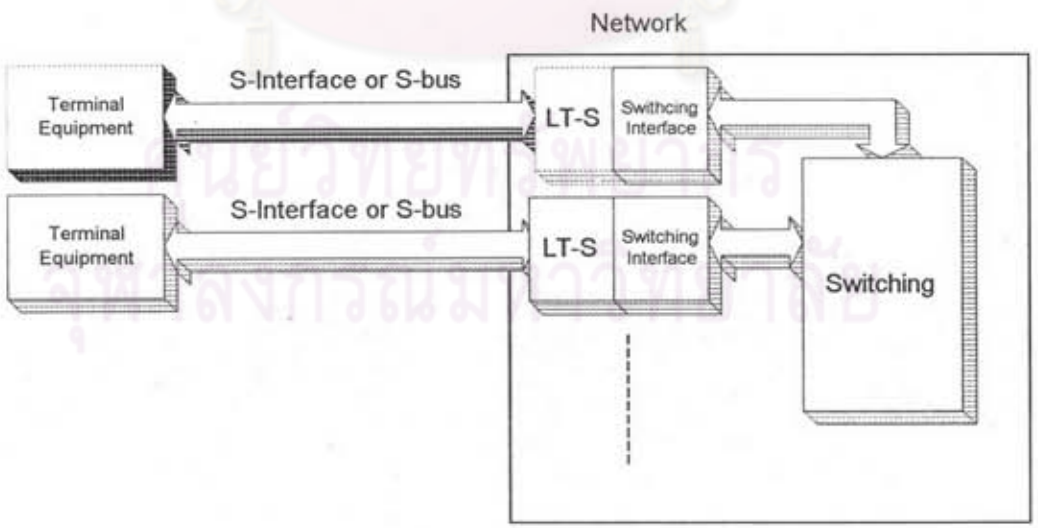
บทที่ 3

การออกแบบอุปกรณ์สื่อสารปลายทางระบบไอเอสดีเอ็นสำหรับการขอใช้บริการส่งสัญญาณเสียงในช่องสัญญาณ B

หลังจากที่ทำความเข้าใจโครงข่ายไอเอสดีเอ็นแบบแถบความถี่แคบในเบื้องต้นแล้ว ในบทนี้จะกล่าวถึงขั้นตอนการออกแบบระบบทั้งหมด รวมทั้งการออกแบบวิธีทดสอบการทำงาน โดยกำหนดให้อุปกรณ์สื่อสารปลายทางที่สร้างขึ้นเชื่อมต่อกับโครงข่ายไอเอสดีเอ็นแบบแถบความถี่แคบแบบจุดต่อจุด (point-to-point) และมีการใช้ช่องสัญญาณแบบ BAI

การสร้างอุปกรณ์สื่อสารปลายทางเพื่อต่อเข้ากับโครงข่ายไอเอสดีเอ็นที่จุดอ้างอิง S นั้น ในการทดสอบการทำงานต้องเชื่อมต่ออุปกรณ์สื่อสารปลายทางเข้ากับโครงข่ายไอเอสดีเอ็นที่จุดอ้างอิง S เพื่อให้อุปกรณ์สื่อสารปลายทางที่ออกแบบและสร้างขึ้นนั้นสามารถทำงานได้ จึงได้สร้างอุปกรณ์ที่ทำหน้าที่จำลองการทำงานของโครงข่ายไอเอสดีเอ็นแบบแถบความถี่แคบที่จุดอ้างอิง S ตามการเชื่อมต่อที่ได้กำหนดไว้ข้างต้นขึ้นมา ซึ่งเรียกว่า Line Termination at S interface (LT-S) (Siemens AG, 1992a) ระบบโดยรวมทั้งหมดสามารถแสดงได้ดังรูปที่ 3.1 โดยส่วนที่เป็นเส้นประเป็นส่วนที่ต้องออกแบบและสร้างขึ้น



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของการเชื่อมต่อกับโครงข่ายไอเอสดีเอ็น

เพื่อความสะดวก คำว่าอุปกรณ์สื่อสารปลายทางในวิทยานิพนธ์นี้ หมายถึงอุปกรณ์สื่อสารปลายทางที่ออกแบบและสร้างขึ้น คำว่าโครงข่ายไอเอสดีเอ็น หมายถึง LT-S ที่สร้างขึ้นเพื่อจำลองการทำงานของโครงข่ายไอเอสดีเอ็นบางส่วน และคำว่าระบบ ในที่นี้หมายถึงอุปกรณ์สื่อสารปลายทางและ LT-S ที่ออกแบบและสร้างขึ้น

จากปัญหาของอุปกรณ์สื่อสารปลายทางที่ได้เคยมีผู้ออกแบบและสร้างไว้และมีข้อจำกัดดังที่กล่าวไว้ในบทที่ 1 วิทยานิพนธ์นี้จึงได้ทำการออกแบบอุปกรณ์สื่อสารปลายทางใหม่ทั้งหมดด้วยการจัดโครงสร้างทางฮาร์ดแวร์และซอฟต์แวร์ของอุปกรณ์สื่อสารปลายทางใหม่ ซึ่งในบางส่วนของอุปกรณ์สื่อสารปลายทางในวิทยานิพนธ์นี้สามารถนำฮาร์ดแวร์และซอฟต์แวร์ของอุปกรณ์สื่อสารปลายทางที่ได้เคยมีผู้ออกแบบและสร้างไว้มาใช้งานได้ ดังจะกล่าวต่อไป

เนื่องจากวิทยานิพนธ์นี้มิได้มุ่งเน้นที่การออกแบบและสร้าง LT-S โครงสร้างทางฮาร์ดแวร์และซอฟต์แวร์ของ LT-S จึงมิได้ถูกออกแบบอย่างเป็นระบบ การออกแบบและสร้าง LT-S มุ่งเน้นให้ทำงานได้เพื่อตรวจสอบขั้นตอนต่างๆ ของการทำงานของอุปกรณ์สื่อสารปลายทางเท่านั้น มิได้จำลองการทำงานของโครงข่ายไอเอสดีเอ็นตามมาตรฐานทั้งหมด การออกแบบและสร้าง LT-S จึงไม่ขอกล่าวถึงในวิทยานิพนธ์นี้

ข้อกำหนด สมบัติและความต้องการทรัพยากรของอุปกรณ์สื่อสารปลายทาง

อุปกรณ์สื่อสารปลายทางที่ออกแบบและสร้างขึ้นมีข้อกำหนดและสมบัติ ดังนี้

- ติดต่อกับผู้ใช้โดยผ่านจอ LCD และปุ่มกด 1 ปุ่ม
- เชื่อมต่อกับโครงข่ายไอเอสดีเอ็นที่จุดอ้างอิง S แบบ BAI
- รูปแบบการเชื่อมต่อเป็นแบบจุดต่อจุด
- การขอใช้บริการส่งสัญญาณเสียงในช่องสัญญาณ B เป็นเพียงการส่งข้อมูลที่จำเป็นในการขอใช้บริการต่อชุมสายเท่านั้น ไม่มีการส่งสัญญาณเสียงในช่องสัญญาณ B หลังจากที่ได้รับบริการจากชุมสายแล้ว
- ซอฟต์แวร์สำหรับควบคุมการทำงานของโปรโตคอลในชั้นที่ 1 และชั้นที่ 2 คลอบคลุมการทำงานตามมาตรฐาน ITU-T ทั้งหมด
- ซอฟต์แวร์สำหรับควบคุมการทำงานของโปรโตคอลในชั้นที่ 3 คลอบคลุมการทำงานตามมาตรฐาน ITU-T บางส่วนเท่านั้น ดังจะได้กล่าวภายหลัง

ในการใช้งาน หลังจากที่จ่ายกำลังไฟฟ้าเข้าสู่อุปกรณ์สื่อสารปลายทาง อุปกรณ์สื่อสารปลายทางจะทดสอบหน่วยความจำภายใน จากนั้นจะ Initialize ระบบ เมื่อ initialize เรียบร้อย

แล้วจะแสดงคำว่า Ready ที่จอ LCD และรอรับคำสั่งจากปุ่มกด เมื่อกดปุ่มอุปกรณ์สื่อสารปลายทางจะเริ่มทำงานตามขั้นตอน ดังนี้

