่งข้อมูลเสียงระหว่างหน่วยความจำข้อมูลภายนอกกับมอดูลเข้ารหัสเสียง

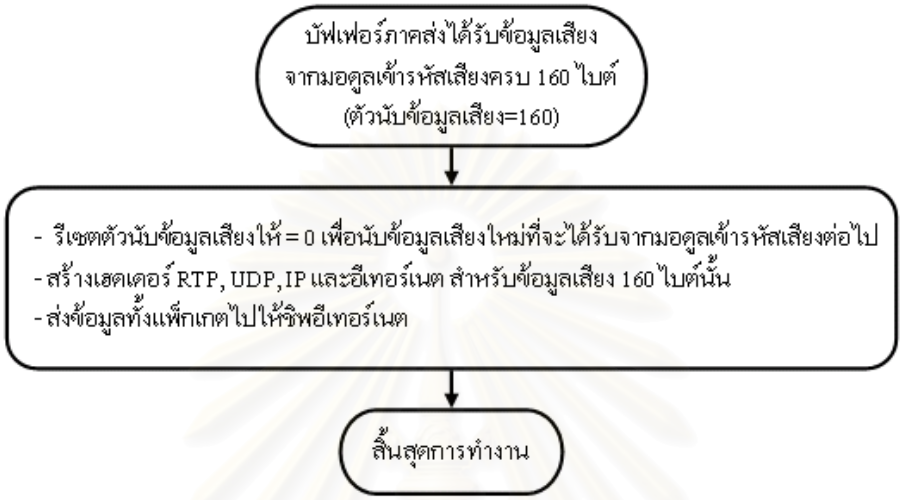
เนื่องจากข้อมูลเสียงนั้นเป็นข้อมูลที่มีการรับส่งอย่างต่อเนื่อง ในการพัฒนาจึงได้ใช้การบัฟเฟอร์แบบเป็นวง เพื่อให้การควบคุมการอ่าน และการเขียนข้อมูลเสียงภายในบัฟเฟอร์ทำได้สะดวกและต่อเนื่อง ในการเขียนข้อมูลไมโครคอนโทรลเลอร์จะเขียนข้อมูลลงบัฟเฟอร์พร้อมกับปรับค่าตัวชี้ตำแหน่งที่ได้เขียนไล่ไปเรื่อย ๆ จนเต็ม แล้ววนกลับมาที่ตำแหน่งเริ่มต้นใหม่อีกครั้งซ้ำดังนี้ไปเรื่อย ๆ ในการอ่านก็เช่นกัน ไมโครคอนโทรลเลอร์จะอ่านข้อมูลจากบัฟเฟอร์พร้อมกับปรับค่าตัวชี้ตำแหน่งที่ได้อ่านไล่ตามการเขียนวนไปเรื่อยดังแสดงในรูปที่ 3.26 ในการพัฒนาได้ใช้บัฟเฟอร์รับข้อมูลเสียงขนาด 640 ไบต์ และบัฟเฟอร์ส่งข้อมูลเสียงขนาด 320 ไบต์ตามลำดับ



รูปที่ 3.26 การบัฟเฟอร์ข้อมูลเสียงภายในหน่วยความจำภายนอก

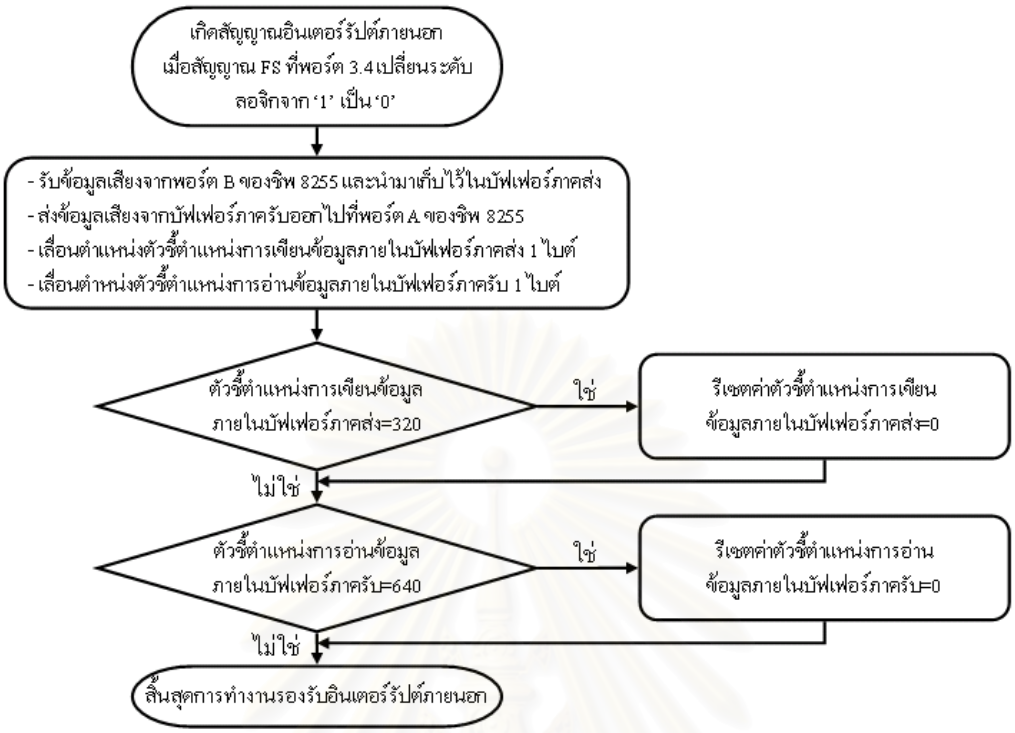
ในการควบคุมการรับส่งข้อมูลเสียงระหว่างหน่วยความจำภายนอก กับชิพอีเทอร์เนต ในภาคส่งเมื่อไมโครคอนโทรลเลอร์นำข้อมูลจากมอดูลเข้ารหัสเสียง มาเก็บในบัฟเฟอร์หน่วยความจำข้อมูลภายนอกจนครบทุก ๆ 160 ไบต์ ไมโครคอนโทรลเลอร์จะสร้าง RTP เฮดเดอร์ และ UDP/IP เฮดเดอร์สำหรับข้อมูลเสียงนั้นแล้วจึงค่อยส่งข้อมูลทั้งแพ็กเกตนั้นต่อไปยังชิพอีเทอร์เนตดังแสดงในรูปที่ 3.27 ส่วนในภาครับเมื่อได้รับข้อมูลจากชิพอีเทอร์เนต ไมโครคอนโทรลเลอร์จะตรวจสอบฟิลด์

destination port ภายใน UDP เฮดเดอร์ โดยถ้ามีค่าเป็น 49170 ซึ่งเป็นพอร์ตที่มีการตกลงไว้ก่อนเปิดเซสชันให้เป็นพอร์ตสำหรับรับข้อมูลเสียง ไมโครคอนโทรลเลอร์จะนำข้อมูลเสียงภายในแพ็กเกตนั้นเขียนลงในบัฟเฟอร์ภาครับ เพื่อรอการอ่านไปให้ชุดมอดูลเข้ารหัสเสียงต่อไป



รูปที่ 3.27 การควบคุมการส่งข้อมูลเสียงให้ชิพอีเทอร์เนต

ในการควบคุมการรับส่งข้อมูลระหว่างหน่วยความจำภายนอก กับมอดูลเข้ารหัสเสียงเนื่องจากการทำงานภายในมอดูลเข้ารหัสเสียงเป็นแบบซิงโครไนซ์กันทั้งหมด ดังนั้นไมโครคอนโทรลเลอร์จึงต้องควบคุมการรับส่งข้อมูลระหว่างหน่วยความจำภายนอก กับชิพ 8255 เป็นจังหวะซิงโครไนซ์กับตัวมอดูลด้วย ซึ่งในการซิงโครไนซ์นี้ ไมโครคอนโทรลเลอร์จะใช้ขอบขาลงของสัญญาณนาฬิกา FS เป็นสัญญาณอินเตอร์รัปต์จากภายนอก เพื่อควบคุมจังหวะการอ่าน และเขียนข้อมูลลงพอร์ต B และ A ของชิพ 8255 ตามลำดับ ทำให้ในภาคส่งข้อมูลเสียงที่ได้รับจากพอร์ต B ของชิพ 8255 จะถูกเขียนลงในวงบัฟเฟอร์ขาส่งครั้งละไบต์ อย่างสม่ำเสมอด้วยอัตราข้อมูล 8000 ไบต์/วินาที ส่วนในภาครับข้อมูลเสียงจะถูกอ่านจากวงบัฟเฟอร์ขารับ เพื่อส่งออกไปที่พอร์ต A ของชิพ 8255 ครั้งละไบต์ อย่างสม่ำเสมอด้วยอัตราข้อมูล 8000 ไบต์/วินาทีเช่นกันดังแสดงในรูปที่ 3.28

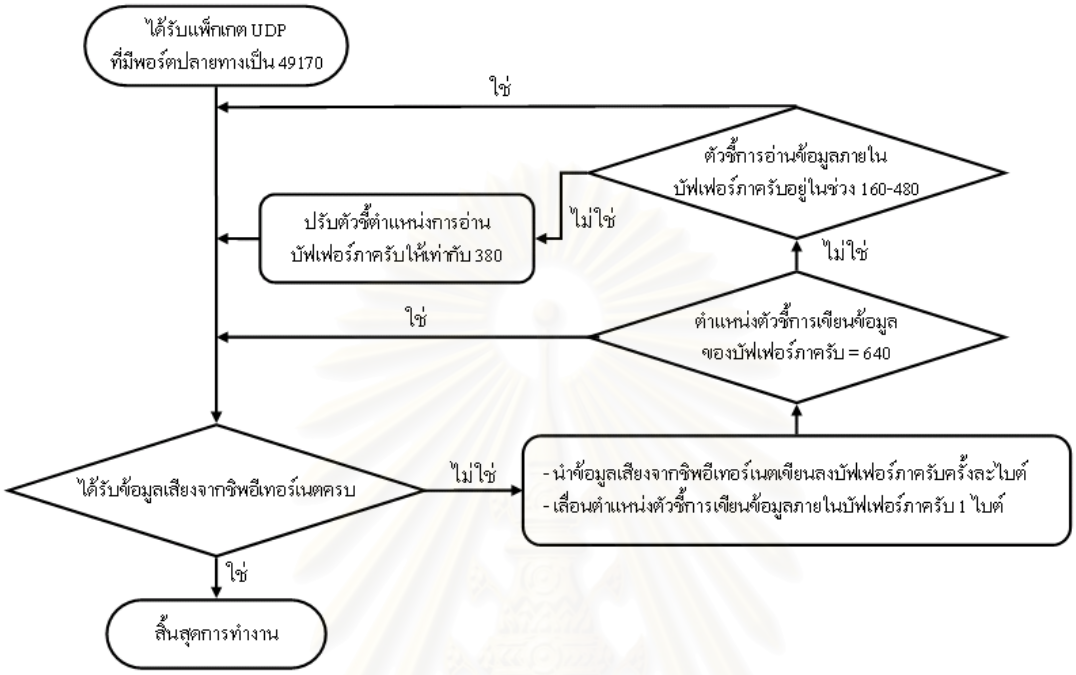


รูปที่ 3.28 การควบคุมการรับส่งข้อมูลระหว่างหน่วยความจำ กับมอดูลเข้ารหัสเสียง

เนื่องจากการส่งข้อมูลเสียงผ่านโครงข่ายอินเทอร์เนตนั้น แพ็กเกตแต่ละแพ็กเกตจะใช้เวลาในการเดินทางผ่านโครงข่ายไม่เท่ากัน ทำให้ข้อมูลเสียงที่ได้รับที่ปลายทางมี Jitter และเกิดเสียงที่เล่นออกไปแบบไม่ต่อเนื่อง ในทางปฏิบัติสามารถลด Jitter ที่เกิดขึ้นนี้ได้โดยการบัฟเฟอร์ข้อมูลที่ได้รับมาในช่วงเวลาหนึ่งก่อน แล้วจึงค่อยเล่นข้อมูลนั้นออกไป โดยถ้ายิ่งมีบัฟเฟอร์ข้อมูลมากขึ้นเพียงไรก็ยิ่งสามารถลด Jitter ได้มากขึ้นเพียงนั้น แต่จะมีผลเสียคือเกิดการประวิงทางเวลาเกิดขึ้น ในการพัฒนาเครื่องโทรศัพท์ ไมโครคอนโทรลเลอร์จะบัฟเฟอร์ข้อมูลที่ได้รับจากชิพอีเทอร์เนตก่อน 320 ไบต์แล้วจึงค่อยส่งข้อมูลนั้นออกไปให้มอดูลเข้ารหัสสัญญาณเสียง ซึ่งในการบัฟเฟอร์ข้อมูลนี้ทำให้เกิดการประวิงเวลาขึ้น 40 ms

ในการรับส่งข้อมูลแบบเวลาจริง นอกจากปัญหา Jitter แล้ว ยังเกิดปัญหาที่เกิดจากการขยับเลื่อนของสัญญาณนาฬิกา (Clock drift) ระหว่างต้นทาง กับปลายทาง ซึ่งทำให้อัตราข้อมูลเสียงที่ได้รับกับอัตราการเล่นข้อมูลเสียงไม่เท่ากัน มีผลให้การอ่าน และการเขียนข้อมูลในวงบัฟเฟอร์เกิดการซ้อนทับกันได้ ซึ่งในการพัฒนาได้แก้ปัญหาโดยเมื่อได้รับข้อมูลเสียงจากชิพอีเทอร์เนตจนครบรอบวงบัฟเฟอร์ทุกรอบ ไมโครคอนโทรลเลอร์จะตรวจสอบตำแหน่งตัวชี้ข้อมูลเสียงที่ได้อ่านไปว่ายังอยู่ในช่วงที่ยอมรับได้ (160-480) หรือไม่ ถ้าตัวชี้ตำแหน่งข้อมูลเสียงที่ได้อ่านไปไม่อยู่ในช่วงที่ยอมรับได้แสดงว่าเกิดการขยับเลื่อนของสัญญาณนาฬิกามาก และอาจเกิดการซ้อนทับของข้อมูลภายในวง

บัฟเฟอร์ ไมโครคอนโทรลเลอร์จะปรับตัวชี้ข้อมูลที่อ่านให้อยู่ในตำแหน่งที่ไบต์ที่ 320 ซึ่งเป็นตำแหน่งที่ถูกต้องอีกครั้งดังแสดงในรูปที่ 3.29

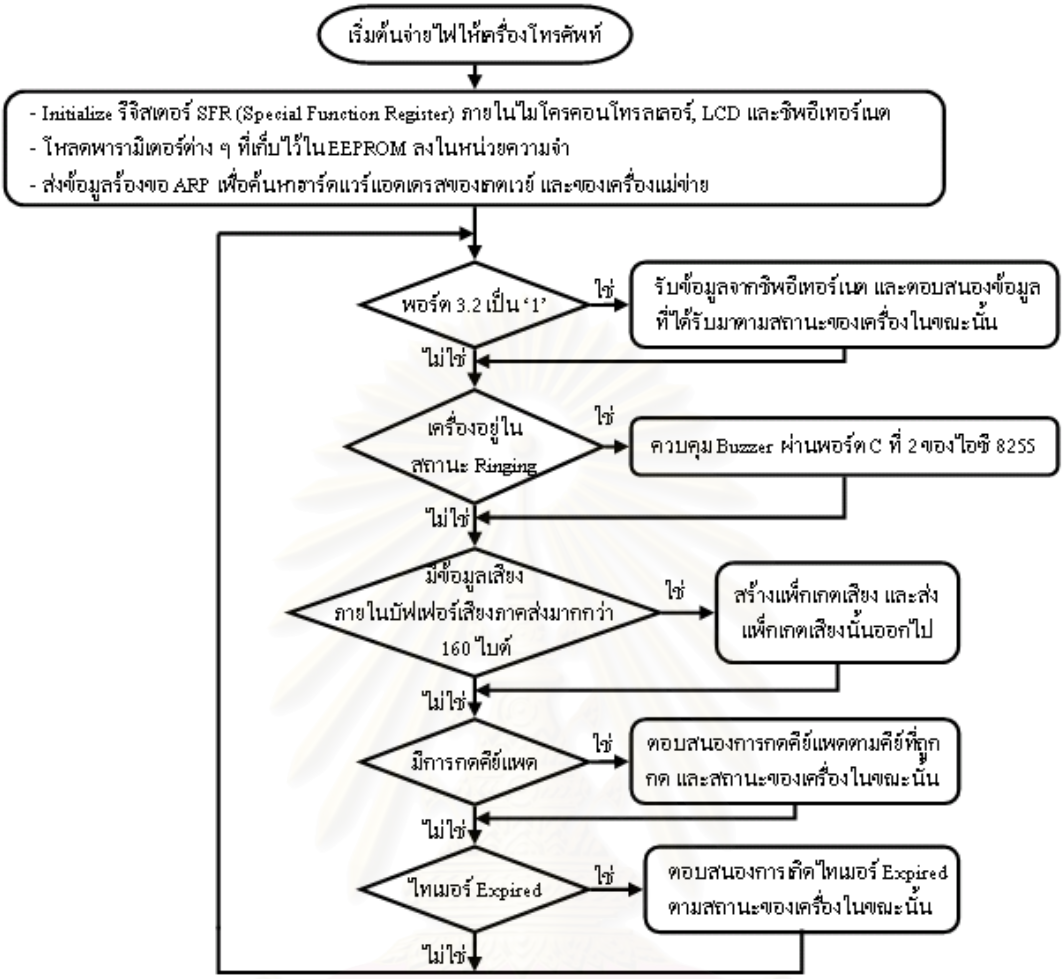


รูปที่ 3.29 การควบคุมการรับข้อมูลเสียงจากชิพอีเทอร์เนต

3.5 การควบคุมการทำงานโดยรวม

ในการควบคุมโดยรวมไมโครคอนโทรลเลอร์จะวนตรวจสอบ อุปกรณ์ภายนอก และตัวแปรที่ใช้ระบุสถานการณ์การทำงานของเครื่องโทรศัพท์ พร้อมกับตอบสนองผลที่ได้จากการตรวจสอบนั้น ๆ ไปเรื่อย ๆ และเมื่อเกิดอินเตอร์รัปต์จากภายนอก หรืออินเตอร์รัปต์จากไทเมอร์ ไมโครคอนโทรลเลอร์จะไปทำงานรองรับการเกิดอินเตอร์รัปต์นั้นก่อนจนเสร็จ แล้วจึงกลับมาวนตรวจสอบอุปกรณ์ภายนอกและตัวแปรที่ใช้ระบุการทำงานต่อไป

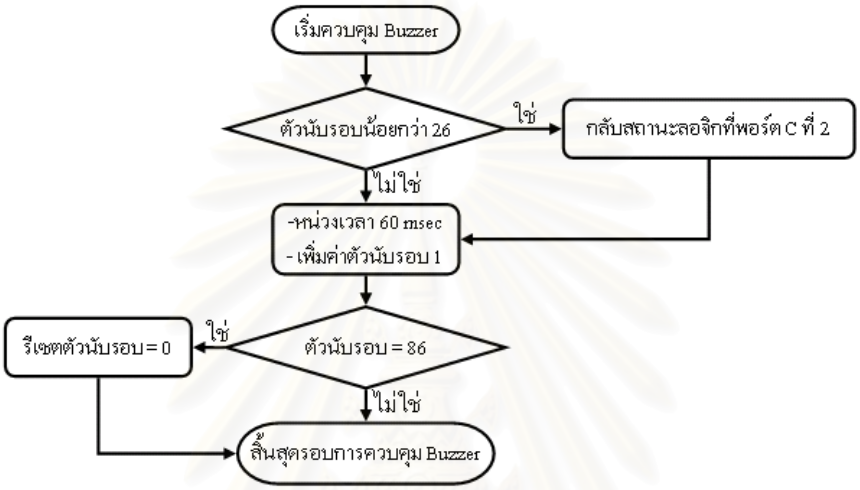
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



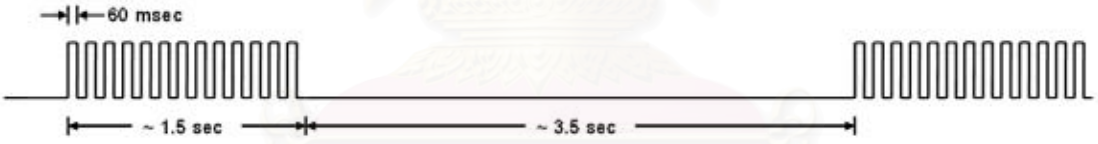
รูปที่ 3.30 การควบคุมการทำงานโดยรวม

เมื่อเริ่มจ่ายไฟให้กับเครื่องโทรศัพท์ดังแสดงในรูปที่ 3.30 ไมโครคอนโทรลเลอร์จะ Initialize รีจิสเตอร์ SFR, LCD, และชิพอีเทอร์เนตให้พร้อมใช้งาน จากนั้นจึงโหลดพารามิเตอร์ ต่าง ๆ ที่เก็บไว้ใน EEPROM ลงในหน่วยความจำ แล้วค่อยส่งข้อมูลร้องขอ ARP เพื่อค้นหาฮาร์แวร์แอดเดรสของเกตเวย์ และของเครื่องแม่ข่าย เมื่อ Initialize ฮาร์ดแวร์ และโหลดข้อมูลเรียบร้อยแล้วไมโครคอนโทรลเลอร์จะตรวจสอบพอร์ต 3.2 ซึ่งต่ออยู่กับขาสัญญาณอินเตอร์รัปต์ของชิพอีเทอร์เนตนั้น เพื่อตรวจจับการรับข้อมูลจากภายนอกผ่านชิพอีเทอร์เนต โดยถ้าพอร์ต 3.2 มีลอจิกเป็น '1' ไมโครคอนโทรลเลอร์จะรับข้อมูลจากชิพอีเทอร์เนต และนำข้อมูลที่ได้รับมาจำแนก และประมวลผลเพื่อตอบสนองข้อมูลนั้นต่อไป จากนั้นไมโครคอนโทรลเลอร์จะตรวจสอบสถานะของเครื่อง เพื่อควบคุมพอร์ต C ที่ 2 ของชิพ 8255 ซึ่งต่อเข้ากับ Buzzer และใช้เป็นกระดิ่งสำหรับเตือนผู้ใช้งานเมื่อมีสายเรียกเข้า โดยในกรณีที่เครื่องอยู่ในสถานะ Ringing ดังแสดงในรูปที่ 3.31 ไมโครคอนโทรลเลอร์จะควบคุม Buzzer ให้เป็นเสียงกระดิ่ง

ดัง 1.5 วินาที หยุด 3.5 วินาทีสลับกันไปเรื่อย ๆ โดยเสียงกระดิ่งที่ดังแต่ละครั้งจะประกอบไปด้วยเสียงดัง และหยุดสลับกันไปทุก ๆ 60 msec ดังแสดงในรูปที่ 3.32 ในการควบคุมแต่ละครั้ง ไมโครคอนโทรลเลอร์จะใช้เวลาเพียงรอบละ 60 msec ควบคุม Buzzer ซึ่งเท่ากับช่วงเวลาเสียงดังหยุดสลับกัน แล้วไปตรวจสอบอย่างอื่นต่อทั้งนี้เนื่องจากเมื่อเครื่องอยู่ในสถานะ Ringing เครื่องโทรศัพท์อาจได้รับข้อมูลจากชิพอีเทอร์เนต หรือผู้ใช้งานอาจกดคีย์แพดซึ่งมีผลต่อการควบคุมเซสชันได้



รูปที่ 3.31 การควบคุม Buzzer



รูปที่ 3.32 สัญญาณควบคุม Buzzer เพื่อใช้เป็นกระดิ่งเตือนผู้ใช้งานเมื่อมีสายเข้า

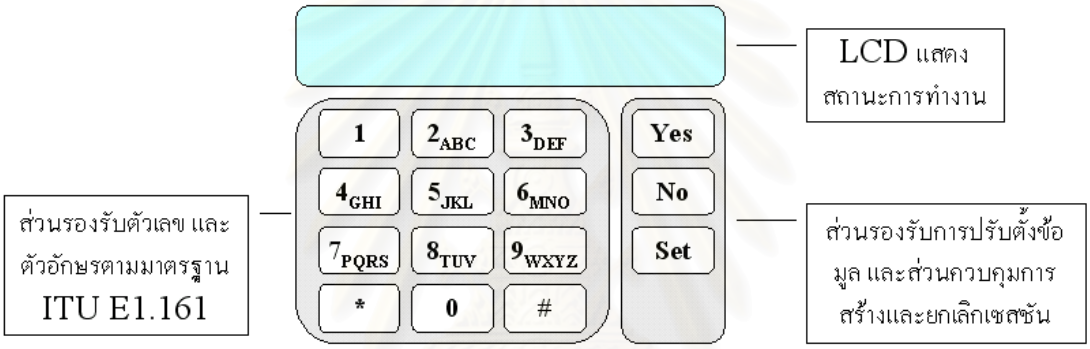
เมื่อตรวจสอบสถานะของเครื่องเรียบร้อยแล้ว ไมโครคอนโทรลเลอร์จะตรวจสอบจำนวนข้อมูลเสียงภายในบัฟเฟอร์ข้อมูลเสียงภาคส่ง ในกรณีที่มีการเขียนข้อมูลเสียงใหม่ครบ 160 ไบต์ ไมโครคอนโทรลเลอร์จะย้ายข้อมูลเสียงทั้ง 160 ไบต์จากบัฟเฟอร์ข้อมูลเสียงมาใส่ลงในบัฟเฟอร์ภาคส่ง แล้วเพิ่มเฮดเดอร์ RTP, UDP, IP, และ Ethernet แล้วค่อยส่งข้อมูลไปให้ชิพอีเทอร์เนต และควบคุมให้ชิพอีเทอร์เนตส่งข้อมูลนั้นออกไป จากนั้นไมโครคอนโทรลเลอร์จะตรวจสอบคีย์แพด โดยถ้ามีการกดคีย์ ไมโครคอนโทรลเลอร์จะตอบสนองการกดคีย์นั้นตามคีย์ที่ได้ถูกกด และสถานะของเครื่องในขณะนั้น เมื่อตรวจสอบการกดคีย์เรียบร้อยแล้วไมโครคอนโทรลเลอร์จะตรวจสอบตัวแปรซึ่งใช้นับจำนวนการเกิดอินเตอร์รัปต์จากไทเมอร์ 1 ซึ่งใช้เป็นไทเมอร์สำหรับการส่งซ้ำ โดยถ้าตัวแปรที่ใช้นับจำนวนการเกิดอินเตอร์รัปต์มีค่าเกินค่าที่กำหนดไว้ตามกฎ exponential backoff ไมโครคอนโทรลเลอร์จะส่งข้อมูลภายในบัฟเฟอร์ภาคส่งไปให้ชิพอีเทอร์เนต และควบคุมให้ชิพอีเทอร์

เนตส่งข้อมูลนั้นซ้ำไปยังปลายทางอีกครั้ง เมื่อตรวจสอบตัวแปรซึ่งใช้นับจำนวนการเกิดอินเตอร์รัปต์เรียบร้อยแล้ว ไมโครคอนโทรลเลอร์จะกลับไปตรวจสอบพอร์ต 3.2 และไล่วนตรวจสอบสถานะของเครื่อง, จำนวนข้อมูลเสียงภายในบัฟเฟอร์ข้อมูลเสียงภาคส่ง, คีย์แพด, และตัวแปรซึ่งใช้นับจำนวนการเกิดอินเตอร์รัปต์จากไทเมอร์ 1 วนซ้ำไปเรื่อย ๆ ต่อไป

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
การออกแบบส่วนติดต่อกับผู้ใช้งานสำหรับระบบโทรศัพท์ผ่านอินเทอร์เน็ต (VOIP) บนพื้นฐานของโพรโทคอลเริ่มต้นเซสชัน

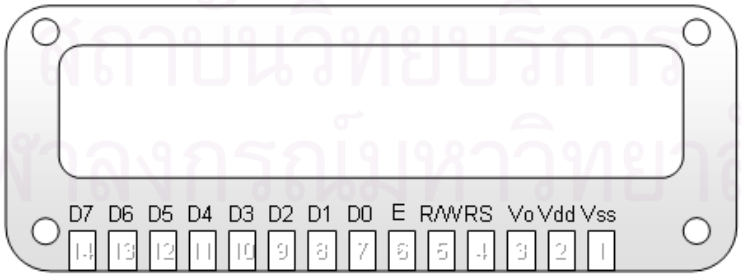
ส่วนติดต่อกับผู้ใช้งานประกอบไปด้วยส่วนแสดงสถานะการทำงานผ่าน LCD (Liquid Crystal Display) และส่วนรับข้อมูลจากผู้ใช้งานผ่านคีย์แพดตามมาตรฐาน ITU E.161 เพื่อรองรับข้อมูลอักขระสำหรับการกำหนด SIP URL ของ เครื่องลูกข่ายปลายทาง และควบคุมการสร้างและยกเลิกเซสชัน แสดงในรูปที่ 4.1



รูปที่ 4.1 ส่วนติดต่อกับผู้ใช้งาน

4.1 ส่วนแสดงผลผ่าน LCD

ในส่วนแสดงผลได้ใช้ LCD ชนิด 16 อักขระ 2 บรรทัด มีขาเชื่อมต่อข้อมูล 8 ขา, ขาควบคุม 3 ขา และขาสำหรับไฟเลี้ยง 3 ขา ซึ่งเป็นมาตรฐานของ LCD ทั่วไป ดังแสดงในรูป ที่ 4.2



รูปที่ 4.2 ขาเชื่อมต่อตามมาตรฐานของ LCD

ขา Vdd และ Vss ใช้ต่อกับไฟเลี้ยง +5V และกราวด์ ส่วน Vo เป็นขาปรับแรงดันไฟเพื่อควบคุมความสว่างของหน้าจอ LCD

ขา RS (Register Select) เป็นขาอินพุตใช้สำหรับเลือกชนิดของข้อมูลที่ติดต่อสื่อสารว่าเป็นข้อมูลคำสั่ง หรือข้อมูลอักขระ โดยถ้า RS = 0 หมายความว่าข้อมูลที่ติดต่อสื่อสารนั้นเป็นข้อมูลคำสั่ง และถ้า RS = 1 หมายความว่าข้อมูลที่ติดต่อสื่อสารนั้นเป็นข้อมูลอักขระที่ต้องการให้แสดงผลออกทางหน้าจอ

ขา R/W (Read/Write) เป็นขาอินพุตใช้สำหรับเลือกการอ่าน หรือเขียนข้อมูล โดยถ้า R/W = 1 จะเป็นการอ่านข้อมูลจาก LCD และถ้า R/W = 0 จะเป็นการเขียนข้อมูลลงใน LCD

ขา E (Enable) เป็นขาอีนาเบิลให้ LCD ทำงาน โดยจะทำงานเมื่อได้รับพัลส์ความกว้างอย่างน้อย 450 nsec เข้ามา

ขา D0-D7 เป็นขารับส่งข้อมูลขนาด 8 บิตระหว่าง LCD กับอุปกรณ์อื่น

โครงสร้างของหน่วยความจำภายในโมดูล LCD ประกอบด้วย

รีจิสเตอร์คำสั่ง (Instruction Register: IR) เป็นรีจิสเตอร์ที่ใช้รับ และส่งข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อควบคุมการทำงาน และการแสดงผลของ LCD

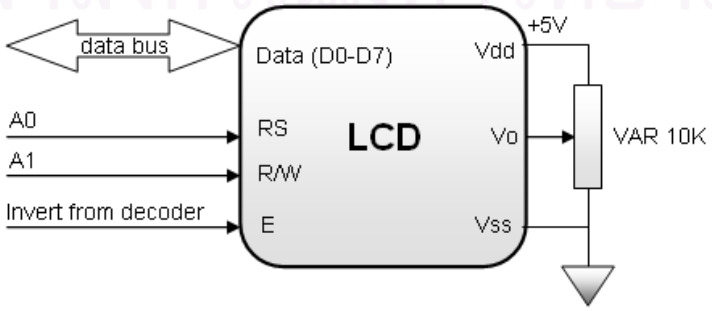
รีจิสเตอร์ข้อมูล (Data Register: DR) เป็นรีจิสเตอร์ที่ใช้รับ และส่งข้อมูลอักขระเพื่อนำไปแสดงผลออกหน้าจอ LCD หรืออ่านค่าอักขระที่ได้แสดงอยู่บนหน้าจอ LCD

แรมเก็บข้อมูลแสดงผล (Display Data RAM: DDRAM) เป็นหน่วยความจำเก็บค่าข้อมูลที่แสดงผลอยู่บนหน้าจอ ในแต่ละตำแหน่ง

รอมเก็บอักขระ (Character Generator ROM: CGROM) เป็นหน่วยความจำที่ใช้เก็บข้อมูลลักษณะของอักขระ และสัญลักษณ์ต่าง ๆ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของ DDRAM

แรมเก็บอักขระ (Character Generator RAM: CGRAM) เป็นหน่วยความจำที่ใช้เก็บอักขระที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่อักขระใน CGROM ไม่เพียงพอ มีขนาด 512 บิต

ในการทำ Memory map เพื่อต่อใช้งาน LCD นั้นได้ต่อขา RS และ R/W เข้ากับสายสัญญาณกำหนดตำแหน่งหน่วยความจำ A0 และ A1 ตามลำดับ ดังแสดงในรูปที่ 4.3 ซึ่งทำให้ไมโครคอนโทรลเลอร์มอง LCD เป็นเสมือนหน่วยความจำขนาด 8 บิตจำนวน 4 ไบต์ดังแสดงในตารางที่ 4.1



รูปที่ 4.3 การทำ Memory Map เพื่อใช้งาน LCD

ตารางที่ 4.1 การทำงานของ LCD เมื่อได้ทำ Memory map

ตำแหน่ง	R/W (A1)	RS (A0)	การทำงาน
0x8000	0	0	เขียนข้อมูลคำสั่งลงในรีจิสเตอร์ควบคุม LCD
0x8001	0	1	เขียนข้อมูลอักขระให้แสดงผลบน LCD
0x8002	1	0	อ่านค่ารีจิสเตอร์ควบคุมภายใน LCD
0x8003	1	1	อ่านข้อมูลอักขระในรีจิสเตอร์แสดงผลของ LCD

ตารางที่ 4.2 ข้อมูลคำสั่งของ LCD

คำสั่ง	Hex	D7	D6	D5	D4	D3	D2	D1	D0
เคลียร์หน้าจอแสดงผล	01	0	0	0	0	0	0	0	1
ให้เคอร์เซอร์มาอยู่ที่ตำแหน่งเริ่มต้น	02-03	0	0	0	0	0	0	1	x
เลือกโมดการป้อนอักขระ	04-07	0	0	0	0	0	1	1/D	S
ควบคุมการแสดงผล	08-0F	0	0	0	0	1	D	U	B
ควบคุมการเลื่อนเคอร์เซอร์และอักขระ	10-1F	0	0	0	1	D/C	R/L	x	x
ตั้งค่าการทำงานของ LCD	20-3F	0	0	1	8/4	2/1	10/7	x	x
x = don't care									

คำสั่งเคลียร์หน้าจอแสดงผล (Clear display) มีข้อมูลคำสั่งเป็น 01H เมื่อ LCD ได้รับคำสั่งนี้ ส่วนควบคุมจะเขียนข้อมูลช่องว่างลงไปใน DDRAM ทุกตำแหน่ง และกำหนดตำแหน่งชี้ข้อมูลของ DDRAM ให้เป็น 0 ซึ่งจะเป็นการลบหน้าจอทั้งหมด และเลื่อนเคอร์เซอร์กลับมาอยู่ที่ตำแหน่งแรกสุดของหน้าจอแสดงผล

คำสั่งให้เคอร์เซอร์มาอยู่ที่ตำแหน่งเริ่มต้น (Return home) มีข้อมูลคำสั่งเป็น 02H หรือ 03H เมื่อ LCD ได้รับคำสั่งนี้ส่วนควบคุมจะกำหนดตำแหน่งชี้ข้อมูลของ DDRAM เป็น 0 ซึ่งจะทำให้เคอร์เซอร์กลับมาอยู่ที่ตำแหน่งแรกสุดของหน้าจอแสดงผล โดยไม่เปลี่ยนแปลงข้อมูลใน DDRAM

คำสั่งตั้งโมดการป้อนอักขระ (Entry mode set) มีข้อมูลคำสั่งเป็น 04H-07H ใช้กำหนดการกระทำต่อตำแหน่งชี้ข้อมูลของ DDRAM เมื่อมีการป้อนอักขระเพื่อแสดงผลบน LCD โดยบิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล โดยถ้า S = 1 เมื่อมีการป้อนข้อมูลใหม่ลงบน LCD เคอร์เซอร์จะอยู่กับที่ และดันข้อมูลอักขระเก่าไปด้านข้าง แต่ถ้า S = 0 เมื่อมีการป้อนข้อมูลใหม่ลงบน LCD อักขระเก่าจะอยู่กับที่ แต่เคอร์เซอร์จะเลื่อนไปด้านข้าง ส่วนบิต I/D นั้นเป็นบิตที่ใช้ในการกำหนดทิศทางในการเลื่อน เมื่อมีการป้อนข้อมูลใหม่ลง LCD โดยถ้า I/D = 0 เมื่อมีการป้อน

อักขระใหม่ลงบน LCD ตำแหน่งชี้ข้อมูล DDRAM จะเพิ่มขึ้น แต่ถ้า I/D = 1 ตำแหน่งชี้ข้อมูล DDRAM จะลดลง คำสั่งตั้งโมดการป้อนอักขระที่ใช้บ่อยคือ 06H คือเมื่อมีการป้อนข้อมูลอักขระใหม่ให้กับ LCD อักขระเก่าจะอยู่กับที่ และเคอร์เซอร์เลื่อนไปทางขวามือ

คำสั่งควบคุมการแสดงผล (Display ON/OFF Control) มีข้อมูลคำสั่งเป็น 08H-0FH ใช้ในการควบคุมหน้าจอแสดงผล และเคอร์เซอร์ โดยบิต D ใช้ควบคุมการเปิดปิดหน้าจอแสดงผล โดยถ้าบิต D = 1 จะเป็นการสั่งให้เปิดหน้าจอ และถ้า D = 0 จะเป็นการสั่งปิดหน้าจอ บิต C ใช้ควบคุมการเปิดปิดเคอร์เซอร์ โดยถ้า C = 1 จะเป็นการเปิดเคอร์เซอร์ และถ้า C = 0 จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์ ส่วนบิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ โดยถ้าบิต B = 1 เคอร์เซอร์จะกระพริบ

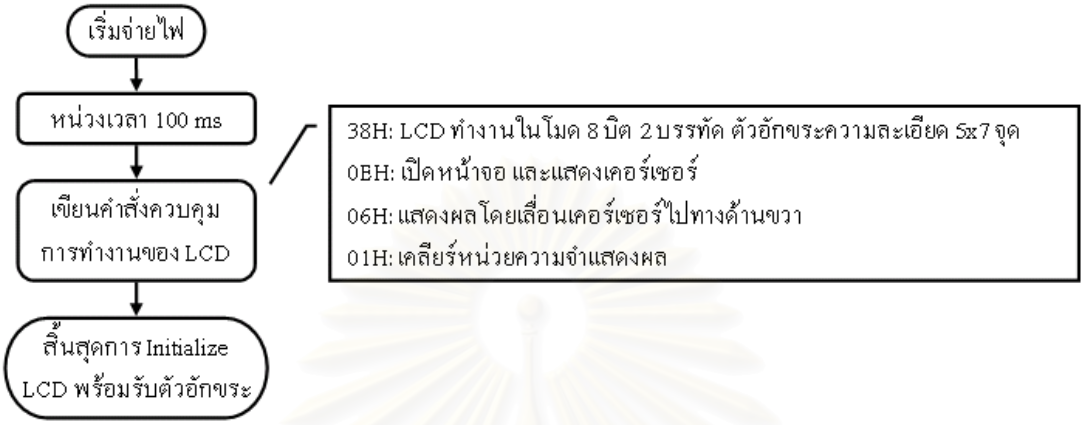
คำสั่งควบคุมการเลื่อนเคอร์เซอร์และอักขระ (Display/Cursor Shift) การควบคุมนั้นขึ้นอยู่กับการกำหนดบิต S/C และ R/L โดยบิต S/C เป็นเป็นบิตเลือกอักขระ และเคอร์เซอร์ และบิต R/L เป็นบิตเลือกทิศทางในการเลื่อน ถ้าบิต S/C = 0 จะเป็นการกระทำกับเคอร์เซอร์ และถ้า S/C = 1 จะเป็นการกระทำกับอักขระ ส่วนบิต R/L = 0 จะเป็นการเลื่อนไปทางซ้าย และถ้าบิต R/L = 1 จะเป็นการเลื่อนไปทางขวา

คำสั่งตั้งค่าการทำงานของของ LCD (Function Set) บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล โดยถ้า DL = 0 จะเป็นการติดต่อแบบ 4 บิต และถ้า DL = 1 จะเป็นการติดต่อแบบ 8 บิต บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผลโดยถ้า N = 0 จะเป็นการตั้งค่าสำหรับการแสดงผลแบบ 1 บรรทัด และถ้า DL = 1 จะเป็นการตั้งค่าสำหรับการแสดงผลแบบ 2 บรรทัด ส่วนบิต F นั้นใช้เลือกความละเอียดของอักขระในการแสดงผล โดยถ้า F = 0 เป็นการตั้งค่าสำหรับการแสดงผลความละเอียดแบบ 5x7 จุดต่ออักขระและถ้า F = 1 จะเป็นการตั้งค่าสำหรับการแสดงผลแบบ 5x10 จุดต่ออักขระ

การเขียนข้อมูลคำสั่ง หรือข้อมูลอักขระลง LCD ทุกครั้ง จำเป็นต้องตรวจสอบสถานะของ LCD ก่อนว่า LCD นั้นพร้อมจะรับข้อมูลหรือไม่ โดยการอ่านบิต BUSY ซึ่งเป็นบิต D7 ของรีจิสเตอร์ควบคุมภายใน LCD โดยถ้าบิต BUSY นี้มีลอจิกเป็น “1” แสดงว่า LCD กำลังประมวลผลภายในอยู่ ไม่สามารถรับข้อมูลได้ในขณะนั้น แต่ถ้าบิต BUSY นี้มีลอจิกเป็น “0” แสดงว่า LCD อยู่ในสถานะว่างและสามารถรับข้อมูลจากอุปกรณ์ภายนอกได้

ในการใช้งาน LCD เริ่มแรกต้องเขียนข้อมูลคำสั่งควบคุมดังแสดงในตารางที่ 2.3 เพื่อกำหนดค่าเริ่มต้นต่าง ๆ ที่จำเป็นต่อการทำงานให้กับ LCD ก่อน ถึงจะสามารถเขียนข้อมูลอักขระให้ LCD แสดงผลได้ และเนื่องจากการทำงานของ LCD นั้นช้ากว่าไมโครคอนโทรลเลอร์มาก ดังนั้นในการใช้งานจริงเมื่อเริ่มจ่ายไฟให้กับไมโครคอนโทรลเลอร์ และ LCD แล้ว ต้องหน่วงเวลา

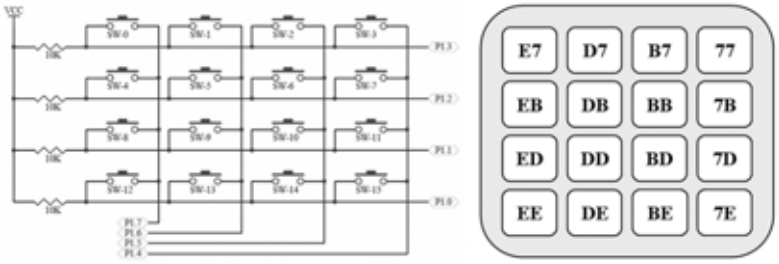
ไมโครคอนโทรลเลอร์ประมาณ 15 ms เพื่อรอให้ LCD อยู่ในสถานะพร้อมทำงาน ถึงจะสามารถเขียนข้อมูลคำสั่งเพื่อกำหนดค่าเริ่มต้นต่างๆให้กับ LCD ได้ ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 การเริ่มต้นใช้งาน LCD

4.2 ส่วนรับข้อมูลจากผู้ใช้งานผ่านคีย์แพด

ในส่วนรับข้อมูลจากผู้ใช้งาน ได้ใช้สวิตช์จำนวน 16 ตัวมาต่อเรียงกันเป็นคีย์แพดขนาด 4x4 แบบเมตริกซ์และต่อเข้ากับพอร์ต 1 ของไมโครคอนโทรลเลอร์ดังแสดงในรูปที่ 4.5(a) การต่อแบบนี้สามารถใช้พอร์ตเพียง 8 บิต ซึ่งน้อยกว่าการต่อสวิตช์เข้ากับพอร์ตโดยตรงที่ต้องใช้พอร์ตถึง 16 บิต แต่ไมโครคอนโทรลเลอร์ต้องทำงานซับซ้อนกว่าดังแสดงในรูปที่ 4.6 พอร์ตบน (P1.4-P1.7) ทำหน้าที่เป็นอินพุต และพอร์ตล่าง (P1.0-P1.3) ทำหน้าที่เป็นเอาต์พุตโดยมีตัวต้านทานขนาด 10 กิโลโอห์มต่อพูลอัปกับแรงดันไฟฟ้า +5โวลต์ไว้ ในการตรวจสอบการกดคีย์ไมโครคอนโทรลเลอร์ จะไล่ส่งลอจิก “0” ออกไปทางพอร์ตล่างครั้งละ 1 บิตอย่างรวดเร็ว และวนซ้ำเรื่อย ๆ ในขณะเดียวกันก็จะอ่านค่าจากพอร์ตบนไปด้วยถ้าค่าที่อ่านได้จากพอร์ตบนเป็นลอจิก ‘1’ หมดแสดงว่าในแถวที่พอร์ตล่างส่งลอจิก “0” อยู่นั้นไม่มีการกดคีย์ แต่ถ้าค่าที่อ่านมาไม่เป็นลอจิก ‘1’ หมดแสดงว่ามีการกดคีย์ที่แถวนั้นเกิดขึ้น ค่าที่อ่านได้จากพอร์ตบนนี้จะเป็นตัวระบุตำแหน่งของสวิตช์ที่ถูกกดในแนวแกน X และตำแหน่งของลอจิก “0” ที่พอร์ตล่างจะเป็นตัวระบุตำแหน่งของสวิตช์ที่ถูกกดในแนวแกน Y โดยมีความสัมพันธ์กันดังแสดงในรูปที่ 4.5 (b)



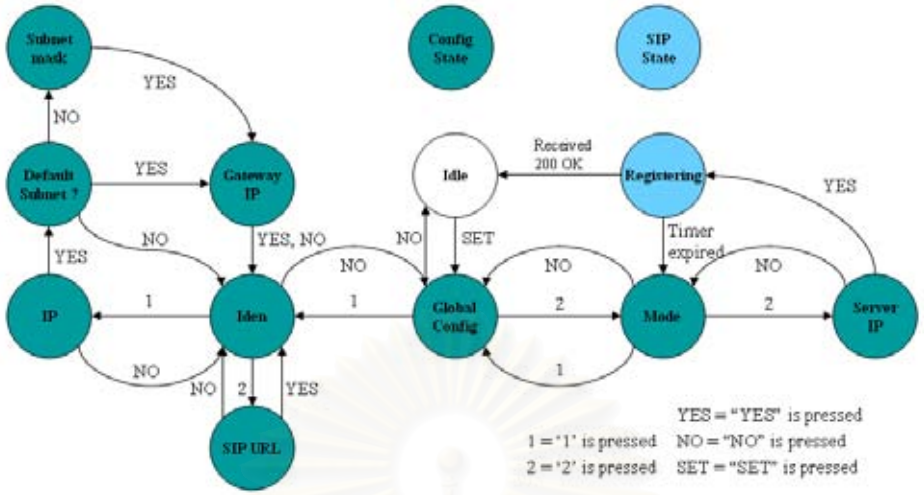
(a) (b)

รูปที่ 4.5 (a) การต่อคีย์แพด 4x4 เพื่อรับอินพุตจากผู้ใช้งาน
(b) รหัสของคีย์แพดแบบเมตริกซ์

รูปที่ 4.6 ขั้นตอนการตรวจสอบคีย์แพด

คีย์แพดเป็นส่วนสำคัญที่ใช้ในการปรับตั้งค่าพารามิเตอร์ที่สำคัญต่อการทำงานของเครื่องโทรศัพท์ ได้แก่ไอพีแอดเดรส, หมายเลขซับเน็ต, SIP URL ของตัวเครื่องโทรศัพท์, ไอพีแอดเดรสของเกตเวย์, และไอพีแอดเดรสของเครื่องแม่ข่าย รวมไปถึงการปรับตั้งโมดการทำงานของตัวเครื่องโทรศัพท์ด้วย

ในการปรับตั้งพารามิเตอร์ต่าง ๆ บนเครื่องโทรศัพท์นั้น ทำได้โดยกดปุ่ม Set ในขณะที่เครื่องโทรศัพท์อยู่ในสภาวะ Idle เมื่อกดปุ่ม Set แล้ว LCD จะแสดงสถานะแจ้งให้ทราบว่าเครื่องโทรศัพท์ได้เข้าสู่โมดปรับตั้งพารามิเตอร์ และสามารถปรับตั้งพารามิเตอร์ต่าง ๆ ได้ดังแสดงในรูปที่ 4.7



รูปที่ 4.7 สถานะการทำงานในโมดปรับตั้งพารามิเตอร์

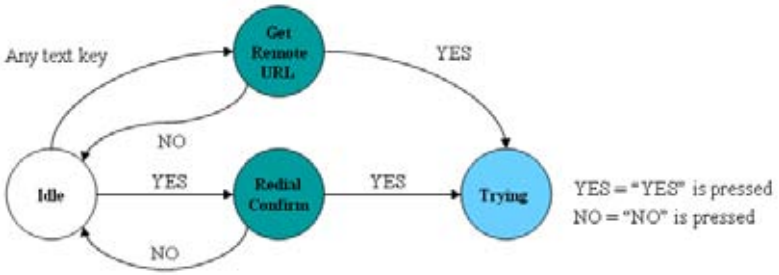
จะเห็นได้ว่าการปรับตั้งพารามิเตอร์นั้นสามารถแบ่งออกได้เป็น 2 ส่วนใหญ่ ๆ คือ พารามิเตอร์ที่ใช้ระบุเครื่องโทรศัพท์ และพารามิเตอร์ที่ใช้บ่งบอกโมดการทำงานของเครื่องโทรศัพท์ พารามิเตอร์ที่ใช้ระบุเครื่องโทรศัพท์ได้แก่ SIP URL, ไอพีแอดเดรสของเครื่องโทรศัพท์, ซับเน็ต, และไอพีแอดเดรสของเกตเวย์ ส่วนพารามิเตอร์ที่ใช้บ่งบอกโมดการทำงานของเครื่องโทรศัพท์ได้แก่ การเลือกทำงานร่วมกับเครื่องแม่ข่าย หรือทำงานโดยไม่อาศัยเครื่องแม่ข่าย ซึ่งในการทำงานร่วมกับเครื่องแม่ข่ายนั้นจำเป็นต้องใช้ไอพีแอดเดรสของเครื่องแม่ข่ายด้วย เมื่อกำหนดค่าไอพีแอดเดรสของเครื่องแม่ข่ายเสร็จแล้ว เครื่องโทรศัพท์อินเทอร์เนตจะลงทะเบียนกับเครื่องแม่ข่ายทันที ในกรณีที่ไม่สามารถลงทะเบียนได้ เครื่องโทรศัพท์จะกลับเข้าสู่โมดปรับตั้งการทำงานของเครื่องโทรศัพท์อีกครั้ง

ในการป้อนค่าไอพีแอดเดรสให้ไมโครคอนโทรลเลอร์ผ่านคีย์แพดนั้น สวิตช์แต่ละตัวใช้ระบุแทนตัวเลขโดยตรงเหมือนคีย์แพดของโทรศัพท์ทั่วไป และใช้สวิตช์ในตำแหน่ง '*' แทนเครื่องหมาย '.' เพื่อแบ่งแยกค่าไอพีแอดเดรสในแต่ละไบต์ และสวิตช์ในตำแหน่ง '#' แทนปุ่มลบ (Delete) เมื่อต้องการลบสิ่งที่ได้พิมพ์ผ่านคีย์แพดไป เนื่องจากการป้อนข้อมูลไอพีแอดเดรสผ่านคีย์แพดนั้นไมโครคอนโทรลเลอร์จะได้รับข้อมูลจากผู้ใช้ครั้งละดิจิต ดังนั้นในขณะพิมพ์ไอพีแอดเดรสไมโครคอนโทรลเลอร์จะเก็บข้อมูลที่พิมพ์ในรูปแบบของตัวอักขระครั้งละตัวไว้ในบัฟเฟอร์ชั่วคราวก่อน และในขณะที่ป้อนคีย์แต่ละครั้ง ไมโครคอนโทรลเลอร์จะตรวจสอบรูปแบบของอักขระให้เป็นไปตามรูปแบบของหมายเลขไอพีแอดเดรสที่ถูกต้อง จนเมื่อป้อนไอพีแอดเดรสครบ และกดปุ่ม "Yes" ไมโครคอนโทรลเลอร์จึงจะนำข้อมูลในบัฟเฟอร์ชั่วคราวนั้นไปแปลงเป็นค่าไอพีแอดเดรสในแต่ละไบต์ และเขียนค่านั้นลงหน่วยความจำข้อมูล RAM และ EEPROM ต่อไป ในกรณีที่ไอพีแอดเดรสที่ปรับตั้งเป็นของเกตเวย์ ไมโครคอนโทรลเลอร์จะตรวจสอบไอพีแอดเดรสที่ได้จากผู้ใช้งาน ว่าอยู่ใน

โครงข่ายพื้นที่ท้องถิ่นเดียวกับตัวเครื่องโทรศัพท์หรือไม่ โดยการนำไอพีแอดเดรสนั้นมาแอนด์ (AND) กับซับเนตมาสค์ ถ้าเนตเวิร์กแอดเดรสที่ได้ตรงกับ เนตเวิร์กแอดเดรสของเครื่องโทรศัพท์แสดงว่าอยู่ในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน แต่ถ้าเนตเวิร์กแอดเดรสที่ได้ไม่ตรงกันแสดงว่าอยู่คนละโครงข่าย ถ้าไม่ได้อยู่ในโครงข่ายเดียวกันเครื่องโทรศัพท์จะกลับเข้าสู่โมดการตั้งค่าไอพีแอดเดรสของเกตเวย์ใหม่อีกครั้ง

ในการพัฒนาส่วนควบคุมเพื่อรับการป้อนค่า SIP URL ซึ่งเป็นอักขระ ผ่านคีย์แพด 4x4 นั้น ได้พัฒนาตามมาตรฐาน ITU E1.161 ซึ่งสวิตช์แต่ละตัวจะต้องรองรับอักขระหลายตัว มีขั้นตอนการทำงานดังนี้ เมื่อผู้ใช้งานกดสวิตช์เพื่อป้อนอักขระ ไมโครคอนโทรลเลอร์จะรับอักขระแรกที่สวิตช์ตำแหน่งนั้นรองรับเข้ามาเก็บไว้ในหน่วยความจำชั่วคราว และแสดงออกทาง LCD พร้อมกับเปิดไทเมอร์เพื่อจับเวลาที่ใช้ในการเลื่อนตำแหน่งเคอร์เซอร์ ถ้าอักขระที่แสดงไม่ใช่อักขระที่ผู้ใช้งานต้องการ ผู้ใช้งานสามารถเปลี่ยนอักขระได้โดยการกดสวิตช์ตำแหน่งเดิมซ้ำ จนกว่าจะถึงอักขระที่ผู้ใช้งานต้องการ โดยทุกครั้งที่ผู้ใช้งานกดสวิตช์ไมโครคอนโทรลเลอร์จะรีเซตไทเมอร์ และเปิดไทเมอร์เพื่อจับเวลาใหม่ จนเมื่อสวิตช์ไม่มีการกดเป็นเวลานานประมาณ 1 วินาทีไมโครคอนโทรลเลอร์จึงจะเขียนอักขระนั้นลงในบัฟเฟอร์ชั่วคราว พร้อมสั่งให้เคอร์เซอร์ LCD เลื่อนไปเพื่อรอรับอักขระถัดไป เมื่อผู้ใช้งานป้อน SIP URL จนเสร็จ และกดปุ่ม “Yes” ไมโครคอนโทรลเลอร์จะนำข้อมูลในบัฟเฟอร์ชั่วคราวนั้นเขียนลงในหน่วยความจำข้อมูล RAM และ EEPROM ต่อไป

นอกจากคีย์แพดจะใช้สำหรับการปรับตั้งพารามิเตอร์ต่าง ๆ แล้ว ยังใช้ในการสร้าง และยกเลิกเซสชันกับเครื่องลูกข่ายอื่นด้วยดังแสดงในรูปที่ 4.8 เมื่อผู้ใช้งานต้องการสร้างเซสชันกับเครื่องลูกข่ายอื่นสามารถทำได้โดยการกดสวิตช์สำหรับป้อนอักขระในขณะที่เครื่องอยู่ในสถานะ Idle เมื่อสวิตช์ถูกกดแล้ว เครื่องโทรศัพท์จะเปลี่ยนสถานะพร้อมกับแสดงผลผ่าน LCD เพื่อรอรับ SIP URL ปลายทางจากผู้ใช้งาน เมื่อผู้ใช้งานป้อน SIP URL ปลายทางเสร็จและกดปุ่ม “YES” เครื่องโทรศัพท์จะเก็บข้อมูล SIP URL นั้นไว้ในหน่วยความจำข้อมูล RAM และส่งข้อมูลเพื่อขอสร้างเซสชันกับปลายทางต่อไป ในกรณีที่ได้สร้างเซสชันและยกเลิกเซสชันจนสำเร็จแล้ว หรือในการสร้างเซสชันไม่สำเร็จ และผู้ใช้ต้องการสร้างเซสชันใหม่ไปยังปลายทางเดิม (Redial) สามารถทำได้โดยกดสวิตช์ “YES” ในขณะที่เครื่องอยู่ในสถานะ Idle เครื่องโทรศัพท์จะเปลี่ยนสถานะและแสดงผลผ่าน LCD เพื่อให้ผู้ใช้งานยืนยันการสร้างเซสชันไปยังปลายทางเดิม เมื่อผู้ใช้งานกดปุ่ม “YES” อีกครั้ง เครื่องโทรศัพท์จะส่งข้อมูลเพื่อขอสร้างเซสชันกับปลายทางเดิมต่อไป



รูปที่ 4.8 การใช้คีย์แพดในการสร้างเซสชัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

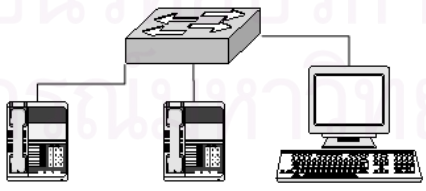
บทที่ 5
การทดสอบ

5.1 ข้อกำหนดในการทดสอบ

ในการทดสอบเครื่องโทรศัพท์อินเทอร์เนตที่ได้พัฒนาขึ้นได้ใช้คอมพิวเตอร์ที่ลงโปรแกรม SCS SIP Proxy ของบริษัท Siemens ทำงานเป็นเครื่องแม่ข่ายทั้งแบบ Proxy และ Redirect ส่วนเครื่องลูกข่ายปลายทางได้ใช้เครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่อง และคอมพิวเตอร์ที่ลงโปรแกรม SIP User Agent เวอร์ชัน 2.0 ของบริษัท Ubiquity ในการร่วมทดสอบ ซึ่งในการทดสอบแต่ละครั้งจะใช้โปรแกรม Ethereal ตรวจจับ Signaling ต่าง ๆ เพื่อบันทึกผล

การทดสอบแบ่งออกเป็น 3 ส่วนคือการทดสอบการทำงานในโมดติดต่อโดยตรง, การทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย, และการทดสอบฟังก์ชันการส่งซ้ำกรณีเครื่องโทรศัพท์ไม่ได้รับข้อความตอบสนอง ในการทดสอบการทำงานในโมดติดต่อโดยตรง และทำงานร่วมกับเครื่องแม่ข่ายนั้นได้ทดสอบการสร้างเซสชันทั้งกรณีที่เครื่องโทรศัพท์เป็นผู้ร้องขอการสร้างเซสชันและเป็นผู้ถูกร้องขอการสร้างเซสชัน ในการขอปิดเซสชันก็เช่นกัน ได้ทดสอบทั้งกรณีที่เครื่องโทรศัพท์เป็นผู้ร้องขอการปิดเซสชัน และเป็นผู้ถูกร้องขอการปิดเซสชัน และเพื่อตรวจสอบความถูกต้องของโพรโทคอลอีเทอร์เนตที่ได้พัฒนาขึ้นจึงได้ทดสอบทั้งในกรณีเครื่องปลายทางอยู่ในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน และอยู่คนละโครงข่ายกัน

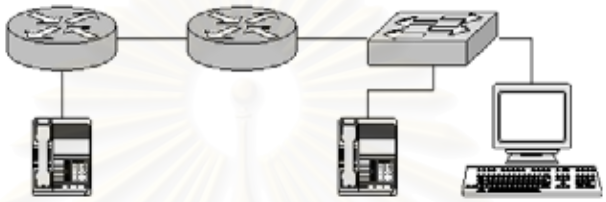
ในการทดสอบการทำงานในโมดติดต่อโดยตรงภายในโครงข่ายเดียวกัน ได้ต่อเครื่องโทรศัพท์ และคอมพิวเตอร์ที่ลงโปรแกรม SIP User Agent เข้ากับสวิตซ์ดังแสดงในรูปที่ 5.1 และปรับตั้งเครื่องโทรศัพท์ที่พัฒนาขึ้น และปรับตั้งโปรแกรม SIP User Agent ให้ทำงานในโมดติดต่อโดยตรง และตั้งค่าไอพีแอดเดรส และซับเนตมาสค์ให้เครื่องโทรศัพท์ และคอมพิวเตอร์อยู่ในโครงข่ายเดียวกัน



รูปที่ 5.1 การทดสอบการทำงานในโมดติดต่อโดยตรงภายในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน

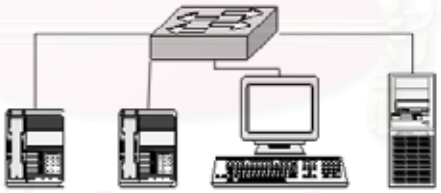
ในการทดสอบการทำงานในโมดติดต่อโดยตรงข้ามโครงข่ายพื้นที่ท้องถิ่น เนื่องจากเราเตอร์ที่ใช้ในการทดสอบมีพอร์ตอีเทอร์เนตเพียงพอร์ตเดียว ดังนั้นในการสร้างโครงข่ายพื้นที่ท้องถิ่น 2โครงข่ายให้มีพอร์ตอีเทอร์เนตทั้งสองโครงข่ายจึงต้องใช้เราเตอร์ 2 ตัว ในการทดสอบได้ต่อเราเตอร์ตัวแรกเข้ากับเครื่องโทรศัพท์ที่ต้องการทดสอบโดยตรง ส่วนเราเตอร์ตัวที่ 2 ต่อเข้ากับ

สวิตซ์เพื่อต่อกับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่องหนึ่ง และคอมพิวเตอร์ที่มีโปรแกรม SIP User Agent ดังแสดงในรูปที่ 5.2 จากนั้นจึงปรับตั้งเครื่องโทรศัพท์ที่พัฒนาขึ้น และปรับตั้งโปรแกรม SIP User Agent ให้ทำงานในโมดติดต่อโดยตรง และตั้งค่าไอพีแอดเดรส และซับเนตมาสค์ของ อินเตอร์เฟสอีเทอร์เนตของเราเตอร์ตัวที่ 2, เครื่องโทรศัพท์, และคอมพิวเตอร์ที่ต่อกับสวิตซ์ให้อยู่ในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน และตั้งค่าไอพีแอดเดรส และซับเนตมาสค์ของเครื่องที่ต้องการทดสอบ และอินเตอร์เฟสอีเทอร์เนตของเราเตอร์ตัวที่ 1 ให้อยู่วงเดียวกัน แต่อยู่คนละวงกับวงแรกที่ได้สร้างขึ้น



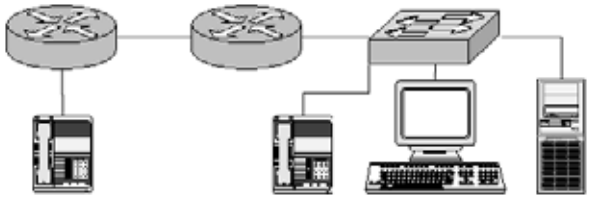
รูปที่ 5.2 การทดสอบการทำงานในโมดติดต่อโดยตรงข้ามโครงข่ายพื้นที่ท้องถิ่น

ในการทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่ายภายในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน ได้ต่อเครื่องโทรศัพท์ และคอมพิวเตอร์ที่ลงโปรแกรม SIP User Agent และคอมพิวเตอร์ที่ลงโปรแกรม SCS SIP Proxy เข้ากับสวิตซ์ดังแสดงในรูปที่ 5.3 และปรับตั้งเครื่องโทรศัพท์ที่พัฒนาขึ้น และโปรแกรม SIP User Agent ให้ทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย พร้อมกับตั้งค่าไอพีแอดเดรส และซับเนตมาสค์ให้อุปกรณ์ทั้งหมดอยู่ในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน



รูปที่ 5.3 การทดสอบการทำงานร่วมกับเครื่องแม่ข่ายภายในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน

การทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย ข้ามโครงข่ายพื้นที่ท้องถิ่น คล้ายกับการทดสอบการทำงานในโมดติดต่อโดยตรงข้ามโครงข่ายพื้นที่ท้องถิ่น แต่เพิ่มเครื่องแม่ข่ายเข้าไปในวงที่ต่ออุปกรณ์ร่วมกันโดยใช้สวิตซ์ ดังแสดงในรูปที่ 5.4 และต้องปรับตั้งเครื่องโทรศัพท์ที่พัฒนาขึ้น และโปรแกรม SIP User Agent ให้ทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย



รูปที่ 5.4 การทดสอบการทำงานร่วมกับเครื่องแม่ข่ายข้ามโครงข่ายพื้นที่ท้องถิ่น

5.2 การทดสอบการทำงานในโมดติดต่อโดยตรง

การทดสอบการทำงานในโมดติดต่อโดยตรง เพื่อให้ครอบคลุมการทำงานขั้นพื้นฐานของการควบคุมเซสชัน จึงได้แบ่งการทดสอบออกเป็น 4 ส่วนคือ การสร้างเซสชัน, การยกเลิกการสร้างเซสชัน, การปฏิเสธการสร้างเซสชัน, และการปิดเซสชัน

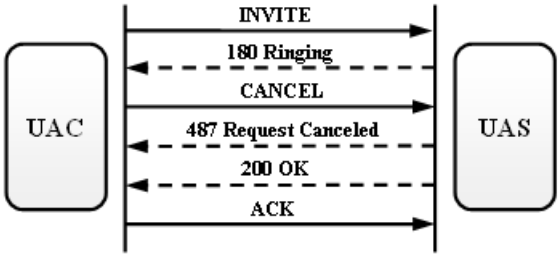
5.2.1 การสร้างเซสชัน

การทดสอบการสร้างเซสชันดังแสดงในรูปที่ 5.5 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน เครื่องโทรศัพท์ต้องสามารถส่งข้อความร้องขอ INVITE และ ACK และตอบสนองข้อความตอบสนอง 180 Ringing และ 200 OK ได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสร้างเซสชัน ในการสร้างเซสชันเครื่องโทรศัพท์ต้องสามารถประมวลผล และตอบสนองข้อความร้องขอ INVITE ที่ได้รับ และสร้างข้อความตอบสนอง 180 Ringing และ 200 OK ได้อย่างถูกต้อง

รูปที่ 5.5 การทดสอบการสร้างเซสชัน

5.2.2 การยกเลิกการสร้างเซสชัน

การทดสอบการยกเลิกการสร้างเซสชันดังแสดงในรูปที่ 5.6 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน ในการยกเลิกการสร้างเซสชันเครื่องโทรศัพท์ต้องสามารถสร้างข้อความ CANCEL และเมื่อได้รับข้อความตอบสนอง 478 Request Canceled และ 200 OK จากเครื่องปลายทางแล้ว เครื่องโทรศัพท์ต้องสามารถประมวลผลข้อความตอบสนองนั้นได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกขอสร้างเซสชัน เมื่อได้รับข้อความ CANCEL เครื่องโทรศัพท์ต้องสามารถตอบสนองโดยสร้างข้อความ 487 Request Canceled และ 200 OK กลับไปได้อย่างถูกต้อง



รูปที่ 5.6 การทดสอบการขอยกเลิกการสร้างเซสชัน

5.2.3 การปฏิเสธการสร้างเซสชัน

การทดสอบการปฏิเสธการสร้างเซสชันดังแสดงในรูปที่ 5.7 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน เมื่อได้รับข้อความตอบสนอง 486 Busy Here เครื่องโทรศัพท์ต้องสามารถตอบสนองโดยสร้างข้อความ ACK กลับไปอย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกขอสร้างเซสชัน เมื่อต้องการปฏิเสธการสร้างเซสชันต้องสามารถสร้างข้อความตอบสนอง 486 Busy Here กลับไปได้อย่างถูกต้อง

รูปที่ 5.7 การทดสอบการปฏิเสธการสร้างเซสชัน

5.2.4 การขอสิ้นสุดเซสชัน

การทดสอบการขอสิ้นสุดเซสชันดังแสดงในรูปที่ 5.8 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสิ้นสุดเซสชัน เครื่องโทรศัพท์ต้องสามารถส่งข้อความร้องขอ BYE ได้อย่างถูกต้อง และในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสิ้นสุดเซสชัน เมื่อได้รับข้อความร้องขอ BYE เครื่องโทรศัพท์ต้องสามารถตอบสนองโดยส่งข้อความตอบสนอง 200 OK กลับไปได้อย่างถูกต้อง

รูปที่ 4.8 การทดสอบการขอสิ้นสุดเซสชัน

5.3 การทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย

การทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย แบ่งออกเป็น 4 ส่วนคือ การลงทะเบียนกับเครื่องแม่ข่าย, การสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy, การยกเลิกการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy, การปฏิเสธการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy, และการสร้างเซสชันผ่านเครื่องแม่ข่าย Redirect จะเห็นได้ว่าไม่ทดสอบการขอสิ้นสุดเซสชัน ทั้งนี้เนื่องจากการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย เมื่อเครื่องลูกข่ายสร้างเซสชันได้แล้ว เครื่องลูกข่ายจะติดต่อกันโดยตรงโดยไม่ผ่านเครื่องแม่ข่าย ดังนั้นกระบวนการขอสิ้นสุดเซสชันจึงมีลักษณะเหมือนการการติดต่อโดยตรงทุกประการ นอกจากนั้นจะเห็นได้ว่าการทดสอบการทำงานร่วมกับเครื่องแม่ข่ายแบบ Redirect นั้นทดสอบเฉพาะการสร้างเซสชันผ่านเครื่องแม่ข่าย Redirect เท่านั้น เนื่องจากเมื่อผู้ร้องขอได้รับข้อความตอบสนอง 302 Moved Temporarily จากเครื่องแม่ข่าย Redirect แล้วผู้ร้องขอจะติดต่อกับเครื่องลูกข่ายปลายทางโดยตรง และมีกระบวนการเหมือนการติดต่อโดยตรงทุกประการเช่นกัน

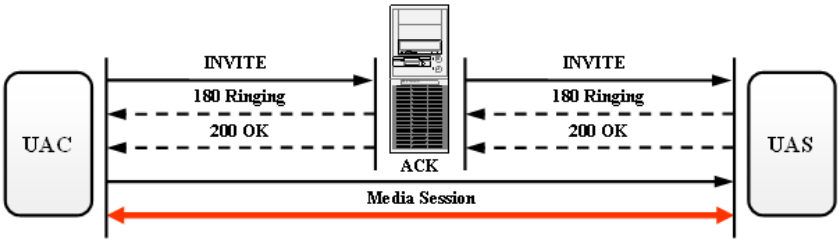
5.3.1 การลงทะเบียนกับเครื่องแม่ข่าย

การทดสอบการลงทะเบียนกับเครื่องแม่ข่ายดังแสดงในรูปที่5.9 เครื่องโทรศัพท์ที่พัฒนาขึ้นจะลงทะเบียนกับเครื่องแม่ข่ายทุกครั้งที่เครื่องโทรศัพท์เริ่มทำงาน หรือมีการปรับตั้งไอพีแอดเดรสของเครื่องแม่ข่ายใหม่ โดยในการทดสอบเครื่องลูกข่ายต้องสามารถสร้างข้อความร้องขอ REGISTER ให้กับเครื่องแม่ข่ายได้อย่างถูกต้อง

รูปที่ 5.9 การทดสอบการลงทะเบียนกับเครื่องแม่ข่าย

5.3.2 การสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy

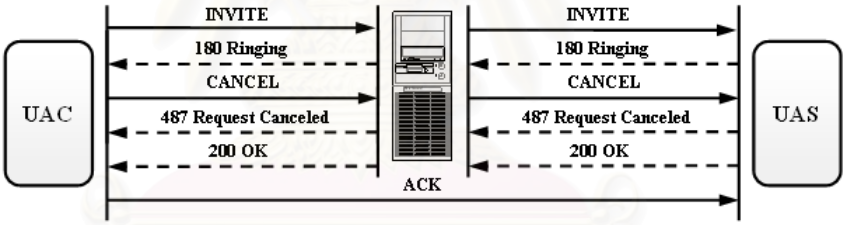
การทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ดังแสดงในรูปที่ 5.10 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน เครื่องโทรศัพท์ต้องสามารถส่งข้อความร้องขอ INVITE ให้กับเครื่องแม่ข่าย Proxy ได้อย่างถูกต้อง และเมื่อได้รับข้อความตอบสนอง 180 Ringing และ 200 OK จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์ต้องสามารถประมวลผลข้อความตอบสนองนั้นเพื่อส่งข้อความ ACK ให้กับเครื่องปลายทางได้โดยตรง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสร้างเซสชัน เมื่อได้รับข้อความร้องขอ INVITE จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์ต้องสามารถประมวลผลข้อความ INVITE นั้นและสร้างข้อความตอบสนอง 180 Ringing และ 200 OK กลับไปให้เครื่องแม่ข่ายได้อย่างถูกต้อง



รูปที่ 5.10 การทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy

5.3.3 การยกเลิกการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy

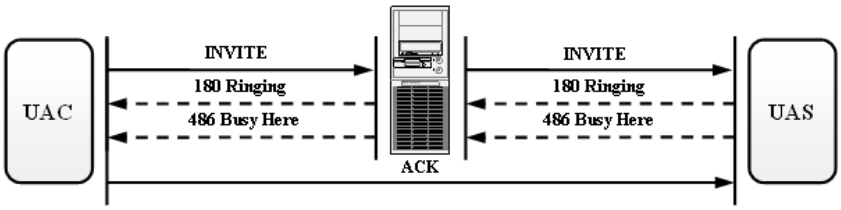
การทดสอบการยกเลิกการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ดังแสดงในรูปที่ 5.11 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน ในการยกเลิกการสร้างเซสชันเครื่องโทรศัพท์ต้องสามารถสร้างข้อความ CANCEL ให้กับเครื่องแม่ข่าย Proxy และเมื่อได้รับข้อความตอบสนอง 478 Request Canceled และ 200 OK จากเครื่องแม่ข่ายแล้ว เครื่องโทรศัพท์ต้องสามารถประมวลผลข้อความตอบสนองนั้น และส่งข้อความ ACK ไปยังเครื่องปลายทางโดยตรงได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกขอสร้างเซสชัน เมื่อได้รับข้อความ CANCEL จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์ต้องสามารถตอบสนองโดยสร้างข้อความ 487 Request Canceled และ 200 OK กลับไปได้อย่างถูกต้อง



รูปที่ 5.11 การทดสอบการยกเลิกการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy

5.3.4 การปฏิเสธการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy

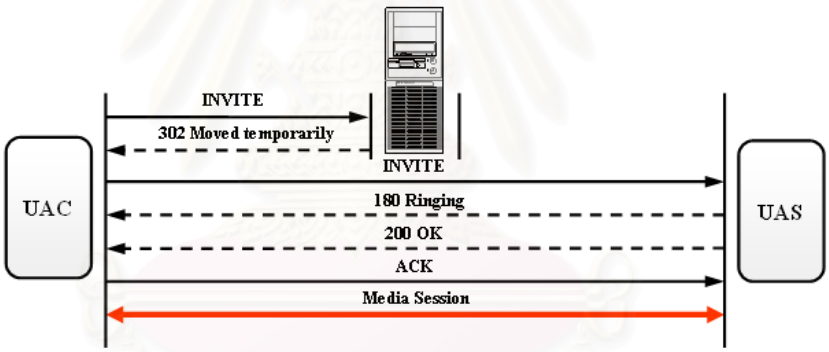
การทดสอบการปฏิเสธการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ดังแสดงในรูปที่ 5.12 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน เมื่อได้รับข้อความตอบสนอง 486 Busy Here จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์ต้องสามารถตอบสนองโดยสร้างข้อความ ACK และส่งกลับไปยังเครื่องปลายทางโดยตรงได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกขอสร้างเซสชัน เมื่อต้องการปฏิเสธการสร้างเซสชันต้องสามารถสร้างข้อความตอบสนอง 486 Busy Here กลับไปให้เครื่องแม่ข่าย Proxy ได้อย่างถูกต้อง



รูปที่ 5.12 การทดสอบการปฏิเสธการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy

5.3.5 การสร้างเซสชันโดยใช้เครื่องแม่ข่าย Redirect

การทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Redirect ดังแสดงในรูปที่ 5.13 ในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน เครื่องโทรศัพท์ต้องสามารถส่งข้อความร้องขอ INVITE ให้กับเครื่องแม่ข่าย Redirect ได้อย่างถูกต้อง และเมื่อได้รับข้อความตอบสนอง 302 Moved Temporarily เครื่องโทรศัพท์ต้องสามารถประมวลผล และส่งข้อความร้องขอ INVITE ใหม่ไปยังเครื่องปลายทางโดยตรงได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอนั้นจะไม่ทดสอบเนื่องจากมีกระบวนการทำงานเหมือนการติดต่อโดยตรงทุกประการ

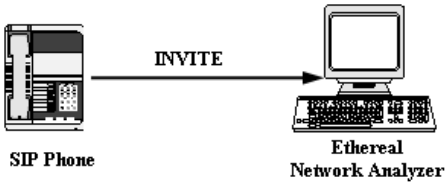


รูปที่ 5.13 การทดสอบการสร้างเซสชันโดยใช้เครื่องแม่ข่าย Redirect

5.4 การทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง

ในการทดสอบได้ต่อเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นเข้ากับเครื่องคอมพิวเตอร์ที่ลงโปรแกรม Ethereal ดังแสดงในรูปที่ 5.14 จากนั้นตั้งไอพีแอดเดรสของเครื่องโทรศัพท์ และเครื่องคอมพิวเตอร์ให้อยู่ภายในโครงข่ายพื้นที่ท้องถิ่นเดียวกัน แล้วให้เครื่องโทรศัพท์ขอสร้างเซสชันไปยังเครื่องคอมพิวเตอร์โดยตรง เครื่องคอมพิวเตอร์จะไม่ตอบสนองการขอเปิดเซสชันเนื่องจากไม่ได้เปิดโปรแกรมโทรศัพท์อินเทอร์เนตรองรับไว้ แต่จะตรวจจับข้อความร้องขอที่ได้รับจากเครื่องโทรศัพท์ พร้อมกับระบุเวลาที่รับข้อความร้องขอนั้นไว้ผ่านโปรแกรม Ethereal

ในการทดสอบเมื่อเครื่องโทรศัพท์ไม่ได้รับข้อความตอบสนองจากปลายทางตามช่วงเวลาที่กำหนด จะต้องสามารถส่งข้อความร้องขอไปยังปลายทางซ้ำด้วยวิธี Exponential Backoff โดยมี T1 = 500 ms และ T2 = 4s ตามลำดับ



รูปที่ 5.14 การทดสอบการส่งซ้ำ

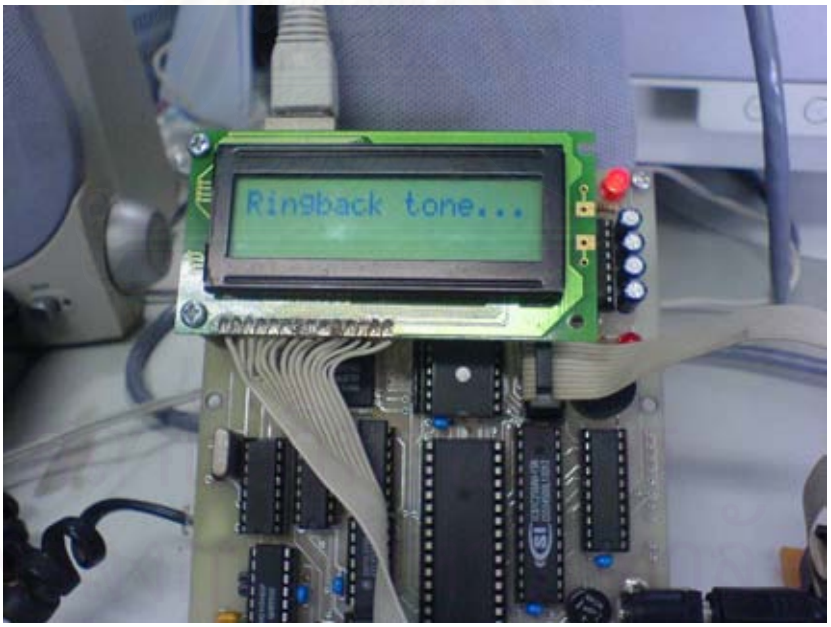
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 6
ผลการทดสอบ และการวิเคราะห์ผลการทดสอบ

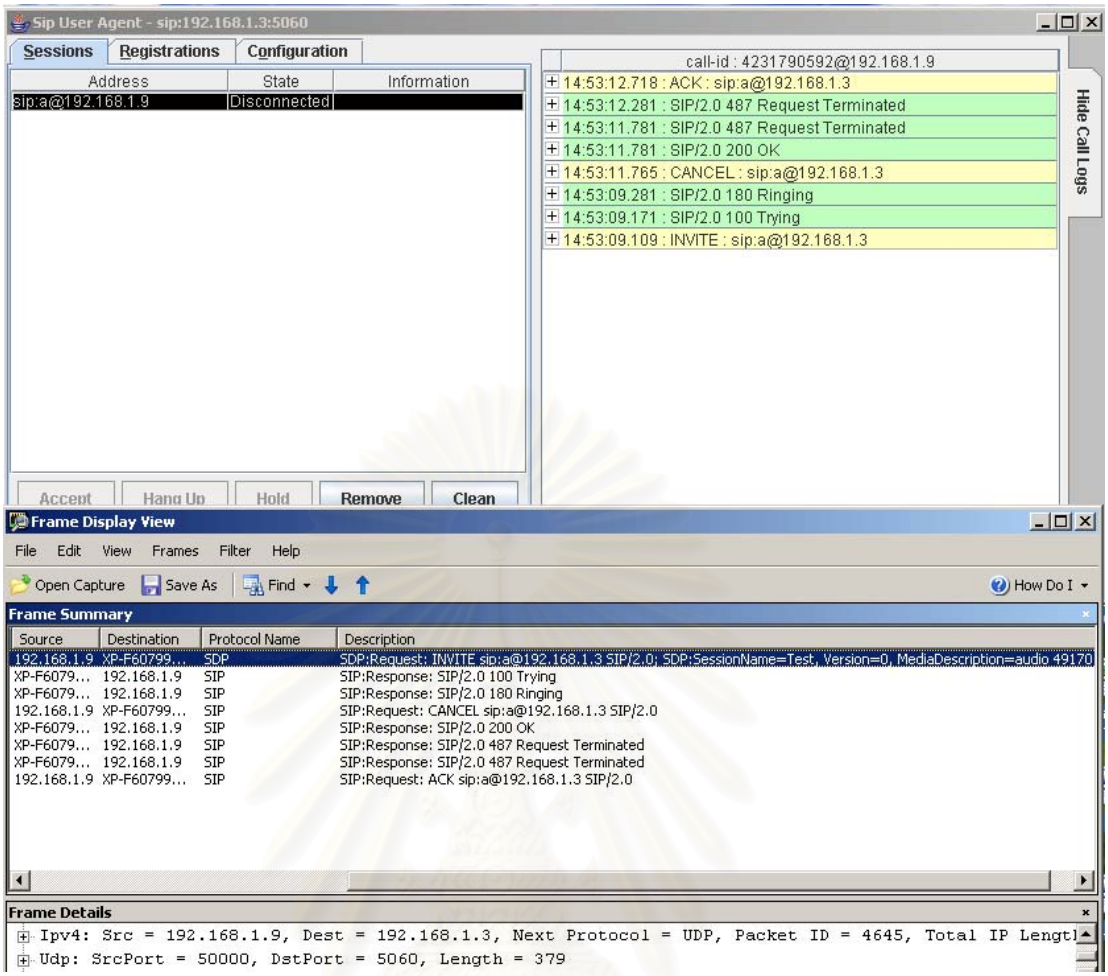
จากการทดสอบตามการทดลองต่างๆของระบบซึ่งถูกติดตั้งแสดงตัวอย่างดังรูปที่ 6.1, 6.2 และ 6.3 จะสามารถตรวจสอบความถูกต้องของการทำงานของเครื่องโทรศัพท์ต้นแบบได้โดยอาศัยบันทึก Signaling ต่างๆที่เกิดขึ้นระหว่างการทำงานของ SIP Protocol ซึ่งถูกบันทึกโดยโปรแกรม SIP User Agent และโปรแกรม Ethereal แสดงตัวอย่างดังรูปที่ 6.4 โดยรายละเอียดของ Signaling แสดงตัวอย่างในรูปที่ 6.5 โดยผลการทดสอบต่างๆได้ดังต่อไปนี้

รูปที่ 6.1 เครื่องต้นแบบของโทรศัพท์อินเทอร์เน็ตที่พัฒนาขึ้น

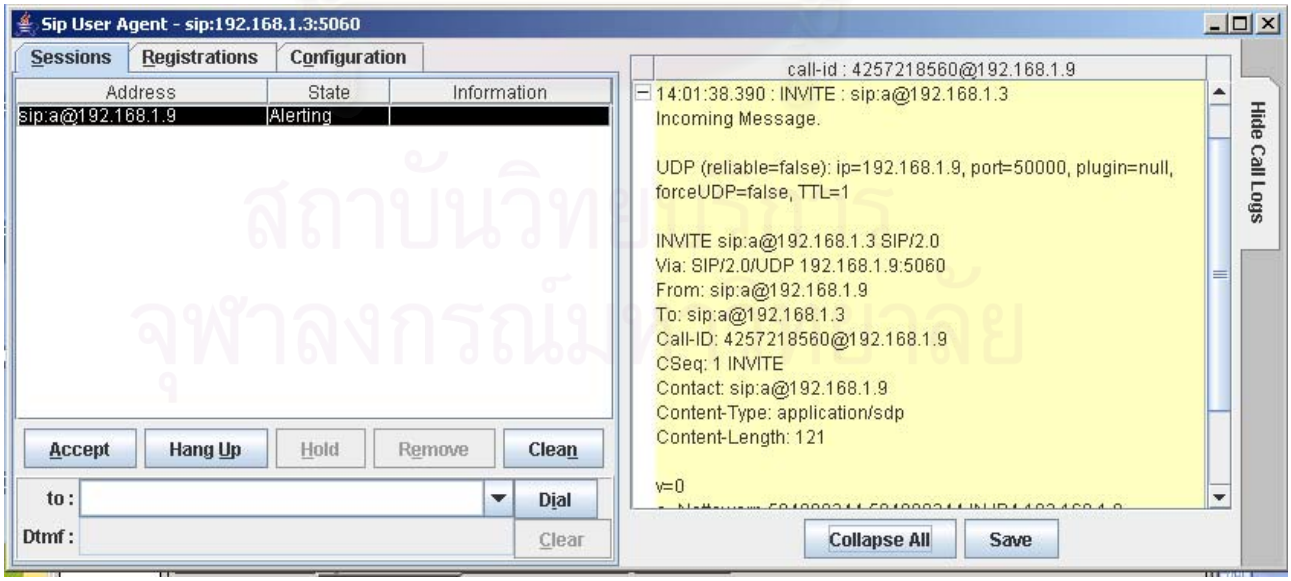
รูปที่ 6.2 การทดสอบเครื่องต้นแบบโทรศัพท์อินเทอร์เน็ตที่พัฒนาขึ้น



รูปที่ 6.3 เครื่องต้นแบบโทรศัพท์อินเทอร์เน็ตขณะดำเนินการทดสอบ



รูปที่ 6.4 ตัวอย่าง Message ที่บันทึกได้จากการทดสอบการทำงานของเครื่องต้นแบบ



รูปที่ 6.5 รายละเอียดของ Message ที่บันทึกได้จากการทดสอบการทำงานของเครื่องต้นแบบ

6.1 ผลการทดสอบการทำงานในโมดติดต่อโดยตรง

6.1.1 การสร้างเซสชัน

รูปที่ 6.6 การทดสอบการตอบสนองการขอสร้างเซสชันกับโปรแกรม SIP User Agent

ตารางที่ 6.1 ผลการทดสอบการตอบสนองการขอสร้างเซสชันกับโปรแกรม SIP User Agent

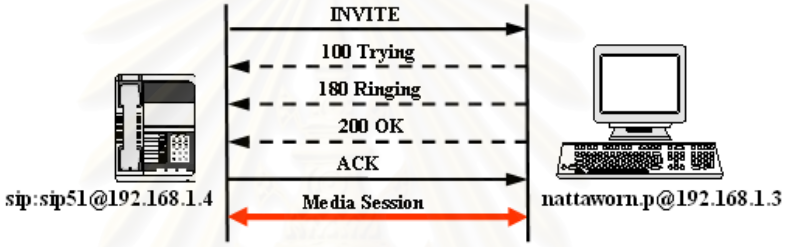
Request messages	Response messages
INVITE sip:sip51@192.168.1.4 SIP/2.0 **CSeq: 3 INVITE** **To: sip:sip51@192.168.1.4** **From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=-66378746** **Call-ID: -1132002475211864l1@192.168.1.3** **Via: SIP/2.0/UDP 192.168.1.3:5060** User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.3> Subject: no subject Content-Type: application/sdp Content-Length: 119 v=0 o=- 1133795177028 1133795177028 IN IP4 192.168.1.3 s=- **c=IN IP4 192.168.1.3** t=0 0 **m=audio 5004 RTP/AVP 8 3 0**	**SIP/2.0 180 Ringing** **Via: SIP/2.0/UDP 192.168.1.3:5060** **From: sip:nattaworn.p@192.168.1.3** **To: sip:sip51@192.168.1.4** **Call-ID: -1132002475211864l1@192.168.1.3** **CSeq: 3 INVITE** **Content-Length: 0**
ACK sip:sip51@192.168.1.4 SIP/2.0 CSeq: 3 ACK To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=-66378746 Call-ID: -1132002475211864l1@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**SIP/2.0 200 OK** **Via: SIP/2.0/UDP 192.168.1.3:5060** **From: sip:nattaworn.p@192.168.1.3** **To: sip:sip51@192.168.1.4** **Call-ID: -1132002475211864l1@192.168.1.3** **CSeq: 3 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 1636171776 1636171776 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**

การทดสอบการสร้างเซสชันระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ที่ 6.6 และตารางที่ 6.1 ได้ผลดังลำดับต่อไปนี้

1. โปรแกรม SIP User Agent ร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ซึ่งบ่งบอกคุณลักษณะการส่งข้อมูล SIP, SIP URL ต้นทาง, SIP URL ปลายทาง, ลำดับการร้องขอ, Call-ID, ไอพีแอดเดรสต้นทาง, ชนิดการเข้ารหัสเสียงที่โปรแกรม SIP User Agent รองรับ, และพอร์ตที่จะใช้รับข้อมูลเสียง

จากเฮดเดอร์ Via, From, To, Cseq, Call-ID, และฟิลพ์ c และ m ภายในข้อความ SDP ตามลำดับ

2. เครื่องโทรศัพท์ที่พัฒนาขึ้นตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 180 Ringing
3. เมื่อกดปุ่ม “Yes” เพื่อตอบรับการขอสร้างเซสชัน เครื่องโทรศัพท์ตอบสนองการร้องขอด้วยข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งมีฟิลด์ m บ่งบอกชนิดการเข้ารหัสเสียงที่เครื่องโทรศัพท์สามารถรองรับได้ และพอร์ต UDP ที่เครื่องโทรศัพท์ใช้รับข้อมูลเสียง
4. โปรแกรม SIP User Agent ตอบรับข้อความตอบสนอง 200 OK นั้นด้วยข้อความร้องขอ ACK และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน



รูปที่ 6.7 การทดสอบการขอสร้างเซสชันกับโปรแกรม SIP User Agent

ตารางที่ 6.2 ผลการทดสอบการขอสร้างเซสชันกับโปรแกรม SIP User Agent

Request messages	Response messages
INVITE sip:nattaworn.p@192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:nattaworn.p@192.168.1.3** **Call-ID: 3890610176@192.168.1.4** **CSeq: 3 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 3275620352 3275620352 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	SIP/2.0 100 Trying To: sip:nattaworn.p@192.168.1.3;tag=-48267952 From: sip:sip51@192.168.1.4 CSeq: 3 INVITE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
	SIP/2.0 180 Ringing To: sip:nattaworn.p@192.168.1.3;tag=-48267952 From: sip:sip51@192.168.1.4 CSeq: 3 INVITE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
Request messages (Continue)	**Response messages (Continue)**
ACK sip:nattaworn.p@192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:nattaworn.p@192.168.1.3;tag=-48267952** **Call-ID: 3890610176@192.168.1.4** **CSeq: 3 ACK** **Content-Length: 0**	**SIP/2.0 200 OK** To: sip:nattaworn.p@192.168.1.3;tag=-48267952 From: sip:sip51@192.168.1.4 CSeq: 3 INVITE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 Content-Length: 117 Content-Type: application/sdp Content-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.3> User-Agent: UbiquityUserAgent/4

	v=0 o=- 1133795177028 1133795177028 IN IP4 192.168.1.3 s=- **c=IN IP4 192.168.1.3** t=0 0 **m=audio 5004 RTP/AVP 0 8**

การทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ที่ 6.7 และตารางที่ 6.2 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ซึ่งบ่งบอกคุณลักษณะการส่งข้อมูล SIP, SIP URL ต้นทาง, SIP URL ปลายทาง, ลำดับการร้องขอ, Call-ID, ไอพีแอดเดรสต้นทาง, ชนิดการเข้ารหัสเสียงที่เครื่องโทรศัพท์รองรับ, และพอร์ตที่เครื่องโทรศัพท์ใช้รับข้อมูลเสียง จากเฮดเดอร์ Via, From, To, Cseq, Call-ID, และฟิลด์ c และ m ภายในข้อความ SDP ตามลำดับ
2. โปรแกรม SIP User Agent ตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 100 Trying และ 180 Ringing
3. โปรแกรม SIP User Agent ยอมรับการขอสร้างเซสชันโดยการส่งข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งมีฟิลด์ m บ่งบอกชนิดการเข้ารหัสเสียงที่โปรแกรม SIP User Agent สามารถรองรับได้ และพอร์ต UDP ที่โปรแกรม SIP User Agent ใช้รับข้อมูลเสียง
4. เครื่องโทรศัพท์ตอบรับข้อความตอบสนอง 200 OK นั้นด้วยข้อความร้องขอ ACK และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน

รูปที่ 6.8 การทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น

ตารางที่ 6.3 ผลการทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น

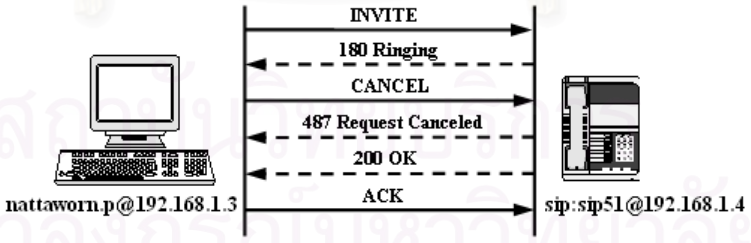
Request messages	Response messages
INVITE sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 4257218560@192.168.1.4 CSeq: 1 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 v=0 o=Nattaworn 504889344 504889344 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	**SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 4257218560@192.168.1.4 CSeq: 1 INVITE Content-Length: 0**
ACK sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 4257218560@192.168.1.4	**SIP/2.0 200 OK Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 4257218560@192.168.1.4**

CSeq: 1 ACK **Content-Length: 0**	**CSeq: 1 INVITE** **Contact: sip:asip51@192.168.1.5** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 3890610176 3890610176 IN IP4 192.168.1.5** **s=Test** **c=IN IP4 192.168.1.5** **t=0 0** **m=audio 49170 RTP/AVP 0 8**

การทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ที่ 6.8 และตารางที่ 6.3 ได้ผลดังต่อไปนี้

1. เครื่องโทรศัพท์เครื่องแรกร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP เพื่อบ่งบอกคุณลักษณะการส่งข้อมูล SIP, SIP URL ต้นทาง, SIP URL ปลายทาง, ลำดับการร้องขอ, Call-ID, ไอพีแอดเดรสต้นทาง, ชนิดการเข้ารหัสเสียงที่เครื่องโทรศัพท์รองรับ, และพอร์ตที่เครื่องโทรศัพท์ใช้รับข้อมูลเสียง
2. เครื่องโทรศัพท์เครื่องที่สองตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 180 Ringing
3. เมื่อเครื่องโทรศัพท์ที่ 2 ยอมรับการขอสร้างเซสชัน เครื่องโทรศัพท์เครื่องที่ 2 ตอบสนองการขอสร้างเซสชันด้วยข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP
4. เครื่องโทรศัพท์เครื่องแรกตอบรับข้อความตอบสนอง 200 OK นั้นด้วยข้อความร้องขอ ACK และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน

6.1.2 การยกเลิกการสร้างเซสชัน



รูปที่ 6.9 การทดสอบการตอบสนองการขอยกเลิกการสร้างเซสชันจากโปรแกรม SIP User Agent

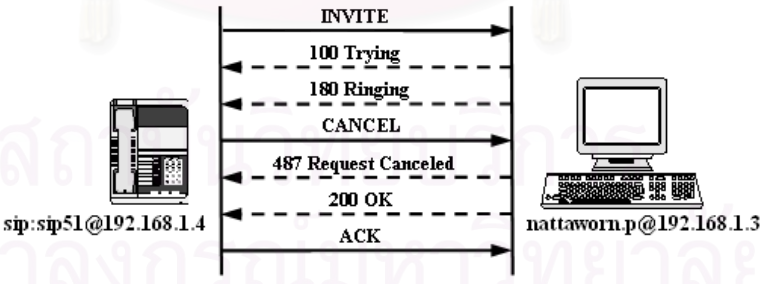
ตารางที่ 6.4 ผลการทดสอบการตอบสนองการขอยกเลิกการสร้างเซสชันจากโปรแกรม SIP User Agent

Request messages	Response messages
INVITE sip:sip51@192.168.1.4 SIP/2.0 CSeq: 1 INVITE To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=1373457655 Call-ID: -400449051-1308826699@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.3> Subject: no subject	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 From: sip:nattaworn.p@192.168.1.3 To: sip:sip51@192.168.1.4 Call-ID: -400449051-1308826699@192.168.1.3 CSeq: 1 INVITE Content-Length: 0

Content-Type: application/sdp Content-Length: 119 v=0 o=- 1133795130351 1133795130351 IN IP4 192.168.1.3 s=- c=IN IP4 192.168.1.3 t=0 0 m=audio 5004 RTP/AVP 8 3 0	
CANCEL sip:sip51@192.168.1.4 SIP/2.0 CSeq: 1 CANCEL To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=1373457655 Call-ID: -400449051-1308826699@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Content-Length: 0	**SIP/2.0 487 Request Canceled** **Via: SIP/2.0/UDP 192.168.1.3:5060** **From: sip:nattaworn.p@192.168.1.3** **To: sip:sip51@192.168.1.4** **Call-ID: -400449051-1308826699@192.168.1.3** **CSeq: 1 INVITE** **Content-Length: 0**
ACK sip:sip51@192.168.1.4 SIP/2.0 CSeq: 1 ACK To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=1373457655 Call-ID: -400449051-1308826699@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**SIP/2.0 200 OK** **Via: SIP/2.0/UDP 192.168.1.3:5060** **From: sip:nattaworn.p@192.168.1.3** **To: sip:sip51@192.168.1.4** **Call-ID: -400449051-1308826699@192.168.1.3** **CSeq: 1 CANCEL** **Content-Length: 0**

การทดสอบการยกเลิกการขอสร้างเซสชันระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น ดังแสดงในรูปที่ 6.9 และตารางที่ 6.4 ได้ผลดังลำดับต่อไปนี้

1. เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ CANCEL ในขณะที่กำลังตัดสินใจเลือกการตอบรับการร้องขอสร้างเซสชัน เครื่องโทรศัพท์ตอบสนองข้อความร้องขอ INVITE และ CANCEL ด้วยข้อความตอบสนอง 487 Request Canceled และ 200 OK ตามลำดับ
2. เมื่อโปรแกรม SIP User Agent ได้รับข้อความตอบสนอง 487 Request Canceled โปรแกรม SIP User Agent ตอบรับข้อความตอบสนองนั้นด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์รับรู้ว่าการยกเลิกการสร้างเซสชันได้สำเร็จเรียบร้อย



รูปที่ 6.10 การทดสอบการขอยกเลิกการสร้างเซสชันกับโปรแกรม SIP User Agent

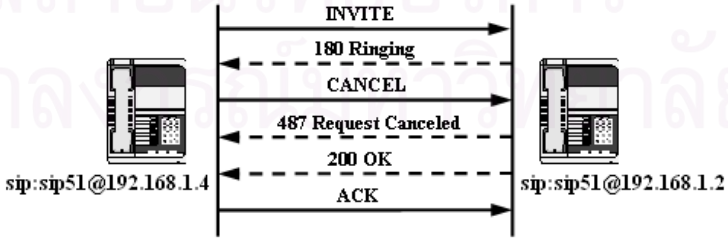
ตารางที่ 6.5 ผลการทดสอบการขอยกเลิกการสร้างเซสชันกับโปรแกรม SIP User Agent

Request messages	Response messages
INVITE sip:nattaworn.p@192.168.1.3 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:nattaworn.p@192.168.1.3 Call-ID: 4257218560@192.168.1.4 CSeq: 1 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 v=0	SIP/2.0 100 Trying To: sip:nattaworn.p@192.168.1.3;tag=1289318417 From: sip:sip51@192.168.1.4 CSeq: 1 INVITE Call-ID: 4257218560@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0

o=Nattaworn 504889344 504889344 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	
	SIP/2.0 180 Ringing To: sip:nattaworn.p@192.168.1.3;tag=1289318417 From: sip:sip51@192.168.1.4 CSeq: 1 INVITE Call-ID: 4257218560@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
CANCEL sip:nattaworn.p@192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:nattaworn.p@192.168.1.3** **Call-ID: 4257218560@192.168.1.4** **CSeq: 1 CANCEL** **Content-Length: 0**	**SIP/2.0 487 Request Terminated** To: sip:nattaworn.p@192.168.1.3;tag=1289318417 From: sip:sip51@192.168.1.4 CSeq: 1 INVITE Call-ID: 4257218560@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
ACK sip:nattaworn.p@192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:nattaworn.p@192.168.1.3;tag=1289318417** **Call-ID: 4257218560@192.168.1.4** **CSeq: 1 ACK** **Content-Length: 0**	**SIP/2.0 200 OK** To: sip:nattaworn.p@192.168.1.3;tag=1289318417 From: sip:sip51@192.168.1.4 CSeq: 1 CANCEL Call-ID: 4257218560@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0

การทดสอบการยกเลิกการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ 6.10 และตารางที่ 6.5 ดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ได้ส่งข้อความร้องขอ CANCELเมื่อปุ่ม “NO” ถูกกดในขณะที่เครื่องโทรศัพท์กำลังรอการตอบรับการสร้างเซสชันจากโปรแกรม SIP User Agent
2. โปรแกรม SIP User Agent ตอบสนองข้อความร้องขอ INVITE และ CANCEL นั้นด้วยข้อความตอบสนอง 487 Request Canceled และ 200 OK ตามลำดับ
3. เมื่อเครื่องโทรศัพท์ได้รับข้อความตอบสนอง 200 OK เครื่องโทรศัพท์ตอบรับด้วยข้อความ ACK เพื่อให้โปรแกรม SIP User Agent รับรู้ว่าการยกเลิกการสร้างเซสชันได้สำเร็จเรียบร้อย



รูปที่ 6.11 การทดสอบการยกเลิกการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น

ตารางที่ 6.6 ผลการทดสอบการยกเลิกการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น

Request messages	Response messages
INVITE sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 4162322432@192.168.1.4 CSeq: 2 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 4245422080 4245422080 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 4162322432@192.168.1.4 CSeq: 2 INVITE Content-Length: 0
CANCEL sip:asip51@192.168.1.5 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 4162322432@192.168.1.4** **CSeq: 2 CANCEL** **Content-Length: 0**	**SIP/2.0 487 Request Canceled** **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 4162322432@192.168.1.4** **CSeq: 2 INVITE** **Content-Length: 0**
ACK sip:asip51@192.168.1.5 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 4162322432@192.168.1.4** **CSeq: 2 ACK** **Content-Length: 0**	**SIP/2.0 200 OK** **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 4162322432@192.168.1.4** **CSeq: 2 CANCEL** **Content-Length: 0**

การทดสอบการยกเลิกการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.11 และตารางที่ 6.6 ดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ส่งข้อความร้องขอ CANCEL เพื่อขอยกเลิกการสร้างเซสชันเมื่อกดปุ่ม “NO” ในขณะที่เครื่องโทรศัพท์รอการตอบรับการขอสร้างเซสชันจากเครื่องโทรศัพท์อีกเครื่อง
2. เครื่องโทรศัพท์อีกเครื่องตอบสนองข้อความร้องขอ INVITE และ CANCEL ด้วยข้อความตอบสนอง 487 Request Canceled และ 200 OK ตามลำดับ
3. เมื่อเครื่องโทรศัพท์ได้รับข้อความตอบสนอง 200 OK เครื่องโทรศัพท์ตอบรับด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์อีกเครื่องรับรู้ว่าการยกเลิกการสร้างเซสชันได้สำเร็จเรียบร้อย

6.1.3 การปฏิเสธการขอสร้างเซสชัน

รูปที่ 6.12 การทดสอบการปฏิเสธการขอสร้างเซสชันกับโปรแกรม SIP User Agent

ตารางที่ 6.7 ผลการทดสอบการปฏิเสธการขอสร้างเซสชันกับโปรแกรม SIP User Agent

Request messages	Response messages
INVITE sip:sip51@192.168.1.4 SIP/2.0 CSeq: 2 INVITE To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=-764670695 Call-ID: -14514060091868510410@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.3> Subject: no subject Content-Type: application/sdp Content-Length: 119 v=0 o=- 1133795138823 1133795138823 IN IP4 192.168.1.3 s=- c=IN IP4 192.168.1.3 t=0 0 m=audio 5004 RTP/AVP 8 3 0	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 From: sip:nattaworn.p@192.168.1.3 To: sip:sip51@192.168.1.4 Call-ID: -14514060091868510410@192.168.1.3 CSeq: 2 INVITE Content-Length: 0
ACK sip:sip51@192.168.1.4 SIP/2.0 CSeq: 2 ACK To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=-764670695 Call-ID: -14514060091868510410@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**SIP/2.0 486 Busy Here** **Via: SIP/2.0/UDP 192.168.1.3:5060** **From: sip:nattaworn.p@192.168.1.3** **To: sip:sip51@192.168.1.4** **Call-ID: -14514060091868510410@192.168.1.3** **CSeq: 2 INVITE** **Content-Length: 0**

การทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.12 และตารางที่ 6.7 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ส่งข้อความตอบสนอง 486 Busy Here เมื่อปุ่ม “NO” ถูกกดหลังจากที่เครื่องโทรศัพท์ได้รับข้อความร้องขอ INVITE จากโปรแกรม SIP User Agent
2. โปรแกรม SIP User Agent ตอบรับการตอบสนองนั้นด้วยด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์รับรู้ว่าการปฏิเสธการสร้างเซสชันได้สำเร็จเรียบร้อย

รูปที่ 6.13 การทดสอบการรองรับการปฏิเสธการขอสร้างเซสชันจากโปรแกรม SIP User Agent

ตารางที่ 6.8 ผลการทดสอบการรองรับการปฏิเสธการขอสร้างเซสชันจากโปรแกรม SIP User Agent

Request messages	Response messages
INVITE sip:nattaworn.p@192.168.1.3 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:nattaworn.p@192.168.1.3 Call-ID: 4231790592@192.168.1.4 CSeq: 2 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 1636171776 1636171776 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	SIP/2.0 100 Trying To: sip:nattaworn.p@192.168.1.3;tag=1839907368 From: sip:sip51@192.168.1.4 CSeq: 2 INVITE Call-ID: 4231790592@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0

	SIP/2.0 180 Ringing To: sip:nattaworn.p@192.168.1.3;tag=1839907368 From: sip:sip51@192.168.1.4 CSeq: 2 INVITE Call-ID: 4231790592@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
ACK sip:nattaworn.p@192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: Nattaworn <sip:sip51@192.168.1.4>** **To: nattaworn.p <sip:nattaworn.p@192.168.1.3>;tag=-1743382141** **Call-ID: 3275620352@192.168.1.4** **CSeq: 2 ACK** **Content-Length: 0**	**SIP/2.0 486 Busy Here** To: sip:nattaworn.p@192.168.1.3;tag=1839907368 From: sip:sip51@192.168.1.4 CSeq: 2 INVITE Call-ID: 4231790592@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0

การทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ 6.13 และตารางที่ 6.8 ได้ผลดังลำดับต่อไปนี้

1. โปรแกรม SIP User Agent ส่งข้อความตอบสนอง 486 Busy Here เพื่อปฏิเสธการขอสร้างเซสชัน
2. เครื่องโทรศัพท์ตอบรับการตอบสนองนั้นด้วยข้อความ ACK เพื่อให้โปรแกรม SIP User Agent รับรู้ว่าการปฏิเสธการสร้างเซสชันได้สำเร็จเรียบร้อย

รูปที่ 6.14 การทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น

ตารางที่ 6.9 ผลการทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น

Request messages	Response messages
INVITE sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 3890610176@192.168.1.4 CSeq: 3 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 3275620352 3275620352 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@192.168.1.4 To: sip:asip51@192.168.1.5 Call-ID: 3890610176@192.168.1.4 CSeq: 3 INVITE Content-Length: 0
Request messages (Continue)	**Response messages (Continue)**
ACK sip:asip51@192.168.1.5 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 3890610176@192.168.1.4** **CSeq: 3 ACK** **Content-Length: 0**	**SIP/2.0 486 Busy Here** **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 3890610176@192.168.1.4** **CSeq: 3 INVITE** **Content-Length: 0**

การทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น ดังแสดงในรูปที่ 6.14 และตารางที่ 6.9 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ปลายทางส่งข้อความตอบสนอง 486 Busy Here เมื่อปุ่ม “NO” ถูกกดหลัง จากที่เครื่องโทรศัพท์ปลายทางได้รับข้อความร้องขอ INVITE จากเครื่องโทรศัพท์ต้นทาง
2. เครื่องโทรศัพท์ต้นทางตอบรับการตอบสนองนั้นด้วยด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์ปลายทางรับรู้ว่าการปฏิเสธการสร้างเซสชันสำเร็จเรียบร้อย

6.1.4 การขอสิ้นสุดเซสชัน

รูปที่ 6.15 การทดสอบการตอบสนองการขอสิ้นสุดเซสชันจากโปรแกรม SIP User Agent

ตารางที่ 6.10 ผลการทดสอบการตอบสนองการขอสิ้นสุดเซสชันจากโปรแกรม SIP User Agent

Request messages	Response messages
BYE sip:sip51@192.168.1.4 SIP/2.0 CSeq: 4 BYE To: sip:sip51@192.168.1.4 From: nattaworn.p<sip:nattaworn.p@192.168.1.3>;tag=-66378746 Call-ID: -1132002477521186411@192.168.1.3 Via: SIP/2.0/UDP 192.168.1.3:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Content-Length: 0	**SIP/2.0 200 OK** **Via: SIP/2.0/UDP 192.168.1.3:5060** **From: sip:nattaworn.p@192.168.1.3** **To: sip:sip51@192.168.1.4** **Call-ID: -1132002477521186411@192.168.1.3** **CSeq: 4 BYE** **Content-Length: 0**

การทดสอบการสิ้นสุดเซสชันระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น ดังแสดงในรูปที่ 6.15 และตารางที่ 6.10 ได้ผลดังลำดับต่อไปนี้

1. โปรแกรม SIP User Agent ส่งข้อความร้องขอ BYE เพื่อขอสิ้นสุดเซสชัน
2. เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ BYE ในขณะที่เปิดเซสชันอยู่ เครื่องโทรศัพท์ตอบสนองข้อความร้องขอ BYE นั้นด้วยข้อความตอบสนอง 200 OK เพื่อตกลงสิ้นสุดเซสชัน

รูปที่ 6.16 การทดสอบการขอสิ้นสุดเซสชันกับโปรแกรม SIP User Agent

ตารางที่ 6.11 ผลการทดสอบการขอสิ้นสุดเซสชันกับโปรแกรม SIP User Agent

Request messages	Response messages
BYE sip:nattaworn.p@192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:nattaworn.p@192.168.1.3;tag=-48267952** **Call-ID: 3890610176@192.168.1.4** **CSeq: 4 BYE** **Content-Length: 0**	**SIP/2.0 200 OK** To: sip:nattaworn.p@192.168.1.3;tag=-48267952 From: sip:sip51@192.168.1.4 CSeq: 4 BYE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0

การทดสอบการสิ้นสุดเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ 6.16 และตารางที่ 6.11 เครื่องโทรศัพท์ส่งข้อความร้องขอ BYE เมื่อปุ่ม “NO” ถูกกดในขณะที่เครื่องโทรศัพท์กำลังเปิดเซสชันอยู่ และโปรแกรม SIP User Agent ตอบสนองข้อความร้องขอ BYE นั้นด้วยข้อความตอบสนอง 200 OK เพื่อตกลงสิ้นสุดเซสชัน

รูปที่ 6.17 การทดสอบการสิ้นสุดเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น

ตารางที่ 6.12 ผลการทดสอบการสิ้นสุดเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น

Request messages	Response messages
BYE sip:asip51@192.168.1.5 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 4257218560@192.168.1.4** **CSeq: 2 BYE** **Content-Length: 0**	**SIP/2.0 200 OK** **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@192.168.1.4** **To: sip:asip51@192.168.1.5** **Call-ID: 4257218560@192.168.1.4** **CSeq: 2 BYE** **Content-Length: 0**

การทดสอบการสิ้นสุดเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.17 และตารางที่ 6.12 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ต้นทางส่งข้อความร้องขอ BYE เมื่อปุ่ม “NO” ถูกกดในขณะที่เครื่องโทรศัพท์กำลังเปิดเซสชันอยู่
2. เครื่องโทรศัพท์ปลายทางตอบสนองข้อความร้องขอ BYE นั้นด้วยข้อความตอบสนอง 200 OK เพื่อตกลงสิ้นสุดเซสชัน

6.2 ผลการทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย

6.2.1 การลงทะเบียนกับเครื่องแม่ข่าย

รูปที่ 6.18 การทดสอบการลงทะเบียนกับเครื่องแม่ข่าย

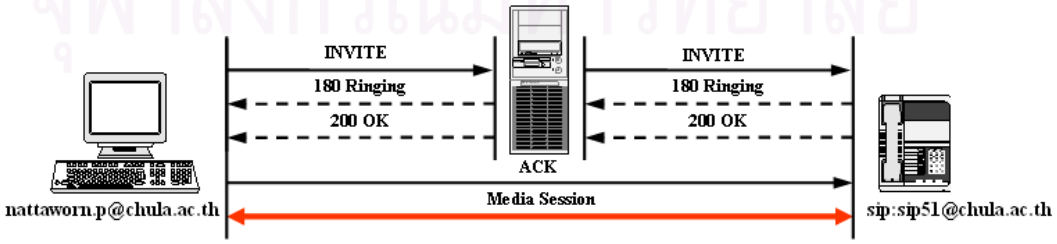
ตารางที่ 6.13 ผลการทดสอบการลงทะเบียนกับเครื่องแม่ข่าย

Request messages	Response messages
REGISTER sip:192.168.1.3 SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:sip51@chula.ac.th** **Call-ID: 4257218560@192.168.1.4** **CSeq: 1 REGISTER** **Contact: sip:sip51@192.168.1.4** **Expires: 3600** **Content-Length: 0**	**SIP/2.0 200 OK** Call-ID: 4257218560@192.168.1.4 CSeq: 1 REGISTER From: sip:sip51@chula.ac.th To: sip:sip51@chula.ac.th;tag=f47d949370c6b74 Via: SIP/2.0/UDP 192.168.1.4:5060 Content-Length: 0 Contact: sip:sip51@192.168.1.4;expires=3600 Date: Mon, 05 Dec 2005 15:31:29 GMT

การทดสอบการลงทะเบียนกับเครื่องแม่ข่ายดังแสดงในรูปที่ 6.18 และตารางที่ 6.13 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ส่งข้อความร้องขอ REGISTER ซึ่งประกอบด้วยเฮดเดอร์ Via, From, To, Call-ID, Cseq, Contact, Expires, และ Content-Length เพื่อบ่งบอกคุณลักษณะการส่งข้อมูล SIP, SIP URL ของตัวเครื่อง, ลำดับการร้องขอ, Call-ID, ไอพีแอดเดรสของเครื่อง
2. เครื่องแม่ข่ายตอบสนองข้อความร้องขอ REGISTER ที่ได้รับจากเครื่องโทรศัพท์ด้วยข้อความตอบสนอง 200 OK

6.2.2 การสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy



รูปที่ 6.19 การทดสอบการตอบสนองการขอสร้างเซสชันจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

ตารางที่ 6.14 ผลการทดสอบการตอบสนองการขอสร้างเซสชันจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

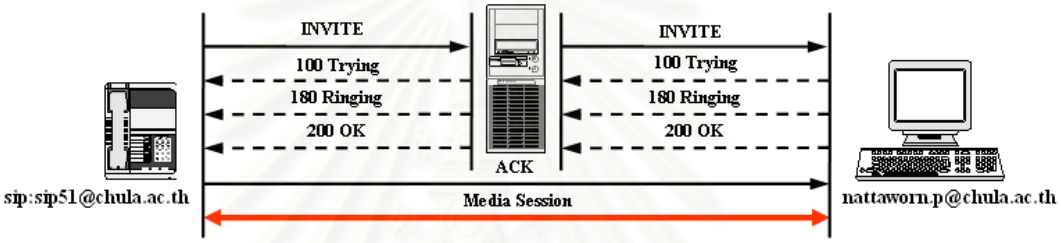
Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:sip51@chula.ac.th SIP/2.0 CSeq: 4 INVITE To: sip:sip51@chula.ac.th From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=-1084079553 Call-ID: -1664972726-576050387@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> Subject: no subject Content-Type: application/sdp Content-Length: 119 v=0 o=- 1133796591171 1133796591171 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2 t=0 0 m=audio 5004 RTP/AVP 8 3 0	**INVITE sip:sip51@192.168.1.4 SIP/2.0** **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK1db278eb** **CSeq: 4 INVITE** **To: sip:sip51@chula.ac.th** **From: nattaworn.p <sip:nattaworn.p@chula.ac.th>;tag=-1084079553** **Call-ID: -1664972726-576050387@192.168.1.2** **Via: SIP/2.0/UDP 192.168.1.2:5060** User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity **Contact: <sip:nattaworn.p@192.168.1.2>** Subject: no subject Content-Type: application/sdp Content-Length: 119 Max-Forwards: 70 v=0 o=- 1133796591171 1133796591171 IN IP4 192.168.1.2 s=- **c=IN IP4 192.168.1.2** t=0 0 **m=audio 5004 RTP/AVP 8 3 0**
SIP/2.0 180 Ringing **Via: SIP/2.0/UDP 192.168.1.2:5060** **From: sip:nattaworn.p@chula.ac.th** **To: sip:sip51@chula.ac.th;tag=-91095040** **Call-ID: -1664972726-576050387@192.168.1.2** **CSeq: 4 INVITE** **Content-Length: 0**	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-91095040 Call-ID: -1664972726-576050387@192.168.1.2 CSeq: 4 INVITE Content-Length: 0
SIP/2.0 200 OK **Via: SIP/2.0/UDP 192.168.1.2:5060** **From: sip:nattaworn.p@chula.ac.th** **To: sip:sip51@chula.ac.th;tag=-91095040** **Call-ID: -1664972726-576050387@192.168.1.2** **CSeq: 4 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 2155741184 2155741184 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	SIP/2.0 200 OK Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-91095040 Call-ID: -1664972726-576050387@192.168.1.2 CSeq: 4 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 2155741184 2155741184 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8
Direct request messages	**Direct response messages**
ACK sip:sip51@192.168.1.4 SIP/2.0 CSeq: 4 ACK To: sip:sip51@chula.ac.th;tag=-91095040 From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=-1084079553 Call-ID: -1664972726-576050387@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	

การทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.19 และตารางที่ 6.14 ได้ผลดังลำดับต่อไปนี้

1. โปรแกรม SIP User Agent ร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ไปยังเครื่องแม่ข่าย
2. เครื่องแม่ข่ายตรวจสอบ SIP URL ปลายทางจากเฮดเดอร์ To ของข้อความร้องขอ INVITE ที่ได้รับ เพื่อค้นหาไอพีแอดเดรสปลายทางจากฐานข้อมูล และเพิ่มเฮดเดอร์

Via ซึ่งบ่งบอกไอพีแอดเดรสของเครื่องแม่ข่ายเข้าไปในข้อความ INVITE เพื่อใช้ในการค้นหาเส้นทางกลับ และส่งข้อความ INVITE นั้นต่อไปยังเครื่องโทรศัพท์

3. เครื่องโทรศัพท์ตรวจพบเฮดเดอร์ Via 2 เฮดเดอร์ และตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 180 Ringing กลับไปโดยผ่านเครื่องแม่ข่าย
4. เมื่อกดปุ่ม “Yes” เพื่อตอบรับการขอสร้างเซสชัน เครื่องโทรศัพท์ส่งข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งซึ่งระบุการเข้ารหัสเสียงที่รองรับ และพอร์ตรับข้อมูลเสียงกลับไปโดยผ่านเครื่องแม่ข่าย
5. โปรแกรม SIP User Agent ตอบรับข้อความตอบสนอง 200 OK นั้นด้วยข้อความร้องขอ ACK โดยตรงและเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน



รูปที่ 6.20 การทดสอบการขอสร้างเซสชันกับโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

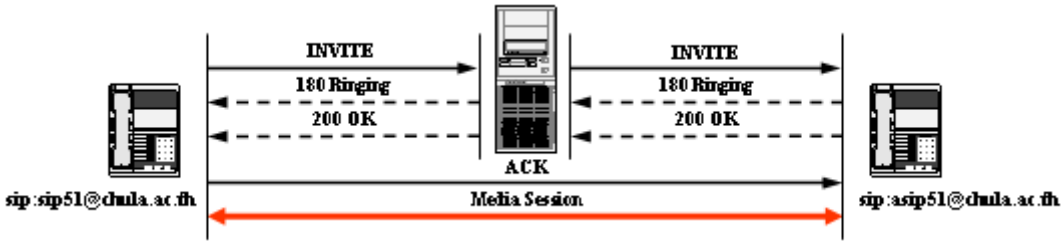
ตารางที่ 6.15 ผลการทดสอบการขอสร้างเซสชันกับโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th** **Call-ID: 838729728@192.168.1.4** **CSeq: 3 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 3858759680 3858759680 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	INVITE sip:nattaworn.p@192.168.1.2 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKc0b3e523 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Call-ID: 838729728@192.168.1.4 CSeq: 3 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 Max-Forwards: 70 v=0 o=Nattaworn 3858759680 3858759680 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8
SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=1662377084 From: sip:sip51@chula.ac.th CSeq: 3 INVITE Call-ID: 838729728@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=1662377084 From: sip:sip51@chula.ac.th CSeq: 3 INVITE Call-ID: 838729728@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKc0b3e523 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
SIP/2.0 180 Ringing To: sip:nattaworn.p@chula.ac.th;tag=1662377084 From: sip:sip51@chula.ac.th CSeq: 3 INVITE Call-ID: 838729728@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**SIP/2.0 180 Ringing** To: sip:nattaworn.p@chula.ac.th;tag=1662377084 From: sip:sip51@chula.ac.th CSeq: 3 INVITE Call-ID: 838729728@192.168.1.4 **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKc0b3e523** **Via: SIP/2.0/UDP 192.168.1.4:5060** User-Agent: UbiquityUserAgent/4 Content-Length: 0

Incoming messages to proxy server (Continue)	Outgoing messages from proxy server (Continue)
SIP/2.0 200 OK To: sip:nattaworn.p@chula.ac.th;tag=1662377084 From: sip:sip51@chula.ac.th CSeq: 3 INVITE Call-ID: 838729728@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 Content-Length: 117 Content-Type: application/sdp Content-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> User-Agent: UbiquityUserAgent/4 v=0 o=- 1133796591171 1133796591171 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2 t=0 0 m=audio 5004 RTP/AVP 0 8	**SIP/2.0 200 OK** To: sip:nattaworn.p@chula.ac.th;tag=1662377084 From: sip:sip51@chula.ac.th CSeq: 3 INVITE Call-ID: 838729728@192.168.1.4 **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKc0b3e523** **Via: SIP/2.0/UDP 192.168.1.4:5060** Content-Length: 117 Content-Type: application/sdp Content-Encoding: identity **Contact: <sip:nattaworn.p@192.168.1.2>** User-Agent: UbiquityUserAgent/4 v=0 o=- 1133796591171 1133796591171 IN IP4 192.168.1.2 s=- **c=IN IP4 192.168.1.2** t=0 0 **m=audio 5004 RTP/AVP 0 8**
Direct request messages	Direct response messages
ACK sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th;tag=1662377084** **Call-ID: 838729728@192.168.1.4** **CSeq: 3 ACK** **Content-Length: 0**	

การทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ 6.20 และตารางที่ 6.15 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ไปยังเครื่องแม่ข่าย
2. เครื่องแม่ข่ายส่งข้อความร้องขอ INVITE ที่ได้รับต่อไปให้โปรแกรม SIP User Agent โดยเพิ่มเฮดเดอร์ Via ซึ่งบ่งบอกไอพีแอดเดรสของเครื่องแม่ข่ายเข้าไปในข้อความ INVITE เพื่อให้ปลายทางทราบว่าข้อความ INVITE นี้ได้ผ่านเครื่องแม่ข่าย Proxy มาและใช้เป็นข้อมูลในส่งข้อความตอบสนองกลับ
3. โปรแกรม SIP User Agent ตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 100 Trying และ 180 Ringing กลับไปโดยผ่านเครื่องแม่ข่าย
4. เมื่อให้โปรแกรม SIP User Agent ตอบรับการขอสร้างเซสชัน โปรแกรม SIP User Agent ส่งข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งระบุการเข้ารหัสเสียงที่รองรับ และพอร์ตรับข้อมูลเสียงกลับไปโดยผ่านเครื่องแม่ข่าย
5. เครื่องโทรศัพท์ตอบรับข้อความตอบสนอง 200 OK ที่ได้รับด้วยข้อความร้องขอ ACK โดยตรง และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน



รูปที่ 6.21 การทดสอบการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Proxy

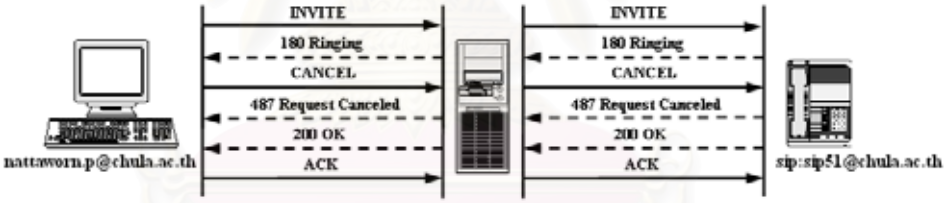
ตารางที่ 6.16 ผลการทดสอบการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Proxy

Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:asip51@chula.ac.th SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 510001152@192.168.1.4 CSeq: 1 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 v=0 o=Nattaworn 822083584 822083584 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	INVITE sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKf708065e Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 510001152@192.168.1.4 CSeq: 1 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 Max-Forwards: 70 v=0 o=Nattaworn 822083584 822083584 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8
SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1983119360 Call-ID: 510001152@192.168.1.4 CSeq: 1 INVITE Content-Length: 0	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1983119360 Call-ID: 510001152@192.168.1.4 CSeq: 1 INVITE Content-Length: 0
SIP/2.0 200 OK Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1983119360 Call-ID: 510001152@192.168.1.4 CSeq: 1 INVITE Contact: sip:asip51@192.168.1.5 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 1431401812 1431401812 IN IP4 192.168.1.5 s=Test c=IN IP4 192.168.1.5 t=0 0 m=audio 49170 RTP/AVP 0 8	SIP/2.0 200 OK Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1983119360 Call-ID: 510001152@192.168.1.4 CSeq: 1 INVITE Contact: sip:asip51@192.168.1.5 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 1431401812 1431401812 IN IP4 192.168.1.5 s=Test c=IN IP4 192.168.1.5 t=0 0 m=audio 49170 RTP/AVP 0 8
Direct request messages	**Direct response messages**
ACK sip:asip51@chula.ac.th SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1983119360 Call-ID: 510001152@192.168.1.4 CSeq: 1 ACK Content-Length: 0	

การทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ที่ 6.21 และตารางที่ 6.16 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์เครื่องแรกร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ไปยังเครื่องแม่ข่าย
2. เครื่องแม่ข่ายตรวจสอบ SIP URL ปลายทางจากเฮดเดอร์ To ของข้อความร้องขอ INVITE ที่ได้รับ เพื่อค้นหาไอพีแอดเดรสปลายทางจากฐานข้อมูล และเพิ่มเฮดเดอร์ Via ซึ่งบ่งบอกไอพีแอดเดรสของเครื่องแม่ข่ายเข้าไปในข้อความ INVITE เพื่อใช้ในการค้นหาเส้นทางกลับ และส่งข้อความ INVITE นั้นต่อไปยังเครื่องโทรศัพท์เครื่องที่ 2
3. เครื่องโทรศัพท์เครื่องที่ 2 ตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 180 Ringing กลับไปโดยผ่านเครื่องแม่ข่าย
4. เมื่อกดปุ่ม “Yes” ที่เครื่องโทรศัพท์เครื่องที่ 2 เพื่อตอบรับการขอสร้างเซสชัน เครื่องโทรศัพท์เครื่องที่ 2 ส่งข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งระบุการเข้ารหัสเสียงที่รองรับ และพอร์ตรับข้อมูลเสียงกลับไปโดยผ่านเครื่องแม่ข่าย
5. เครื่องโทรศัพท์เครื่องแรกตอบรับข้อความตอบสนอง 200 OK ที่ได้รับด้วยข้อความร้องขอ ACK โดยตรง และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน

6.2.3 การยกเลิกการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy



รูปที่ 6.22 การทดสอบการตอบสนองการขอยกเลิกการสร้างเซสชันจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

ตารางที่ 6.17 ผลการทดสอบการตอบสนองการขอยกเลิกการสร้างเซสชันจากจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

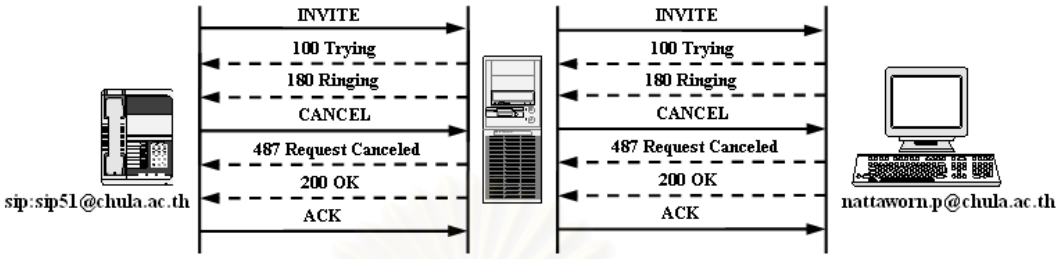
Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:sip51@chula.ac.th SIP/2.0 CSeq: 2 INVITE To: sip:sip51@chula.ac.th From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=316075616 Call-ID: 119171739962048806 9@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> Subject: no subject Content-Type: application/sdp Content-Length: 119 v=0 o=- 1133796577652 1133796577652 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2 t=0 0 m=audio 5004 RTP/AVP 8 3 0	INVITE sip:sip51@192.168.1.4 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK77386d18 CSeq: 2 INVITE To: sip:sip51@chula.ac.th From: nattaworn.p <sip:nattaworn.p@chula.ac.th>;tag=316075616 Call-ID: 119171739962048806 9@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> Subject: no subject Content-Type: application/sdp Content-Length: 119 Max-Forwards: 70 v=0 o=- 1133796577652 1133796577652 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2 t=0 0

	m=audio 5004 RTP/AVP 8 3 0
Incoming messages to proxy server (Continue)	Outgoing messages from proxy server (Continue)
SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1926627328 Call-ID: 11917173996204880 69@192.168.1.2 CSeq: 2 INVITE Content-Length: 0	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1926627328 Call-ID: 11917173996204880 69@192.168.1.2 CSeq: 2 INVITE Content-Length: 0
CANCEL sip:sip51@chula.ac.th SIP/2.0 CSeq: 2 CANCEL To: sip:sip51@chula.ac.th From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=316075616 Call-ID: 11917173996204880 69@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Content-Length: 0	**CANCEL sip:sip51@192.168.1.4 SIP/2.0** **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK77386d18** CSeq: 2 CANCEL To: sip:sip51@chula.ac.th From: nattaworn.p <sip:nattaworn.p@chula.ac.th>;tag=316075616 Call-ID: 11917173996204880 69@192.168.1.2 **Via: SIP/2.0/UDP 192.168.1.2:5060** User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Content-Length: 0 Max-Forwards: 70
SIP/2.0 487 Request Canceled **Via: SIP/2.0/UDP 192.168.1.2:5060** **From: sip:nattaworn.p@chula.ac.th** **To: sip:sip51@chula.ac.th;tag=-1926627328** **Call-ID: 11917173996204880 69@192.168.1.2** **CSeq: 2 INVITE** **Content-Length: 0**	SIP/2.0 487 Request Canceled Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1926627328 Call-ID: 11917173996204880 69@192.168.1.2 CSeq: 2 INVITE Content-Length: 0
SIP/2.0 200 OK **Via: SIP/2.0/UDP 192.168.1.2:5060** **From: sip:nattaworn.p@chula.ac.th** **To: sip:sip51@chula.ac.th;tag=-1926627328** **Call-ID: 11917173996204880 69@192.168.1.2** **CSeq: 2 CANCEL** **Content-Length: 0**	SIP/2.0 200 OK Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1926627328 Call-ID: 11917173996204880 69@192.168.1.2 CSeq: 2 CANCEL Content-Length: 0
ACK sip:sip51@chula.ac.th SIP/2.0 CSeq: 2 ACK To: sip:sip51@chula.ac.th;tag=-1926627328 From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=316075616 Call-ID: 11917173996204880 69@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**ACK sip:sip51@192.168.1.4 SIP/2.0** Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK2b64931b CSeq: 2 ACK To: sip:sip51@chula.ac.th;tag=-1926627328 From: nattaworn.p <sip:nattaworn.p@chula.ac.th>;tag=316075616 Call-ID: 11917173996204880 69@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0 Max-Forwards: 70

การทดสอบการยกเลิกการขอสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น ดังแสดงในรูปที่ 6.22 และตารางที่ 6.17 ได้ผลดังลำดับต่อไปนี้

1. เมื่อโปรแกรม SIP User Agent ขอยกเลิกการสร้างเซสชันโดยส่งข้อความร้องขอ CANCELผ่านเครื่องแม่ข่ายไปยังเครื่องโทรศัพท์ในขณะที่เครื่องโทรศัพท์กำลังตัดสินใจตอบรับ
2. เครื่องโทรศัพท์ตอบสนองข้อความร้องขอ INVITE และ CANCEL ผ่านเครื่องแม่ข่ายด้วยข้อความตอบสนอง 487 Request Canceled และ 200 OK ตามลำดับ

3. โปรแกรม SIP User Agent ตอบรับข้อความตอบสนอง 200 OK ผ่านเครื่องแม่ข่ายด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์รับรู้ว่าการยกเลิกการสร้างเซสชันได้สำเร็จเรียบร้อย



รูปที่ 6.23 การทดสอบการยกเลิกการสร้างเซสชันกับโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

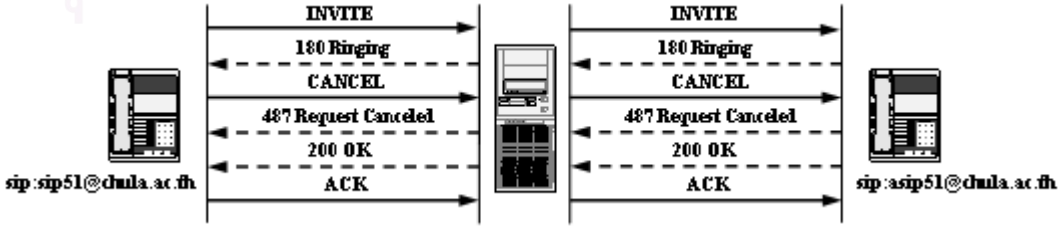
ตารางที่ 6.18 ผลการทดสอบการยกเลิกการสร้างเซสชันกับโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:nattaworn.p@chula.ac.th SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Call-ID: 822083584@192.168.1.4 CSeq: 1 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 v=0 o=Nattaworn 536739840 536739840 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	INVITE sip:nattaworn.p@192.168.1.2 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKb061e399 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Call-ID: 822083584@192.168.1.4 CSeq: 1 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 Max-Forwards: 70 v=0 o=Nattaworn 536739840 536739840 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8
SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 822083584@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 822083584@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKb061e399 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
SIP/2.0 180 Ringing To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 822083584@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	SIP/2.0 180 Ringing To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 822083584@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKb061e399 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
CANCEL sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th** **Call-ID: 822083584@192.168.1.4** **CSeq: 1 CANCEL** **Content-Length: 0**	CANCEL sip:nattaworn.p@192.168.1.2 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKb061e399 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Call-ID: 822083584@192.168.1.4 CSeq: 1 CANCEL Content-Length: 0 Max-Forwards: 70
SIP/2.0 487 Request Terminated To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th	**SIP/2.0 487 Request Terminated** To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th

CSeq: 1 INVITE Call-ID: 822083584@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	CSeq: 1 INVITE Call-ID: 822083584@192.168.1.4 **Via: SIP/2.0/UDP** **192.168.1.3:5060;branch=z9hG4bKb061e399** **Via: SIP/2.0/UDP 192.168.1.4:5060** User-Agent: UbiquityUserAgent/4 Content-Length: 0
Incoming messages to proxy server (Continue)	Outgoing messages from proxy server (Continue)
SIP/2.0 200 OK To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th CSeq: 1 CANCEL Call-ID: 822083584@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**SIP/2.0 200 OK** To: sip:nattaworn.p@chula.ac.th;tag=1191717399 From: sip:sip51@chula.ac.th CSeq: 1 CANCEL Call-ID: 822083584@192.168.1.4 **Via: SIP/2.0/UDP** **192.168.1.3:5060;branch=z9hG4bKb061e399** **Via: SIP/2.0/UDP 192.168.1.4:5060** User-Agent: UbiquityUserAgent/4 Content-Length: 0
ACK sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th;tag=1191717399** **Call-ID: 822083584@192.168.1.4** **CSeq: 1 ACK** **Content-Length: 0**	ACK sip:nattaworn.p@192.168.1.2 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKf698ed20 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th;tag=1191717399 Call-ID: 822083584@192.168.1.4 CSeq: 1 ACK Content-Length: 0 Max-Forwards: 70

การทดสอบการยกเลิกการขอสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ 6.23 และตารางที่ 6.18 ได้ผลดังลำดับต่อไปนี้

1. เมื่อปุ่ม “NO” ของเครื่องโทรศัพท์ถูกกดในขณะที่กำลังรอการตอบรับการสร้างจากโปรแกรม SIP User Agent เครื่องโทรศัพท์ส่งข้อความร้องขอ CANCEL ผ่านเครื่องแม่ข่ายไปยังเครื่องโทรศัพท์เพื่อขอยกเลิกการสร้างเซสชัน
2. โปรแกรม SIP User Agent ตอบสนองข้อความร้องขอ INVITE และ CANCEL ผ่านเครื่องแม่ข่ายด้วยข้อความตอบสนอง 487 Request Canceled และ 200 OK ตามลำดับ
3. เครื่องโทรศัพท์ตอบรับข้อความตอบสนอง 200 OK ผ่านเครื่องแม่ข่ายด้วยข้อความ ACK เพื่อให้โปรแกรมโปรแกรม SIP User Agent รับรู้ว่าการยกเลิกการสร้างเซสชันได้สำเร็จเรียบร้อย



รูปที่ 6.24 การทดสอบการยกเลิกการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Proxy

ตารางที่ 6.19 ผลการทดสอบการยกเลิกการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Proxy

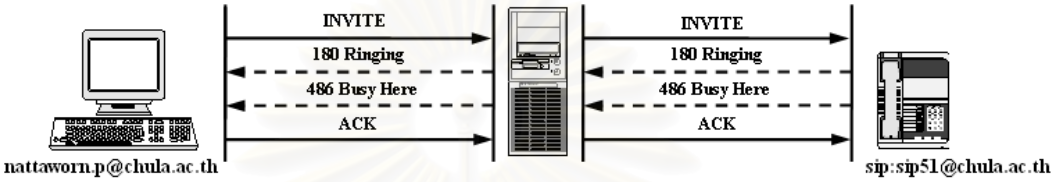
Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:asip51@chula.ac.th SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 4245422080@192.168.1.4 CSeq: 2 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 3890610176 3890610176 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	INVITE sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK617e6265 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 4245422080@192.168.1.4 CSeq: 2 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 Max-Forwards: 70 v=0 o=Nattaworn 3890610176 3890610176 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8
SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=-226951168 Call-ID: 4245422080@192.168.1.4 CSeq: 2 INVITE Content-Length: 0	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=-226951168 Call-ID: 4245422080@192.168.1.4 CSeq: 2 INVITE Content-Length: 0
CANCEL sip:asip51@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th** **Call-ID: 4245422080@192.168.1.4** **CSeq: 2 CANCEL** **Content-Length: 0**	**CANCEL sip:asip51@192.168.1.5 SIP/2.0** **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK617e6265** **Via: SIP/2.0/UDP 192.168.1.4:5060** From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 4245422080@192.168.1.4 CSeq: 2 CANCEL Content-Length: 0 Max-Forwards: 70
SIP/2.0 487 Request Canceled **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th;tag=-226951168** **Call-ID: 4245422080@192.168.1.4** **CSeq: 2 INVITE** **Content-Length: 0**	**SIP/2.0 487 Request Canceled** **Via: SIP/2.0/UDP 192.168.1.3:5060** **Via: SIP/2.0/UDP 192.168.1.4:5060** From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=-226951168 Call-ID: 4245422080@192.168.1.4 CSeq: 2 INVITE Content-Length: 0
SIP/2.0 200 OK **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th;tag=-226951168** **Call-ID: 4245422080@192.168.1.4** **CSeq: 2 CANCEL** **Content-Length: 0**	SIP/2.0 200 OK Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=-226951168 Call-ID: 4245422080@192.168.1.4 CSeq: 2 CANCEL Content-Length: 0
ACK sip:asip51@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th;tag=-226951168** **Call-ID: 4245422080@192.168.1.4** **CSeq: 2 ACK** **Content-Length: 0**	**ACK sip:asip51@192.168.1.5 SIP/2.0** **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK3335c4c4** **Via: SIP/2.0/UDP 192.168.1.4:5060** From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=-226951168 Call-ID: 4245422080@192.168.1.4 CSeq: 2 ACK Content-Length: 0 Max-Forwards: 70

การทดสอบการยกเลิกการขอสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.24 และตารางที่ 6.19 ได้ผลดังลำดับต่อไปนี้

1. เมื่อปุ่ม “NO” ของเครื่องโทรศัพท์ต้นทางถูกกดในขณะที่กำลังรอการตอบรับการสร้างจาก เครื่องโทรศัพท์ปลายทาง เครื่องโทรศัพท์ต้นทางส่งข้อความร้องขอ CANCEL ผ่านเครื่องแม่ข่ายไปยังเครื่องโทรศัพท์ปลายทางเพื่อขอยกเลิกการสร้างเซสชัน

2. เครื่องโทรศัพท์ปลายทางตอบสนองข้อความร้องขอ INVITE และ CANCEL ผ่านเครื่องแม่ข่ายด้วยข้อความตอบสนอง 487 Request Canceled และ 200 OK ตามลำดับ
3. เครื่องโทรศัพท์ต้นทางตอบรับข้อความตอบสนอง 200 OK ผ่านเครื่องแม่ข่ายด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์ต้นทางรับรู้ว่าการยกเลิกการสร้างเซสชันได้สำเร็จเรียบร้อย

6.2.4 การปฏิเสธการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy



รูปที่ 6.25 การทดสอบการปฏิเสธการขอสร้างเซสชันจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

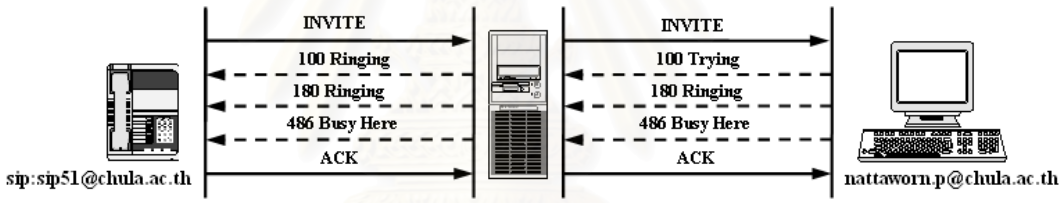
ตารางที่ 6.20 ผลการทดสอบการปฏิเสธการขอสร้างเซสชันจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:sip51@chula.ac.th SIP/2.0 CSeq: 3 INVITE To: sip:sip51@chula.ac.th From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=1447445578 Call-ID: 1662377084440790769@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> Subject: no subject Content-Type: application/sdp Content-Length: 119 v=0 o=- 1133796585934 1133796585934 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2 t=0 0 m=audio 5004 RTP/AVP 8 3 0	INVITE sip:sip51@192.168.1.4 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK11d54e0c CSeq: 3 INVITE To: sip:sip51@chula.ac.th From: nattaworn.p <sip:nattaworn.p@chula.ac.th>;tag=1447445578 Call-ID: 166237708440790769@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Accept: text/plain Accept: application/sdp Accept-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> Subject: no subject Content-Type: application/sdp Content-Length: 119 Max-Forwards: 70 v=0 o=- 1133796585934 1133796585934 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2 t=0 0 m=audio 5004 RTP/AVP 8 3 0
SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1440546816 Call-ID: 166237708440790769@192.168.1.2 CSeq: 3 INVITE Content-Length: 0	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1440546816 Call-ID: 166237708440790769@192.168.1.2 CSeq: 3 INVITE Content-Length: 0
SIP/2.0 486 Busy Here **Via: SIP/2.0/UDP 192.168.1.2:5060** **From: sip:nattaworn.p@chula.ac.th** **To: sip:sip51@chula.ac.th;tag=-1440546816** **Call-ID: 166237708440790769@192.168.1.2** **CSeq: 3 INVITE** **Content-Length: 0**	SIP/2.0 486 Busy Here Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.2:5060 From: sip:nattaworn.p@chula.ac.th To: sip:sip51@chula.ac.th;tag=-1440546816 Call-ID: 166237708440790769@192.168.1.2 CSeq: 3 INVITE Content-Length: 0

Incoming messages to proxy server (Continue)	Outgoing messages from proxy server (Continue)
ACK sip:sip51@chula.ac.th SIP/2.0 CSeq: 3 ACK To: sip:sip51@chula.ac.th;tag=-1440546816 From: nattaworn.p<sip:nattaworn.p@chula.ac.th>;tag=1447445578 Call-ID: 1662377084440790769@192.168.1.2 Via: SIP/2.0/UDP 192.168.1.2:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**ACK sip:sip51@192.168.1.4 SIP/2.0** **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKd6ca3ddd** CSeq: 3 ACK To: sip:sip51@chula.ac.th;tag=-1440546816 From: nattaworn.p <sip:nattaworn.p@chula.ac.th>;tag=1447445578 Call-ID: 1662377084440790769@192.168.1.2 **Via: SIP/2.0/UDP 192.168.1.2:5060** User-Agent: UbiquityUserAgent/4 Content-Length: 0 Max-Forwards: 70

ทดสอบการปฏิเสธการขอสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างโปรแกรม SIP User Agent กับเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.25 และตารางที่ 6.20 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ส่งข้อความตอบสนอง 487 Busy Here ไปยังโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy เมื่อปุ่ม “NO” ของเครื่องโทรศัพท์ถูกกดหลังจากที่เครื่องโทรศัพท์ได้รับข้อความร้องขอ INVITE จากโปรแกรม SIP User Agent
2. โปรแกรม SIP User Agent ตอบรับการตอบสนองนั้นผ่านเครื่องแม่ข่ายด้วยข้อความ ACK เพื่อให้เครื่องโทรศัพท์รับรู้ว่าการปฏิเสธการสร้างเซสชันได้สำเร็จเรียบร้อย



รูปที่ 6.26 การทดสอบการรองรับการปฏิเสธการขอสร้างเซสชัน
จากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

ตารางที่ 6.21 ผลการทดสอบการรองรับการปฏิเสธการขอสร้างเซสชันจากโปรแกรม SIP User Agent ผ่านเครื่องแม่ข่าย Proxy

Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:nattaworn.p@chula.ac.th SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Call-ID: 47054848@192.168.1.4 CSeq: 2 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 v=0 o=Nattaworn 2021523456 2021523456 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	INVITE sip:nattaworn.p@192.168.1.2 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK6e5d6d1b Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Call-ID: 47054848@192.168.1.4 CSeq: 2 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 123 Max-Forwards: 70 v=0 o=Nattaworn 2021523456 2021523456 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8

Incoming messages to proxy server (Continue)	Outgoing messages from proxy server (Continue)
SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=620488069 From: sip:sip51@chula.ac.th CSeq: 2 INVITE Call-ID: 47054848@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=620488069 From: sip:sip51@chula.ac.th CSeq: 2 INVITE Call-ID: 47054848@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK6e5d6d1b Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
SIP/2.0 180 Ringing To: sip:nattaworn.p@chula.ac.th;tag=620488069 From: sip:sip51@chula.ac.th CSeq: 2 INVITE Call-ID: 47054848@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	SIP/2.0 180 Ringing To: sip:nattaworn.p@chula.ac.th;tag=620488069 From: sip:sip51@chula.ac.th CSeq: 2 INVITE Call-ID: 47054848@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK6e5d6d1b Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
SIP/2.0 486 Busy Here To: sip:nattaworn.p@chula.ac.th;tag=620488069 From: sip:sip51@chula.ac.th CSeq: 2 INVITE Call-ID: 47054848@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0	**SIP/2.0 486 Busy Here** To: sip:nattaworn.p@chula.ac.th;tag=620488069 From: sip:sip51@chula.ac.th CSeq: 2 INVITE Call-ID: 47054848@192.168.1.4 **Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK6e5d6d1b** **Via: SIP/2.0/UDP 192.168.1.4:5060** User-Agent: UbiquityUserAgent/4 Content-Length: 0
ACK sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th;tag=620488069** **Call-ID: 47054848@192.168.1.4** **CSeq: 2 ACK** **Content-Length: 0**	ACK sip:nattaworn.p@192.168.1.2 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bKd8ee4c6c Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th;tag=620488069 Call-ID: 47054848@192.168.1.4 CSeq: 2 ACK Content-Length: 0 Max-Forwards: 70

ทดสอบการปฏิเสธการขอสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ 6.26 และตารางที่ 6.21 ได้ผลดังลำดับต่อไปนี้

1. โปรแกรม SIP User Agent ส่งข้อความตอบสนอง 487 Busy Here ไปยังเครื่องโทรศัพท์ ผ่านเครื่องแม่ข่าย Proxy เมื่อโปรแกรม SIP User Agent ปฏิเสธการขอสร้างเซสชันจากเครื่องโทรศัพท์
2. เครื่องโทรศัพท์ตอบรับการตอบสนองนั้นด้วยข้อความ ACK ผ่านเครื่องแม่ข่ายเพื่อให้โปรแกรม SIP User Agent รับรู้ว่าการปฏิเสธการสร้างเซสชันได้สำเร็จเรียบร้อย

รูปที่ 6.27 การทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Proxy

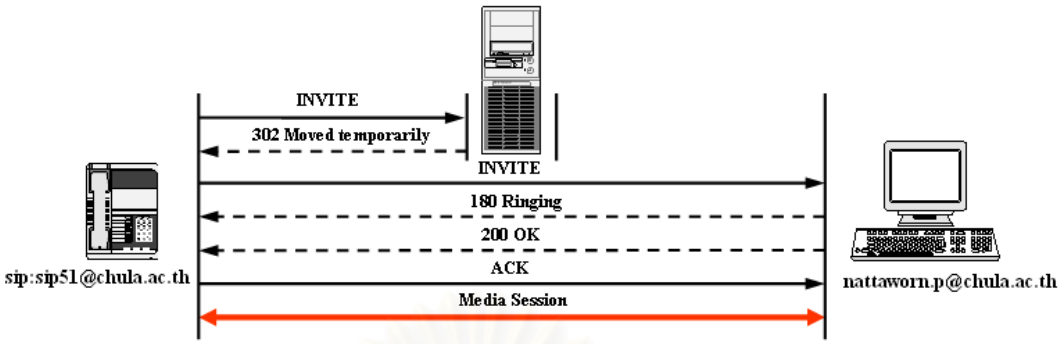
ตารางที่ 6.22 ผลการทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Proxy

Incoming messages to proxy server	Outgoing messages from proxy server
INVITE sip:asip51@chula.ac.th SIP/2.0 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 3275620352@192.168.1.4 CSeq: 3 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 v=0 o=Nattaworn 510001152 510001152 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8	INVITE sip:asip51@192.168.1.5 SIP/2.0 Via: SIP/2.0/UDP 192.168.1.3:5060;branch=z9hG4bK70f8dfb1 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Call-ID: 3275620352@192.168.1.4 CSeq: 3 INVITE Contact: sip:sip51@192.168.1.4 Content-Type: application/sdp Content-Length: 121 Max-Forwards: 70 v=0 o=Nattaworn 510001152 510001152 IN IP4 192.168.1.4 s=Test c=IN IP4 192.168.1.4 t=0 0 m=audio 49170 RTP/AVP 0 8
SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1892810752 Call-ID: 3275620352@192.168.1.4 CSeq: 3 INVITE Content-Length: 0	SIP/2.0 180 Ringing Via: SIP/2.0/UDP 192.168.1.3:5060 Via: SIP/2.0/UDP 192.168.1.4:5060 From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1892810752 Call-ID: 3275620352@192.168.1.4 CSeq: 3 INVITE Content-Length: 0
SIP/2.0 486 Busy Here **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th;tag=1892810752** **Call-ID: 3275620352@192.168.1.4** **CSeq: 3 INVITE** **Content-Length: 0**	**SIP/2.0 486 Busy Here** **Via: SIP/2.0/UDP 192.168.1.3:5060** **Via: SIP/2.0/UDP 192.168.1.4:5060** From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1892810752 Call-ID: 3275620352@192.168.1.4 CSeq: 3 INVITE Content-Length: 0
ACK sip:asip51@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th;tag=1892810752** **Call-ID: 3275620352@192.168.1.4** **CSeq: 3 ACK** **Content-Length: 0**	**ACK sip:asip51@192.168.1.5 SIP/2.0** **Via: SIP/2.0/UDP** **192.168.1.3:5060;branch=z9hG4bK88f14523** **Via: SIP/2.0/UDP 192.168.1.4:5060** From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th;tag=1892810752 Call-ID: 3275620352@192.168.1.4 CSeq: 3 ACK Content-Length: 0 Max-Forwards: 70

ทดสอบการปฏิเสธการขอสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่พัฒนาขึ้น ดังแสดงในรูปที่ 6.27 และตารางที่ 6.22 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ปลายทางส่งข้อความตอบสนอง 487 Busy Here ผ่านเครื่องแม่ข่ายกลับไปยังเครื่องโทรศัพท์ต้นทาง เพื่อปฏิเสธการขอสร้างเซสชัน
2. เครื่องโทรศัพท์ต้นทางตอบรับการตอบสนองนั้นด้วยข้อความ ACK ผ่านเครื่องแม่ข่าย เพื่อให้เครื่องโทรศัพท์ปลายทางรับรู้ว่าการปฏิเสธการสร้างเซสชันได้สำเร็จเรียบร้อย

6.2.5 การสร้างเซสชันโดยใช้เครื่องแม่ข่าย Redirect



รูปที่ 6.28 การทดสอบการขอสร้างเซสชันกับโปรแกรม SIP User Agent โดยใช้เครื่องแม่ข่าย Redirect

ตารางที่ 6.23 ผลการทดสอบการขอสร้างเซสชันกับโปรแกรม SIP User Agent โดยใช้เครื่องแม่ข่าย Redirect

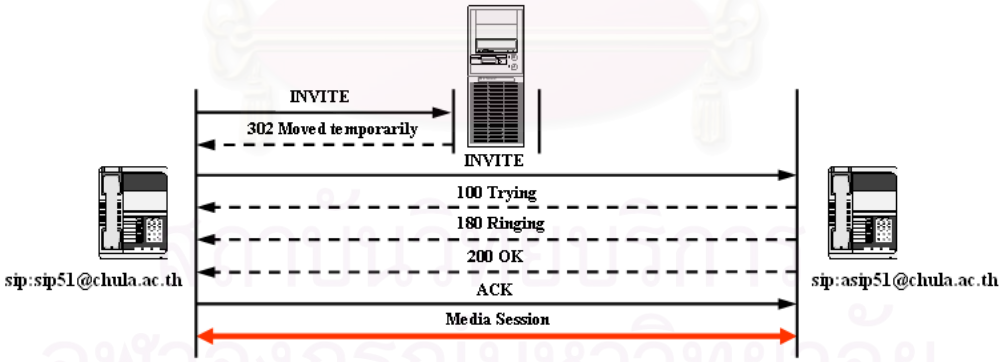
Incoming messages to redirect server	Outgoing messages from redirect server
INVITE sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th** **Call-ID: 4162322432@192.168.1.4** **CSeq: 1 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 4245422080 4245422080 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	**SIP/2.0 302 Moved Temporarily** Call-ID: 4162322432@192.168.1.4 CSeq: 1 INVITE From: sip:sip51@chula.ac.th To: sip:nattaworn.p@chula.ac.th Via: SIP/2.0/UDP 192.168.1.4:5060 Content-Length: 0 **Contact: <sip:nattaworn.p@192.168.1.2>;expires=3229**
Direct request messages	**Direct response messages**
INVITE sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th** **Call-ID: 3890610176@192.168.1.4** **CSeq: 1 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 3275620352 3275620352 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	SIP/2.0 100 Trying To: sip:nattaworn.p@chula.ac.th;tag=-485925142 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
	SIP/2.0 180 Ringing To: sip:nattaworn.p@chula.ac.th;tag=-485925142 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 User-Agent: UbiquityUserAgent/4 Content-Length: 0
ACK sip:nattaworn.p@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:nattaworn.p@chula.ac.th;tag=-485925142** **Call-ID: 3890610176@192.168.1.4** **CSeq: 1 ACK** **Content-Length: 0**	**SIP/2.0 200 OK** To: sip:nattaworn.p@chula.ac.th;tag=-485925142 From: sip:sip51@chula.ac.th CSeq: 1 INVITE Call-ID: 3890610176@192.168.1.4 Via: SIP/2.0/UDP 192.168.1.4:5060 Content-Length: 117 Content-Type: application/sdp Content-Encoding: identity Contact: <sip:nattaworn.p@192.168.1.2> User-Agent: UbiquityUserAgent/4 v=0 o=- 1134911145109 1134911145109 IN IP4 192.168.1.2 s=- c=IN IP4 192.168.1.2

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

	t=0 0 **m=audio 5004 RTP/AVP 0 8**

การทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้น กับโปรแกรม SIP User Agent ดังแสดงในรูปที่ที่ 6.28 และตารางที่ 6.23 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ไปยังเครื่องแม่ข่าย Redirect
2. เครื่องแม่ข่ายตรวจสอบ SIP URL ปลายทางจากเฮดเดอร์ To ของข้อความร้องขอ INVITE ที่ได้รับ เพื่อค้นหาไอพีแอดเดรสปลายทางจากฐานข้อมูล และตอบสนองข้อความ INVITE นั้นด้วยข้อความตอบสนอง 302 Moved Temporarily ซึ่งบ่งบอกไอพีแอดเดรสของเครื่องปลายทางภายในเฮดเดอร์ Contact
3. เครื่องโทรศัพท์ตรวจสอบข้อความตอบสนอง 302 Moved Temporarily ที่ได้รับ และส่งข้อความร้องขอ INVITE ไปยังโปรแกรม SIP User Agent โดยตรง
4. โปรแกรม SIP User Agent ตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 100 Trying และ 180 Ringing กลับไปยังเครื่องโทรศัพท์โดยตรง
5. เมื่อให้โปรแกรม SIP User Agent ตอบรับการขอสร้างเซสชัน โปรแกรม SIP User Agent ส่งข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งระบุการเข้ารหัสเสียงที่รองรับ และพอร์ตรับข้อมูลเสียงกลับไปยังเครื่องโทรศัพท์โดยตรง
6. เครื่องโทรศัพท์ตอบรับข้อความตอบสนอง 200 OK ที่ได้รับด้วยข้อความร้องขอ ACK โดยตรง และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน



รูปที่ 6.29 การทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Redirect

ตารางที่ 6.24 ผลการทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นผ่านเครื่องแม่ข่าย Redirect

Incoming messages to redirect server	Outgoing messages from redirect server
INVITE sip:asip51@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th** **Call-ID: 504889344@192.168.1.4** **CSeq: 1 INVITE**	**SIP/2.0 302 Moved Temporarily** Call-ID: 504889344@192.168.1.4 CSeq: 1 INVITE From: sip:sip51@chula.ac.th To: sip:asip51@chula.ac.th Via: SIP/2.0/UDP 192.168.1.4:5060

Contact: sip:sip51@192.168.1.4 **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 4231790592 4231790592 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	Content-Length: 0 **Contact: sip:asip51@192.168.1.5;expires=3462**
Direct request messages	Direct response messages
INVITE sip:asip51@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th** **Call-ID: 1636171776@192.168.1.4** **CSeq: 1 INVITE** **Contact: sip:sip51@192.168.1.4** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 4072669184 4072669184 IN IP4 192.168.1.4** **s=Test** **c=IN IP4 192.168.1.4** **t=0 0** **m=audio 49170 RTP/AVP 0 8**	**SIP/2.0 180 Ringing** **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th** **Call-ID: 1636171776@192.168.1.4** **CSeq: 1 INVITE** **Content-Length: 0**
ACK sip:asip51@chula.ac.th SIP/2.0 **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th** **Call-ID: 1636171776@192.168.1.4** **CSeq: 1 ACK** **Content-Length: 0**	**SIP/2.0 200 OK** **Via: SIP/2.0/UDP 192.168.1.4:5060** **From: sip:sip51@chula.ac.th** **To: sip:asip51@chula.ac.th** **Call-ID: 1636171776@192.168.1.4** **CSeq: 1 INVITE** **Contact: sip:asip51@192.168.1.5** **Content-Type: application/sdp** **Content-Length: 123** **v=0** **o=Nattaworn 1431401812 1431401812 IN IP4 192.168.1.5** **s=Test** **c=IN IP4 192.168.1.5** **t=0 0** **m=audio 49170 RTP/AVP 0 8**

การทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นดังแสดงในรูปที่ 6.29 และตารางที่ 6.23 ได้ผลดังลำดับต่อไปนี้

1. เครื่องโทรศัพท์ต้นทางร้องขอการสร้างเซสชันโดยส่งข้อความร้องขอ INVITE พร้อมกับข้อความ SDP ไปยังเครื่องแม่ข่าย Redirect
2. เครื่องแม่ข่ายตรวจสอบ SIP URL ปลายทางจากเฮดเดอร์ To ของข้อความร้องขอ INVITE ที่ได้รับ เพื่อค้นหาไอพีแอดเดรสปลายทางจากฐานข้อมูล และตอบสนองข้อความ INVITE นั้นด้วยข้อความตอบสนอง 302 Moved Temporarily ซึ่งบ่งบอกไอพีแอดเดรสของเครื่องปลายทางภายในเฮดเดอร์ Contact
3. เครื่องโทรศัพท์ต้นทางตรวจสอบเฮดเดอร์ Contact ภายในข้อความตอบสนอง 302 Moved Temporarily ที่ได้รับ และส่งข้อความร้องขอ INVITE ไปยังเครื่องโทรศัพท์ปลายทางโดยตรง
4. เครื่องโทรศัพท์ปลายทางตอบสนองข้อความร้องขอ INVITE ที่ได้รับด้วยข้อความตอบสนอง 100 Trying และ 180 Ringing กลับไปยังเครื่องโทรศัพท์โดยตรง

5. เมื่อปุ่ม “YES” ของเครื่องโทรศัพท์ปลายทางถูกกด เครื่องโทรศัพท์ปลายทางส่งข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งระบุการเข้ารหัสเสียงที่รองรับ และพอร์ตรับข้อมูลเสียงกลับไปยังเครื่องโทรศัพท์ต้นทางโดยตรง
6. เครื่องโทรศัพท์ตอบรับข้อความตอบสนอง 200 OK ที่ได้รับด้วยข้อความร้องขอ ACK โดยตรง และเริ่มเปิดเซสชันเพื่อส่งข้อมูลเสียงระหว่างกัน

6.3 ผลการทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง

การทดสอบการส่งซ้ำได้ผลดังแสดงในรูปที่ 6.30 โปรแกรม Ethereal สามารถตรวจจับข้อมูลร้องขอ INVITE ได้ทั้งหมด 5 ครั้ง ซึ่งในการส่งครั้งที่ 2, 3, 4, และ 5 นั้นมีระยะเวลาห่างจากการส่งครั้งแรกประมาณ 0.54, 1.60, 3.74, และ 8 วินาทีตามลำดับ

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.4	192.168.1.3	SIP/SDP	Request: INVITE
2	0.537504	192.168.1.4	192.168.1.3	SIP/SDP	Request: INVITE
3	1.604181	192.168.1.4	192.168.1.3	SIP/SDP	Request: INVITE
4	3.737474	192.168.1.4	192.168.1.3	SIP/SDP	Request: INVITE
5	8.004130	192.168.1.4	192.168.1.3	SIP/SDP	Request: INVITE

รูปที่ 6.30 ผลการทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง

6.4 อภิปรายผลการทดสอบการทำงานในโมดติดต่อโดยตรง

ในการทดสอบการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่องในโมดติดต่อโดยตรง ในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ INVITE จากปลายทาง เครื่องโทรศัพท์สามารถตรวจสอบ และประมวลผล SIP เฮดเดอร์ และฟิลด์ SDP เพื่อเก็บข้อมูลไอพีแอดเดรสปลายทาง, SIP URL ปลายทาง, Call-id, Cseq, ชนิดการเข้ารหัสสัญญาณเสียงที่เครื่องปลายทางรองรับ, และพอร์ตรับข้อมูลเสียงปลายทาง พร้อมกับสร้างข้อความตอบสนอง 180 Ringing กลับไปได้อย่างถูกต้อง และเมื่อผู้ใช้งานรับการขอเปิดเซสชันโดยกดปุ่ม “YES” เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งบ่งบอกชนิดการเข้ารหัสสัญญาณเสียงที่เครื่องโทรศัพท์สามารถรองรับได้กลับไป เพื่อตอบรับการขอเปิดเซสชันได้อย่างถูกต้อง ส่วนในกรณีเครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชันกับปลายทาง เมื่อผู้ใช้งานป้อน SIP URL ปลายทาง และกดปุ่ม “YES” เพื่อเริ่มสร้างเซสชัน เครื่องโทรศัพท์สามารถสร้างข้อความร้องขอ INVITE พร้อมกับข้อความ SDP และส่งไปยังเครื่องปลายทางได้อย่างถูกต้อง ในกรณีที่ผู้ใช้ปลายทางยอมรับการขอสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความตอบสนอง 200 OK เครื่องโทรศัพท์สามารถตรวจสอบพอร์ตรับข้อมูลเสียงปลายทาง และชนิดการเข้ารหัสเสียงที่เครื่องปลายทางรองรับจากข้อความ SDP และส่งข้อความ ACK เพื่อตอบรับการเปิดเซสชันได้อย่างถูกต้อง

ในการทดสอบการยกเลิกการสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่องในโมดติดต่อโดยตรง ในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกขอยกเลิกการสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ CANCEL ในระหว่างที่ผู้ใช้งานกำลังตัดสินใจเลือกรับสาย เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 487 Request Cancel และ 200 OK เพื่อตอบสนองข้อความร้องขอ INVITE และ CANCEL กลับไปยังปลายทางได้อย่างถูกต้องตามลำดับ ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอยกเลิกการขอสร้างเซสชัน เมื่อผู้ใช้งานกดปุ่ม “NO” ในช่วงที่ผู้ใช้ปลายทางตัดสินใจเลือกรับสาย เครื่องโทรศัพท์สามารถสร้างข้อความร้องขอ CANCEL ส่งไปยังปลายทาง และเมื่อได้รับข้อความตอบสนอง 200 OK จากโปรแกรม SIP User Agent แล้วเครื่องโทรศัพท์สามารถส่งข้อความ ACK เพื่อตอบรับ และเป็นการสิ้นสุดกระบวนการขอสร้างเซสชันได้อย่างถูกต้อง

ในการทดสอบการปฏิเสธการขอสร้างเซสชันระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่องในโมดติดต่อโดยตรง ในกรณีเครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับการร้องขอการสร้างเซสชันจากจากโปรแกรมปลายทางแล้วผู้ใช้กดปุ่ม “NO” เพื่อปฏิเสธการเปิดเซสชัน เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 486 Busy Here และส่งกลับไปให้โปรแกรม SIP User Agent ได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ร้องขอการสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความตอบสนอง 486 Busy Here จากปลายทาง เครื่องโทรศัพท์สามารถสร้างข้อความ ACK ส่งกลับไปได้อย่างถูกต้อง

ในการทดสอบการสิ้นสุดการสนทนาระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่องในโมดติดต่อโดยตรง ในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสิ้นสุดเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ BYE ในขณะที่กำลังเปิดเซสชันกับปลายทางอยู่ เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 200 OK เพื่อตอบรับการขอสิ้นสุดเซสชันนั้นได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ร้องขอการสิ้นสุดการสนทนา เมื่อผู้ใช้งานกดปุ่ม “NO” ในขณะที่เซสชันเปิดอยู่ เครื่องโทรศัพท์สามารถส่งข้อความร้องขอ BYE ไปยังเครื่องปลายทางได้อย่างถูกต้อง

6.5 อภิปรายผลการทดสอบการทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย

ในการทดสอบการลงทะเบียนกับเครื่องแม่ข่าย เมื่อเครื่องโทรศัพท์ที่ทำงานในโมดทำงานร่วมกับเครื่องแม่ข่ายถูกเปิดขึ้น หรือมีการปรับตั้งให้เครื่องโทรศัพท์ทำงานในโมดทำงานร่วมกับเครื่องแม่ข่าย เครื่องโทรศัพท์สามารถลงทะเบียนกับเครื่องแม่ข่ายโดยสร้างข้อความร้องขอ REGISTER และส่งให้กับเครื่องแม่ข่ายได้อย่างถูกต้อง

ในการทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่อง ในกรณีที่เครื่องโทรศัพท์เป็นผู้ถูกร้องขอการสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ INVITE จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์สามารถตรวจสอบ และประมวลผล SIP เฮดเดอร์ และฟิลด์ SDP เพื่อเก็บข้อมูลต่าง ๆ ที่สำคัญต่อการสร้างเซสชัน ได้แก่ ไอพีแอดเดรสปลายทาง, SIP URL ปลายทาง, Call-id, Cseq, ชนิดการเข้ารหัสสัญญาณเสียงที่เครื่องปลายทางรองรับ, และพอร์ตรับข้อมูลเสียงปลายทาง พร้อมกับสร้างข้อความตอบสนอง 180 Ringing กลับไปยังเครื่องแม่ข่าย Proxy ได้อย่างถูกต้อง ในกรณีที่ผู้ใช้งานยอมรับการขอเปิดเซสชัน โดยกดปุ่ม “YES” เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 200 OK พร้อมกับข้อความ SDP ซึ่งบ่งบอกชนิดการเข้ารหัสสัญญาณเสียงที่เครื่องโทรศัพท์สามารถรองรับได้กลับไปให้เครื่องแม่ข่าย Proxy เพื่อตอบรับการขอเปิดเซสชันได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชัน เมื่อผู้ใช้งานป้อน SIP URL ปลายทาง และกดปุ่ม “YES” เพื่อเริ่มสร้างเซสชัน เครื่องโทรศัพท์สามารถสร้างข้อความร้องขอ INVITE พร้อมกับข้อความ SDP และส่งให้กับเครื่องแม่ข่าย Proxy ได้อย่างถูกต้อง ในกรณีที่ผู้ใช้ปลายทางยอมรับการขอสร้างเซสชัน เมื่อเครื่องโทรศัพท์ได้รับข้อความตอบสนอง 200 OK จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์เก็บข้อมูล ไอพีแอดเดรสเครื่องปลายทาง, พอร์ตรับข้อมูลเสียงปลายทาง, และชนิดการเข้ารหัสเสียงที่เครื่องปลายทางรองรับจากข้อความ SDP และส่งข้อความ ACK กลับไปยังเครื่องปลายทางโดยตรง เพื่อตอบรับ และเริ่มเปิดเซสชันได้อย่างถูกต้อง

ในการทดสอบการยกเลิกการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่อง เมื่อเครื่องโทรศัพท์ได้รับข้อความร้องขอ CANCEL จากเครื่องแม่ข่าย Proxy ในระหว่างที่ผู้ใช้กำลังตัดสินใจเลือกรับสาย เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 487 Request Cancel และ 200 OK ส่งกลับไปให้เครื่องแม่ข่าย Proxy เพื่อตอบสนองข้อความร้องขอ INVITE และ CANCEL ได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ร้องขอการสร้างเซสชัน เมื่อผู้ใช้งานต้องการยกเลิกการขอสร้างเซสชันโดยกดปุ่ม “NO” ในระหว่างที่ผู้ใช้ปลายทางตัดสินใจเลือกรับสาย เครื่องโทรศัพท์สามารถสร้างข้อความร้องขอ CANCEL ส่งไปยังเครื่องแม่ข่าย Proxy และเมื่อได้รับข้อความตอบสนอง 200 OK จากเครื่องแม่ข่าย Proxy เครื่องโทรศัพท์สามารถส่งข้อความ ACK กลับไปให้เครื่องแม่ข่าย Proxy เพื่อเป็นการตอบรับ และสิ้นสุดกระบวนการขอสร้างเซสชันได้อย่างถูกต้อง

ในการทดสอบการปฏิเสธการสร้างเซสชันผ่านเครื่องแม่ข่าย Proxy ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่อง เมื่อเครื่องโทรศัพท์ได้รับการร้องขอการสร้างเซสชันจากเครื่องแม่ข่าย Proxy แล้วผู้ใช้งาน

กดปุ่ม “NO” เพื่อปฏิเสธการขอสร้างเซสชัน เครื่องโทรศัพท์สามารถสร้างข้อความตอบสนอง 486 Busy Here กลับไปยังเครื่องแม่ข่าย Proxy ได้อย่างถูกต้อง ส่วนในกรณีที่เครื่องโทรศัพท์เป็นผู้ขอสร้างเซสชันกับปลายทางผ่านเครื่องแม่ข่าย Proxy เมื่อเครื่องโทรศัพท์ได้รับข้อความตอบสนอง 486 Busy Here จากเครื่องแม่ข่าย Proxyในขณะที่กำลังรอการตอบรับจากปลายทาง เครื่องโทรศัพท์สามารถสร้างข้อความ ACK ส่งกลับไปยังปลายทาง โดยผ่านเครื่องแม่ข่าย Proxy ได้อย่างถูกต้อง

ในการทดสอบการสร้างเซสชันผ่านเครื่องแม่ข่าย Redirect ระหว่างเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นกับโปรแกรม SIP User Agent และเครื่องโทรศัพท์ที่ได้พัฒนาขึ้นอีกเครื่อง เมื่อเครื่องโทรศัพท์ส่งข้อความร้องขอ INVITE ให้กับเครื่องแม่ข่าย Redirect และได้รับข้อความตอบสนอง 302 Moved Temporarily แล้ว เครื่องโทรศัพท์สามารถตรวจสอบไอพีแอดเดรสของปลายทางได้จากเฮดเดอร์ Contact ที่อยู่ภายในข้อความตอบสนอง 302 Moved Temporarily และส่งข้อความร้องขอ INVITE ใหม่ไปยังเครื่องปลายทางโดยตรงได้อย่างถูกต้อง

6.6 อภิปรายผลการทดสอบการส่งซ้ำกรณีไม่ได้รับข้อความตอบสนอง

ในมูลของเครื่องโทรศัพท์ได้ผลดังแสดงในรูปที่ 6.25 ซึ่งจะเห็นได้ว่าเครื่องโทรศัพท์สามารถส่งซ้ำในกรณีที่ไม่ได้รับข้อความตอบสนองได้ โดยโปรแกรม Ethereal สามารถตรวจจับข้อมูลร้องขอ INVITE ได้ทั้งหมด 5 ครั้ง ซึ่งในการส่งครั้งที่ 2, 3, 4, และ 5 นั้นมีระยะเวลาห่างจากการส่งครั้งแรกประมาณ 0.54, 1.60, 3.74, และ 8 วินาทีตามลำดับ ซึ่งเมื่อคิดเป็นช่วงเวลาผลต่างจากการส่งครั้งก่อนหน้า จะเห็นว่าการส่งซ้ำแต่ละครั้งเกิดขึ้นหลังจากการส่งครั้งก่อนหน้า 0.54, 1.06, 2.14, และ 4.26 วินาทีตามลำดับ คลาดเคลื่อนจากที่ได้ออกแบบไว้ให้ T1=0.5 วินาที และ T2=4 วินาทีเล็กน้อย ทั้งนี้เนื่องจากในการส่งข้อมูลแต่ละครั้งไมโครคอนโทรลเลอร์ต้องใช้เวลาช่วงหนึ่งในการย้ายข้อมูลจากหน่วยความจำภายนอก ไปยังชิพอีเทอร์เนต ซึ่งในการนำช่วงเวลาในส่วนนี้มาร่วมคำนวณด้วย และเป็นที่น่าสังเกตว่าการคลาดเคลื่อนนั้นเพิ่มขึ้นเมื่อช่วงเวลาการส่งซ้ำนานขึ้น ทั้งนี้อาจเกิดจากการคลาดเคลื่อนของความถี่สัญญาณนาฬิกาของคริสตอลที่ป้อนให้กับไมโครคอนโทรลเลอร์ หรือปัจจัยอื่นด้วย อย่างไรก็ตามความคลาดเคลื่อนของช่วงเวลาในการส่งซ้ำตรงนี้มีผลต่อการทำงานของเครื่องโทรศัพท์น้อยมาก หรือแทบจะไม่มีผลเลย

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 7
ต้นทุนการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตต้นแบบและแนวทางในการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตในอนาคต

7.1 ต้นทุนการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตต้นแบบ

การพัฒนาเครื่องโทรศัพท์อินเทอร์เนตต้นแบบ ได้ใช้อุปกรณ์ที่หาซื้อได้จากภายในประเทศ มีรายละเอียดดังแสดงในตารางที่ 7.1 การพัฒนาเครื่องโทรศัพท์อินเทอร์เนตต้นแบบมีต้นทุนอยู่ที่ประมาณ 2,500 บาทต่อเครื่อง แต่ถ้าผลิตจำนวนมากสามารถลดต้นทุนการทำ PCB (Print Circuit Board) และต้นทุนอุปกรณ์ได้

ตารางที่ 7.1 รายการอุปกรณ์ที่ใช้ในการพัฒนาเครื่องโทรศัพท์อินเทอร์เนตต้นแบบ

Electronic parts	Part No.	Description	Supplier	Unit Price (Baht)	Quantity	Price (Baht)
MCU, Memory &	AT89C51ED2	MCU	ES	230	1	230
	61C256AH	32 kbytes SRAM	ES	65	1	65
	82C55	4 Extended Port IC	ES	67	1	67
	74HC373	Latch IC	ES	6	1	6
	74HC139	Memory Decoder	ES	4	1	4
	MAX232	RS-232 Interface	ES	33	1	33
	X'tal 18.432 MHz	For MCU	ES	7	1	7
Ethernet Interface	RTL8019as	Ethernet Controller	ETT	310	1	310
	20F001N	10BaseT Low Past Filter	ETT	80	1	80
	RJ-45 Connector	Ethernet Interface	ES	12	1	12
	X'tal 20 Mhz	For Ethernet Controller	ES	7	1	7
Codec Module	MC145480	5V PCM μ/A Law Codec	WesTech	90	1	90
	74HC4060	14 State Binary Counter	ES	7	1	7
	74HC165	8 bit P/S	ES	7	1	7
	74HC164	8 bit S/P	ES	7	1	7
	74HC02	Nor-Gate x 4	ES	3	1	3
	74HC04	Inverter x 6	ES	4	1	4
	74HC08	And-Gate x 4	ES	4	1	4
	RJ-11 Connector	Handset interface	ES	6	1	6
	X'tal 4.096 MHz	For Clock Module	ES	7	1	7
User Interface	LCD 16x2	User Display	ES	284	1	284
	5mm Switch	User Input (Keypad)	ES	1	16	16
	Connector 14 pin	For LCD Connection	ES	3	1	3
	Connector 10 pin	For Keypad Connection	ES	2	1	2
Power Supply	LM7805	5V-1A Regulator	ES	6	1	6
	Bridge Diode	Full Wave Rectifier	ES	4	1	4
Miscellaneous (Resistor, Capacitor, LED, etc)			ES			20
Print Circuit Board (4x5 inch)			WARA	1,230	1	1,230
Total Price (B)						2,521

ในการสั่งทำ PCB ต้นแบบนั้นต้องเสียค่าใช้จ่ายในการ SETUP และค่าฟิล์มทำให้ต้นทุนในการผลิต PCB ต้นแบบสูง แต่ในการผลิตเพิ่มเป็นจำนวนมากนั้นมีค่าใช้จ่ายเพียงค่าแผ่น PCB และค่าทำ

PCB สำเร็จเท่านั้นซึ่งมีราคาต้นทุนต่อตารางนิ้วละประมาณ 6 บาท ทำให้ต้นทุนการทำ PCB ต่อเครื่องลดลงเหลือประมาณ 120 บาท ส่วนอุปกรณ์อื่น ๆ มีต้นทุนประมาณ 1,200 บาท แต่ในกรณีที่สั่งซื้อเป็นจำนวนมากสามารถลดต้นทุนอุปกรณ์ต่อหน่วยได้ประมาณ 10-30% (อ้างอิงราคาจาก Electronic Source) ทำให้ต้นทุนอุปกรณ์ต่าง ๆ ลดลงเหลือประมาณ 1,000 บาทต่อเครื่อง รวมต้นทุนPCB และอุปกรณ์ต่อเครื่องประมาณ 1,000-1,200 บาท

7.2 แนวทางในการพัฒนาเครื่องโทรศัพท์อินเทอร์เน็ตในอนาคต

7.2.1 ฟังก์ชันการรับและส่ง Email

ในเพิ่มเติมความสามารถให้เครื่องโทรศัพท์อินเตอร์เน็ตที่พัฒนาขึ้นสามารถให้บริการการรับและส่ง Email ได้นั้นจำเป็นต้องอาศัยการพัฒนา Firmware ให้รองรับการทำงานของโพรโทคอล SMTP (Simple Mail Transfer Protocol) ซึ่งเป็นโพรโทคอลมาตรฐานในการส่งอีเมลบนโครงข่ายอินเทอร์เน็ต โดยโพรโทคอลดังกล่าวทำงานบนโพรโทคอล TCP โดยใช้ TCP Port เท่ากับ 25 เป็น พอร์ตมาตฐานในการทำงาน โดยในการพัฒนาดังกล่าวสามารถพัฒนาลงบนหน่วยประมวลผลเดิมได้ยกเว้นในส่วนของการแสดงผลและการป้อนข้อมูลสำหรับตัวเครื่องโทรศัพท์ซึ่งจำเป็นต้องเปลี่ยนแปลง

โดยในการใช้บริการ E-mail ต้องการการแสดงผลที่ชัดเจนอีกทั้งยังต้องการการป้อนข้อมูลที่สะดวก ดังนั้นงานพัฒนาเครื่องโทรศัพท์ในอนาคต อาจเปลี่ยนระบบจอแสดงผลจากจอ LCD แบบเดิมที่ใช้อยู่ไปเป็นจอ LCDในแบบของจอแบบ Touch Screen ซึ่งในปัจจุบันมีราคาที่ต่ำลงมาก และอำนวยความสะดวกสบายแก่ผู้ใช้งาน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 8

สรุปผลการทดสอบและข้อเสนอแนะของการวิจัย

8.1 สรุปผลการพัฒนาและทดสอบ

จากการทดสอบการทำงานของเครื่องโทรศัพท์อินเทอร์เนตที่ได้พัฒนาขึ้น เครื่องโทรศัพท์สามารถรองรับการทำงานขั้นพื้นฐานทั้งในกรณีติดต่อโดยตรง และทำงานร่วมกับเครื่องแม่ข่าย รวมไปถึงการทำงานข้ามโครงข่ายพื้นที่ท้องถิ่นได้อย่างถูกต้อง เราสามารถสรุปคุณลักษณะของเครื่องโทรศัพท์ที่พัฒนาขึ้นได้ดังนี้

1. ติดต่อกับผู้ใช้งานโดยแสดงผลการทำงานผ่านจอ LCD 16x2 และรับข้อมูลผ่านคีย์แพดขนาด 4x4
2. สามารถแสดงชื่อผู้เรียกสาย และโทรซ้ำได้
3. รองรับโพรโทคอล SIP, IPv4, UDP, ARP, ICMP, RTP, และ SDP
4. รองรับข้อความร้องขอ INVITE, CANCEL, ACK, BYE, และ REGISTER
5. รองรับข้อความตอบสนอง 180 Ringing, 200 OK, 302 Moved Temporarily, 406 Busy Here, 407 Request Canceled, และสามารถแยกแยะข้อความตอบสนองอื่น ๆ ที่ได้รับออกเป็นข้อความตอบสนองทั้ง 6 คลาสได้
6. รองรับ SIP เฮดเดอร์ Via, From, To, Call-ID, Content-Length, Content-Type, Cseq, และ Contact
7. รองรับฟิลด์ v, o, a, c, และ m ในโพรโทคอล SDP
8. รองรับการเข้ารหัสเสียงแบบ PCM μ-law และ A-law
9. สามารถอัปเกรดซอฟต์แวร์ผ่านพอร์ต RS-232 ได้

8.2 ปัญหาในการพัฒนาและการแก้ไข

8.2.1 ข้อจำกัดในด้านความเร็วของไมโครคอนโทรลเลอร์

ชิพเข้ารหัสเสียงที่ใช้ในงานวิจัยนี้มีการรับส่งข้อมูลแบบอนุกรม และไม่ได้ถูกออกแบบมาให้รับส่งข้อมูลเสียงกับไมโครคอนโทรลเลอร์โดยตรง เมื่อให้ไมโครคอนโทรลเลอร์รับส่งข้อมูลเสียงแบบอนุกรมกับชิพเข้ารหัสเสียงโดยตรง ไมโครคอนโทรลเลอร์ต้องซิงโครไนซ์กับชิพเข้ารหัสเสียงโดยต้องอินเตอร์รัปต์ 64,000 ครั้ง/วินาที เพื่อรับส่งข้อมูลเสียงกับชิพเข้ารหัสเสียงครั้งละบิต นอกจากนี้ยังต้องรับส่งข้อมูลเสียงกับชิพอีเทอร์เนตอีกด้วย ซึ่งเมื่อทดสอบใช้งานจริง ไมโครคอนโทรลเลอร์ไม่สามารถดึงข้อมูลจากชิพอีเทอร์เนตได้ทัน และทำให้บัฟเฟอร์ภายในชิพ

อีเทอร์เนตล้น (Buffer overflow) จึงได้แก้ไขโดยการแปลงข้อมูลเสียงจากชิพเข้ารหัสเสียงให้เป็นแบบขนานก่อน ทั้งนี้เพื่อลดการทำงานของไมโครคอนโทรลเลอร์ โดยสามารถลดอัตราการเกิดอินเตอร์รัปต์ภายในไมโครคอนโทรลเลอร์ให้เหลือ 8,000 ครั้ง/วินาที และทำให้ไมโครคอนโทรลเลอร์สามารถรับส่งข้อมูลกับชิพอีเทอร์เนตได้ทันอีกด้วย

8.2.2 การซิงโครไนซ์ของสัญญาณนาฬิกาที่ป้อนให้กับชิพเข้ารหัสเสียง

ในการพัฒนาในตอนแรกได้ใช้ไมโครคอนโทรลเลอร์สร้างสัญญาณนาฬิกา MCLK, BCLK, และ FS ความถี่ 256, 64, 8 kHz ตามลำดับให้กับชิพเข้ารหัสเสียง แต่เมื่อทดสอบโดยการลูปสัญญาณดิจิทัลจากภาคส่งเข้าภาครับ แล้วส่งสัญญาณเสียงเข้าทางพอร์ตรับสัญญาณแอนะลอก ปรากฏว่าสัญญาณเสียงที่ได้จากชิพมีความเพี้ยนมาก คาดว่าเกิดจากการประวิงทางเวลา ซึ่งเป็นผลจากช่วงแมชชีนไซเคิลของการทำงานของไมโครคอนโทรลเลอร์ ทำให้สัญญาณนาฬิกาที่ได้แต่ละสัญญาณไม่ซิงโครไนซ์กันพอดี จึงได้แก้ไขโดยเพิ่มวงจรสร้างสัญญาณนาฬิกาขึ้นมาต่างหาก และจ่ายให้กับชิพเข้ารหัสเสียงโดยตรง เมื่อทดสอบใหม่จึงได้สัญญาณเสียงที่ไม่ผิดเพี้ยน และเป็นการลดการทำงานของไมโครคอนโทรลเลอร์ในช่วงรับส่งข้อมูลเสียงด้วย

8.2.3 การขยับเคลื่อนของสัญญาณนาฬิกา

ในการทดสอบคุณภาพของเสียงเมื่อเปิดเซสชัน ในตอนแรกเสียงที่ได้มีคุณภาพชัดเจน และเพี้ยนสลับกันไป และมีคาบของการเปลี่ยนแปลงคุณภาพเสียงคงที่ และเมื่อได้ลองทดสอบเพิ่มขนาดบัฟเฟอร์ข้อมูลเสียงภาครับ เสียงที่ได้รับก็ยังคงมีคุณภาพชัดเจน และเพี้ยนสลับกันไปอยู่ แต่มีคาบของการเปลี่ยนแปลงคุณภาพเสียงนานขึ้น จึงสันนิษฐานว่าเกิดจากการขยับเคลื่อนของสัญญาณนาฬิกา (clock drift) ระหว่างต้นทาง กับปลายทาง ทำให้อัตราข้อมูลเสียงที่ได้รับกับอัตราการเล่นข้อมูลเสียงไม่เท่ากัน มีผลทำให้บัฟเฟอร์เกิดการซ้อนทับเป็นช่วง ๆ จึงได้แก้ไขโดยเพิ่มฟังก์ชันการปรับพอยเตอร์ (pointer justification) ในส่วนรองรับการส่งข้อมูลเสียงให้กับชิพเข้ารหัสเสียง หลังการแก้ไขเมื่อทดสอบใหม่ปรากฏว่าสัญญาณเสียงที่เพี้ยนเป็นช่วง ๆ ก็หายไป

8.2.4 ความเสถียรของเครื่องโทรศัพท์

ในการพัฒนาในตอนแรก เนื่องจากต้องมีการปรับเปลี่ยนการต่อทางฮาร์ดแวร์บ่อย จึงได้พัฒนาโดยใช้บอร์ดอเนกประสงค์ และต่อขาไอซีโดยใช้สายไวร์แลป ในการทดสอบหลาย ๆ ครั้ง เครื่องโทรศัพท์ทำงานผิดพลาดอันเนื่องมาจากการต่อที่ไม่ดี หรือเกิดการลัดวงจร จึงทำให้การพัฒนาเป็นไปได้อย่างล่าช้า ในตอนหลังเมื่อปรับเปลี่ยนฮาร์ดแวร์จนแน่ใจแล้วจึงได้ทำบอร์ด PCB (Print Circuit Board) ซึ่งทำให้เครื่องโทรศัพท์ทำงานได้เสถียร และสามารถพัฒนาได้รวดเร็วกว่าช่วงแรกมาก

8.3 ข้อเสนอแนะ

เครื่องโทรศัพท์อินเทอร์เนตที่ได้พัฒนาขึ้นในงานวิจัยนี้ สามารถใช้งานขั้นพื้นฐานได้เท่านั้น อย่างไรก็ตามงานวิจัยนี้ได้ออกแบบ และพัฒนาฮาร์ดแวร์ของเครื่องโทรศัพท์ให้สามารถรองรับการเปลี่ยนแปลงซอฟต์แวร์เพื่อเพิ่มความสามารถของเครื่องโทรศัพท์ได้โดยง่าย และสะดวก โดยมีข้อจำกัดอยู่ที่ขนาดของหน่วยความจำโปรแกรมแบบแฟลชภายในไมโครคอนโทรลเลอร์ 64 กิโลไบต์, ขนาดของหน่วยความจำข้อมูลภายนอก 16 กิโลไบต์, และความเร็วในการประมวลผลข้อมูลของไมโครคอนโทรเลอร์ ซึ่งการพัฒนาในงานวิจัยนี้ได้ใช้หน่วยความจำโปรแกรมแบบแฟลชภายในไมโครคอนโทรลเลอร์ไปประมาณ 22 กิโลไบต์ และใช้หน่วยความจำข้อมูลภายนอกประมาณ 3.3 กิโลไบต์เท่านั้น จะเห็นได้ว่าสามารถพัฒนาซอฟต์แวร์เพื่อเพิ่มความสามารถให้กับเครื่องโทรศัพท์ได้อีกมาก ทั้งนี้ผู้วิจัยได้เสนอแนะ และแบ่งการพัฒนาส่วนซอฟต์แวร์ของเครื่องโทรศัพท์ต่อออกเป็น 3 ส่วนคือ ส่วนรองรับข้อมูลเสียง, ส่วนรองรับโพรโทคอล และส่วนจัดสรรการเก็บข้อมูลภายในเครื่องโทรศัพท์

1. ส่วนรองรับข้อมูลเสียง

งานวิจัยนี้ ได้ทดสอบเครื่องโทรศัพท์เฉพาะส่วนรองรับโพรโทคอล SIP ในส่วนรองรับข้อมูลเสียงได้ตรวจฟังเสียงที่ได้รับที่ต้นทาง และปลายทางไม่ให้ผิดเพี้ยนเท่านั้น แต่การนำไปใช้งานบนโครงข่ายจริงที่มีปริมาณทราฟฟิกสูง อาจทำให้คุณภาพเสียงที่เครื่องโทรศัพท์ได้รับต่ำลงไปได้ ดังนั้นในการพัฒนาต่อควรทดสอบคุณภาพของสัญญาณเสียง เมื่อข้อมูลเสียงที่ได้รับเกิดการสูญหาย, การประวิงทางเวลา, และ Jitter ที่แตกต่างกัน รวมไปถึงการทดสอบการประวิงทางเวลาของข้อมูลเสียงที่เกิดขึ้นภายในเครื่องโทรศัพท์ และนำผลที่ได้จากการทดสอบมาปรับปรุง และพัฒนาการทำงานในส่วนรองรับข้อมูลเสียงให้สามารถทำงานบนโครงข่ายที่มีคุณลักษณะต่าง ๆ กันได้ และมีประสิทธิภาพมากที่สุด

2. ส่วนรองรับโพรโทคอล

ในส่วนรองรับโพรโทคอล SIP นั้นควรพัฒนาให้สามารถรองรับการทำงานได้เพิ่มมากขึ้น เช่นการตรวจพิสูจน์ผู้ใช้ (authentication), การโอนสาย, การพักสาย, การประชุมหลายสาย, และการส่งข้อความ SMS เป็นต้น ในส่วนโพรโทคอลอื่น ๆ ควรพัฒนาให้เครื่องโทรศัพท์สามารถรองรับโพรโทคอลอื่น ๆ ที่เพิ่มความสามารถของเครื่องโทรศัพท์ให้มากขึ้นเช่นโพรโทคอล TCP, IPv6, DHCP, RSVP เป็นต้น

3. ส่วนจัดสรรการเก็บข้อมูลภายในเครื่องโทรศัพท์

พัฒนาส่วนเก็บข้อมูลภายในเครื่องโทรศัพท์ให้มีความสามารถสูงขึ้น เช่นการจดจำสายเรียกเข้า, สายที่ไม่ได้รับ หรือบันทึก SIP URL ปลายทางที่สำคัญภายในตัวเครื่องโทรศัพท์เป็นต้น

นอกจากการพัฒนาทางซอฟต์แวร์แล้ว อาจพัฒนาฮาร์ดแวร์เพื่อเพิ่มความสามารถบางส่วนที่นอกเหนือจากการพัฒนาทางด้านซอฟต์แวร์สามารถทำได้เช่น เพิ่มการรองรับการเข้ารหัสเสียงที่มีอัตราการบีบอัดข้อมูลสูงกว่าแบบ PCM เช่น G.723, G729 หรืออาจพัฒนาให้รองรับการเข้ารหัสสัญญาณภาพวิดีโอ ซึ่งทั้งนี้ต้องเพิ่มชิพเข้ารหัสสัญญาณเสียง หรือภาพ รวมไปถึงปรับเปลี่ยนส่วนแสดงผลเป็น LCD แบบกราฟฟิก และพัฒนาซอฟต์แวร์เพื่อรองรับเพิ่ม ในกรณีที่ข้อมูลภาพมีอัตราข้อมูลสูง ไมโครคอนโทรลเลอร์อาจไม่สามารถรับส่งข้อมูลกับชิพอีเทอร์เนตได้ทัน อาจเปลี่ยนไปใช้ไมโครคอนโทรลเลอร์ที่มีความสามารถสูงกว่าเช่น ARM7 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีหน่วยประมวลผลแบบ 32 บิต, มีผู้นำมาพัฒนาอย่างแพร่หลาย, ราคาไม่สูงมาก และสามารถรองรับหน่วยความจำได้มากกว่าไมโครคอนโทรลเลอร์ตระกูล 8051 ได้

ในส่วนวงจรดิจิทัลซึ่งประกอบด้วย ไอซี Latch, วงจรถอดรหัสตำแหน่งหน่วยความจำ, วงจรสร้างสัญญาณนาฬิกา, วงจรแปลงการส่งข้อมูลอนุกรมเป็นขนาน (S/P), และวงจรแปลงการส่งข้อมูลขนานเป็นอนุกรม (P/S) สามารถปรับเปลี่ยนมาใช้ชิพ CPLD (Complex Programmable Logic Device) ชิพเดียวแทนได้ ทั้งนี้เพื่อลดจำนวนของอุปกรณ์, ขนาดของบอร์ด, และต้นทุนรวมไปถึงการทำ Memory Map ที่มีประสิทธิภาพที่ดีกว่าได้

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

โทรศัพท์ผ่านอินเทอร์เน็ต (IP Phone) ที่พัฒนาและออกแบบโดยคนไทยที่สามารถใช้งานได้จริง และนำไปใช้งานในเชิงพาณิชย์

บริษัท ฟอร์ท คอร์ปอเรชั่น จำกัด (มหาชน) ได้ออกแบบและพัฒนาโทรศัพท์ผ่านอินเทอร์เน็ต (IP Phone) เครื่องแรกของโลกที่สามารถแสดงหน้าจอเป็นภาษาไทย ซึ่งออกแบบโดยวิศวกรชาวไทย สามารถปรับแต่งได้ตามต้องการ ทำงานด้วยหน้าจอขนาด 4.3 นิ้ว ความละเอียดสูง 480 x 272 pixel 65,536 สี หน้าจอแบบ Touch Screen

ในการออกแบบมุ่งเน้นการใช้ประโยชน์การเชื่อมต่อกับระบบ VoIP ด้วยคุณภาพเสียงที่ดี พร้อมทั้งแสดงหน้าจอรูปภาพความละเอียดสูงพร้อมแสดงภาษาไทยสมบูรณ์แบบ ซึ่งเป็นรายแรกและรายเดียวของโลก ที่หาไม่ได้จากโทรศัพท์ผ่านอินเทอร์เน็ตชั้นนำจากต่างประเทศ และเนื่องจากใช้หน้าจอ LCD TFT ความละเอียดสูง ขนาด 4.3 นิ้วที่ 65,536 สีทำให้โทรศัพท์ผ่านอินเทอร์เน็ตของบริษัท ฟอร์ท สามารถแสดงข้อมูลการใช้งานที่บันทึกไว้อย่างละเอียด ทั้งภาพถ่ายคู่สนทนา พร้อมรายละเอียดต่างๆของคู่สนทนา ทั้งจากสายในหรือสายนอก โดยโทรศัพท์ผ่านอินเทอร์เน็ตของบริษัทฟอร์ทเหมาะสมอย่างยิ่งสำหรับธุรกิจที่ต้องการระบบการสื่อสารชั้นสูง เช่นระบบ Call Center, ระบบโรงแรม, รวมทั้งบริษัทต่างๆที่ต้องการริเริ่มนำระบบโทรศัพท์ผ่านอินเทอร์เน็ตมาใช้แทนระบบโทรศัพท์แบบดั้งเดิม

โทรศัพท์ผ่านอินเทอร์เน็ตของบริษัท ฟอร์ท ทำงานโดยอาศัยโพรโทคอล SIP (SIP Protocol) มาตรฐาน โดยโทรศัพท์ดังกล่าวสามารถใช้งานร่วมกับระบบ IP-PBX ได้ทุกยี่ห้อที่ใช้โพรโทคอล SIP นอกจากนี้ยังสามารถนำมาใช้เป็นเครื่องโทรศัพท์ระบบ Operator หรือโทรศัพท์สำหรับผู้บริหารได้อย่างมีประสิทธิภาพมากกว่าระบบ Key Telephone ทั่วไป ด้วยการใช้งานอันแสนชาญฉลาดและง่ายดาย ทั้งจากปุ่ม Softkey หรือการกดบนหน้าจอโดยตรง (Touch Screen) อีกทั้งยังรองรับระบบ CRM ต่างๆในอนาคตอีกด้วย

ในด้านการออกแบบโทรศัพท์ผ่านอินเทอร์เน็ต บริษัท ฟอร์ท ออกแบบโดยใช้ Dual-core CUP RISC ขนาด 32 bits และชิพ DSP ที่มีชื่อเสียงในการออกแบบระบบ VoIP เพื่อให้ได้คุณภาพของเสียงที่ดี โดยเครื่องสามารถใช้งานได้ทั้งระบบ Full Duplex Hand Free แบบพูดสวนทางกันได้, ระบบ Head-set สำหรับพนักงานรับสาย และรองรับการขยายด้วย USB Port นอกจากนี้ยังสามารถเลือกใช้ระบบ Codec ชั้นนำต่างๆได้เช่น G.711, G.723, G.726, G.727, GSM, iLBC และยังการเพิ่มคุณภาพเสียงด้วยวิธี Adaptive Echo Cancellation, VAD, CNG โดยการออกแบบดังกล่าวเพื่อทำให้โทรศัพท์

ผ่านอินเทอร์เน็ตดังกล่าวให้คุณภาพเสียงสูงสุดโดยไม่ใช้แบนด์วิดท์ในการทำงานมากนัก นอกจากนี้ยังออกแบบให้รองรับระบบจ่ายไฟผ่านสาย LAN (PoE) เพื่อไม่ให้ระบบหยุดชะงักเมื่อเกิดภาวะไฟฟ้าดับ

โดยโทรศัพท์ผ่านอินเทอร์เน็ตของบริษัท ฟอร์ท มีจุดเด่นดังต่อไปนี้

- The best field proven of VoIP chipsets for Voice & Data Networking
- SIP protocol compliant (RFC3261) support by any SIP server & IP-PBX
- G.711, G.723.1, G.726, G.727, G.729, Full Rate GSM6.10, Linear 16 bit 8 Khz coder, iLBC
- Voice Enhancement Features VAD, DTMF, CNG, G.168
- Large LCD – TFT 4.2” at 65535 color
- Soft-key and Touch Screen
- Dual Ethernet Port
- Power over Ethernet
- Full Duplex Speaker Phone
- Headset Interface
- USB Expansion Ports

รูปที่ ผ.1.1 IP Phone ของบริษัท ฟอร์ท

รูปที่ ผ.1.2 หน้าจอการใช้งาน IP Phone ของบริษัท ฟอร์ท

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] สำนักงานเลขานุการคณะกรรมการเทคโนโลยีสารสนเทศแห่งชาติ กรอบนโยบายเทคโนโลยีสารสนเทศ ระยะ พ.ศ. 2544-2553 ของประเทศไทย, NECTEC, เมษายน 2545

[2] B. Khasnabish, Implementing Voice over IP, John Wiley & Sons, 2003

[3] M. A. Miller, Voice over IP Technologies: Building the Converged Network, John Wiley & Sons, 2002

[4] J. Ryan, The Technology Guide Series: Voice over IP, The Applied Technologies Group, Inc., 1998

[5] R. Arcomano, "Voice over IP How to," online <URL: http://web.tiscalinet.it/bertolinux>, 12 Nov. 2000

[6] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," Request For Comments 2543, Internet Draft Internet Engineering Task Force, Mar. 1999

[7] International Telecommunication Union, H.323 Recommendation, online <URL: http://www.itu.int/itudoc/itu-t/rec/h/h323.html>

[8] I. E. del Pozo, "An Implementation of the Internet Call Waiting Service Using SIP", Laboratory of Telecommunications Technology, Dec. 1999

[9] A. B. Johnston, SIP: Understanding the Session Initiation Protocol, Artech House, 2001

[10] FORTH Professional IP Phone: IP Telephony System, www.Forth.co.th

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย