บทที่ 2

ระบบการสื่อสารข้อมูลและระบบเครือข่าย

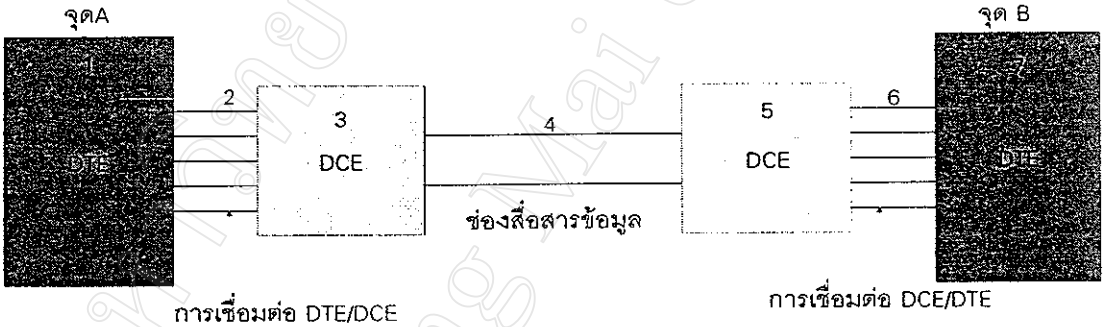
การรับส่งข้อมูลเป็นรูปแบบหนึ่งของการสื่อสาร ซึ่งข้อมูลที่รับส่งกันนั้นเป็นสัญญาณไฟฟ้าในรูปเลขฐานสอง “0” หรือ “1”

2.1 เจ็ดส่วนสำคัญในการสื่อสารข้อมูล

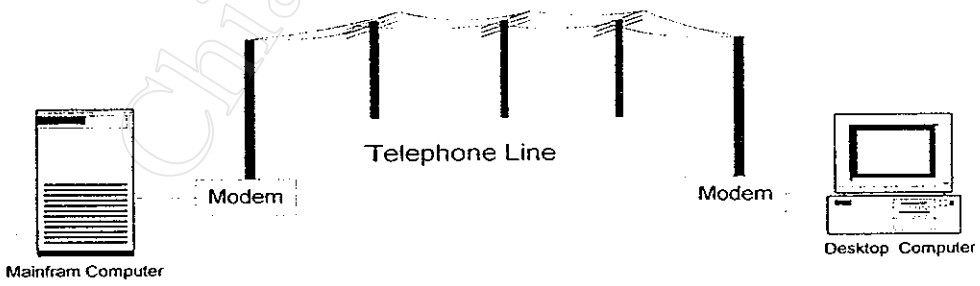
ระบบการสื่อสารข้อมูลสามารถกล่าวได้ว่ามีส่วนประกอบหลัก 3 ส่วนคือ

- ตัวส่ง (Transmitter หรือ Source)
- ตัวกลางหรือทางเดินของข้อมูล (Media, Channel หรือ Line)
- ตัวรับ (Receiver หรือ Sink)

ในระบบการสื่อสารข้อมูลของคอมพิวเตอร์จะต้องมีส่วนประกอบคือ 7 ส่วน



(ก)



(ข)

รูปที่ 2.1 วงจรเจ็ดส่วนสำคัญในการสื่อสารข้อมูล [2]

(ก) บล็อกไดอะแกรมของการส่ง

(ข) ลักษณะการใช้งานจริง

เช่น จากรูปที่ 2.1 จุด A เป็นแหล่งต้นกำเนิด จุด B เป็นแหล่งรับ วงจรการสื่อสารระหว่างจุด A และจุด B จะเป็นวงจรที่ประกอบด้วยอุปกรณ์ 7 ส่วน (Universal Seven-Part Data Circuit) ซึ่งประกอบด้วย

1) Data Terminal Equipment (DTE) ที่จุด A
2) การเชื่อมต่อ (Interface) ระหว่าง Data Terminal Equipment (DTE) หรือกับ Data Communication Equipment (DCE) ที่จุด A
3) DCE ที่จุด A
4) ช่องสื่อสารข้อมูลระหว่างจุด A และ B
5) DCE ที่จุด B
6) เชื่อมต่อระหว่าง DCE กับ DTE ที่จุด B
7) DTE ที่จุด B

ในที่นี้ DTE ที่จุด A และ B คืออุปกรณ์ปลายทาง (Terminal Devices) เช่น มอนิเตอร์ (Monitor) เครื่องโทรพิมพ์ (Teleprinter) หรือคอมพิวเตอร์ ส่วน DCE จะเป็นโมเด็ม ถ้าเส้นทางการสื่อสารเป็นแบบที่ใช้กับสัญญาณแอนะล็อก และจะเป็น Data Service Unit (DSU) ถ้าเส้นทางการสื่อสารเป็นแบบสัญญาณดิจิตอล

และ DTE ที่จุด A มีหน้าที่เป็น Source หรือ Sink หรือทั้งสองอย่างในขณะเดียวกันก็ได้ คือ รับ/ส่งข้อมูลผ่านเส้นทางการส่ง โดยให้ DCE เป็นผู้จัดการรับ/ส่ง ให้คอมพิวเตอร์ขนาดใหญ่หรืออุปกรณ์ใดก็ตามที่สามารถรับ/ส่งข้อมูลได้จัดเป็น DTE ทั้งสิ้น

จุดมุ่งหมายหลักของระบบการสื่อสารข้อมูลก็คือ การรับ/ส่งข้อมูลระหว่างจุด A กับ B ต้องมีความถูกต้อง ซึ่งข้อมูลเหล่านั้นอาจจะเป็นข้อมูลที่ DTE เป็นผู้นำไปใช้เอง หรืออาจเป็นเพียงผู้ประมวลผลให้มนุษย์ใช้ก็ได้

DCE มีหน้าที่เพียงเคลื่อนย้ายข้อมูลจากจุด A ไปยังจุด B โดยไม่สนใจเลยว่าข้อมูลที่ผ่านเข้ามาในตัวมันนั้น ประกอบด้วยข้อความอะไรบ้าง

2.2 วิธีการถ่ายโอนข้อมูล

การสื่อสารระหว่างเครื่องควรจะพิจารณาสัญญาณที่ส่งออกมาจากเครื่องและรับเข้าไปในเครื่องว่าเป็นอย่างไร

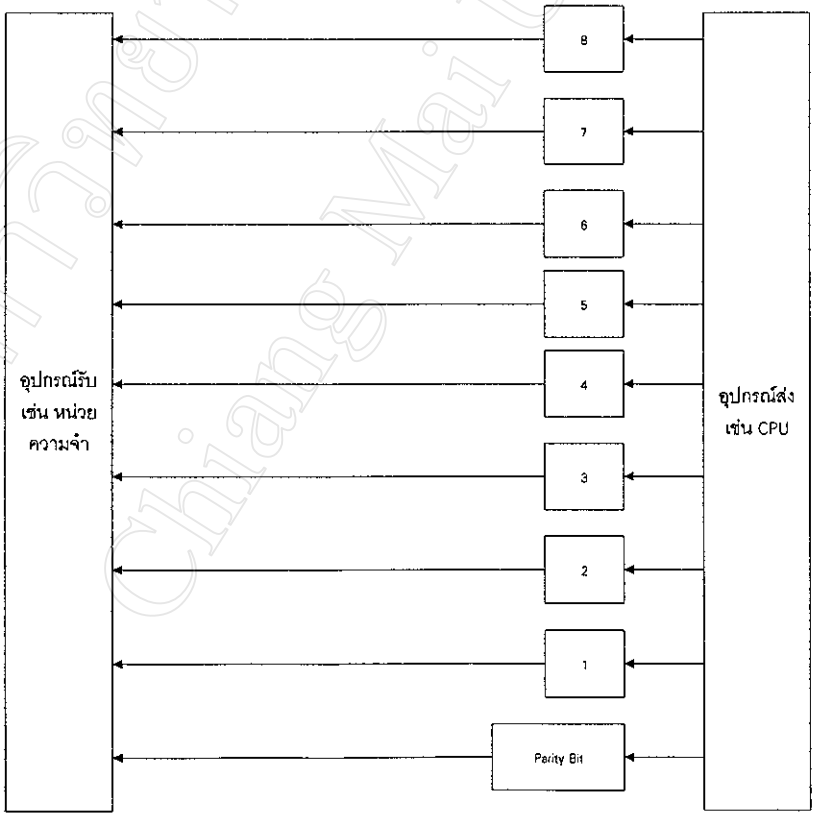
2.2.1 การถ่ายโอนข้อมูลแบบขนาน

ลักษณะของการส่งข้อมูลแบบขนาน ทำได้โดยการส่งข้อมูลออกมาทีละ 1 ไบต์ คือ 8 บิต

จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ตัวกลางระหว่าง 2 เครื่องจะต้องมีช่องทางให้ข้อมูลเดินทางอย่างน้อย 8 ช่องทาง โดยมากจะเป็นสายขนานให้กระแสไฟฟ้าวิ่งมากกว่าจะเป็นตัวกลางชนิดอื่น เนื่องจากมีการบันทอนสัญญาณไปกับความต้านทานของสาย ระยะทางระหว่าง 2 เครื่อง ไม่ควรจะเกิน 100 ฟุตเฉพาะในกรณีของการเชื่อมต่อแบบ Centronics ปัญหาที่เกิดขึ้นหากระยะทางของสายมากกว่านี้ก็คือ ระดับของกราวด์ในทางไฟฟ้าที่จุดรับผิดไปจากจุดส่ง ทำให้เกิดการผิดพลาดในการรับสัญญาณลอจิกทางฝ่ายรับ

นอกจากสายที่เป็นทางเดินของข้อมูลแล้วอาจจะมีทางเดินของสัญญาณควบคุมอื่น ๆ อีก เป็นต้นว่าบิตที่บอกภาวะของสัญญาณ เพื่อเป็นการตรวจสอบความผิดพลาดของการรับสัญญาณที่ปลายทาง หรือสายที่ควบคุมการโต้ตอบ (Hand-shake)

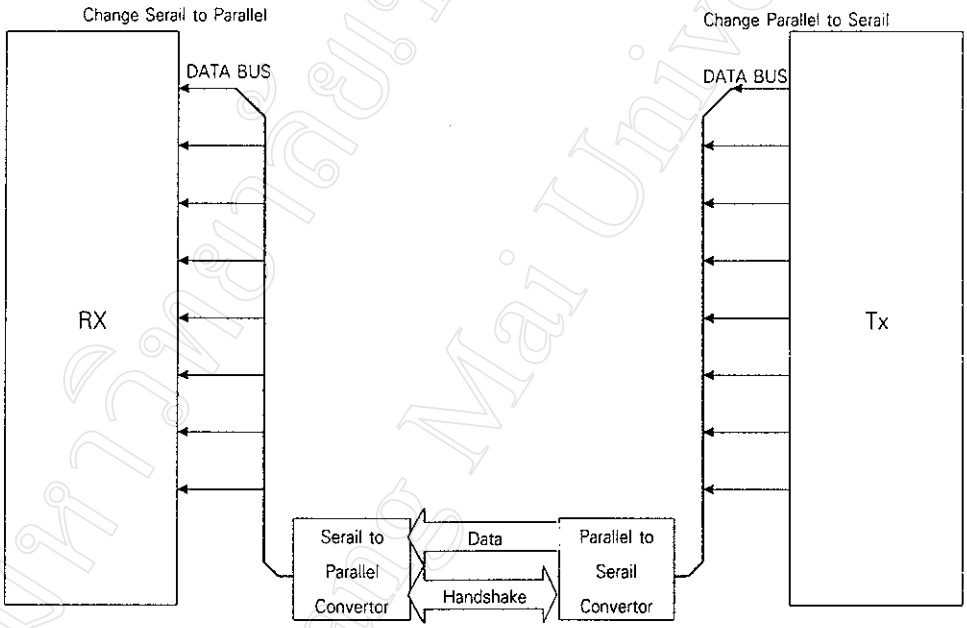
จะเห็นว่าการส่งแบบขนานส่วนมากจะทำในระยะใกล้ ๆ เนื่องจากจะต้องมีช่องทางเดินของสัญญาณมากกว่า 8 สาย และอุปกรณ์ที่ติดต่อแบบขนานกับคอมพิวเตอร์ก็เห็นจะได้แก่เครื่องพิมพ์และหน่วยความจำ



รูปที่ 2.2 การส่งข้อมูลแบบขนาน [2]

2.2.2 การถ่ายโอนข้อมูลแบบอนุกรม

ในการถ่ายโอนข้อมูลแบบอนุกรม ข้อมูลถูกส่งออกมาทีละบิต ระหว่างจุดส่งและจุดรับ จะเห็นว่าการส่งข้อมูลแบบนี้จะช้ากว่าแบบขนานที่กล่าวมาแล้ว แต่ตัวกลางการสื่อสารต้องการเพียงช่องเดียวหรือสายเพียงคู่เดียว ค่าใช้จ่ายในสื่อกลางจะต้องถูกกว่าแบบขนานอย่างแน่นอน สำหรับการส่งระยะทางไกล ๆ โดยเฉพาะเมื่อเรามีระบบการสื่อสารทางโทรศัพท์ไว้ใช้งานอยู่แล้วย่อมจะเป็นการประหยัดกว่าที่จะทำการติดต่อสื่อสารทีละ 8 ช่อง เพื่อการถ่ายโอนข้อมูลแบบขนานอย่างแน่นอน



รูปที่ 2.3 การส่งข้อมูลแบบอนุกรม [2]

จากรูปแสดงให้เห็นการส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นอนุกรมเสียก่อนแล้วค่อยทยอยส่งออกทีละบิตไปยังจุดรับ ณ ที่จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาทีละบิต ให้เป็นสัญญาณแบบขนานซึ่งลงตัวพอดี นั่นคือ บิตที่ 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดี การที่จะทำให้การแปลงสัญญาณจากอนุกรมทีละบิตให้ลงพอดีนั้นจำเป็นจะต้องมีกลไกที่เหมาะสม เพื่อป้องกันการผิดพลาดในการรับ กลไกที่ว่านี้แบ่งเป็น 2 แบบ

1) การสื่อสารแบบซิงโครนัส

2) การสื่อสารแบบอะซิงโครนัส

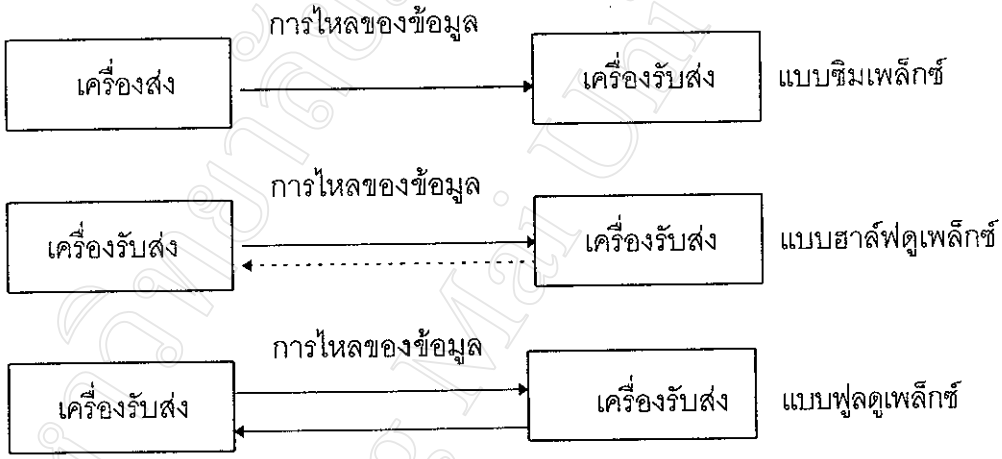
2.2.3 รูปแบบของการติดต่อสื่อสารแบบอนุกรม

การติดต่อแบบอนุกรมอาจจะแบ่งตามรูปลักษณะได้ 3 แบบ

1) **แบบซิมเพล็กซ์** (Simplex) ข้อมูลส่งได้ในทางเดียวเท่านั้นบางครั้งก็เรียกว่าการส่งทิศทางเดียว (Unidirectional)

2) **แบบฮาล์ฟดูเพล็กซ์** (Half Duplex) ข้อมูลสามารถส่งได้ทั้งสองสถานีแต่จะต้องผลัดกันส่งและผลัดกันรับ จะส่งและรับพร้อมกันไม่ได้

3) **แบบฟูลดูเพล็กซ์** (Full Duplex) ทั้งสองสถานีสามารถรับและส่งได้ในเวลาเดียวกัน



รูปที่ 2.4 แบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม [2]

การส่งแบบฟูลดูเพล็กซ์และฮาล์ฟดูเพล็กซ์ ไม่ขึ้นกับจำนวนของสายในการติดต่อ บางครั้งคำว่า ทูไวร์ (Two Wire) หรือสองเส้น และโฟร์ไวร์ (Four Wire) หรือสี่เส้น ใช้ในการบรรยายถึงลักษณะการสื่อสารข้อมูลซึ่งอาจจะทำให้เข้าใจผิด ฮาล์ฟดูเพล็กซ์สายโทรศัพท์ทั่วไปเป็นแบบ 2 เส้น ส่วนในสายที่เป็นแบบเช่า (Lease Line) นั้นส่วนมากจะเป็น 4 เส้น

2.2.4 ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม

ความเร็วของการถ่ายโอนข้อมูลแบบอนุกรม หน่วยวัดเป็นบิตต่อวินาที (bps) หน่วยที่บรรยายถึงการเปลี่ยนแปลงของสัญญาณใน 1 วินาที เรียกว่าบอดเรต (Baud Rate) หรืออัตราบอด คนทั่วไปยังเข้าใจสับสนระหว่างอัตราบอดและอัตราบิต (Bit Rate) การเปลี่ยนแปลงของสัญญาณ

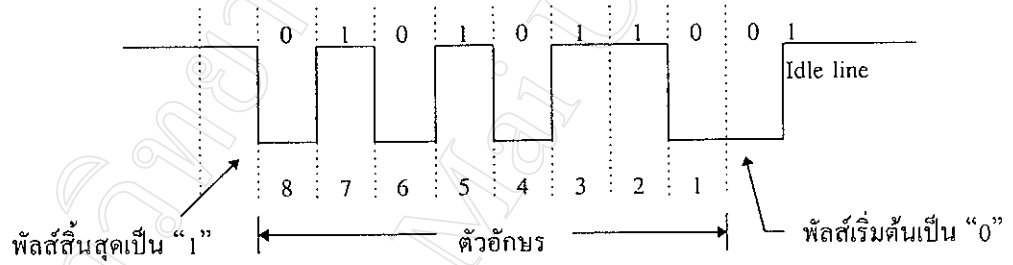
1 ครั้ง อาจจะแสดงถึงการส่งข้อมูลแบบอนุกรมมากกว่า 1 บิต ดังนั้นเราอาจเขียนรูปของสมการทางคณิตศาสตร์แสดงความสัมพันธ์ระหว่างอัตราบิตกับอัตราบอดได้ดังนี้

อัตราบิต (Bit Rate) = จำนวนบิตทั้งหมดที่ทำการรับส่งข้อมูลใน 1 วินาที

หรือ = 1/Td ;Td = ค่าคาบเวลาของสัญญาณข้อมูลที่รับส่ง

2.2.5 การสื่อสารแบบอะซิงโครนัส

การส่งแบบอะซิงโครนัส ตัวอักขระจะถูกส่งออกไปในเวลาใด ๆ ก็ได้ โดยไม่จำเป็นต้องมีความสัมพันธ์ระหว่างตัวอักขระ หรือจะต้องมีเวลาที่แน่นอนอย่างไร ตัวอักขระแต่ละตัว อาจจะทิ้งช่วงห่างกันเท่าใดก็ได้ เพราะอักขระแต่ละตัวจะมีบิตที่คอยบอกการเริ่มต้น และการสิ้นสุดของอักขระตัวนั้น (Start Bit, Stop Bit) ในทางปฏิบัติ ทางด้านรับจะทราบได้ว่ามีบิตเริ่มต้นเข้ามาแล้วเริ่มด้วยการเปลี่ยนสถานะจาก "1" เป็น "0" ดังแสดงในรูปที่ 2.5



รูปที่ 2.5 การส่งแบบอะซิงโครนัส [2]

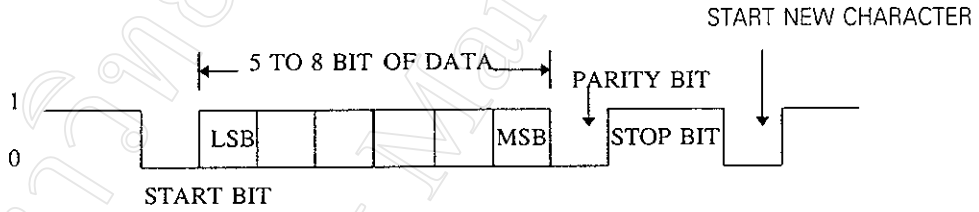
ทางด้านส่งจะเริ่มส่งบิตเริ่มต้นโดยการเปลี่ยนสถานะของค่าปกติที่เป็น "1" เป็น "0" หลังจากนั้นจะส่งบิตของข่าวสารสำหรับอักขระหนึ่งตัวออกไป ในขณะที่ด้านรับทราบถึงการเปลี่ยนสถานะจาก "1" เป็น "0" เป็นการบอกจุดเริ่มต้นของการทำงานว่า จะเริ่มภายหลังเวลาหรือเท่ากับครึ่งเวลาของความกว้าง 1 บิต หลังจากนั้นทางด้านรับก็จะสุ่มตัวอย่างสถานะของสายทุก ๆ ช่วงเวลา 1 บิต เพื่อหาค่ารหัสของตัวอักขระที่ส่งมาจนกว่าจะครบ 8 บิต (ถ้าเป็นรหัสแอสกี) และบิตถัดไปจะเป็นบิตสิ้นสุด หลังจากนั้นจะเว้นช่วงเวลาไปอีกนานเท่าไรก็ได้ในการเริ่มส่งตัวอักขระตัวใหม่ เพราะบิตเริ่มต้นจะคอยบอกอยู่แล้ว

ด้วยวิธีการส่งแบบมีบิตเริ่มต้นและจบด้วยบิตสิ้นสุดนี้ บางครั้งจึงเรียกการส่งแบบนี้ว่า Start-Stop Transmission บิตสิ้นสุดนั้นจะแตกต่างกันไปในแต่ละระบบ เช่น การส่งรหัสแอสกี บิตสิ้นสุดจะมีขนาดความกว้างเท่ากับความกว้างของข้อมูล 1,1.5 หรือ 2 บิต ซึ่งจะเป็นช่วงเวลาที่ทาง

ด้านรับทำงานหลังจากรับข้อมูลมาครบหนึ่งอักขระ เช่น การพิมพ์ในเครื่องพิมพ์ การเจาะรูบนกระดาษ ของเครื่องเจาะเทปกระดาษ เป็นต้น ช่วงเวลาของบิตสิ้นสุดนี้ขึ้นอยู่กับการทำงานของเครื่องอิเล็กทรอนิกส์ซึ่งมีความเร็วช้า โดยอาจต้องมีเวลาของบิตสิ้นสุดเท่ากับ 2 บิต แต่ในอุปกรณ์ประเภทเครื่องอิเล็กทรอนิกส์นั้น จะมีความเร็วในการทำงานมากซึ่งอาจจะมีความกว้างของบิตดังกล่าวเพียง 1 บิตเท่านั้น

ในการส่งแบบอะซิงโครนัสนี้ เนื่องจากอักขระทุกตัวที่ส่งออกมาต่างเป็นอิสระต่อกัน และทุก ๆ อักขระจะมีบิตบอกการเริ่มต้นและสิ้นสุด ดังนั้นหากเกิดสัญญาณรบกวน จะก่อให้เกิดความผิดพลาดจนทำลายความถูกต้องของตัวอักขระ ความผิดพลาดนี้ถ้าเกิดขึ้นในการส่งแบบซิงโครนัสอาจจะทำลายข่าวสารหมดทั้งบล็อกก็ได้

การส่งแบบอะซิงโครนัสนี้ พัฒนามาจากการส่งโทรพิมพ์ในสมัยก่อน ลักษณะของสัญญาณแสดงไว้ดังรูปที่ 2.6 เพื่อเพิ่มกลไกในการรับส่งอย่างถูกต้อง สัญญาณอะซิงโครนัส จะประกอบด้วยบิตเริ่มต้นหรือสตาร์ต และบิตสิ้นสุดหรือบิตหยุด



รูปที่ 2.6 ฟอร์แมตการสื่อสารแบบอะซิงโครนัส [2]

ขณะที่สถานะของการส่งเป็นแบบว่าง (Idle) คือ ยังไม่มีสัญญาณส่งออกมา จะมีสัญญาณหรือมีแรงดัน (หรือกระแส) ตลอดเวลา เพื่อความแน่ใจว่าฝ่ายรับยังติดต่ออยู่กับฝ่ายส่ง เมื่อเริ่มจะส่งข้อมูล สัญญาณของอะซิงโครนัสจะเป็น 0 หนึ่งช่วงสัญญาณนาฬิกา บิตนี้เรียกว่า บิตเริ่ม ตามหลังของบิตเริ่มก็จะเป็นข้อมูลสำหรับ 1 ตัวอักษร ซึ่งอาจจะมีขนาดตั้งแต่ 5 บิต จนถึง 8 บิต โดยบิตที่มีค่านัยสำคัญน้อยที่สุด (LSB) จะถูกส่งออกมาก่อนไล่ไปจนถึงบิตที่มีค่านัยสำคัญมากที่สุด (MSB) การเข้ารหัสอักขระนี้ส่วนมากจะนิยมใช้รหัส ASCII แรกเริ่มทีเดียวในงานของโทรพิมพ์เขาใช้รหัส Baudot ซึ่งใช้ 5 บิต ในการแทนอักขระ 1 ตัว ตามหลังข้อมูลก็จะเป็นบิตภาวะ (Parity Bit) ซึ่งอาจจะใช้หรือไม่ใช้ก็ได้ บิตภาวะทำหน้าที่เป็นตัวตรวจสอบความถูกต้องของสัญญาณที่ได้รับ บิตภาวะอาจจะเป็นแบบคู่ (Even Parity) หรือแบบคี่ (Odd Parity) หมายความว่าถ้าหากเป็นภาวะคู่

จำนวนบิตที่เป็น 1 ในช่วงบิตข้อมูลกับบิตภาวะรวมแล้วจะต้องเป็นจำนวนคู่ ผู้ส่งจะต้องทำหน้าที่ตรวจสอบข้อมูลแล้วใส่บิตภาวะเอง ฝ่ายรับเมื่อรับแล้วก็จะต้องตรวจสอบดูว่าเป็นจริงดังสถานะที่ตั้งเอาไว้หรือไม่ หากผิดพลาดก็หมายความว่าสัญญาณที่รับนั้นผิดพลาดไปจากที่สถานีส่งส่งออกมา แต่ความผิดพลาดที่จะตรวจพบได้นั้นต้องผิดเป็นจำนวนคี่เท่านั้น คือผิดไป 1 บิต 3 บิต หรือ 5 บิต พร้อมกัน เพราะถ้าผิดเป็นจำนวนคู่ ผลรวมของจำนวนบิตที่เป็นหนึ่งก็ยังเป็นคู่อยู่ จึงตรวจไม่พบความผิดพลาด ในกรณีของภาวะคี่ ก็มีผลเช่นเดียวกันกับภาวะคู่ อย่างไรก็ตามโอกาสที่จะผิดพลาด 2 บิตพร้อมกันมีน้อยมาก

ย้อนกลับมาดูสัญญาณอะซิงโครนัสใหม่ หลังจากบิตภาวะแล้วก็จะต้องมีบิตหยุดซึ่งเป็น 1 ความกว้างของบิตหยุดอาจจะเป็น 1,1.5 หรือ 2 พัลส์ของสัญญาณนาฬิกา แล้วแต่ผู้รับและผู้ส่งจะตกลงใช้กันเอง การเริ่มใช้พอร์ตอนุกรม (ทางออกอนุกรม) จึงจำเป็นจะต้องตั้งค่าต่าง ๆ สำหรับเป็นการส่งแบบอนุกรมอันได้แก่

1) ความเร็วในการส่ง

2) ความยาวรหัส 1 อักขระ

3) บิตตรวจสอบ

4) จำนวนบิตหยุด

ในการส่งโทรพิมพ์หรือโทรเลขสมัยก่อนนี้ใช้ความเร็วเพียง 70 บอด และ 110 บอด สำหรับคอมพิวเตอร์ความเร็วในการส่งมีให้เลือกตั้งแต่ 110,200,300,1200,2400,4800,9600 บอด และสูงไปกว่านั้น เนื่องจากมีไอซีหลายเบอร์ทำหน้าที่รับส่งแบบอะซิงโครนัสให้ใช้ การส่งแบบอนุกรมจึงสะดวกสบายสำหรับคนออกแบบพอร์ตอนุกรม

จะเห็นว่ากลไกในการซิงโครไนซ์ของการสื่อสารอะซิงโครนัส มีลักษณะเป็นไปทีละตัวอักขระ จำนวนพัลส์ของสัญญาณที่ส่งออกยังมีบางส่วนใช้ในการควบคุมการส่งได้แก่ บิตเริ่ม บิตหยุด และบิตภาวะ ทำให้ความเร็วการส่งอักขระต่อวินาทีน้อยลงไป การส่งสัญญาณด้วยความเร็ว 300 บอด สำหรับการเข้ารหัส 7 บิต จึงไม่ได้หมายความว่าส่งได้ 300 หารด้วย 7 อักขระต่อวินาที

2.2.6 การสื่อสารข้อมูลแบบอะซิงโครนัสที่มีการแมตช์ความเร็ว

การถ่ายโอนข้อมูลจากแบบอนุกรมจากอุปกรณ์เครื่องหนึ่งไปยังอีกเครื่องหนึ่งซึ่งอาจจะเป็นคอมพิวเตอร์หรืออุปกรณ์สื่อสารชนิดอื่น โดยตั้งสมมุติฐานว่าความเร็วในการเปลี่ยนสัญญาณจากขนานเป็นอนุกรมได้เร็วพอ และฝ่ายรับเปลี่ยนจากอนุกรมเป็นขนานแล้วนำไปแสดงบนจอพิมพ์ออกที่เครื่องพิมพ์หรือเก็บไว้ในดิสก์ทันทีได้ทันเวลา ด้วยความเร็วในแต่ละขั้นตอนของการทำงานเท่ากันทั้งฝ่ายรับและฝ่ายส่ง ไม่มีการหน่วงเวลาหรือการอินเตอร์รัพต์ระหว่างกลาง อย่างไร

ก็ตามสมมุติฐานนี้ย่อมไม่เป็นความจริง ฝ่ายส่งทำหน้าที่ส่งอย่างเดียวแต่ฝ่ายรับอาจต้องทำหน้าที่หลายอย่าง เช่น รับ แสดงผล เก็บ พิมพ์ เป็นต้น ความเร็วของฝ่ายรับหากไม่เพียงพอที่จะทำหลายอย่างให้ทันกับฝ่ายส่ง (ทั้งนี้ขึ้นอยู่กับความเร็วในการส่ง) จำเป็นจะต้องมีกลไกในการควบคุมการถ่ายโอน เทคนิคในการควบคุมความเร็วในการส่งมีอยู่หลายรูปแบบซึ่งอาจจะแบ่งออกได้เป็น 2 ลักษณะคือ แบบอะซิงโครนัส กับแบบซิงโครนัส

2.2.7 การควบคุมการส่งความเร็วในการทำงานของฝ่ายรับและฝ่ายส่งไม่เท่ากัน

เนื่องจากการใช้ภาษาในระดับสูงเขียนเป็นโปรแกรมสำหรับการควบคุมการทำงานของการถ่ายโอนข้อมูลแบบอนุกรม อาจจะใช้เวลามากกว่าที่จะรับข้อมูลเข้ามาได้ทันกับสถานีส่งข้อมูลมา จำเป็นจะต้องมีวิธีการควบคุมไม่ให้เกิดการสูญหายของข้อมูลที่สถานีส่งส่งมา วิธีการดังกล่าวนี้มีอยู่หลายวิธีในที่นี้จะกล่าวถึงวิธีการใช้บัฟเฟอร์

บัฟเฟอร์สำหรับการสื่อสารก็คือหน่วยความจำในคอมพิวเตอร์ซึ่งแบ่งแยกออกมาจากหน่วยความจำหลักสำหรับเก็บพักข้อมูลในการติดต่อชั่วคราว บัฟเฟอร์สำหรับการสื่อสารนี้ส่วนมากใช้สำหรับฝ่ายรับเท่านั้น เนื่องจากฝ่ายรับจำเป็นจะต้องตามฝ่ายส่งให้ทัน ถ้าหากฝ่ายรับใช้โปรแกรมภาษาเอสเซมบลีควบคุม มีความเร็วพออาจไม่จำเป็นต้องใช้บัฟเฟอร์สำหรับการสื่อสาร เนื่องจากภาษาเอสเซมบลีมีความเร็ว

ข้อมูลที่จัดส่งให้คอมพิวเตอร์ที่เป็นฝ่ายรับ ส่วนมากจะอ่านมาจากแฟ้มที่บันทึกไว้ในดิสก์ หากพิจารณาระหว่างการส่งข้อมูลออก ข้อมูลที่อ่านมาจากดิสก์จะมีลักษณะเป็นกลุ่มได้รับการนำมาสู่บัฟเฟอร์ การอ่านแต่ละกลุ่มดำเนินไปจนกระทั่งบัฟเฟอร์เต็ม การอ่านจะหยุดลงจนกระทั่งบัฟเฟอร์ถูกส่งออกไปหมดในลักษณะเข้าก่อนออกก่อน ข้อมูลก็จะถูกอ่านออกมาใส่ในบัฟเฟอร์ส่งอีกครั้ง โดยปกติบัฟเฟอร์ส่งจะมีขนาด 255 ตัวอักษรหรือประมาณ 3 บรรทัดของ 80 อักษร

บัฟเฟอร์รับของฝ่ายรับมีผลกระทบต่อการรับ-ส่งข้อมูลมากกว่าบัฟเฟอร์ส่ง บัฟเฟอร์รับทำหน้าที่เช่นเดียวกับบัฟเฟอร์ส่ง แต่ทิศทางของการไหลของข้อมูลอยู่ในทางตรงกันข้าม ฝ่ายรับรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์รับก่อนจนกว่าโปรแกรมควบคุมการสื่อสารจะนำข้อมูลออกไปจากบัฟเฟอร์รับเพื่อไปแสดงหรือพิมพ์หรือเก็บไว้ในแฟ้ม ในระบบควบคุมการทำงานของไมโครคอมพิวเตอร์อย่างเช่น IBM/PC มีกลไกบัฟเฟอร์รับส่งนี้ไว้อยู่ โปรแกรมในระดับสูงจึงเพียงแต่ทำหน้าที่ดึงเอาข้อมูลจากบัฟเฟอร์นี้ไปใช้ จะเห็นได้ชัดถึงความจำเป็นในการใช้บัฟเฟอร์เมื่อความเร็วในการส่งสูงเกินกว่า 600 บอด ระบบควบคุมการทำงานจึงถูกออกแบบมาเพื่อการสื่อสารข้อมูล โดยการใช้อินเตอร์รัพต์เข้าช่วย เมื่อมีข้อมูลเข้ามาที่พอร์ตอนุกรมเมื่อไร ระบบควบคุมจะอินเตอร์รัพต์การ

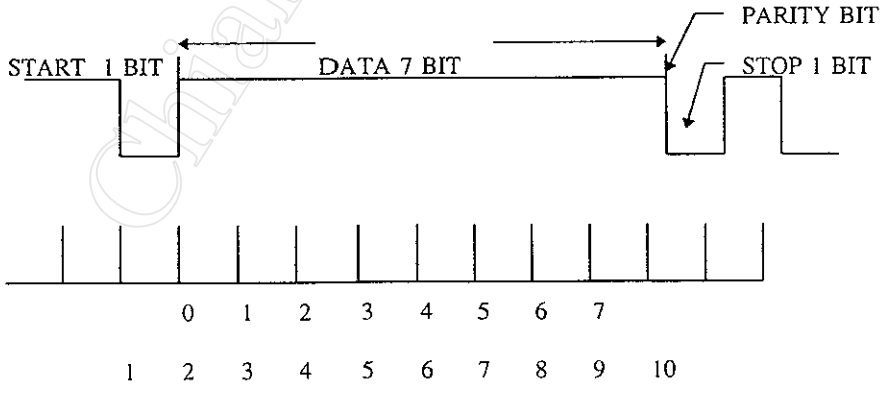
ทำงานเพื่อดึงข้อมูลไปใส่ในบัฟเฟอร์รับทันที เพื่อไม่ให้ข้อมูลที่รับหายไปก่อนที่ข้อมูลใหม่ส่งมาทับที่พอร์ตอนุกรม

หน้าที่ของโปรแกรมควบคุมการรับส่งก็คือการอ่านข้อมูลจากบัฟเฟอร์รับไปใช้ เมื่อถูกอ่านจากบัฟเฟอร์รับไปแล้ว ตัวที่อ่านออกไปก็จะหายไปจากบัฟเฟอร์ สรุปได้ว่าฝ่ายหนึ่งคือระบบควบคุมการทำงาน (OS) รับข้อมูลจากพอร์ตอนุกรมใส่บัฟเฟอร์ อีกฝ่ายหนึ่งคือโปรแกรมควบคุมการรับส่งดึงข้อมูลออกจากบัฟเฟอร์ เปรียบเสมือนคนหนึ่งตักน้ำใส่ตุ่มอีกคนหนึ่งตักออกจากตุ่ม ถ้าฝ่ายที่ตักออกมีความเร็วมากกว่าตุ่มก็จะมีโอกาสแห้ง ในทางตรงกันข้ามถ้าฝ่ายตักออกช้ากว่าฝ่ายตักเข้าโอกาสที่จะล้นตุ่มก็ย่อมจะมี ในทางสื่อสารเรียกว่าบัฟเฟอร์รับโอเวอร์โฟล์ว (Receive Buffer Overflow) การไหลล้นดังกล่าวทำให้ข้อมูลที่ได้รับหายไป

โปรแกรมที่เขียนด้วยภาษาเอสเซมบลีสามารถทำงานได้เร็ว และอาจไม่จำเป็นต้องใช้บัฟเฟอร์สำหรับการรับส่งเลยก็ได้ แต่ถ้าหากเขียนด้วยภาษาเบสิกจำเป็นจะต้องมีบัฟเฟอร์อย่างน้อย 1024 ไบต์ สำหรับการสื่อสารที่ความเร็วไม่เกิน 600 บอด

2.2.8 การควบคุมโดยใช้ XON/XOFF

ถึงแม้ว่าเราจะมีบัฟเฟอร์สำหรับการสื่อสารแล้วก็ตาม ในบางครั้งการถ่ายโอนข้อมูลด้วยความเร็วสูงและด้วยขนาดของแฟ้มที่จะทำการถ่ายโอนมีขนาดใหญ่กว่าบัฟเฟอร์สื่อสาร โอกาสที่ข้อมูลจะหายไปมีอยู่มาก สมมุติว่า เราใช้ความเร็วในการถ่ายโอนข้อมูล 9600 บิตต่อวินาที บิตหยุดเป็น 1 บิต ข้อมูล 7 บิต และภาวะเป็นคู่ใน 1 ตัวอักษร จะต้องใช้ 11 บิต



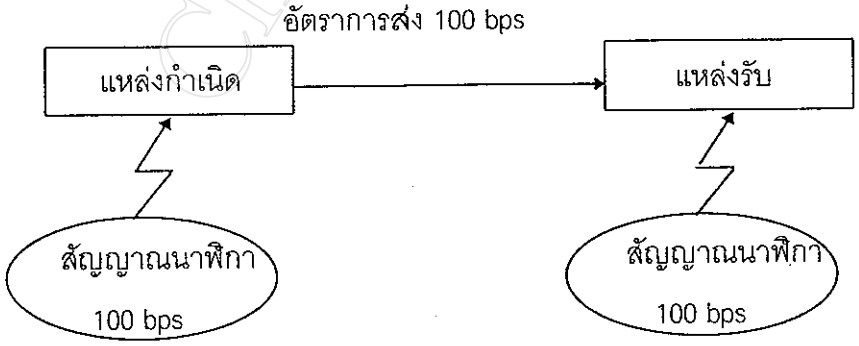
รูปที่ 2.7 รูปแบบของข้อมูล 1 ตัวอักษร [2]

เพราะฉะนั้นฝ่ายรับจะต้องอ่านข้อมูลจากพอร์ตอนุกรมทุก 110/9600 ได้ผลประมาณ .001 วินาที หรือ 1 มิลลิวินาที ถ้าเปรียบเทียบเป็นพัลส์ ได้ประมาณ 5000 พัลส์นาฬิกา (ความเร็วนาฬิกาของ IBM PC = 4.77 MHz หรือประมาณ .2 ไมโครวินาทีต่อหนึ่งพัลส์ 1000/.2 = 5000) ในเมื่อบัฟเฟอร์ไม่เพียงพอ เราก็จำเป็นจะต้องควบคุมการรับส่ง โดยการบอกให้ฝ่ายส่งหยุดส่งชั่วคราว (XOFF) จนกว่าฝ่ายรับจะจัดการเอาข้อมูลออกจากบัฟเฟอร์สื่อสารหมดเสียก่อน จึงบอกให้ฝ่ายส่งจัดการส่งต่อไป (XON) ในรหัส ASCII XON มีค่าเท่ากับ 17 XOFF มีค่าเท่ากับ 19 เป็นหน้าที่ของนักเขียนโปรแกรมที่จะต้องจัดการส่ง XOFF ออกไปให้ฝ่ายส่งได้รู้ ก่อนที่บัฟเฟอร์สื่อสารจะเต็มเสียก่อน

คอมพิวเตอร์เมนเฟรมส่วนมากจะมีระบบ XON/XOFF ให้ สำหรับการเชื่อมต่อทางด้านความเร็ว (Speed Matching) แต่โปรแกรมสื่อสารที่มีขายสำหรับ IBM/PC ไม่มี XON/XOFF ทุกตัว โดยมากโปรแกรมที่เขียนโดยภาษาเอสเซมบลีจะมี XON/XOFF ให้ โปรแกรมที่เขียนโดยภาษาเบสิกจะมีเพียงบางโปรแกรมเท่านั้น โปรแกรมที่เขียนโดยภาษาเอสเซมบลีความเร็วพอที่จะคอยตรวจสอบดู XOFF ที่ส่งมาจากฝ่ายรับ แต่ถ้าเป็นภาษาเบสิกความเร็วอาจจะไม่เพียงพอต่อการตรวจสอบ XOFF ที่ฝ่ายส่ง

2.2.9 การสื่อสารแบบซิงโครนัส

การส่งแบบซิงโครนัสเราแบ่งได้เป็น 2 แบบ คือ แบบซิงโครนัสบิตและแบบซิงโครนัสอักขระ ซิงโครนัสบิตเป็นการรับส่งที่ด้านรับต้องรู้ว่าจะรับบิตแรกจากสายส่งนั้นเมื่อใด และหลังจากรับมาแล้ว จะรับบิตที่ 2, 3,...เมื่อใด ซึ่งสามารถกระทำโดยเพิ่มสัญญาณนาฬิกา (Clock) เข้าไปที่จุดปลายทางของระบบทั้งสองด้านดังรูปที่ 2.8



รูปที่ 2.8 การให้สัญญาณนาฬิกาเพื่อการรับส่งแบบซิงโครนัส [2]

ซึ่งในบางระบบอาจใช้วิธีส่งสัญญาณนาฬิกาจากด้านส่งไปทางด้านรับด้วย และทางด้านรับจะนำสัญญาณที่ได้รับมาเป็นตัวขับสัญญาณนาฬิกาของตัวมันเอง เพื่อให้สัญญาณนาฬิกาทางด้านรับและส่งเท่ากันพอดี

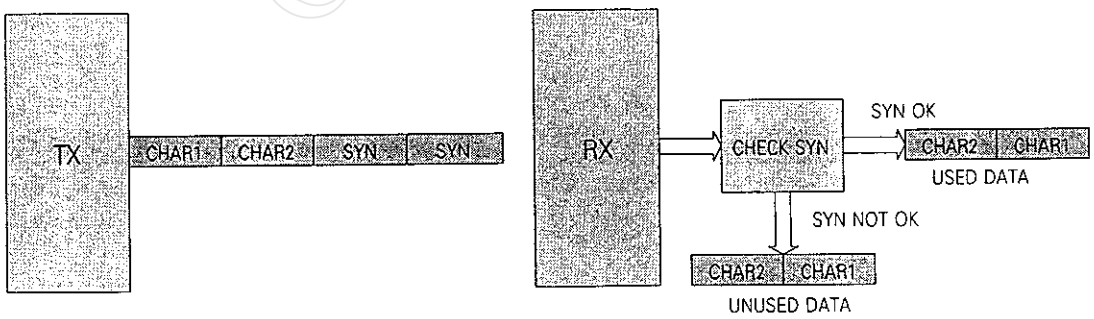
2.2.10 ซิงโครนัสอักขระ

ในการรับส่งข้อมูลตามสายนั้น แม้ว่าจะมีการจัดการกับบิตได้ดีแล้วก็ตามแต่ก็มีปัญหาอีกคือการนำเอาบิตของอักขระ (Character) หลาย ๆ ตัวมารวมกันเป็นบล็อก ดังนั้นถึงแม้ว่าบิตต่าง ๆ จะได้รับมาอย่างถูกต้องแล้วก็ตาม เรายังต้องทราบว่ากลุ่มของบิตที่แสดงถึงตัวอักขระต่าง ๆ นั้นเริ่มต้นที่บิตใด เพื่อที่เราจะแก้ปัญหาได้ ถ้าเราทราบว่าบิตใดเป็นบิตเริ่มต้นของตัวอักขระ และถ้าหากทราบว่าในตัวอักขระหนึ่งตัวนั้นมีกี่บิต รวมทั้งความเร็วของการสั่งของบิตต่าง ๆ โดยการนับจำนวนบิตที่ได้รับมาตามสายหลังจากทราบบิตแรกก็จะสามารถแยกตัวอักษรออกจากกันได้

เทคนิคการส่งข้อมูลแบบซิงโครนัสนั้น ใช้สำหรับส่งข้อมูลทั้งชุดไปครั้งเดียวโดยในการส่งแบบนี้ช่วงความกว้างของเวลาระหว่างบิตแต่ละบิตจะมีค่าเท่ากัน

ในรูปที่ 2.9 แสดงให้เห็นถึงการส่งอักขระต่าง ๆ แบบซิงโครนัส โดยมีการส่งชุดข้อมูลชุดหนึ่งก่อนหน้าการส่งชุดข้อมูลตัวอักขระ เพื่อทำการหาบิตแรกของอักขระตัวแรกให้เป็นไปอย่างถูกต้อง ข้อมูลชุดนี้เรียกว่า SYN Transmission Control Character (TC) ซึ่งประกอบด้วยข้อมูลขนาด 8 บิต เช่น 00010110 (มีภาวะเป็นคี่) และทางด้านรับก็จะคอยมองหาชุดของบิตดังกล่าว (Looking for SYNC) ซึ่งจะกระทำทุกครั้งเมื่อรับบิตใหม่เข้ามา จนกว่าจะได้รับข้อมูลตัวอักขระที่กล่าวมาแล้วข้างต้น

ในระบบนี้ส่วนมากมักจะมี SYN นำหน้าข้อมูล 3-4 ตัว เพื่อให้แน่ใจได้ว่าชุดของบิตตัวอักขระที่รับเข้าคือข้อมูลที่ถูกต้อง



รูปที่ 2.9 การส่งแบบซิงโครนัสที่มี SYN หลายตัว

ข้อแตกต่างระหว่างวงจรส่งข้อมูลอนุกรมแบบซิงโครนัสและอะซิงโครนัสก็คือ ความต่อเนื่องของข้อมูลที่ส่ง ในแบบซิงโครนัสข้อมูลที่ส่งออกมาแบบต่อเนื่อง ไม่มีบิตเริ่มหรือบิตหยุดหรือแม้กระทั่งบิตภาวะ โพรโทคอลที่ใช้ในการส่งแบบซิงโครนัสจึงแตกต่างไปจากโพรโทคอลแบบอะซิงโครนัส มีโพรโทคอลหลายแบบที่ใช้ในการส่งแบบซิงโครนัสดังจะกล่าวต่อไปนี้

2.3 โพรโทคอล (Protocol)

โพรโทคอลคือ กฎข้อบังคับที่กำหนดรูปแบบและขั้นตอนการทำงานของฮาร์ดแวร์ (Hard ware) และซอฟต์แวร์ (Software) เพื่อให้ทราบว่าจะเกิดข้อผิดพลาดในการรับส่งข้อมูลขึ้นมาเมื่อใด และในบางโพรโทคอลยังสามารถแก้ไขข้อผิดพลาดเหล่านั้นได้ด้วย

2.3.1 การใช้โพรโทคอล

การควบคุมการรับส่งก็คือ การส่งโพรโทคอล (Protocol Transfer) เทคนิคนี้จำเป็นจะต้องมีเหมือนกันทั้งฝ่ายรับและฝ่ายส่ง โดยการใช้อักขระควบคุมในตารางของ ASCII สำหรับควบคุมการส่งข้อมูลออกมาเป็นกลุ่มที่มีขนาดคงที่

การใช้โพรโทคอลอาจจะใช้อักขระต่อไปนี้ ในการควบคุมการส่งข้อมูลเป็นกลุ่ม ๆ

ETB End of Transmission Block

ETB มีค่าเท่ากับ 23 ในตาราง ASCII เป็นการบอกฝ่ายรับว่าขณะนี้สิ้นสุดการส่งข้อมูลกลุ่มหนึ่งแล้ว

ETX End of Text

ETX มีค่าเท่ากับ 03 ในตาราง ASCII เป็นการบอกฝ่ายรับว่าขณะนี้สิ้นสุดการส่งแล้ว

ENQ Enquiry

ENQ มีค่าเท่ากับ 05 ในตาราง ASCII เป็นอักขระที่ส่งมาจากฝ่ายรับ ขอให้ฝ่ายส่งส่งข้อมูลมา

NAK Negative Acknowledge

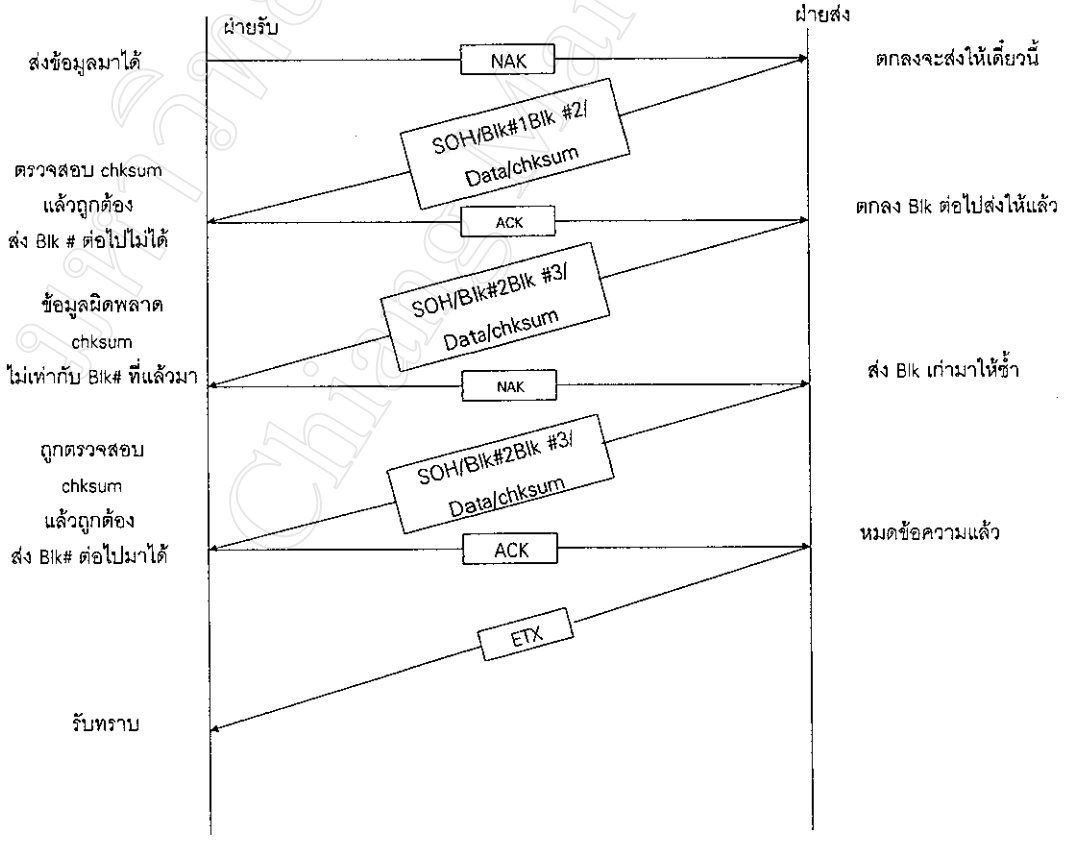
ในตารางรหัส ASCII มีค่าเท่ากับ 021 เป็นการบอกฝ่ายส่งว่าข้อมูลที่ได้รับนั้นผิดพลาด

ACK Acknowledge เป็นการบอกฝ่ายส่งว่าข้อมูลที่ได้รับนั้นถูกต้องแล้ว เป็นรหัส ASCII เท่ากับ 06

โพรโทคอลที่ใช้ใน IBM PC ส่วนมากจะเป็นโพรโทคอลที่มีชื่อว่า XMODEM ลักษณะการทำงานไว้ดังรูป 2.10

2.3.2 โพรโทคอล XMODEM

ฝ่ายส่งจะยังไม่ส่งข้อมูลจนกว่าจะได้รับ NAK จากฝ่ายรับ ฝ่ายส่งจะส่งข้อมูลออกไปโดยมีรูปแบบเริ่มด้วย SOH ตามด้วยอักขระ 2 ตัว สำหรับบอกกลุ่มของข้อมูลที่ส่งและตามด้วยส่วนเติมเต็ม 1 (1's complement) ของกลุ่มต่อไปที่จะส่ง ต่อจากนั้นก็จะเป็นข้อมูล 128 ไบต์ ตามหลังด้วย



รูปที่ 2.10 โพรโทคอล XMODEM [2]

การตรวจสอบข้อผิดพลาดโดยวิธีตรวจสอบผลบวก (Checksum) การตรวจสอบผลบวกคำนวณมาจากการบวกค่า ASCII ของข้อมูลที่ส่งออกไปทั้งหมด 128 ไบต์ แล้วหารด้วย 255 เศษที่เหลือก็คือค่า Checksum ข้างฝ่ายรับเมื่อแยกเอา SOH และหมายเลขบล็อก (Block Number) ทั้งสองออกไปแล้วก็จะเอาข้อมูลทั้ง 128 ไบต์มารวมกันเพื่อหาค่าตรวจสอบผลบวกเอา ค่าตรวจสอบผลบวกที่หาได้เปรียบเทียบกับค่าที่ได้รับ หากตรงกันก็ถือว่าข้อมูลที่ได้รับถูกต้อง จึงส่งสัญญาณ ACK ไปให้ฝ่ายส่งได้รู้ว่าขณะนี้ได้รับข้อมูลไว้ถูกต้องแล้วส่งกลุ่มของข้อมูลต่อไปมาได้ ถ้าหากค่าตรวจสอบผลบวกไม่ถูกต้อง ฝ่ายรับก็จะส่ง NAK ให้ฝ่ายส่งเพื่อเป็นการบอกให้รู้ว่าข้อมูลที่ได้รับผิดพลาด ขอให้ส่งกลุ่มของข้อมูลอันเก่ามาให้ใหม่ ฝ่ายส่งก็จะส่งข้อมูลกลุ่มเก่ามาให้ใหม่ การส่งใหม่จะดำเนินไป 9 ครั้ง หากยังคงได้รับแต่สัญญาณ NAK ฝ่ายส่งจะหยุดทำงานแสดงว่าตัวกลางการสื่อสารบกพร่อง

การที่ XMODEM ใช้เลขบอกกลุ่ม (Block Number) 2 ตัว (ตัวหนึ่งบอกกลุ่มที่ส่งขณะนี้ อีกตัวหนึ่งเป็นส่วนเติมเต็ม 1 ของกลุ่มต่อไป) เพื่อความแน่นอนว่ากลุ่มเดียวกันจะไม่ถูกส่งออกไปสองครั้ง ถ้าหากอักขระควบคุมการส่งเกิดสูญหายไประหว่างการส่ง ฝ่ายรับจะตรวจสอบดูว่ากลุ่มของข้อมูลที่ส่งมาเป็นกลุ่มที่ฝ่ายรับต้องการหรือไม่ ถ้าหากกลุ่มเก่าเกิดส่งมาใหม่อีกด้วยความผิดพลาดจาก ACK เป็น NAK ของฝ่ายส่ง ฝ่ายรับก็จะนำข้อมูลที่รับมาโยนทิ้งไป เมื่อทุกอย่างดำเนินไปอย่างเรียบร้อยจนสิ้นสุดแฟ้มที่จะส่ง ฝ่ายส่งก็จะส่ง ETX เป็นการบอกฝ่ายรับว่าหมดข้อความที่จะส่งแล้ว

XMODEM เหมาะสำหรับ IBM PC และมีคุณสมบัติเหนือกว่าโพรโทคอลชนิดอื่น 4 ข้อ คือ

1) ใช้อักขระควบคุมที่มีอยู่แล้วใน ASCII

2) สามารถจะใช้ภาษาในระดับสูงควบคุมได้ เช่น ภาษาเบสิก ภาษาปาสคาล

3) ต้องการบัฟเฟอร์สื่อสารแค่ 256 ไบต์

4) ระบบบริการข่าวสารด้วยคอมพิวเตอร์ โดยทั่วไปใช้โพรโทคอล XMODEM

ข้อดีของระบบการควบคุมการรับส่งข้อมูลแบบอนุกรมโดยการใช้ระบบการใช้โพรโทคอลก็คือ

1) โพรโทคอลบางชนิดสามารถเลือกขนาดของกลุ่มข้อมูลได้

2) สามารถส่งข้อมูลที่ไม่ใช่ ASCII ได้ โดยไม่ต้องกลัวว่ารหัสนั้นจะไปทับกับรหัสควบคุมของ ASCII

3) การตรวจสอบโดยวิธีตรวจสอบผลบวก มีความสามารถตรวจสอบความผิดพลาดได้ดีกว่า ใช้พาริตี้ในอะซิงโครนัสในขณะที่บิตพาริตี้สามารถให้ประสิทธิผลได้ 95 เปอร์เซ็นต์ หากพบ

การผิดพลาดด้วยบิตพาริตี้ ไม่ทำให้เกิดการส่งใหม่เกิดขึ้น แต่ข้อผิดพลาดจากตรวจสอบผลบวกทำให้เกิดการส่งข้อมูลมาใหม่

2.3.2 โพรโทคอลไบซิงก์ (Bisync Protocol)

Bisync ย่อมาจาก Binary Synchronous Communication [11] เป็นผลผลิตของบรษัท IBM Bisync เป็นโพรโทคอลในระดับอักขระซึ่งหมายความว่าอักขระแต่ละตัวมีขอบเขตที่แน่นอน แต่ละอักขระไม่มีบิตเริ่มหรือบิตหยุดเหมือนกับอะซิงโครนัส การซิงโครนัสกระทำกันที่จุดเริ่มต้นของการส่งข้อมูลเลยทีเดียว สถานีส่งจะส่งสัญญาณที่เรียกว่าตัวอักษรนำ (Leading Pad Character) ไปยังสถานีรับก่อนที่จะเริ่มส่งข้อมูล ตัวอักษรนำ จะประกอบด้วย 0 และ 1 สลับกันเพื่อให้สถานีรับจัดสัญญาณนาฬิกาให้ตรงกัน นอกจากนั้นก่อนข้อมูลจะส่งออกมาจะต้องมีอักขระที่เรียกว่า SYN ตามหลัง PAD มาก่อน และสถานีส่งจำเป็นจะต้องบอกความยาวของข้อมูลมาในกลุ่มนี้และเครื่องหมายที่เป็นตัวบอกจุดเริ่มต้นของข้อมูลมาด้วยอักขระ SYN ในโพรโทคอล Bisync ทำหน้าที่คล้ายกับบิตเริ่มต้นในอะซิงโครนัสซึ่งทำหน้าที่ "ปลุก" สถานีรับให้ตื่นมารับข้อมูล ขณะที่สถานีรับกำลังรอรับสัญญาณจากสถานีส่ง เครื่องรับอยู่ในสถานภาพที่เรียกว่า "Hunt" บิตทุกบิตที่ผ่านเข้ามาจะถูกค้นหาอักขระ SYN ก่อน เมื่อได้รับอักขระ SYNแล้วจึงจะเริ่มนับบิตที่เข้ามาเพื่อจุดเริ่มต้นของสัญญาณ เพื่อป้องกันโอกาสที่จะเกิดความผิดพลาด อักขระ SYN จะส่งมา 2 ตัวก่อนที่เริ่มส่งข้อมูล อักขระควบคุมที่ใช้ในการส่งใน Bisync แสดงไว้ในตารางที่ 2.1 ไม่มีขอบเขตจำกัดของจำนวนกลุ่มที่จะส่งไปในการส่งแต่ละครั้ง กลุ่มของข้อมูลอาจจะมีส่วนหัวนำหน้าเพื่อบรรยายข้อมูลที่ส่งมา

ส่วนที่เป็นข้อมูลจริง ๆ ในสัญญาณ Bisync มีกฎเกณฑ์อย่างเดียวกับระบบอะซิงโครนัส นั่นคือ อักขระแต่ละตัวอาจจะใช้ 5 บิต 6 บิต 7 บิต หรือ 8 บิต และอาจจะตามด้วยบิตพาริตี้ ในเครื่องเมนเฟรม IBM รหัสที่ใช้แทนที่จะเป็น ASCII กลับเป็น EBCDIC ซึ่งรหัสควบคุมจะแตกต่างไปจาก ASCII ฉะนั้นในการรับส่งระหว่างเมนเฟรมคอมพิวเตอร์กับไมโครคอมพิวเตอร์จำเป็นต้องให้ไมโครคอมพิวเตอร์รับส่งเป็นแบบ EBCDIC เพื่อให้เข้ากับเมนเฟรมคอมพิวเตอร์ หน้าที่ของไมโครคอมพิวเตอร์จึงต้องเพิ่มขึ้นอีกอย่างหนึ่งคือ การแปลงรหัส EBCDIC และ ASCII

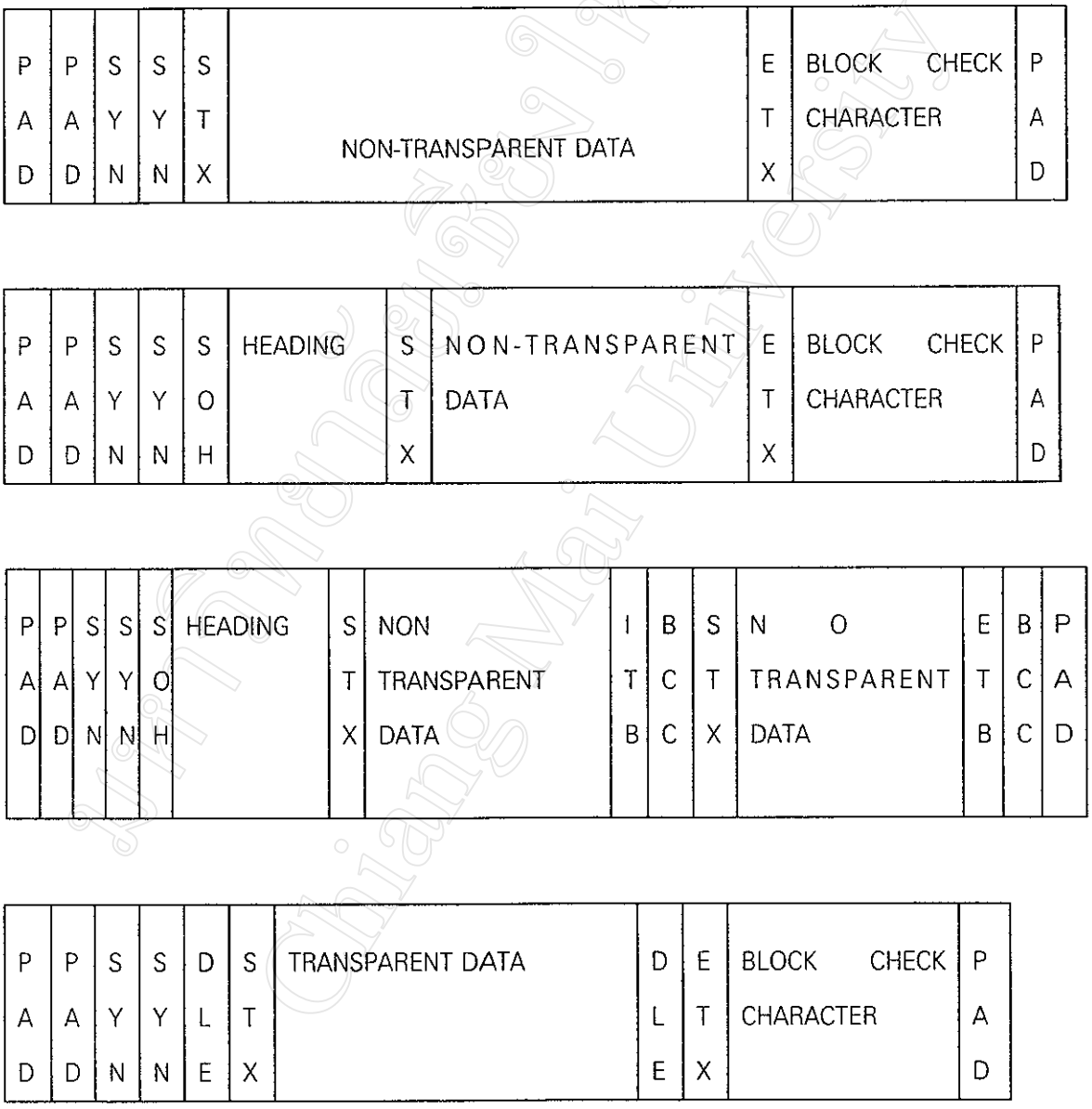
อักขระ	เลขฐาน 16 (Hex value)	เลขฐาน 10 (Decimal value)	อธิบาย
SYN	32	22	Synchronous Idle บอกการซิงโครนัส
PAD	55	85	PAD เริ่มต้นของ Frame
PAD	FF	255	PAD บอกท้าย Frame
DLE	10	16	Data Link Escape บอกว่าใช้อักขระที่ตามหลังมาในการควบคุม
ENQ	2D	5	Enquiry ขอให้ส่ง
SOH	01	1	Start of Heading:เริ่มส่วนหัว
STX	02	2	Start of Text:เริ่มต้นข้อความ
ITB	1F	15	End of Intermediate Block หมดสิ้นกลุ่มของข้อมูลระหว่างกลาง
ETB	26	23	End of Transmission Block สิ้นสุดการส่งของกลุ่ม
ETX	03	3	End of Text สิ้นสุดข้อความ

ตารางที่ 2.1 อักขระควบคุมในไบซิงค์

ไม่มีขอบเขตจำกัดของจำนวนกลุ่มที่จะส่งไปในการส่งแต่ละครั้ง กลุ่มของข้อมูลอาจจะมีส่วนหัวนำหน้าเพื่อบรรยายข้อมูลที่ส่งมา

ส่วนที่เป็นข้อมูลจริง ๆ ในสัญญาณ Bisync มีกฎเกณฑ์อย่างเดียวกับระบบอะซิงโครนัส นั่นคือ อักขระแต่ละตัวอาจจะใช้ 5 บิต 6 บิต 7 บิต หรือ 8 บิต และอาจจะตามด้วยบิตพาริตี้ ในเครื่องเมนเฟรม IBM รหัสที่ใช้แทนที่จะเป็น ASCII กลับเป็น EBCDIC ซึ่งรหัสควบคุมจะแตกต่างไป

จาก ASCII ฉะนั้นในการรับส่งระหว่างเมนเฟรมคอมพิวเตอร์กับไมโครคอมพิวเตอร์จำเป็นต้องให้ไมโครคอมพิวเตอร์รับส่งเป็นแบบ EBCDIC เพื่อให้เข้ากับเมนเฟรมคอมพิวเตอร์ หน้าที่ของไมโครคอมพิวเตอร์จึงต้องเพิ่มขึ้นอีกอย่างหนึ่งคือ การแปลงรหัส EBCDIC และ ASCII



รูปที่ 2.11 โครงสร้างข้อมูลของไบซิงค์ [11]

PAD	PAD	SYN	SYN	SOH	HEADING	DLE	STX	TRANS-PARENT DATA	DLE	SYN	TRANS-PARENT DATA	DLE	ETX	BLOCK CHECK CHARACTER

PAD	PAD	SYN	SYN	SOH	HEADING	DLE	STX	TRANS-PARENT DATA	DLE	ITB	BCC	DLE	STX	TRANS-PARENT DATA	DLE	ETB	BCC	PAD

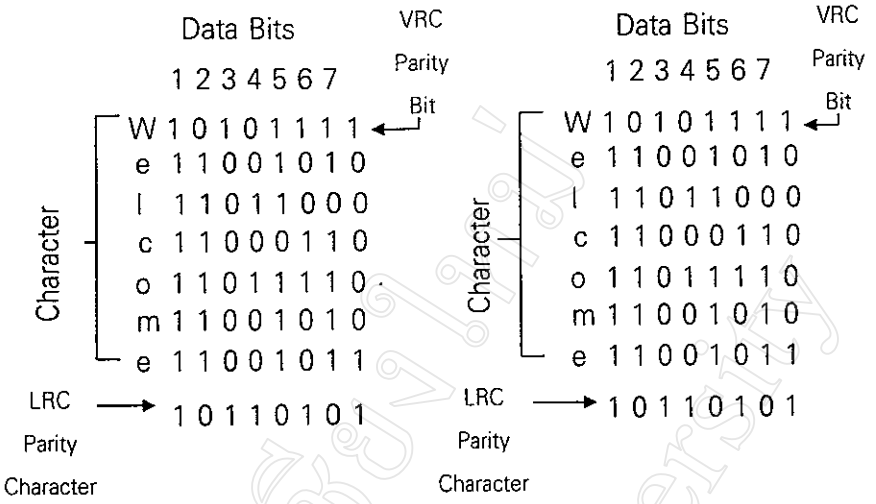
รูปที่ 2.11 โครงสร้างข้อมูลของไบซิงค์ (ต่อ)

2.3.3 โครงสร้างของข้อมูลไบซิงค์

แต่ละกลุ่มของข้อมูลที่ถูกส่งออกจะได้รับการตรวจสอบความถูกต้องที่ฝ่ายรับโดย Block Check Character (BCC) ซึ่งเป็นกลุ่มของตัวอักษรสำหรับการตรวจสอบความถูกต้อง โดยอาจจะมีวิธีในการตรวจสอบได้ 3 วิธีด้วยกันคือ

1) VRC หรือ Vertical Redundancy Checking เป็นวิธีการตรวจสอบข้อมูลในการส่งและรับข้อมูลโดยเพิ่มข้อมูล 1 บิต โดยข้อมูลจะเป็น คู่ (Odd) หรือ คี่ (Even) ก็ได้ในแต่ละการส่งข้อมูลทางด้านรับจะทำการตรวจสอบ ข้อมูลที่ได้รับและทำการตรวจสอบว่าถูกต้องหรือไม่ ดังในรูปที่ 2.11

2) LRC หรือ Longitudinal Redundancy Checking เป็นวิธีการตรวจสอบข้อมูลโดยทำการเพิ่มข้อมูลจำนวน 1 ไบต์ ต่อการส่งข้อมูล LRC ไบต์ที่จะส่ง ส่งตามมาตรฐาน ISO-1155 ซึ่งจะมีการส่งแบบ พาริตี้คู่ (Odd Parity) และ พาริตี้คี่ (Even Parity) บางครั้งเรียกการส่งแบบ LRC นี้ว่าเป็นการส่งแบบ Checksum คือตรวจสอบโดยการรวมค่าข้อมูลที่ทำการส่งก่อน แล้วจึงนำค่าที่รวมได้ส่งตามไปเป็นไบต์สุดท้ายของการส่งข้อมูลแต่ละครั้งดังในรูปที่ 2.11

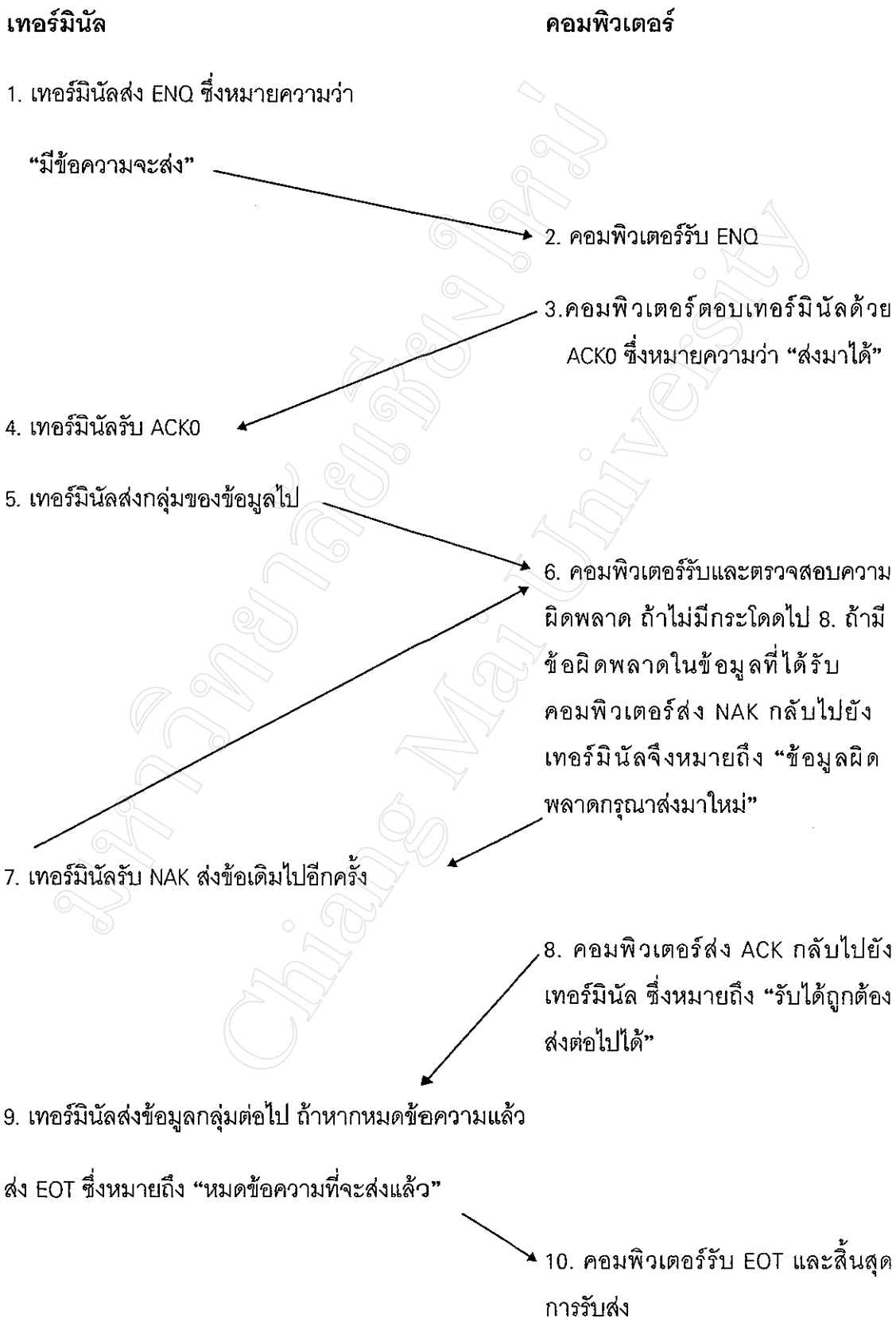


รูปที่ 2.12 วิธีการตรวจสอบ Vertical และ Longitudinal Redundacy [10]

3) CRC หรือ Cyclic Redundancy Checking เป็นวิธีการตรวจสอบโดยใช้วิธีของการคำนวนหาแบบโพลีโนเมียล (Polynomial) โดยจะตรวจสอบทุกบิตที่ทำการส่งข้อมูล เมื่อหาโดยหลักการของ CRC แล้วจะได้ค่าซึ่งเรียกว่า Error Checking ก็จะถูกส่งข้อมูลไปยังทางด้านรับโดยมีรูปแบบดังนี้

CRC TYPE	CRC Polynomial
CRC-12	$X^{12}+X^{11}+X^{3}+X^{2}+X+1$
CRC-16	$X^{16}+X^{12}+X^{2}+1$
CRC-CCITT	$X^{16}+X^{12}+X^{5}+1$
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+$ $X^{10}+X^{8}+X^{7}+X^{5}+X^{4}+X^{2}+X+1$

ตารางที่ 2.2 สมการทางคณิตศาสตร์สำหรับดิฟเฟอร์เรลเชียล CRC



รูปที่ 2.13 แบบของการแลกเปลี่ยนโพรโทคอลของไบซิงค์ [2]

2.3.4 โพรโทคอล SDLC และ HDLC

SDLC ย่อมาจาก Synchronous Data Link Control และ HDLC ย่อมาจาก High Level Data Link Control โพรโทคอลทั้งสองนี้ไม่เหมาะสำหรับมือสมัครเล่นเพราะราคาเริ่มต้นสูง แต่ทั้งสองนี้เป็นมาตรฐานโพรโทคอลที่ใช้ในงานธุรกิจสำหรับเครื่องระดับเมนเฟรม อย่างไรก็ตามยังมีความจำเป็นที่ไมโครคอมพิวเตอร์จะต้องใช้โพรโทคอลทั้งสอง จึงขอกล่าวถึงโพรโทคอลทั้งสองไว้ด้วยกัน ข้อแตกต่างที่เห็นชัดจะบอกไว้ในขณะที่กล่าว

ข้อมูลที่ส่งโดย SDLC/HDLC เรียกว่า Information field และฟิลด์ดังกล่าวก็คือ เลขฐานสองที่วิ่งต่อเนื่องกันมาแบบอนุกรม ขนาดของฟิลด์อาจจะมีตั้งแต่ 0 จนถึงค่าสูงสุดที่หน่วยความจำจะรับได้ สายของข้อมูลดังกล่าวจะอยู่ในรูปของบิตข้อมูล Bisync เป็นโพรโทคอลแบบเชิงตัวอักษร ซึ่งหมายความว่าไม่มีขอบเขตระหว่างอักขระ ถ้าหากข้อมูลเป็นอักขระฝ่ายรับจะต้องแยกออกเอาเองเมื่อรับข้อมูลทั้งหมดแล้ว

ความต่อเนื่องของข้อมูลใน SDLC/HDLC แตกต่างไปจาก Bisync การหยุดชั่วขณะของการส่งสามารถทำได้ใน Bisync แต่ SDLC/HDLC ไม่ยอมให้ทำเช่นนั้น ความต่อเนื่องของสายข้อมูลจะต้องต่อเนื่องไปจนสิ้นสุด Information Field ถ้าหากมีการหยุดหรือเบรกก่อนที่จะสิ้นสุด Information Field ถือว่ามีความผิดพลาดเกิดขึ้น

ใน SDLC/HDLC สถานีรับส่งแบ่งเป็น 2 ฝ่ายคือ ฝ่ายปฐมภูมิและฝ่ายทุติยภูมิ สถานีปฐมภูมิจะเป็นตัวควบคุมการเชื่อมต่อข้อมูล

SDLC format

Begining flag	Address	Control	Information	Frame check	Ending flag
01111110	8 bit	8 bit		16 bit	01111110
8 bit					8 bit

1frame

Begining flag	Address	Control	Packet heading	Information	frame check	Ending flag

HDLC format

รูปที่ 2.14 ฟอร์แมตของ SDLC และ HDLC [8]

แฟล็กเริ่มต้น (Beginning flag) เป็นตัวอ้างอิงถึงตำแหน่งเริ่มต้นสำหรับฟิลด์แอดเดรส (Address Field) และฟิลด์ควบคุม (Control Field) แฟล็กเริ่มต้นประกอบด้วย 0101110 ทำนองเดียวกัน แฟล็กท้ายขบวน (Ending Flag) เป็นตัวบอกว่า 16 บิตที่รับก่อนหน้านี้เป็นการตรวจสอบเฟรม ในขณะเดียวกันถ้าหากเฟรมที่ส่งต่อเนื่องกัน แฟล็กท้ายขบวน จะทำหน้าที่เป็นแฟล็กเริ่มต้นและแฟล็กท้ายขบวน จะเห็นว่าฮาร์ดแวร์ของโพรโทคอล SDLC/HDLC จะต้องคอยตรวจสอบรูปแบบของ 01111110 อยู่เสมอ เพราะถือว่าเป็นแฟล็ก เพื่อป้องกันความผิดพลาดในการตีความข้อมูลเป็นแฟล็ก SDLC/HDLC จะจัดการแทรกบิต "0" ลงในข้อมูลที่มี "1" ต่อเนื่องมากกว่า 5 บิต ฝ่ายรับจะต้องคอยแยก "0" ที่แทรกใส่เข้าไปเอาเองด้วยเหตุนี้เองฮาร์ดแวร์จึงมีราคาสูง

ฟิลด์แอดเดรส มีอยู่ 8 บิต เป็นการบอกสถานีทุติยภูมิว่าต้องการส่งให้ตัวไหน ฟิลด์ควบคุม ประกอบไปด้วย 8 บิต ซึ่งจะคอยแยกการส่งข้อความเป็น 3 รูปแบบใหญ่ ๆ คือ

1) ฟอร์แมตการโอนย้ายข้อมูล

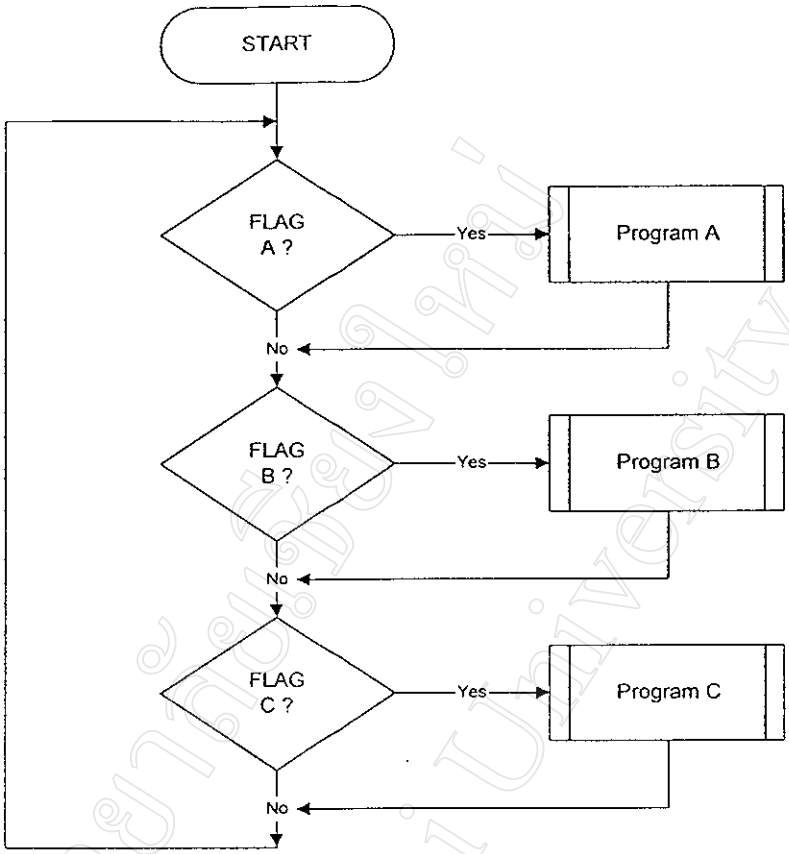
2) ฟอร์แมต Supervisory

3) ฟอร์แมต Non Sequence

รูปแบบที่ 1 ใช้ทั่วไปในการส่งข้อมูล ในจำนวน 8 บิตนี้จะมีจำนวนเฟรมที่กำลังส่งและจำนวนเฟรมที่ได้รับอยู่ด้วย

2.4 วิธีการเวียนตรวจสอบ (Polling)

วิธีการเวียนตรวจสอบเมื่อ CPU (Central Processing Unit) ต้องการติดต่อกับอุปกรณ์จะใช้วิธีการเวียนตรวจเช็คแฟล็กที่พอร์ตต่างๆ ที่ตำแหน่งของอุปกรณ์นั้นๆ โดยเริ่มต้นด้วยวิธีการอ่านสภาวะของแฟล็กทุกครั้งก่อน การส่งถ่ายข้อมูลจำเป็นต้องร่วมทำกับการตรวจสอบสเตตัสซึ่งกันและกันระหว่าง CPU กับอุปกรณ์ที่ CPU จะติดต่อโดยใช้วิธีการแฮนเชคกิ้ง (Handshaking) วิธีการเวียนตรวจสอบ โดยการทำงานจะเป็นลักษณะเวียนตรวจสอบสถานะแฟล็กที่เป็นอุปกรณ์ต่างๆ อยู่ตลอดเวลา เริ่มต้นด้วยการตรวจสอบของอุปกรณ์ A ถ้าแฟล็กไม่ถูกเซท จะผ่านมาตรวจสอบแฟล็กอุปกรณ์ B ซึ่งถ้าพร้อมก็จะเข้าไปทำงานในโปรแกรมจัดการอุปกรณ์ B และเมื่อทำงานเสร็จก็จะเคลียร์แฟล็กของ B และตรวจสอบต่อมาถึง C แล้วจึงกลับไปอุปกรณ์ A ใหม่ สำหรับโปรแกรมจัดการอุปกรณ์ A B และ C จะกำหนดให้เป็นอะไรขึ้นอยู่กับอุปกรณ์และความต้องการของผู้ใช้ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 วิธีการเวียนตรวจสอบ

2.5 การอินเตอร์รัพต์ (Interrupt)

การที่ CPU จะติดต่อกับอุปกรณ์ภายนอกได้นั้นมีด้วยกันหลายวิธี การอินเตอร์รัพต์ก็เป็นวิธีหนึ่งที่ใช้กันอย่างกว้างขวาง

การอินเตอร์รัพต์หมายถึง การขัดจังหวะการทำงาน CPU เพื่อให้ CPU หยุดการทำงานของโปรแกรมการทำงานที่กำลังทำอยู่ เมื่อได้รับสัญญาณอินเตอร์รัพต์จากอุปกรณ์ภายนอก CPU จะทำงานในคำสั่งที่กำลังทำงานอยู่จนจบคำสั่งนั้นเสียก่อนแล้ว CPU จึงไปบริการให้กับงานที่ส่งสัญญาณอินเตอร์รัพต์มา เมื่อบริการงานอินเตอร์รัพต์เสร็จแล้วก็จะกลับไปทำงานที่โปรแกรมเดิมต่อไป การอินเตอร์รัพต์ต่างกับการกระโดดภายในโปรแกรม ถ้าเป็นการกระโดดภายในโปรแกรมจะกระโดดเมื่อมีคำสั่ง Jump แต่การอินเตอร์รัพต์จะกระโดดไปทำงานในอีกโปรแกรมหนึ่ง เมื่อ CPU ได้รับสัญญาณจากอุปกรณ์ภายนอก

ข้อดีของการอินเตอร์รัพต์ คือ ไม่ต้องเสียเวลาในการรับส่งข้อมูลให้กับอุปกรณ์ภายนอกในขณะที่อุปกรณ์นั้นมีความเร็วต่ำ เป็นการเพิ่มประสิทธิภาพให้กับ CPU เพราะว่าในขณะที่ยังไม่มี

การอินเตอร์รัพต์ นั้น CPU สามารถทำงานให้กับโปรแกรมหลักได้โดยไม่ต้องเสียเวลา CPU สามารถควบคุมและติดต่อกับอุปกรณ์ ภายนอกหลายชนิดในเวลาเดียวกันได้ด้วยการอินเตอร์รัพต์ และสามารถจัดบริการตามลำดับความสำคัญก่อนหลังได้อีกด้วย

ข้อเสียการอินเตอร์รัพต์ คือ การเขียนโปรแกรมเพื่อบริการให้กับการอินเตอร์รัพต์นั้น จะต้องมีการเก็บค่าสเตตัส (Status) แฟล็ก (Flag) และแอดเดรส (Address) ของโปรแกรมเดิมไว้ในสแตคเมื่อเกิดการอินเตอร์รัพต์ขึ้น ถ้าไม่เก็บสิ่งเหล่านี้ไว้แล้วจะทำให้ CPU จะกลับมาทำงานต่อจากเดิมไม่ได้ นอกจากนั้นระบบการอินเตอร์รัพต์ยังต้องมีวงจรประกอบภายนอกซึ่งต้องเพิ่มเติมในการออกแบบสำหรับการอินเตอร์รัพต์

2.6 การเรียกขานและการตอบรับการเรียกขาน

การตอบรับการเรียกขานนี้เป็นวิธีการใหม่ที่เสนอในงานวิจัยนี้ ภายใต้ระบบนี้แม่ข่ายจะเรียกลูกข่ายทุกตัวตามลำดับในแต่ละรอบ โดยส่งรหัสประจำตัวออกไป ถ้าลูกข่ายตัวใดได้รับเรียกขานที่มีรหัสประจำตัวตรงกับของตัวเอง จะส่งข้อมูลตอบรับการเรียกขานกลับมาพร้อมทั้งรายงานสถานะการทำงานของตนเอง ในกรณีที่รหัสประจำตัวที่ได้รับไม่ตรงกับของลูกข่ายเองก็จะไม่ตอบ หรือในกรณีที่แม่ข่ายเรียกไปแล้วลูกข่ายไม่ตอบกลับมายังแม่ข่ายจะทำการเรียกลูกข่ายอีกครั้งหนึ่ง ถ้าลูกข่ายยังเฉยอยู่จะถือว่าลูกข่ายชะงักงัน แม่ข่ายจะทำการอินเตอร์รัพต์ลูกข่ายตัวนั้น ในกรณีฉุกเฉินแม่ข่ายสามารถทำการอินเตอร์รัพต์ลูกข่ายได้ตลอดเวลา หรือลูกข่ายก็สามารถร้องขออินเตอร์รัพต์แม่ข่ายได้ตลอดเวลา ซึ่งจะไม่ขึ้นกับการเวียนตรวจสอบของแม่ข่าย วิธีนี้จึงเป็นการนำข้อดีทั้งของระบบพอลลิ่งและระบบอินเตอร์รัพต์มาใช้ร่วมกัน ซึ่งนอกจากจะแก้ไขข้อบกพร่องของระบบอินเตอร์รัพต์ในเรื่องการชะงักงันของลูกข่ายและข้อบกพร่องของระบบพอลลิ่งในเรื่องการตอบสนองได้ทันต่อกรณีฉุกเฉินแล้ว ระบบใหม่ยังมีข้อดีเพิ่มเติมคือ สามารถนำไปใช้กับระบบการผลิตแบบยืดยุ่นอีกด้วย เช่น ในสายการผลิตที่ใช้เครื่องปักลายซึ่งสามารถเปลี่ยนลายได้โดยการเปลี่ยนโปรแกรมในการปักลายเท่านั้น สมมติมีเครื่องจักรอยู่ 10 เครื่อง ทำการปักลายต่างกัน 3 แบบ ตามจำนวนมากน้อยที่ได้รับการสั่งซื้อ ถ้าเกิดมีการยกเลิกการสั่งซื้อแบบ ก และเพิ่มการสั่งซื้อแบบ ข แทน ระบบควบคุมแบบใหม่ นี้จะสามารถสั่งงานโดยวิธีการอินเตอร์รัพต์ไปยังเครื่องที่กำลังปักลายแบบ ก ว่าเมื่อจบขั้นตอนการของงานที่ทำอยู่ให้หยุด คอมพิวเตอร์แม่ข่ายก็จะส่งโปรแกรมปักลายแบบ ข ไปลงที่หน่วยความจำของไมโครคอนโทรล เลอร์ลูกข่ายนั้น เพื่อปรับเปลี่ยนการทำงานเป็นแบบ ข ได้อย่างรวดเร็ว สายการผลิตนี้จึงมีความยืดหยุ่นสูง สามารถตอบสนองการปรับเปลี่ยนได้รวดเร็วมากตามหลักการของระบบโรงงานแบบใหม่ ซึ่งเป็นที่ต้องการในระบบอุตสาหกรรมสมัยใหม่

2.7 มาตรฐานการสื่อสารข้อมูล RS-485

RS-485 เป็นมาตราฐาน EIA (The Electronics Industries Association :EIA) [14] การสื่อสารข้อมูลแบบสมดุล ที่ได้พัฒนามาจากมาตรฐาน RS-422-A เพื่อให้ตัวรับและตัวส่งจำนวนมากคู่สามารถใช้สายสัญญาณรับส่งร่วมกันได้ ซึ่งกรณีของ RS-422-A คู่สายสัญญาณรับส่งหนึ่งคู่ จะมีตัวรับได้ไม่เกิน 10 ชุดและมีตัวส่ง 1 ชุด

Feature	RS-232	RS-423	RS-422	RS-485
Mode	Single-ended	Single-ended	Differential	Differential
No. drivers and receivers	1 Driver 1 Receiver	1 Driver 10 Receivers	1 Driver 10 Receivers	32 Driver 32 Receivers
Max. distance (ft)	50	4000	4000	4000
Max. data rate (bits/s)	20k	100k	10M	10M
Single levels (V)	3 to 25 -3 to -25	3.6 to 6 -3.6 to -6	2 to 6 -2 to -6	1.5 to 6 -1.5 to -6
Receiver decision point (V)	+3 -3	0.2 -0.2	0.2 -0.2	0.2 -0.2

ตารางที่ 2.3 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลของ EIA

สำหรับ RS-485 สามารถใช้ตัวรับ 32 ชุด และตัวส่ง 32 ชุด ร่วมกันได้ภายในคู่สายสัญญาณหนึ่งคู่ (multi-point interface) โดยทั่วไป RS-485 คุณลักษณะทางไฟฟ้าของตัวรับและตัวส่ง มีระดับสัญญาณ -1.5V ถึง -6V และ +1.5V ถึง +6V ตามลำดับ และไม่จำกัดรูปแบบโพรโทคอลที่จะใช้ในการพัฒนา นอกจากนี้ยังเป็นระบบที่มีความเร็วในการส่งข้อมูลสูงส่งได้ไกล อีกทั้งอุปกรณ์มีราคาไม่แพง จึงทำให้ RS-485 ถูกนำมาพัฒนาใช้งานในระบบการสื่อสารข้อมูลอนุกรมแบบเครือข่ายอย่างแพร่หลาย ตาราง 2.3 แสดงข้อมูลเปรียบเทียบระหว่าง RS-485 กับระบบอื่นๆ