1. หลังจากกดปุ่ม ที่จอ LCD จะแสดงคำว่า Call Initialize

2. เริ่มการ Activate ตามโปรโตคอลในชั้นที่ 1 (ITU-T, 1988a)

3. ขอ ID ด้วย ID assignment procedure และสร้าง logical link กับโครงข่ายไอเอสดีเอ็นตามโปรโตคอลในชั้นที่ 2 (ITU-T, 1988b)

4. ขอบริการจากโครงข่ายไอเอสดีเอ็นตามโปรโตคอลในชั้นที่ 3 (ITU-T, 1988c)

5. เมื่อได้รับการตอบรับการขอใช้บริการจากโครงข่ายไอเอสดีเอ็น จอ LCD ของอุปกรณ์สื่อสารปลายทางจะแสดงคำว่า Calling สำหรับขั้นตอนนี้ในการใช้งานจริงจะเป็นขั้นที่โครงข่ายได้ให้บริการแก่ผู้ขอใช้บริการ มีการส่งสัญญาณกระดิ่ง (Ringing) ที่ปลายทาง ส่วนทางผู้เรียกจะได้ยินสัญญาณเรียกติด (Ring back signal) และรอให้ผู้ใช้บริการปลายทางตอบรับการเรียก หากอุปกรณ์สื่อสารปลายทางนั้นเป็นโทรศัพท์ดิจิตอล

6. หลังจากที่ผู้ใช้บริการปลายทางตอบรับการเรียกด้วยการจำลองการตอบรับโดย LT-S ที่ออกแบบและสร้างขึ้น จอ LCD ของอุปกรณ์สื่อสารปลายทางจะแสดงคำว่า Connect

7. หน่วงเวลาชั่วครู่ แล้วจำลองการขอเลิกการใช้บริการด้วยการขอเลิกการใช้บริการต่อโครงข่ายไอเอสดีเอ็นตามโปรโตคอลในชั้นที่ 3 (ITU-T, 1988c) จอ LCD ของอุปกรณ์สื่อสารปลายทางจะแสดงคำว่า Disconnecting...

8. เมื่อได้รับการตอบรับการยกเลิกการขอใช้บริการจากโครงข่ายไอเอสดีเอ็นแล้ว อุปกรณ์สื่อสารปลายทางก็จะขอยกเลิก logical link กับโครงข่ายตามโปรโตคอลในชั้นที่ 2 (ITU-T, 1988b)

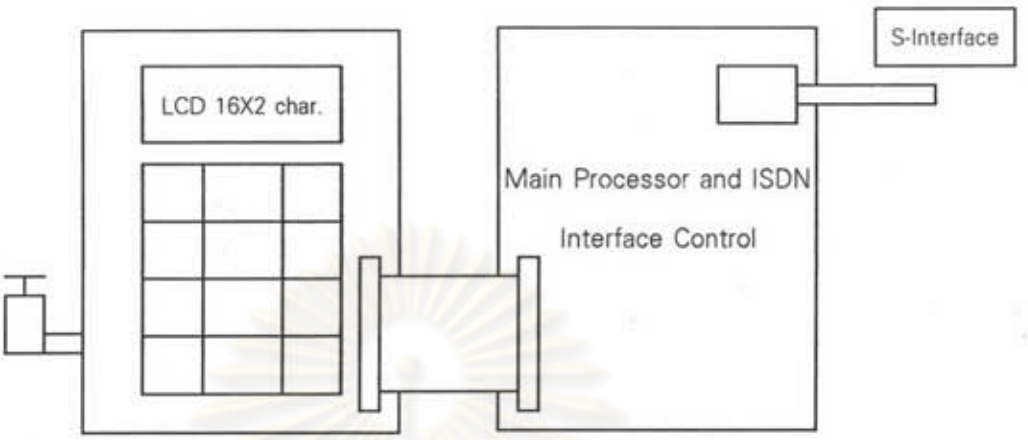
9. โครงข่ายไอเอสดีเอ็นทำการ Deactivate ตามโปรโตคอลในชั้นที่ 1 (ITU-T, 1988a) เมื่อ Deactivate สมบูรณ์ ที่จอ LCD ของอุปกรณ์สื่อสารปลายทางจะแสดงคำว่า Ready อีกครั้ง เป็นการเสร็จสิ้นขั้นตอนการทำงาน

หลังจากกำหนดขั้นตอนการใช้งานแล้ว ต่อไปเป็นการกำหนดรูปแบบของอุปกรณ์สื่อสารปลายทางที่จะสร้างขึ้นมา ซึ่งแบ่งเป็น 2 ส่วน ได้แก่

1. ส่วนควบคุมหลักและส่วนควบคุมการเชื่อมต่อกับโครงข่ายไอเอสดีเอ็นที่จุดอ้างอิง S

2. ส่วนติดต่อกับผู้ใช้ ส่วนนี้ประกอบด้วยจอ LCD ขนาด 16 ตัวอักษร 2 แถว ปุ่มกดสำหรับควบคุมการทำงานและแผงปุ่มกดขนาด 12 ปุ่ม ซึ่งในที่นี้ไม่ได้ใช้งานแต่เผื่อไว้สำหรับอนาคต ส่วนนี้จะทำด้วยแผ่นวงจรพิมพ์ (PCB) แบบมีลายทองแดงเพียงด้านเดียว โดยวาง

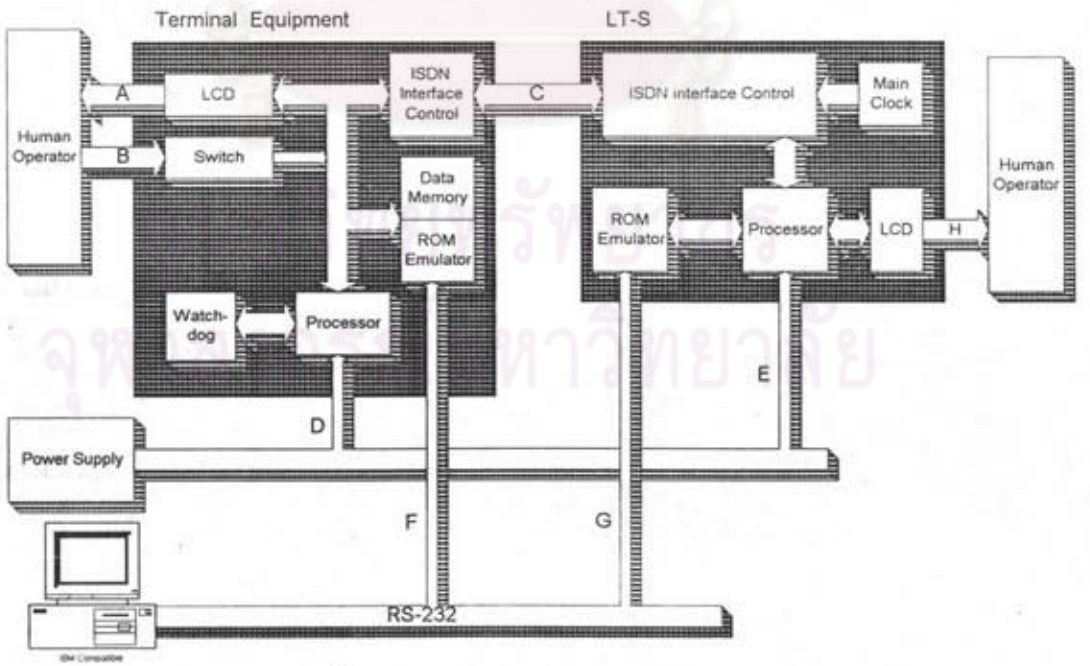
อุปกรณ์และเดินสายที่ด้านที่มีลายวงจรพิมพ์ เพื่อใช้อีกด้านหนึ่งที่ไม่มีลายวงจรพิมพ์เป็นหน้าปัดติดต่อกับผู้ใช้ ดังรูปที่ 3.2



รูปที่ 3.2 แสดงรูปแบบของอุปกรณ์สื่อสารปลายทาง

<u>บล็อกไดอะแกรมของระบบ</u>

จากข้อกำหนดทางรูปแบบและขั้นตอนการใช้งานของอุปกรณ์สื่อสารปลายทางข้างต้น สามารถออกแบบบล็อกไดอะแกรมของระบบได้ ดังแสดงในรูที่ 3.3



รูปที่ 3.3 แสดงบล็อกไดอะแกรมของระบบ

จากบล็อกไดอะแกรมของระบบจะเห็นว่ามีจุดที่ติดต่อกับภายนอก ดังนี้

A, B, H เป็นส่วนที่ติดต่อกับผู้ใช้ ประกอบด้วย จอ LCD ที่ TE และ LT-S สำหรับแสดงผลหรือขั้นตอนการทำงาน และปุ่มกดสำหรับสั่งงานให้ระบบเริ่มทำงาน

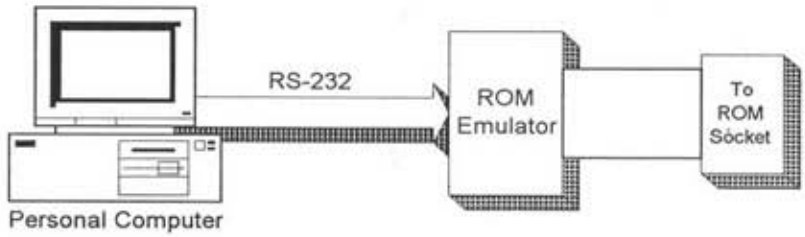
C คือจุดอ้างอิง S ที่จำลองขึ้นมาโดย LT-S เป็นจุดที่อุปกรณ์สื่อสารปลายทางเชื่อมต่อเข้ากับโครงข่ายไอเอสดีเอ็น

D, E เป็นจุดจ่ายกำลังไฟฟ้า ขนาดแรงดัน 5 โวล์ทให้แก่ระบบ

F, G เป็น RS-232 ที่ต่อระหว่างเครื่องไมโครคอมพิวเตอร์กับ ROM Emulator ของทั้งอุปกรณ์สื่อสารปลายทางและ LT-S

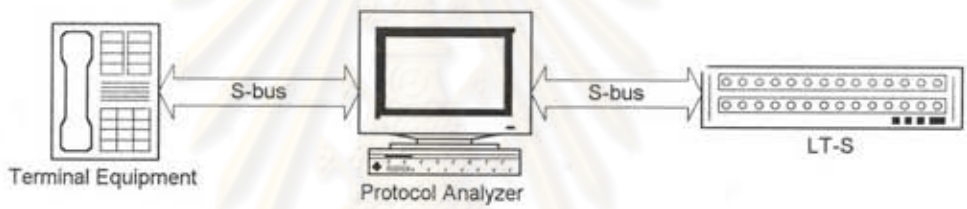
<u>การพัฒนาและการทดสอบการทำงานของระบบ</u>

ในการพัฒนาระบบ หลังจากที่เขียนโปรแกรมของระบบแล้วต้องเขียนโปรแกรมเหล่านั้นลงหน่วยความจำแบบอ่านได้อย่างเดียว (Read Only Memory, ROM) (John B. Peatman, 1988) จากนั้นจึงนำหน่วยความจำที่เขียนแล้วนั้นไปใส่ในระบบ หากโปรแกรมมีข้อผิดพลาดต้องนำหน่วยความจำนั้นมาลบด้วยแสง UV แล้วจึงนำโปรแกรมที่แก้ไขและผ่านการแปลด้วยตัวแปลภาษา (Compiler) แล้วเขียนลงไปใหม่ แต่ในการพัฒนาระบบต้องมีการแก้ไขโปรแกรมหลายครั้ง การทำตามขั้นตอนขั้นต้นจึงไม่สะดวก นอกจากนี้การเขียนโปรแกรมลงหน่วยความจำและลบหลายๆ ครั้ง จะทำให้หน่วยความจำนั้นเสียหายได้ ดังนั้นในการพัฒนาระบบจึงใช้ ROM Emulator แทน ซึ่งเป็นอุปกรณ์ที่จำลองการทำงานของหน่วยความจำแบบอ่านอย่างเดียว โดยด้านหนึ่งของอุปกรณ์มีการจัดเรียงขาและมีลักษณะการโต้ตอบของสัญญาณที่เหมือนกับหน่วยความจำแบบอ่านอย่างเดียว อีกด้านหนึ่งเป็นส่วนที่ต่อเข้ากับเครื่องไมโครคอมพิวเตอร์ผ่านทาง RS-232 ในการพัฒนาระบบเพียงแต่นำโปรแกรมที่เขียนขึ้นมา แปลด้วยตัวแปลภาษา (Compiler) แล้วเขียนลงใน ROM Emulator ผ่านทาง RS-232 หากโปรแกรมมีข้อผิดพลาด ก็เพียงแต่แก้ไขโปรแกรมแล้วเขียนลง ROM Emulator ใหม่เท่านั้น (John B. Peatman, 1988) ดังรูปที่ 3.4 ROM Emulator ที่ใช้เป็น ROM Emulator ของบริษัทศิลา รีเสิร์ช จำกัด ที่จำลองการทำงานของหน่วยความจำแบบอ่านอย่างเดียวเบอร์ 27256 ซึ่งเป็นหน่วยความจำแบบอ่านอย่างเดียวขนาด 32 กิโลไบท์



รูปที่ 3.4 แสดงการเชื่อมต่อ ROM Emulator

การทดสอบการทำงานของระบบ ใช้ Protocol Analyzer ของบริษัท NAVTEL รุ่น 9440 ต่อระหว่างอุปกรณ์สื่อสารปลายทางและ LT-S เพื่อติดตามสถานะและข้อมูลในจุดอ้างอิง S ดังรูปที่ 3.5

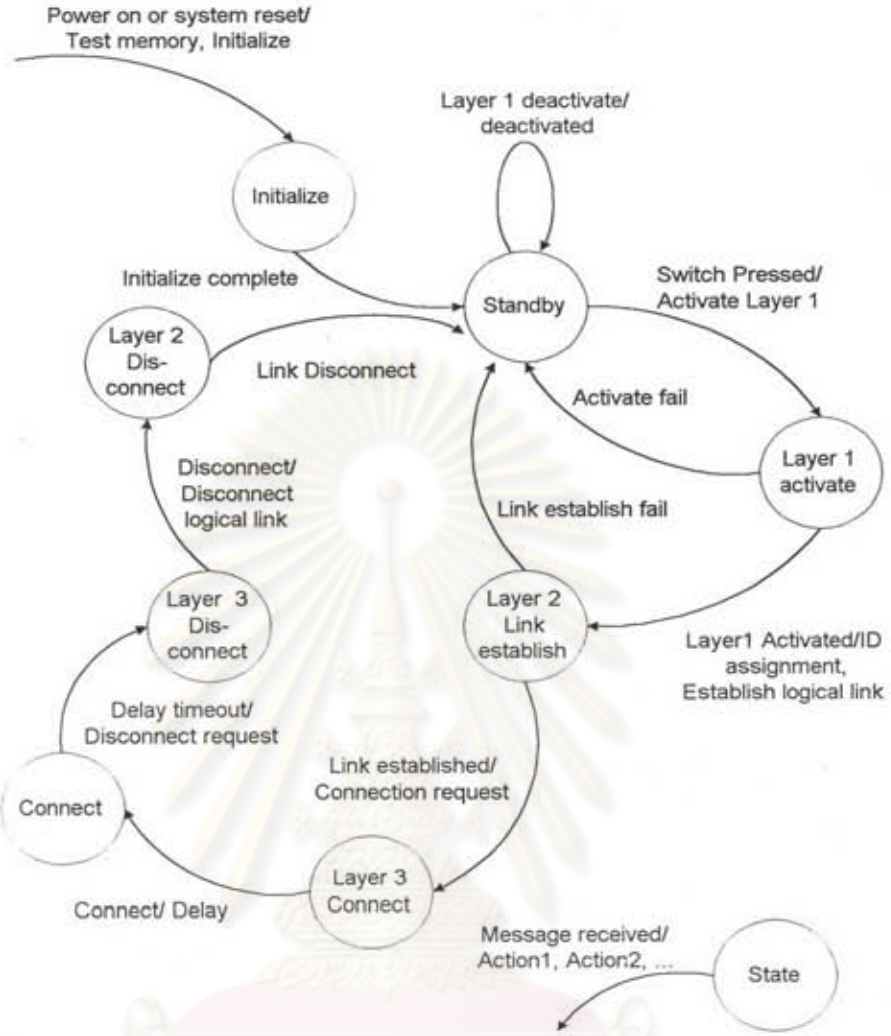


รูปที่ 3.5 แสดงการจุดต่อ Protocol Analyzer เข้ากับอุปกรณ์สื่อสารปลายทางและ LT-S

Protocol Analyzer นี้สามารถที่จะติดตาม แสดงผลสถานะและข้อมูลที่จุดอ้างอิง S ได้ทั้งชั้นที่ 1 2 และ 3 แต่ไม่สามารถควบคุมสถานะและข้อมูลที่จุดอ้างอิง S ได้ (Navtel Canada Inc., n.d.) จึงเป็นอีกเหตุผลหนึ่งที่ต้องออกแบบและสร้าง LT-S ขึ้นมาเอง

Control Flow ของอุปกรณ์สื่อสารปลายทาง

หลังจากที่ได้ออกแบบบล็อกไดอะแกรมของระบบทั้งหมดแล้ว ขั้นตอนต่อไปเป็นการออกแบบ Control Flow (John B. Peatman, 1988) ซึ่งบอกถึงขั้นตอนการทำงานที่สำคัญของอุปกรณ์สื่อสารปลายทางและลักษณะของการทำงานของซอฟต์แวร์ Control Flow ของอุปกรณ์สื่อสารปลายทางสามารถแสดงได้ดังรูปที่ 3.6



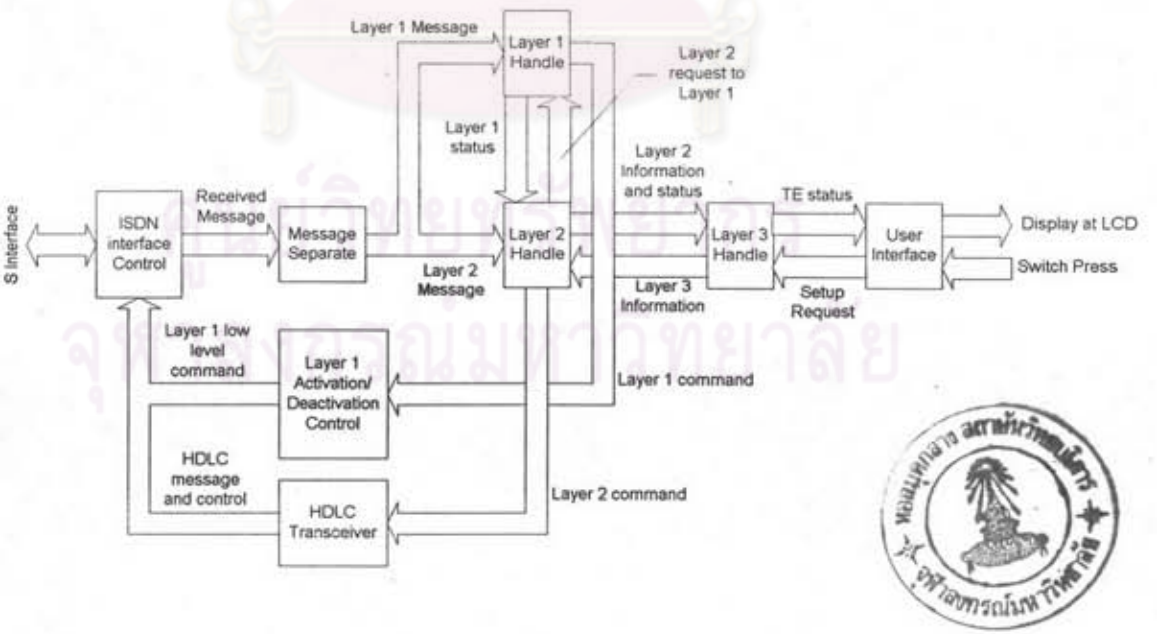
รูปที่ 3.6 แสดง Control Flow ของอุปกรณ์สื่อสารปลายทาง

จาก Control flow จะเห็นว่าอุปกรณ์สื่อสารปลายทางที่ออกแบบมีการทำงานในส่วนของชั้นที่ 1 และ 2 คลอบคลุมการทำงานทั้งหมด หากเกิดข้อผิดพลาดในการทำงานในระดับชั้นที่ 1 หรือ 2 เช่น ในชั้นที่ 1 เกิด deactivate (ITU-T, 1988a) ในขณะที่ชั้นที่ 2 กำลังสร้าง logical link (ITU-T, 1988b) อุปกรณ์สื่อสารปลายทางก็ยังคงทำงานอยู่ได้ เนื่องจากมีสถานะปลายทางรองรับ ดังรูปที่ 3.6 ในกรณีตัวอย่างข้างต้นนี้อุปกรณ์สื่อสารปลายทางอยู่ที่สถานะ Layer 2 Link establish และได้รับข้อความ Link establish fail ก็จะไปที่สถานะ standby ส่วนการทำงานในชั้นที่ 3 ซึ่งประกอบด้วยสถานะ Layer 3 connect, Connect, Layer 3 Disconnect ทั้ง 3 สถานะนี้หากเกิดข้อผิดพลาดในระหว่างการทำงาน เช่น ไม่ได้รับบริการจากโครงข่าย จะไม่มีสถานะรองรับปลายทาง ทำให้อุปกรณ์สื่อสารปลายทางทำงานต่อไปไม่ได้

การออกแบบอุปกรณ์สื่อสารปลายทางในลักษณะนี้ เนื่องจากต้องการให้อุปกรณ์สื่อสารปลายทางทำงานในชั้นที่ 1 และ 2 ได้อย่างสมบูรณ์ เพื่อสร้างซอฟต์แวร์พื้นฐานที่จำเป็นของอุปกรณ์สื่อสารปลายทาง ซึ่งจะทำให้การพัฒนาอุปกรณ์สื่อสารปลายทางที่มีประสิทธิภาพสูงขึ้นสามารถทำได้ง่าย ผู้ที่ต้องการพัฒนาอุปกรณ์สื่อสารปลายทางให้ทำงานต่างๆ สามารถนำอุปกรณ์สื่อสารปลายทางที่ออกแบบและสร้างขึ้นนี้ไปใช้งานได้โดยไม่ต้องแก้ไขซอฟต์แวร์ที่ทำงานในระดับชั้นที่ 1 และ 2 อีก ส่วนการทำงานในชั้นที่ 3 ได้ออกแบบให้ทำงานตามที่ได้กล่าวไว้กล่าวข้างต้น เพื่อเป็นตัวอย่างหนึ่งในการที่จะเรียกใช้งานซอฟต์แวร์ของอุปกรณ์สื่อสารปลายทางที่สร้างขึ้นเท่านั้น

Data Flow ของอุปกรณ์สื่อสารปลายทาง

เมื่อได้ออกแบบ Control Flow แล้ว ขั้นตอนต่อไปเป็นการออกแบบ Data Flow (John B. Peatman, 1988) ซึ่งเป็นการแสดงการไหลของข้อมูลทั้งหมดและแสดงการเปลี่ยนแปลงของข้อมูลเมื่อผ่านฟังก์ชั่นต่างๆ ฟังก์ชั่นเหล่านี้อาจเป็นซอฟต์แวร์ ฮาร์ดแวร์ หรือทั้งซอฟต์แวร์และฮาร์ดแวร์ที่ทำงานร่วมกันก็ได้ ซึ่งสามารถนำไปประมาณขนาดของหน่วยความจำข้อมูลและหน่วยความจำโปรแกรม เพื่อนำไปเป็นข้อมูลพื้นฐานในการเลือกใช้ตัวประมวลผล (Processor) ต่อไป



รูปที่ 3.7 แสดง Data Flow ของอุปกรณ์สื่อสารปลายทาง

Data Flow ของอุปกรณ์สื่อสารปลายทางสามารถแบ่งได้เป็น 2 ส่วนคือ ส่วนที่ข้อมูลไหลจากจุดอ้างอิง S จนถึงการแสดงผลที่จอ LCD และส่วนของข้อมูลที่ไหลจากสัญญาณของปุ่มกดไปยังจุดอ้างอิง S

1. ส่วนของข้อมูลที่ไหลจากจุดอ้างอิง S จนถึงการแสดงผลที่จอ LCD สัญญาณไฟฟ้าที่คู่สายที่จุดอ้างอิง S จะถูกเปลี่ยนเป็นค่าแสดงสถานะของชั้นที่ 1 และข้อมูลในชั้นที่ 2 โดย ISDN Interface Control จากนั้นสถานะของชั้นที่ 1 และข้อมูลในชั้นที่ 2 จะถูกแยกออกจากกันโดย Message Separate

สถานะของชั้นที่ 1 ที่รับได้จากจุดอ้างอิง S จะผ่าน Layer 1 Handle และเปลี่ยนเป็นคำสั่งสำหรับสั่งงานในชั้นที่ 1 โดยใช้ข้อมูล Layer 2 request to Layer 1 จาก Layer 2 Handle ประกอบเพื่อให้ได้คำสั่งที่ถูกต้อง คำสั่งสั่งงานนี้ (Layer 1 command ในรูปที่ 3.7) จะถูกเปลี่ยนเป็น Layer 1 low level command เพื่อนำไปเปลี่ยนเป็นสัญญาณไฟฟ้าออกสู่จุดอ้างอิง S โดยผ่าน ISDN Interface control ต่อไป

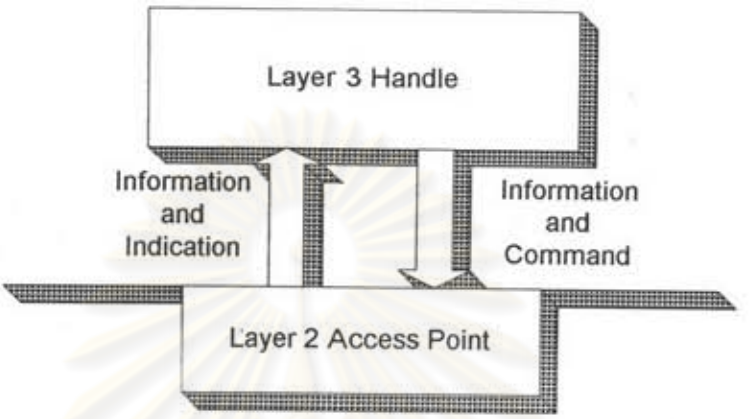
ข้อมูลอีกส่วนจาก Message Separate ได้แก่ส่วนที่เป็นข้อมูลของชั้นที่ 2 จะผ่านเข้าสู่ Layer 2 Handle และให้ข้อมูลออกมา 3 แบบขึ้นอยู่กับข้อมูลจาก Layer 1 Handle และ Layer 3 Handle ข้อมูลส่วนแรกเป็นข้อมูลที่แสดงความต้องการขอใช้บริการในชั้นที่ 1 ได้แก่ข้อมูล Layer 2 request to Layer 1 ตามรูปที่ 3.7

ส่วนที่ 2 เป็นส่วนที่ส่งเข้า HDLC Transceiver เพื่อนำไปเปลี่ยนเป็นสัญญาณไฟฟ้าออกสู่จุดอ้างอิง S โดยผ่าน ISDN Interface control อย่างไรก็ตามการที่จะได้ข้อมูลส่วนนี้ออกมานั้นต้องอาศัยข้อมูลจาก Layer 1 Handle และ Layer 3 Handle ประกอบด้วย ข้อมูลส่วนนี้เป็นข้อมูลและคำสั่งที่ใช้ในการควบคุมส่วน HDLC Transceiver ข้อมูลที่ส่งจะเป็นไปตามมาตรฐาน Q.920-Q.921 (ITU-T, 1988b)

ส่วนที่ 3 เป็นส่วนที่ส่งเข้า Layer 3 Handle ได้แก่ข้อมูลที่แสดงสถานะในชั้นที่ 2 เช่น มีหรือไม่มี logical link และข้อมูลของชั้นที่ 3 ที่มากับเฟรมของชั้นที่ 2 ข้อมูลส่วนนี้จะส่งเข้า Layer 3 Handle และ Layer 3 Handle จะให้ข้อมูลออกมา 2 ส่วน คือ TE status ตามรูปที่ 3.7 เพื่อผ่านไปเข้าส่วน User Interface เพื่อแสดงผลที่จอ LCD อีกส่วนหนึ่งจะเป็นข้อมูลที่ส่งกลับไปที่ Layer 2 Handle เพื่อส่งออกไปที่จุดอ้างอิง S ต่อไป

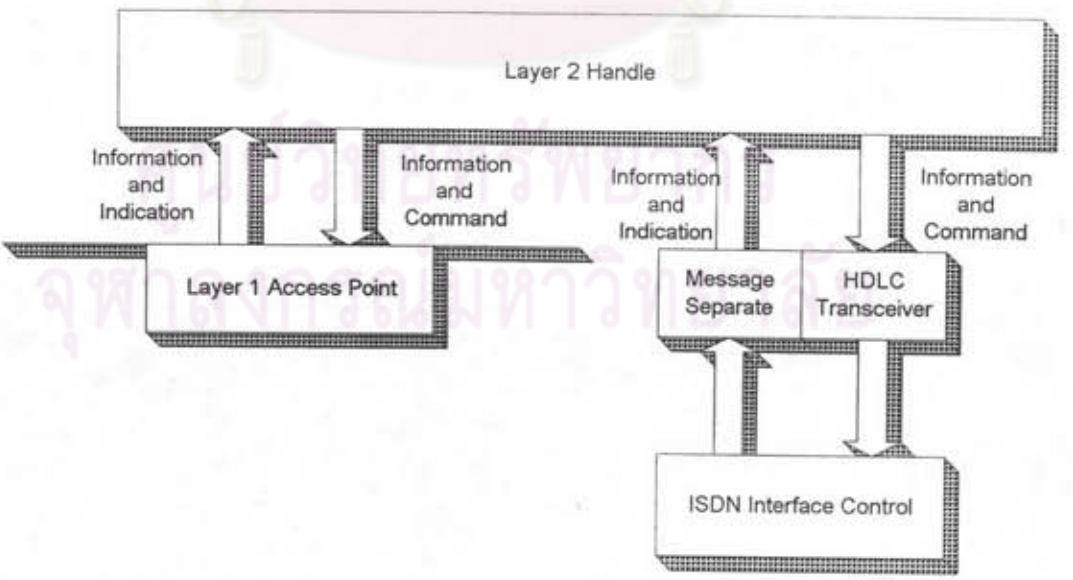
2. ส่วนของข้อมูลที่ไหลจากสัญญาณของปุ่มกดไปยังจุดอ้างอิง S สัญญาณไฟฟ้าที่ปุ่มกดจะถูกเปลี่ยนเป็นข้อมูลทางซอฟต์แวร์ ไปสู่ Layer 3 Handle ซึ่งส่วน Layer 3 Handle นี้จะนำไปเปลี่ยนเป็นข้อมูลที่เหมาะสมไปสู่ส่วน Layer 2 Handle และทำงานต่อไปดังอธิบายข้างต้น

หากสังเกตที่ส่วน Layer 3 Handle จะเห็นว่าส่วนตั้งแต่ส่วนนี้ลงไปจนถึงจุดอ้างอิง S ส่วน Layer 3 Handle จะไม่เห็นการทำงานต่างๆ เลย จะมองเห็นแต่เพียงจุดที่ต้องส่งข้อมูลเข้าไปเท่านั้น ดังรูปที่ 3.8



รูปที่ 3.8 แสดงจุดเข้าของชั้นที่ 2 เมื่อมองจากชั้นที่ 3

ในส่วนของชั้นที่ 2 ก็เช่นเดียวกัน ต่างกันที่จุดเข้าเมื่อมองจาก Layer 2 Handle มี 2 จุด ได้แก่ หากต้องการเข้าไปสู่ชั้นที่ 1 ก็เข้าทาง Layer 1 Handle หรือหากต้องการติดต่อกับชั้นที่ 2 ของโครงข่ายไอเอสดีเอ็นโดยตรงก็เข้าทาง HDLC Transceiver ดังรูปที่ 3.9



รูปที่ 3.9 แสดงจุดเข้าของ Layer 2 Handle

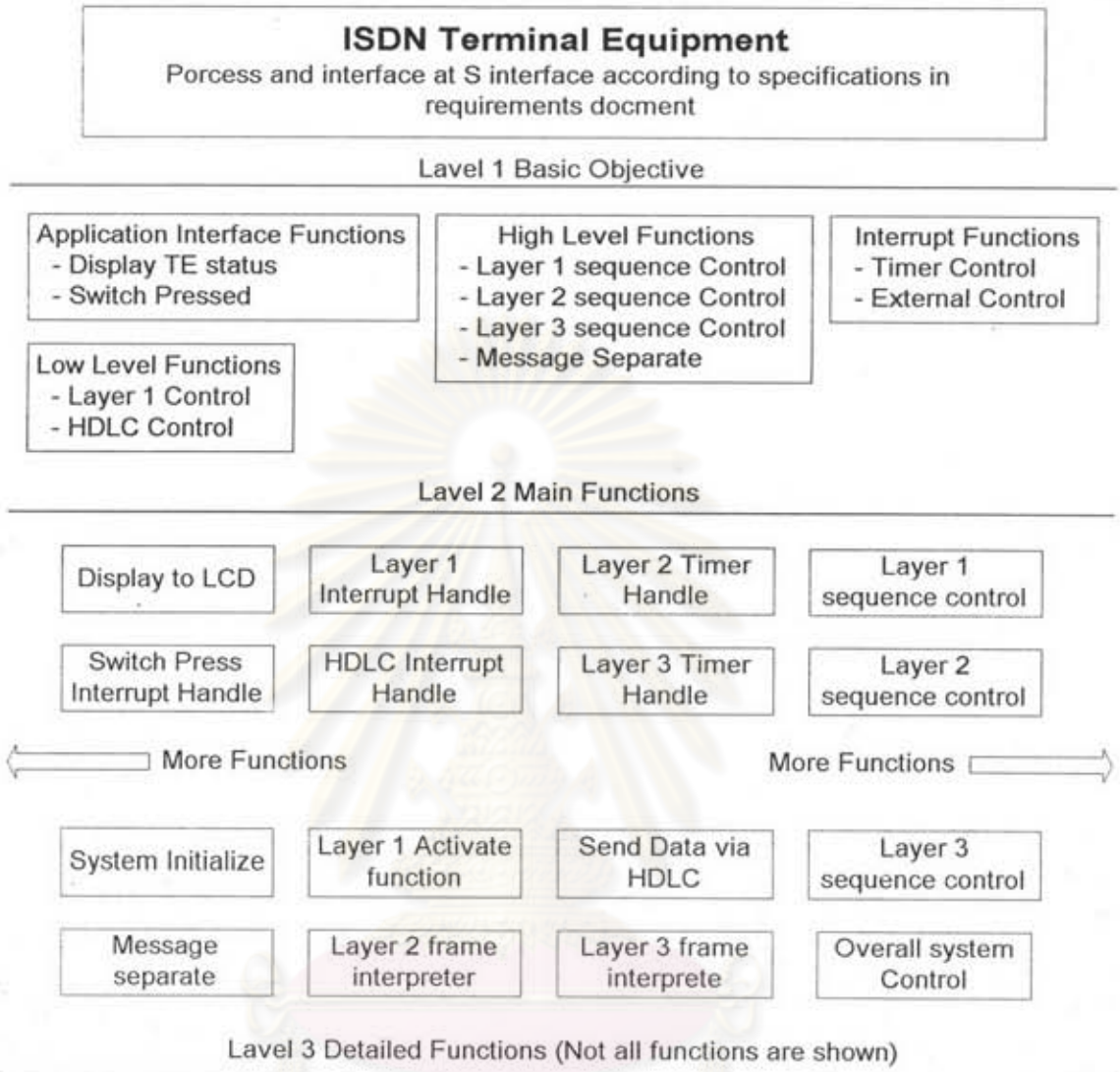
จากการออกแบบเช่นนี้ทำให้การติดต่อกันตั้งแต่ของชั้นที่ 2 ขึ้นไป ตามแบบจำลองของ OSI สามารถติดต่อกันได้เสมือนชั้นต่างๆ ของอุปกรณ์สื่อสารปลายทางติดต่อกับชั้นเดียวกันของโครงข่ายไอเอสดีเอ็นโดยตรง โดยมีการรับส่งข้อมูลผ่านจุดเข้าที่กำหนด นอกจากนี้การพัฒนาอุปกรณ์สื่อสารปลายทางต่อก็สามารถนำส่วนที่พัฒนาขึ้นมาแล้วนี้ไปใช้ได้โดยไม่ต้องแก้ไขซอฟต์แวร์ในที่ทำงานตั้งแต่ชั้นที่ 2 ลงมา ผู้พัฒนาเพียงแต่ส่งข้อมูลเข้าที่จุดเข้าเท่านั้น

Functional Decomposition ของอุปกรณ์สื่อสารปลายทาง

Control Flow และ Data Flow ที่ออกแบบไว้ข้างต้น ไม่ได้แสดงรายละเอียดมากพอที่จะบอกถึงการทำงานของอุปกรณ์สื่อสารปลายทาง Control Flow เป็นเพียงการขั้นตอนหลักๆ ของการทำงานของอุปกรณ์สื่อสารปลายทางเท่านั้น ส่วน Data Flow บอกถึงความสัมพันธ์ของข้อมูลหรือความเปลี่ยนแปลงของข้อมูลเท่านั้น แต่ Functional Decomposition จะบอกรายละเอียดที่มากขึ้น โดยบอกฟังก์ชั่นหรือโปรแกรมย่อยที่ต้องสร้างขึ้นมา ดังรูปที่ 3.10 แบ่งเป็น 3 ระดับ คือ

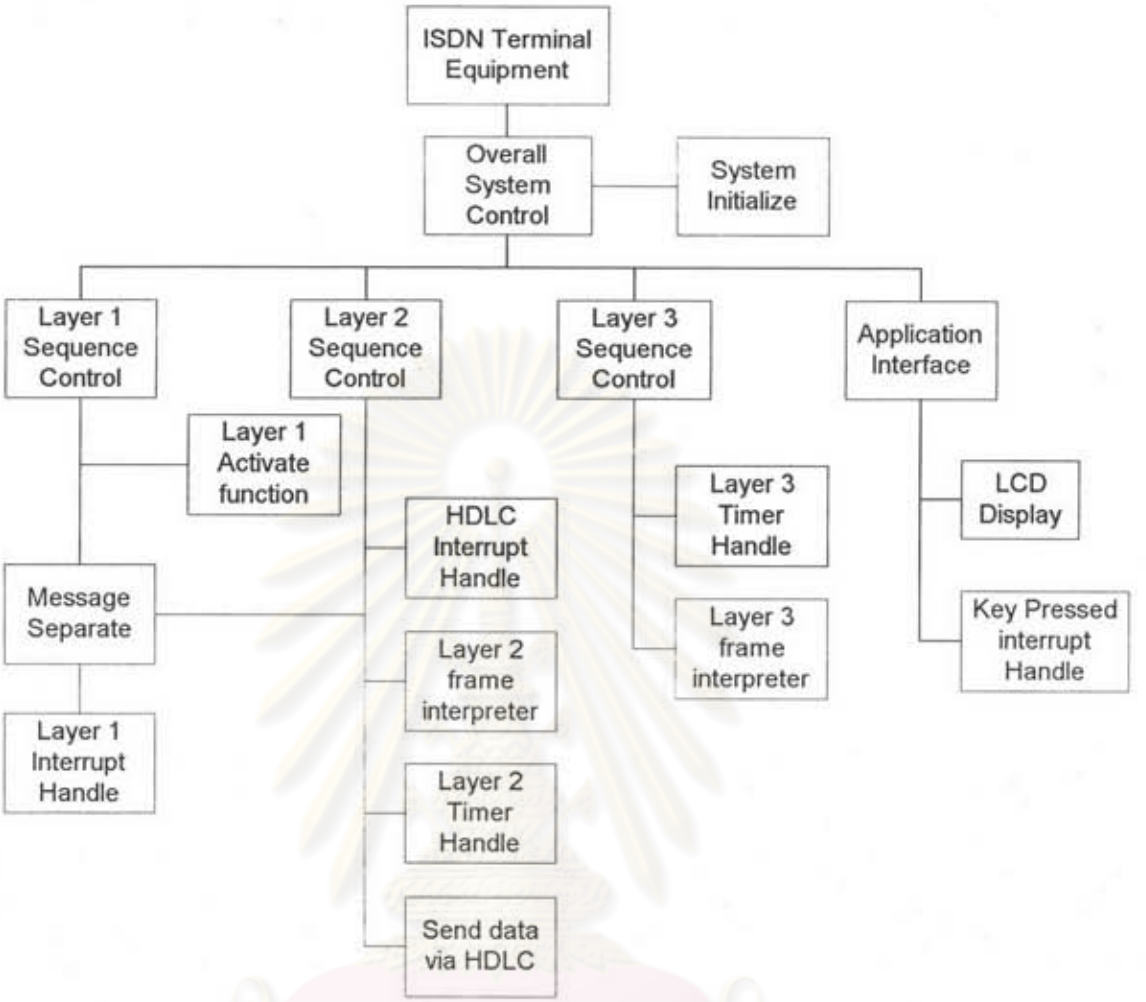
1. ระดับที่ 1 บอกถึงหน้าที่การทำงานหลักของอุปกรณ์สื่อสารปลายทาง
2. ระดับที่ 2 บอกถึงฟังก์ชั่นหรือโปรแกรมหลักที่จำเป็น
3. ระดับที่ 3 บอกถึงฟังก์ชั่นหรือโปรแกรมที่ต้องมีในรายละเอียดและจำนวนที่มากขึ้น

ฟังก์ชั่นหรือโปรแกรมใน Functional Decomposition นี้อาจมีหรือไม่มีในการเขียนโปรแกรมจริงก็ได้ Functional Decomposition เป็นเพียงการคาดคะเนหรือทำนายฟังก์ชั่นที่ต้องใช้งานเท่านั้น เนื่องจากฟังก์ชั่นที่ใช้งานจริงในโปรแกรมขึ้นอยู่กับข้อจำกัดหรือปัจจัยอื่นอีกหลายอย่าง เช่น โครงสร้างของโปรแกรม ขนาดของหน่วยความจำ เป็นต้น อย่างไรก็ตาม หากฟังก์ชั่นที่ใช้งานจริงในโปรแกรมใกล้เคียงกับ Functional Decomposition มากเท่าใด ก็จะยิ่งทำให้การใช้งานตัวประมวลผล (Processor) ใช้งานได้เต็มประสิทธิภาพมากขึ้นเท่านั้น ไม่ใช้งานมากหรือน้อยกว่าประสิทธิภาพของตัวประมวลผลเกินไป เปรียบเสมือนการนำเอาเครื่องซุปเปอร์คอมพิวเตอร์ไปใช้เป็นเครื่องคิดเลขหรือใช้ไมโครคอนโทรลเลอร์ไปควบคุมระบบขนาดใหญ่ ดังนั้นหาก Functional Decomposition ใกล้เคียงกับฟังก์ชั่นที่ใช้งานในโปรแกรมจริงๆ มากเท่าใด ก็จะทำให้ใช้งานตัวประมวลผลได้เต็มความสามารถและเหมาะสมกับงานนั้นๆ ได้มากขึ้นเท่านั้น



รูปที่ 3.10 แสดง Functional Decomposition ของอุปกรณ์สื่อสารปลายทาง

รูปที่ 3.10 แสดงถึงฟังก์ชั่นที่คาดว่าต้องใช้งาน แต่ไม่ได้แสดงความสัมพันธ์ของฟังก์ชั่นต่างๆ ความสัมพันธ์สัมพันธ์ของฟังก์ชั่นต่างๆ สามารถแสดงได้ด้วยแผนภูมิต้นไม้ ดังรูปที่ 3.11



รูปที่ 3.11 แผนภูมิต้นไม้แสดงความสัมพันธ์ของฟังก์ชั่นต่างๆ

จากรูปที่ 3.11 จะพบว่าฟังก์ชั่น Message Separate ขึ้นอยู่กับฟังก์ชั่นที่สูงกว่าถึง 2 ฟังก์ชั่น ได้แก่ ฟังก์ชั่น Layer 1 Handle และ Layer 2 Handle ในขณะที่ฟังก์ชั่นอื่นขึ้นอยู่กับฟังก์ชั่นที่สูงกว่าเพียงฟังก์ชั่นเดียว จึงพอที่จะคาดคะเนได้ว่าฟังก์ชั่น Message Separate ต้องมีความซับซ้อนกว่าฟังก์ชั่นอื่น ข้อผิดพลาดที่เกิดที่ฟังก์ชั่นนี้อาจส่งผลต่อการทำงานของระบบในวงกว้าง ในขณะที่หากเกิดข้อผิดพลาดที่ฟังก์ชั่นอื่นจะส่งผลเพียงฟังก์ชั่นที่สูงขึ้นไปเพียงฟังก์ชั่นเดียวเท่านั้น จึงจำเป็นต้องติดตามการทำงานของฟังก์ชั่นนี้เป็นพิเศษหากอุปกรณ์สื่อสารปลายทางไม่ทำงาน หรือออกแบบให้มีความซับซ้อนน้อยลง เช่น แบ่งเป็นฟังก์ชั่นย่อย เป็นต้น

การประมาณความยาวโปรแกรมและหน่วยความจำที่ต้องการ

ก่อนที่จะประมาณความยาวโปรแกรม ต้องเลือกภาษาที่ใช้ในการพัฒนาอุปกรณ์สื่อสารปลายทางเสียก่อน ในวิทยานิพนธ์นี้ได้เลือกใช้ภาษาแอสเซมบลี (Assembly) เนื่องจากภาษาสูงอื่นๆ เช่น ภาษาซี หรือภาษาเบสิก หากทำการแปลด้วยตัวแปลภาษาแล้ว โปรแกรมที่ได้จะมีขนาดใหญ่เกินความจำเป็น เนื่องจากตัวแปลภาษาเหล่านี้มีการเก็บคำสั่งของภาษาไว้ในส่วนที่เรียกว่า Library เมื่อทำการแปลโปรแกรม ตัวแปลภาษาจะนำคำสั่งที่เก็บไว้ใน Library นี้มาใส่ไว้ในโปรแกรมที่ถูกแปลด้วย เพื่อให้โปรแกรมที่ถูกแปลแล้วสามารถทำงานทำงานได้โดยไม่ต้องพึ่งตัวแปลภาษาอีกต่อไป การทำเช่นนี้ทำให้โปรแกรมที่ได้มีขนาดใหญ่เกินความจำเป็น เนื่องจากคำสั่งต่างๆ จะอยู่ในโปรแกรมที่ถูกแปลนั้นทั้งคำสั่งโปรแกรมที่ใช้งานและไม่ได้ใช้งาน ผลของการที่โปรแกรมมีขนาดใหญ่ขึ้นโดยไม่จำเป็น นอกจากทำให้สิ้นเปลืองหน่วยความจำที่ใช้เก็บโปรแกรมแล้ว ยังทำให้โปรแกรมทำงานช้าลงโดยไม่จำเป็นอีกด้วย

ในการประมาณความยาวของโปรแกรม ต้องประมาณจำนวนชุดคำสั่งที่เป็นอิสระต่อกันและจำนวน Operand ที่ใช้ในฟังก์ชั่นต่างๆ เสียก่อน ชุดคำสั่งที่เป็นอิสระต่อกันหมายความว่าชุดคำสั่งที่มีหน้าที่การทำงานที่แตกต่างกัน ตัวอย่างเช่น คำสั่ง MOV ด้วยการอ้างตำแหน่งแบบต่างๆ กันถือว่าเป็นชุดคำสั่งเพียงคำสั่งเดียว (Boehm, B.W., 1988) สำหรับภาษาแอสเซมบลีของ 8031 ได้ประมาณให้มีจำนวนชุดคำสั่งประมาณ 60 ชุดคำสั่ง ส่วนการประมาณจำนวน Operand แสดงดังตารางที่ 3.1 โดยนำข้อมูลที่ใช้ในการประมาณมาจากการออกแบบข้างต้นทั้งหมด

ตารางที่ 3.1 แสดงการประมาณ Operand ของฟังก์ชั่นต่างๆ

ฟังก์ชั่น	จำนวน Operand
Display to LCD	50
- pointer at string position	2
- length to be displayed	1
Switch Press Interrupt Handle	10
Layer 1 Interrupt Handle	20
- Command from ISDN interface control	1

ตารางที่ 3.1 แสดงการประมาณ Operand ของฟังก์ชั่นต่างๆ (ต่อ)

ฟังก์ชั่น	จำนวน Operand
HDLC Interrupt Handle	60
- Received data buffer	4096
- Transmit buffer	4096
Layer 2 Timer Handle	20
- T200 timer and N200	2
- T202 timer and N202	2
- T203 timer and N203	2
- Temporary variable	2
Layer 3 Timer Handle	40
- T302 timer	1
- T303 timer	1
- T305 timer	1
- T308 timer	1
- T310 timer	1
- T313 timer	1
- Temporary variable	2
Layer 1 sequence Control	50
- Layer 1 current state	1
Layer 2 sequence Control	200
- Layer 2 current state	1
Layer 3 sequence Control	250
- Layer 3 current state	1
System Initialize	80
- Test memory	5
- Temporary Variable	5

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.1 แสดงการประมาณ Operand ของฟังก์ชั่นต่างๆ (ต่อ)

ฟังก์ชั่น	จำนวน Operand
Layer 1 Activate function	50
- Command to activate	1
- Activate class	1
- Result of activation procedure	1
Send data via HDLC	50
- pointer at data to be sent	4
- length to be sent	2
Message Separate	100
- received message	1
- pointer for scanning received message	2
- Temporary variable	6
Layer 2 frame Interpreter	30
- pointer for scanning received message	4
- Layer 2 frame type	1
Layer 3 frame Interpreter	50
- pointer for scanning received message	4
- Layer 3 frame type	1
Overall System Control	10
- System status	5
- System command	2
รวม	1070+60+4096+4096

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ค่าในตารางมีความหมายดังนี้

ฟังก์ชั่น	จำนวน Operand
ฟังก์ชั่น - ตัวแปร	
รวม	จำนวน Operand + จำนวนหน่วยความจำสำหรับตัวแปร + จำนวนหน่วยความจำสำหรับเก็บข้อมูลที่รับได้จากจุดอ้างอิง S + จำนวนหน่วยความจำสำหรับเก็บข้อมูลที่จะส่งออกสู่จุดอ้างอิง S

จากตารางที่ 3.1 ได้จำนวน operand ประมาณ 1,070 operand รวมกับจำนวนชุดคำสั่งอีก 60 ชุดคำสั่ง ได้ k=1,070+60=1,130 เพื่อนำไปแทนค่าในสมการ (Boehm, B.W., 1988)

$$N = k(0.5572+\ln k)$$

จะได้

$$N = 1,170*(0.5772+\ln 1,170)$$

$$N = 8,941.1$$

$$N = 8,940 \text{ บรรทัด}$$

ค่า N ที่คำนวณได้เป็นจำนวนบรรทัดของโปรแกรมที่เขียนด้วยภาษาซี แต่อุปกรณ์สื่อสารปลายทางนี้พัฒนาด้วยภาษาแอสเซมบลี จึงกำหนดให้ค่าตัวคูณสำหรับปรับค่าให้เป็นจำนวนบรรทัดในภาษาแอสเซมบลีเป็น 30 เปอร์เซ็นต์ จะได้จำนวนบรรทัดในภาษาแอสเซมบลีเท่ากับ 2,682 บรรทัด ค่าที่คำนวณได้เป็นค่าประมาณเท่านั้นมิได้หมายความว่าโปรแกรมที่พัฒนาขึ้นมาจะมีขนาด 2,682 บรรทัด อาจกล่าวได้ว่าโปรแกรมที่พัฒนาขึ้นมีความยาวประมาณ 2,600 ถึง 3,000 บรรทัด

ในภาษาแอสเซมบลี คำสั่ง 1 บรรทัดใช้เนื้อที่ของหน่วยความจำโปรแกรม 1 ถึง 4 ไบท์ (byte) แต่โดยส่วนใหญ่จะใช้เนื้อที่ประมาณ 2 ไบท์ต่อ 1 บรรทัด (Intel Corporation, 1986) ดังนั้นอุปกรณ์สื่อสารปลายทางนี้จึงต้องการหน่วยความจำโปรแกรมประมาณ 2*3,000 = 6,000 ไบท์

จากตารางที่ 3.1 สามารถประมาณหน่วยความจำข้อมูลได้เท่ากับ 60 ไบท์ และให้ค่าตัวคูณเพื่อความปลอดภัย (Safety factor) เป็น 200 เปอร์เซ็นต์ จะได้ขนาดของหน่วยความจำข้อมูล

ประมาณ 60*2=120 ไบท์ สำหรับจำนวนหน่วยความจำสำหรับเก็บข้อมูลที่รับได้จากจุดอ้างอิง S และจำนวนหน่วยความจำสำหรับเก็บข้อมูลที่จะส่งออกสู่จุดอ้างอิง S นั้น เป็นหน่วยความจำที่กำหนดให้อุปกรณ์สื่อสารปลายทางสามารถเก็บได้จริงๆ มิได้เป็นค่าประมาณ

จากการออกแบบและการประมาณทรัพยากรที่ระบบต้องการทั้งหมดดังได้กล่าวมาแล้วข้างต้น จึงได้เลือกใช้ตัวประมวลผล (Processor) ตระกูล MCS-51 เบอร์ 8031 ของบริษัท Intel Corporation. ซึ่งมีหน่วยความจำข้อมูล 128 ไบท์ สามารถขยายหน่วยความจำข้อมูลภายนอกได้เป็น 64 กิโลไบท์ หน่วยความจำโปรแกรม 64 กิโลไบท์ และมี timer ภายในให้ใช้งาน 2 ชุด (พิพัฒน์, 2537) นอกจากนี้ยังสามารถเชื่อมต่อเข้ากับไอซีที่ทำหน้าที่เชื่อมต่อกับจุดอ้างอิง S ของบริษัท Siemens เบอร์ ISAC-S PEB2085 ได้โดยตรง ซึ่งจะกล่าวถึงในบทที่ 4

การประมาณจำนวน Operand และหน่วยความจำข้อมูลดังตารางที่ 3.1 นั้น ต้องศึกษารายละเอียดของ PEB2085 และโปรโตคอลในชั้นที่ 1 2 และ 3 เสียก่อน จึงสามารถประมาณค่าได้ การศึกษารายละเอียดของ PEB2085 ทำให้สามารถประมาณจำนวน Operand และจำนวนหน่วยความจำข้อมูลของฟังก์ชั่นที่ทำงานเกี่ยวข้องกับ PEB2085 ได้ ส่วนการศึกษารายละเอียดของโปรโตคอลในชั้นที่ 1 2 และ 3 จะทำให้สามารถออกแบบรูปแบบการทำงานของฟังก์ชั่นที่เกี่ยวข้อง เช่น ฟังก์ชั่น Layer 2 Handle เป็นต้น ซึ่งนำไปสู่การประมาณจำนวน Operand และจำนวนหน่วยความจำข้อมูลได้ สำหรับขั้นตอนการปรับเปลี่ยนโปรโตคอลในชั้นที่ 1 2 และ 3 ไปเป็นฟังก์ชั่นจะได้กล่าวถึงในบทที่ 6 ต่อไป

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย