บทที่ 4

ระบบการอินเตอร์เฟสมาตรฐาน (IEEE-488) และ NI-488

4.1 กล่าวนำ

ในปี ค.ศ.1970 บริษัท Hewlett Packard ได้แนะนำผลิตภัณฑ์ที่ชื่อว่า Hewlett-Packard Interface Bus (HP-IB) ซึ่ง HP-IB นี้ เป็นตัวเชื่อมต่อที่ใช้เชื่อมต่อระหว่างเครื่องมือสองชุดโดยผ่านสายสัญญาณบัสมาตรฐาน ซึ่งประกอบขึ้นจากสายไฟเป็นจำนวนมาก ต่อมาในปี ค.ศ.1975 HP-IB มาตรฐาน ได้ถูกปรับปรุงโดยคณะกรรมการ IEEE (Institute of Electrical and Electronic Engineers) เรียกว่า IEEE Standard 488-1975 (สุนันต์ แสงโพธิ์ และคณะ, ม.ป.ป.)

ในปี ค.ศ.1978 ได้มีการปรับปรุง IEEE Standard 488-1975 ให้มีการรับ-ส่งข้อมูลดิจิตอลระหว่างเครื่องมือและอุปกรณ์ต่าง ๆ ในลักษณะของบิตขนาน ไบต์อนุกรม (byte-serial bit-perallel) ระบบการอินเตอร์เฟสติดต่อเชื่อมโยงกันทางสายในรูปแบบของบัส (contiguous party-line bus system) เรียกว่า IEEE Standard 488-1978, Digital Interface for Programmable Instrumentation และ ANSI (American National Standards Institute) ได้ยอมรับให้เป็นมาตรฐานในปี ค.ศ.1979 ดังนั้นจึงเรียกมาตรฐานนี้ว่า ANSI/IEEE Standard 488-1978 นอกจากนี้ยังเป็นมาตรฐานของ Internation Electrotechnical Commission คือ IEC Std 625-1

คำว่า "programmable" หมายความว่า ความสามารถของเครื่องมือที่โปรแกรมได้โดยสัญญาณดิจิตอล แต่ระบบอินเตอร์เฟสในปัจจุบัน สามารถประยุกต์ใช้กับเครื่องมือที่โปรแกรมได้และโปรแกรมไม่ได้ เช่น เครื่องคิดเลข เครื่องวาดภาพ (plotter) ดิสต์ คอมพิวเตอร์ และอื่น ๆ (Allocca, 1983)

วัตถุประสงค์ของการอินเตอร์เฟส ก็เพื่อให้อุปกรณ์ต่าง ๆ ที่ต่อเชื่อมกัน สามารถติดต่อสื่อสาร หรือส่งข่าวสาร (message) กันได้ โดยไม่มีการก่อกวนซึ่งกันและกัน และไม่กำกวมข่าวสาร แบ่งออกได้เป็น 2 ชนิด คือ ข่าวสารที่ใช้ในการจัดการ หรือควบคุมการอินเตอร์เฟส

เรียกว่า interface messages และข่าวสารที่ส่งไปให้อุปกรณ์ต่าง ๆ ซึ่งเป็นข่าวสารที่ไม่ได้ใช้ในการจัดการในระบบอินเตอร์เฟส เรียกว่า device dependent messages

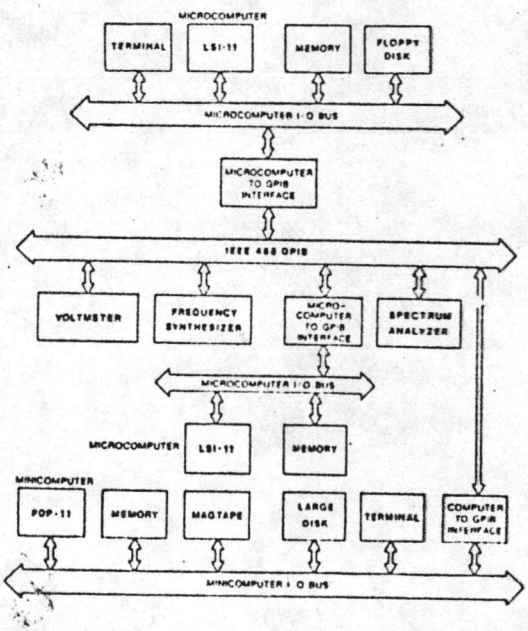
สายสัญญาณบัสที่ใช้เชื่อมต่อระหว่างอุปกรณ์ต่าง ๆ คือ IEEE-488 bus และมีชื่อเรียกทั่วไปว่า General-Purpose Interface Bus (GPIB) มีการกำหนดกลไก (mechanical) กระแสไฟ (electrical) และฟังก์ชันมาตรฐานสำหรับการอินเตอร์เฟส ดังนี้

1) ข้อมูลต้องเป็นดิจิตอลเท่านั้น

2) จำนวนอุปกรณ์ (device) ที่จะต่อกันในแต่ละบัส ต้องไม่เกิน 15 อุปกรณ์

3) ความยาวของสายเคเบิลทั้งหมดไม่เกิน 20 เมตร

4) ความเร็วของการส่งข้อมูลไม่เกิน 1 เมกะบิตต่อวินาที (IEEE, 1978)



รูป 4.1 ตัวอย่างแสดงการอินเตอร์เฟสระหว่างคอมพิวเตอร์และอุปกรณ์ต่าง ๆ

4.2 อุปกรณ์ที่ต่อกับ IEEE-488 bus

อุปกรณ์ที่ต่อกับ IEEE-488 bus สามารถแบ่งตามหน้าที่พื้นฐานได้ 3 ประเภท คือ

1) **ผู้ควบคุม (controller)** คืออุปกรณ์ที่ทำหน้าที่ควบคุมอุปกรณ์อื่น ๆ ในระบบให้ฟังหรือพูด โดยการส่งข่าวสารที่ใช้ในการจัดการ เพื่อสั่งให้อุปกรณ์เหล่านั้นปฏิบัติงานต่าง ๆ ได้

2) **ผู้พูด** (talker) คืออุปกรณ์ที่มีความสามารถส่งข้อมูลข่าวสารไปยังอุปกรณ์อื่น ที่อยู่ใน IEEE-488 bus เดียวกัน

3) **ผู้ฟัง** (listener) คือ อุปกรณ์ที่สามารถรับข้อมูลข่าวสาร จากอุปกรณ์อื่นที่ต่อเชื่อมกันใน IEEE-488 bus (IEEE, 1978)

ผู้ควบคุมจะทำหน้าที่เปรียบเสมือนพนักงานรับสายโทรศัพท์ ในการที่จะกำหนดเส้นทางรับข้อมูลจากผู้พูด และส่งข้อมูลไปยังผู้ฟัง อุปกรณ์ที่ทำหน้าที่ผู้ควบคุมส่วนใหญ่จะใช้เครื่องคอมพิวเตอร์ ทำให้สามารถที่จะเป็นทั้งผู้ควบคุม ผู้พูด และผู้ฟังพร้อมกัน

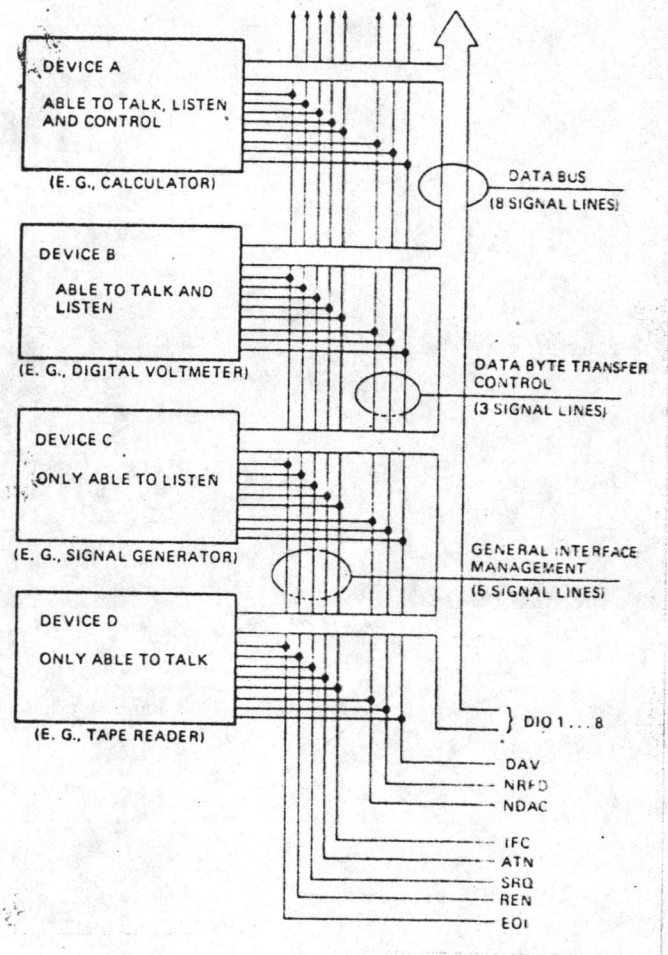
ผู้พูดซึ่งสามารถแทนได้โดยอุปกรณ์ "พูดอย่างเดียว" สามารถส่งข้อมูลผ่านบัสแต่มันไม่สามารถรับข้อมูลหรือว่าคำสั่งได้ ตัวอย่างทั่ว ๆ ไปของอุปกรณ์ "พูดอย่างเดียว" คือโวลต์มิเตอร์ ซึ่งจะส่งแค่ผลของการวัดออกไปเท่านั้น โดยทั่วไปแล้ว ระบบตรวจสอบเครื่องวัดทางไฟฟ้าที่ใช้ IEEE-488 bus ไม่จำเป็นต้องมีผู้พูดก็ได้เนื่องจากผู้ควบคุมไม่สามารถปิดหรือเปิดผู้พูดได้ อุปกรณ์เหล่านี้ ต้องถูกกระตุ้นโดยแถวของสัญญาณทริกที่แยกออกจากกัน (separate trigger line) หรือปุ่มที่อยู่บนแผงหน้าปัด อย่างไรก็ตาม ถ้ามีอุปกรณ์แบบ "พูดอย่างเดียว" มากกว่า 1 อุปกรณ์ขึ้นไป ที่ต่อกับ IEEE-488 bus อุปกรณ์แบบพูดอย่างเดียวจะใช้งานบัสเสียผู้เดียว และก็อาจจะแทรกแซงคำสั่งที่ออกมาจากตัวควบคุม และ/หรือข้อมูลบนบัสที่มาจากอุปกรณ์อื่นได้อีกด้วย

อุปกรณ์ที่เป็นผู้ฟังหรืออุปกรณ์ "ฟังอย่างเดียว" สามารถรับข้อมูลที่ส่งมาตามบัสได้เท่านั้น ไม่สามารถส่งข้อมูลได้ ตัวอย่างของอุปกรณ์แบบฟังอย่างเดียว คือเครื่องกำเนิดสัญญาณ ซึ่งทำหน้าที่รับคำสั่งหรือข้อมูลจากบัสเพียงอย่างเดียว เพื่อส่งสัญญาณความถี่ต่าง ๆ ออกมา

นอกจากอุปกรณ์ "พูดอย่างเดียว" และ "ฟังอย่างเดียว" แล้ว ยังมีอุปกรณ์ที่มีคุณสมบัติในการพูดและฟังได้ภายในเครื่องเดียวกัน ซึ่งสามารถทั้งรับและส่งข้อมูลในบัสได้ ทำให้การโปรแกรม การตรวจสอบหาจุดบกพร่อง และการบำรุงซ่อมแซมทำได้ง่ายยิ่งขึ้น เครื่องวัดทางไฟฟ้าในปัจจุบัน ส่วนใหญ่จะมีคุณสมบัติในการพูดและฟังทั้งสิ้น เพื่อการประยุกต์ใช้งานต่าง ๆ ต้องการควบคุมที่เคร่งครัดยิ่งกว่าเดิม เครื่องวัดทางไฟฟ้าเหล่านี้ สามารถทำงานแบบ "พูดอย่างเดียว" หรือ "ฟังอย่างเดียว" ก็ได้ โดยใช้เพียงคำสั่งทางซอฟต์แวร์ หรือสวิตช์ทางฮาร์ดแวร์อย่างใดอย่างหนึ่งเท่านั้น

จากความสามารถในการต่อเชื่อมของเครื่องวัดทางไฟฟ้า และคุณสมบัติต่าง ๆ ของ IEEE-488 ทำให้การตรวจสอบเครื่องวัดทางไฟฟ้าทำได้โดยอัตโนมัติ ทำการวัดได้หลายอย่างในเวลาเดียวกัน และสามารถนำข้อมูลมาทำการเก็บ พิมพ์ และคำนวณค่าต่าง ๆ ได้ตามต้องการ

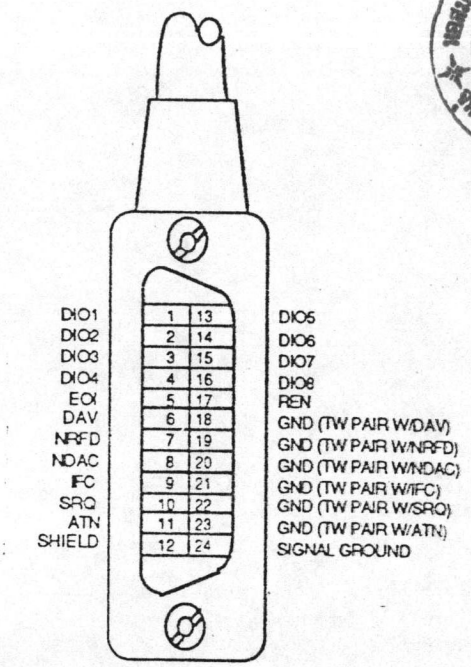
4.3 สายสัญญาณของ IEEE-488 bus



รูป 4.2 โครงสร้างของบัส และการอินเตอร์เฟส

บัสที่ใช้ต่อเชื่อมระหว่างอุปกรณ์ต่าง ๆ มีทั้งหมด 24 สาย ประกอบด้วยสายสัญญาณ 16 สาย สายดิน 1 สาย สายชิลด์ 1 สาย และคู่สายสำหรับสายสัญญาณอีก 6 สาย (ดังรูป 4.3) สายสัญญาณทั้ง 16 สาย ใช้สำหรับรับ-ส่งข้อมูล ข่าวสารของการอินเตอร์เฟส และข่าวสารของอุปกรณ์ ไปยังอุปกรณ์ต่าง ๆ ภายในระบบ โดยแบ่งออกเป็น 3 ส่วน คือ

1) บัสข้อมูล (data bus) 8 สาย ได้แก่ DIO1 - DIO8 ใช้สำหรับรับส่งข่าวสารสำหรับการอินเตอร์เฟส และข่าวสารของอุปกรณ์



รูป 4.3 แสดงโครงสร้างของสายเคเบิลของ IEEE-488 bus

2) **สายสัญญาณควบคุมการส่งข้อมูล (data byte transfer control bus)** 3 สาย ใช้ในการควบคุมข้อมูลจากผู้พูดไปยังผู้ฟัง หรือจากผู้ควบคุมไปยังอุปกรณ์ต่าง ๆ ประกอบด้วย DAV (data valid) ใช้แสดงความถูกต้องของข้อมูลภายในบัสข้อมูล NRFD (not ready for data) แสดงถึงความไม่พร้อมที่จะรับข้อมูลอุปกรณ์ และ NDAC (not data accepted) บ่งบอกถึงการที่ยังไม่ได้รับข้อมูลของผู้ฟัง

3) **สายสัญญาณจัดการอินเตอร์เฟส (general interface management bus)** 5 สาย ประกอบด้วย

(1) ATN (attention) ใช้แสดงว่า ข้อมูลนั้นเป็นของผู้ควบคุมที่ซึ่งอุปกรณ์ต่าง ๆ ต้องตอบสนองข้อมูลนั้น ถ้า ATN = 1 แสดงว่าผู้ควบคุมส่งข่าวสารการอินเตอร์เฟสไปยังอุปกรณ์ แต่ถ้า ATN = 0 หมายถึง ผู้พูดส่งข่าวสารไปยังผู้ฟัง

(2) IFC (interface clear) ผู้ควบคุมระบบสั่งให้อุปกรณ์ต่าง ๆ อยู่ในสถานะเริ่มต้น

(3) REN (remote enable) เป็นการบอกให้อุปกรณ์รับการควบคุมโดยผู้ควบคุม และไม่ให้รับข้อมูลจากแผงหน้าปัด (โดยมนุษย์)

(4) SRQ (sevice request) ถูกใช้โดยอุปกรณ์เพื่อขอส่งข้อมูลหรือขัดจังหวะเหตุการณ์ที่กำลังดำเนินการอยู่

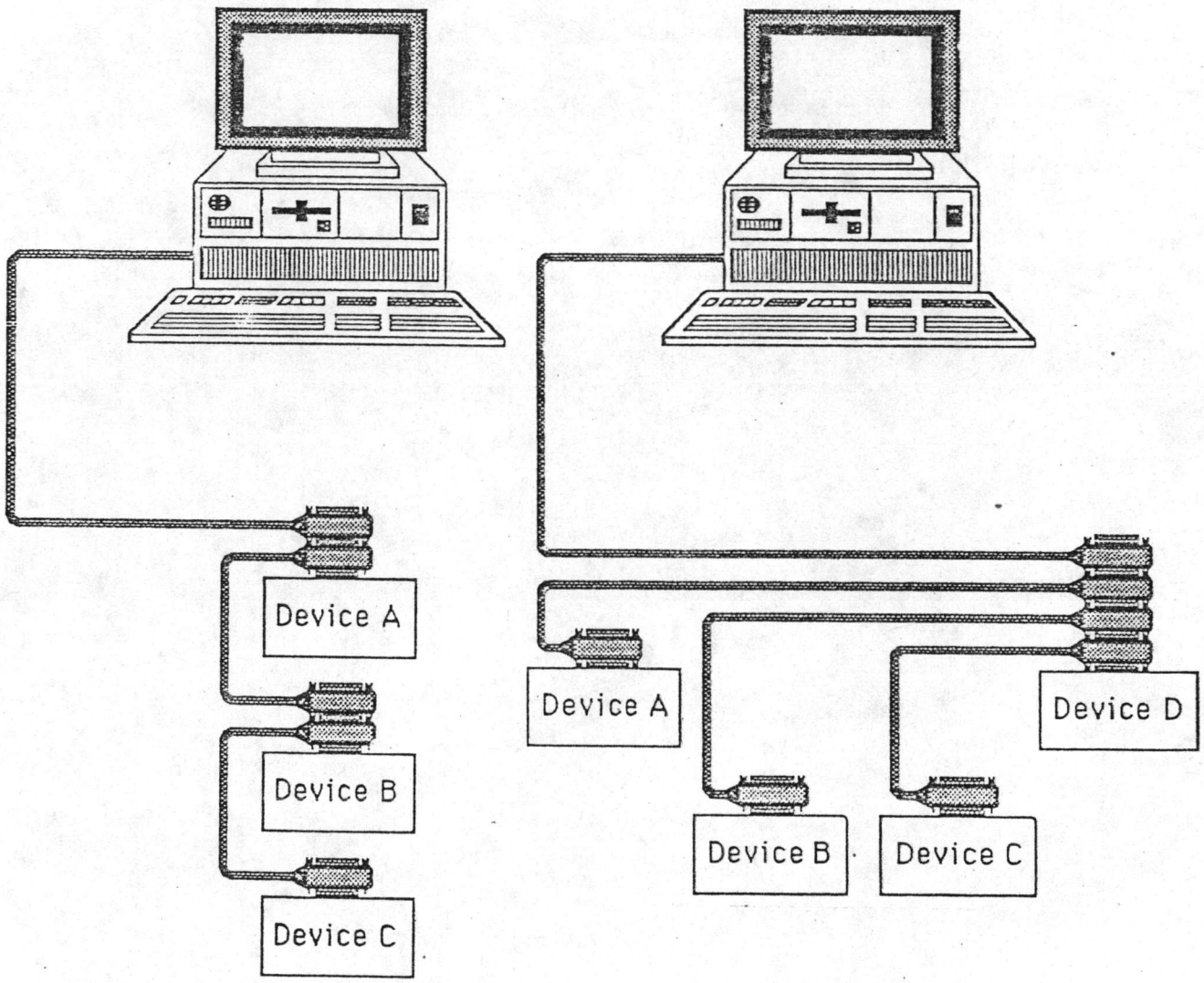
(5) EOI (end of identify) กำหนดโดยผู้พูด เพื่อแสดงถึงข้อมูลหมดแล้ว

ซึ่งผู้ควบคุมก็จะหยุดการทำงานนั้น (IEEE, 1978) (Hall, 1992)

จากความยาวของสายเคเบิลถูกจำกัดให้มีความยาวได้สูงสุด 20 เมตร โดยขึ้นอยู่กับขีดจำกัดทางด้านเสียง แต่สามารถยืดความยาวออกได้ โดยใช้อุปกรณ์ที่เรียกว่า IEEE-488 Extend ทำให้ความยาวของสายเคเบิลนั้นสามารถเพิ่มขึ้นได้เป็นหลายร้อยเมตร และสามารถที่จะต่ออุปกรณ์เพิ่มมากขึ้นได้ ตัวอย่างของ IEEE-488 Extend คือ GPIB-100 ของบริษัทเนชั่นแนลอินสทรูเมนต์ (National Instrument)

IEEE-488 กำหนดให้แต่ละบัสสามารถต่ออุปกรณ์ได้สูงสุด 15 อุปกรณ์ ซึ่งอุปกรณ์นั้นอาจเป็นคอมพิวเตอร์หรือเครื่องมือวัดต่าง ๆ อย่างไรก็ตามในระบบอาจมีการใช้บัสมากกว่า 1 บัสโดยข้อมูลจะถูกถ่ายโอนจากบัสหนึ่งไปยังบัสอื่น ๆ โดยซอฟต์แวร์ระดับผู้ควบคุม (controller level)

ลักษณะการต่อสายสัญญาณบัสระหว่างเครื่องคอมพิวเตอร์และอุปกรณ์ต่าง ๆ สามารถต่อแบบเส้นตรง (Linear) หรือต่อแบบสตาร์ (Star) ก็ได้ ดังรูป 4.4



รูป 4.4 ลักษณะการต่อสายสัญญาณบัส

4.4 การอินเตอร์เฟสกับ IEEE-488 bus

เมื่อเริ่มต้นระบบ ผู้ควบคุมจะส่งสัญญาณ IFC ไปยังทุกอุปกรณ์ที่ต่อเชื่อมอยู่กับบัส หลังจากนั้นก็จะปฏิบัติงานต่าง ๆ ตามลำดับโปรแกรม โดยผู้ควบคุมจะออกจากการควบคุม โดยกำหนด

สัญญาณ ATN เป็นศูนย์ ในรูป 4.5 แสดงรูปแบบของคำสั่ง จะเห็นว่าบิตที่ 8 ไม่ได้ถูกกำหนดไว้ บิตที่ 7 บ่งบอกว่าเป็นคำสั่งกำหนดที่อยู่ของผู้พูด ส่วนบิตที่ 6 เป็นคำสั่งกำหนดที่อยู่ของผู้ฟัง และบิตที่ 1-5 เป็นรหัสที่อยู่ของผู้พูดหรือผู้ฟัง ขึ้นอยู่กับบิตที่ 6 และ 7 เช่นต้องการให้อุปกรณ์หมายเลข 4 เป็นผู้พูด ผู้ควบคุมก็จะเปลี่ยนสัญญาณ ATN ให้เป็นศูนย์ก่อนแล้วส่งรหัส X1000100 ไปยังบัสข้อมูล และส่งรหัส $X01A_5A_4A_3A_2A_1$ ไปให้ผู้ฟัง โดยบิตที่ 1-5 เป็นรหัสที่อยู่ของผู้ฟัง เมื่อ ส่งข้อมูลเสร็จแล้ว ผู้ควบคุมจะส่งรหัส X0111111 ไปให้แก่ผู้ฟังเพื่อให้หยุดฟัง (unlisten) และส่งรหัส X1011111 ไปให้ผู้พูดเพื่อให้หยุดพูด (untalk) ส่วนคำสั่งทั่ว ๆ ไปที่ส่งโดยผู้ควบคุมไปยังผู้พูดและผู้ฟัง โดยกำหนดให้บิตที่ 5-7 มีค่าเป็นศูนย์ และบิตที่ 1-4 จะเป็นรหัสของคำสั่งต่าง ๆ (Hall, 1992)

D8	D7	D6	D5	D4	D3	D2	D1	MEANING
X	0	0	0	B4	B3	B2	B1	UNIVERSAL COMMANDS
X	0	1	A5	A4	A3	A2	A1	LISTEN ADDRESSES
X	0	1	1	1	1	1	1	UNLISTEN COMMAND
X	1	0	A5	A4	A3	A2	A1	TALK ADDRESSES
X	1	0	1	1	1	1	1	UNTALK COMMAND
X	1	1	A5	A4	A3	A2	A1	SECONDARY COMMANDS
X	1	1	1	1	1	1	1	IGNORED

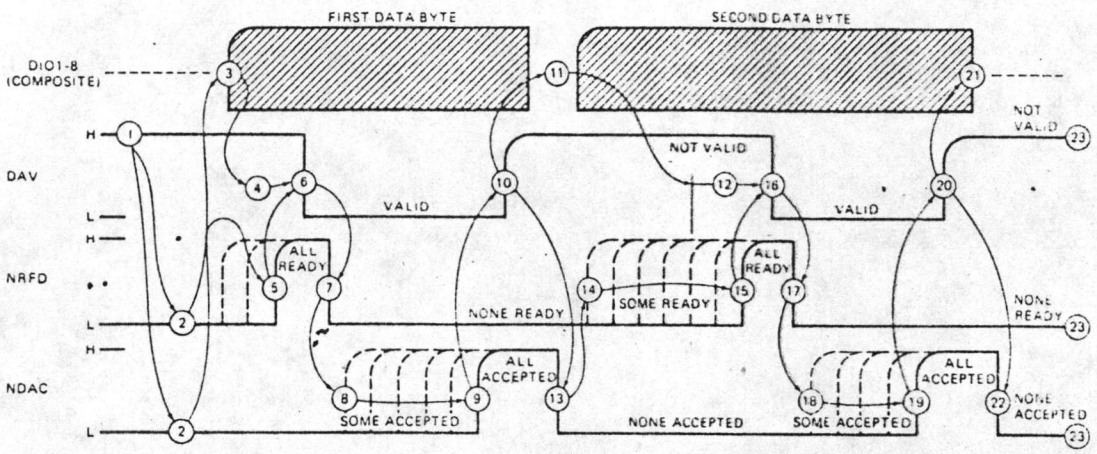
CODE FOR TYPE OF COMMAND

NOTES: These codes are only valid when ATN is low. Address 11111 cannot be used for a listener or a talker

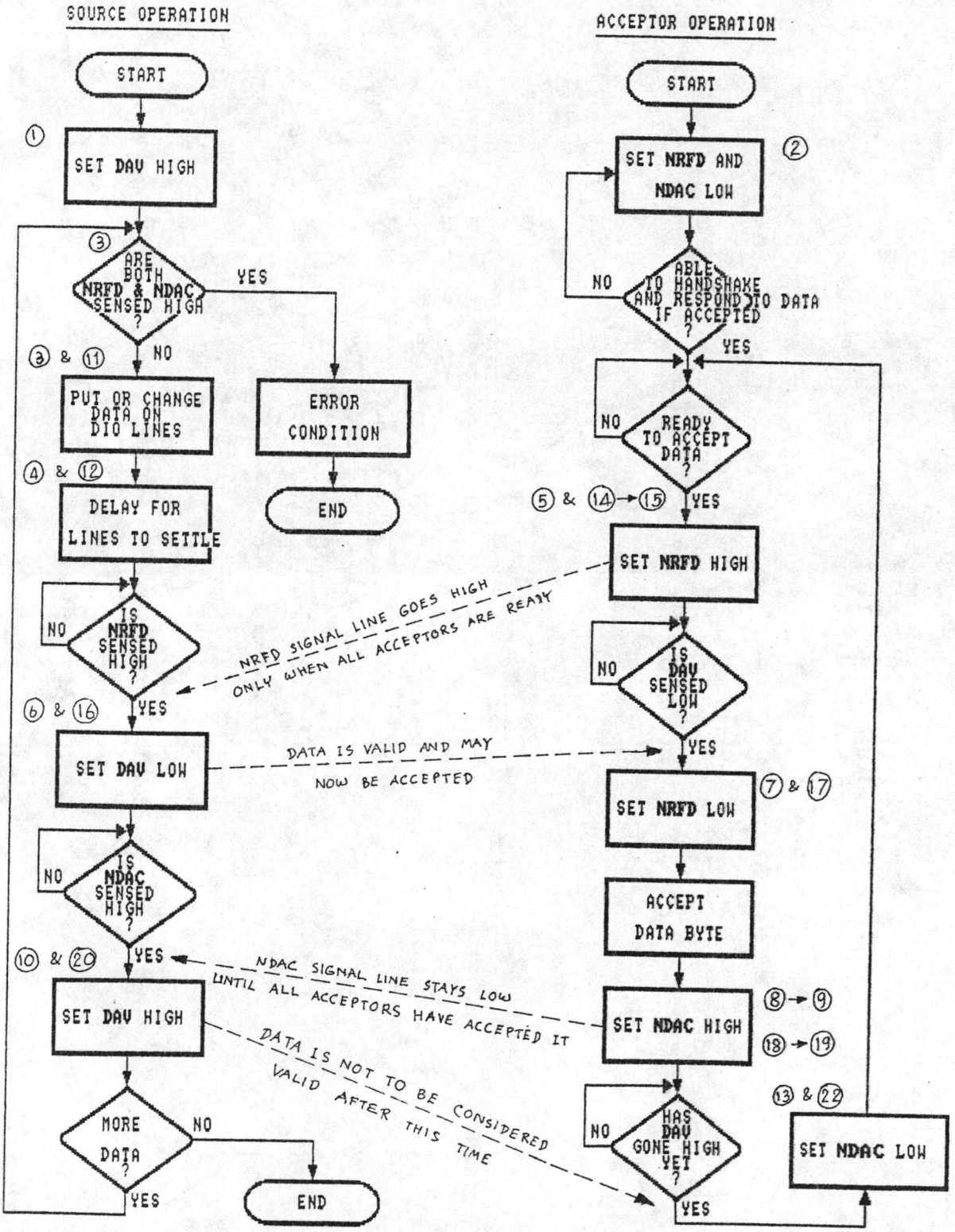
รูป 4.5 รูปแบบของคำสั่ง

4.5 การจับมือ (handshake)

การเคลื่อนย้ายข้อมูลในระบบอินเตอร์เฟส จะใช้วิธีการจับมือระหว่างผู้ส่ง (ผู้พูด) และผู้รับ (ผู้ฟัง) ในรูป 4.6 แสดงรูปคลื่นสัญญาณที่เกิดขึ้นภายในสายสัญญาณ DAV NRFD และ NDAC จากกระบวนการจับมือ เพื่อจะส่งข้อมูลจากผู้ส่งไปยังผู้รับหลาย ๆ ผู้รับในเวลาที่ต่างกัน ขึ้นอยู่กับความยาวของสาย และความเร็วในการรับข้อมูลของแต่ละผู้รับ ลำดับขั้นตอนการจับมือระหว่างผู้ส่งและผู้รับเป็นไปตามแผนผังในรูป 4.7



รูป 4.6 แสดงรูปคลื่นสัญญาณที่เกิดขึ้นภายในสายสัญญาณ DAV, NRFD และ NDAC ในการส่งข้อมูลจากผู้ส่งไปยังผู้รับ



รูป 4.7 แผนผังการส่งข้อมูลจากผู้ส่งไปยังผู้รับโดยขบวนการจับมือ

4.6 ฟังก์ชันการอินเตอร์เฟสที่กำหนดโดยมาตรฐาน IEEE standard 488-1978

SH (source handshake) และ AH (acceptor handshake) เป็นฟังก์ชันในการรับ-ส่งข่าวสารต่าง ๆ ภายในบัส ข่าวสารต่าง ๆ นี้ อาจจะเป็นข้อมูลของอุปกรณ์ หรือเป็นคำสั่งในการอินเตอร์เฟสขึ้นอยู่กับสายสัญญาณ ATN (attention) ถ้า ATN=1 บัสข้อมูล นั้นเป็นคำสั่งในการอินเตอร์เฟส แต่ถ้าเป็น 0 บัสข้อมูลนั้นจะเป็นข้อมูลของอุปกรณ์

T (talker) และ TE (extended talker) เป็นฟังก์ชันที่ทำให้อุปกรณ์สามารถส่งข้อมูลเข้าสู่บัส ฟังก์ชัน T จะใช้ 1 ไบต์ ในการกำหนดที่อยู่ (address) แต่ฟังก์ชัน TE จะใช้ 2 ไบต์ ในการกำหนดที่อยู่

L (listener) และ LE (extended listener) เป็นฟังก์ชันที่ทำให้อุปกรณ์สามารถรับข้อมูลจากอุปกรณ์อื่น

SR (service request) เป็นฟังก์ชันที่อุปกรณ์แจ้งความต้องการใช้บริการให้ผู้ควบคุมทราบ

RL (remote local) เป็นฟังก์ชันที่บอกให้อุปกรณ์สามารถรับข้อมูลจากแผงหน้าปัดเครื่อง หรือให้รับข้อมูลจากบัส

PP (parallel poll) เป็นฟังก์ชันที่ให้อุปกรณ์ที่มีความสามารถในการส่งสัญญาณ PPR (parallel poll response) ให้แก่ผู้ควบคุม โดยที่ไม่มีการอ้างที่อยู่การทำ parallel poll อุปกรณ์ต่าง ๆ จะส่งสถานภาพของอุปกรณ์ทางสายสัญญาณ DIO1-DIO8 ซึ่งผู้ควบคุมจะกำหนดให้สายสัญญาณ DIO 1 สาย ต่ออุปกรณ์ 1 ชิ้น นอกจากนี้ PP ยังสามารถทราบถึงความต้องการใช้บริการของอุปกรณ์ โดยต่างจากการขอใช้บริการตามปกติ คือผู้ควบคุมเป็นผู้เริ่มต้นทำ parallel poll แต่ถ้าเป็นอุปกรณ์จะเริ่มต้นโดยการทำ serial poll และ parallel poll สามารถที่จะส่งสถานภาพของอุปกรณ์พร้อม ๆ กันได้หลายอุปกรณ์ แต่ serial poll จะเลือกเฉพาะอุปกรณ์ใดอุปกรณ์หนึ่งเท่านั้น

DC (device clear) เป็นฟังก์ชันที่ให้อุปกรณ์ทำการล้าง (clear) สถานภาพที่เป็นอยู่

DT (device trigger) เป็นฟังก์ชันให้อุปกรณ์เริ่มต้นปฏิบัติงาน

C (controller) เป็นฟังก์ชันใช้ส่งที่อยู่ของอุปกรณ์ คำสั่งทั่ว ๆ ไป และการกำหนดที่อยู่ให้กับอุปกรณ์ ฟังก์ชันในการควบคุมสามารถปฏิบัติงานได้โดยส่งรหัส ATN เข้าสู่ IEEE-488

bus กรณีที่มีอุปกรณ์หลาย ๆ อุปกรณ์ที่มีฟังก์ชันในการควบคุม (C) โดยแต่ละอุปกรณ์จะอยู่ในสถานะ CIDS (controller idle state) แต่จะมีอุปกรณ์หนึ่งที่ไม่อยู่ในสถานะ CIDS คืออยู่ในสถานะ SACS (system control active state) เรียกว่า controller-in-charge ซึ่งจะเป็นผู้ส่ง IFC และ REN ไปยัง IEEE-488 bus เรียกอุปกรณ์นี้ว่าผู้ควบคุมระบบ (system controller) (IEEE, 1978)

4.7 NI-488

บริษัทเนชั่นแนลอินสทรูเมนต์ ได้นำเอามาตรฐาน IEEE-488 มาสร้างเป็นผลิตภัณฑ์สำหรับไมโครคอมพิวเตอร์ ประกอบด้วยฮาร์ดแวร์ คือ GPIB-PCII และ GPIB-PCIIA สำหรับติดตั้งกับเครื่องประเภทไอบีเอ็มพีซี และส่วนที่เป็นซอฟต์แวร์คือ NI-488 controller software เดิมเรียกว่า NI-488 MS-DOS ภายใต้ระบบปฏิบัติการ MS-DOS (National Instrument Coparation, 1988)

4.7.1 GPIB-PCII และ GPIB-PCIIA

เป็นแผงวงจรที่ติดตั้งภายในเครื่องไมโครคอมพิวเตอร์ โดย GPIB-PCII แตกต่างจาก GPIB-PCIIA เล็กน้อย คือ GPIB-PCIIA มีแบตเตอรี่ภายในสำหรับนาฬิกา

4.7.2 NI-488 Software Package ประกอบด้วย

1) GPIB.COM เป็นซอฟต์แวร์ประเภทตัวขับอุปกรณ์ (device driver) โดยติดตั้ง device = GPIB.COM ไว้ในไฟล์ CONFIG.SYS

2) IBDIAG.EXE เป็นโปรแกรมที่ใช้สำหรับตรวจสอบการติดตั้งฮาร์ดแวร์

3) IBTEST.EXE เป็นโปรแกรมที่ใช้ในการตรวจสอบการติดตั้งซอฟต์แวร์

4) IBCONF.EXE เป็นซอฟต์แวร์สำหรับเปลี่ยนโครงแบบ (config.) ของโปรแกรม ซึ่งใช้กำหนดค่าพารามิเตอร์ต่าง ๆ ของ NI-488 software เช่น primary address, secondary address, time limit เป็นต้น

5) IBIC.EXE เป็นโปรแกรมแบบ interactive ที่ให้ป้อนคำสั่งหรือฟังก์ชันได้โดยตรงจากแป้นพิมพ์ และจะทำงานทันที ช่วยให้สามารถเรียนรู้ฟังก์ชันและการใช้งานตลอดจนทดสอบคำสั่งของอุปกรณ์ต่าง ๆ

6) APPMON.COM เป็นโปรแกรมเพื่อใช้ในการตรวจสอบการทำงานของ

โปรแกรม และค้นหาข้อผิดพลาดที่เกิดขึ้นในระหว่างการทดสอบโปรแกรม

7) IBTRAP.EXE เป็นโปรแกรมที่ทำหน้าที่ติดตั้ง APPMON.COM

8) ULI.COM เป็นซอฟต์แวร์ประเภทตัวขับอุปกรณ์ สำหรับอินเตอร์เฟสกับภาษาที่ใช้กันทั่วไป (universal language)

9) ไฟล์ที่ทำหน้าที่อินเตอร์เฟสกับ microsoft C ได้แก่ DECL.H ซึ่งเป็นไฟล์ที่ประกอบไปด้วยการกำหนดตัวแปรต่าง ๆ และการกำหนดค่าคงที่ และไฟล์ MCIBS.OBJ MCIBM.OBJ, MCIBC.OBJ ที่ใช้ในการลิงค์รวมกับโปรแกรมภาษาซี ที่พัฒนาขึ้นมาตามโมเดล small, medium, large และ compact ตามลำดับซึ่งมีฟังก์ชันการใช้งานตามตาราง 4.1 (National Instrument Coparation, 1989)

Description	Call Syntax
Change access board of device	ibbna (ud, bname)
Become Active Controller	ibcac (ud, v)
Clear specified device	ibclr (ud)
Send commands from string	ibcmd (ud, cmd, cnt)
Send commands asynchron. from string	ibcmda (ud, cmd, cnt)
Enable/disable DMA	ibdma (ud, v)
Change/disable EOS mode (write)	ibeos (ud, v)
Enable/disable END message	ibeot (ud, v)
Open device and return unit descriptor	ud = ibfind (udname)
Go from Active Controller to Standby	ibgts (ud, v)
Set/clear individual status bit for Parallel Polls	ibist (ud, v)
Go to Local	ibloc (ud)
Place device or board online/offline	ibonl (ud, v)
Change Primary Address	ibpad (ud, v)
Pass Control	ibpct (ud)
Parallel Poll Configure	ibppc (ud, v)
Read data to string	ibrd (ud, rd, cnt)
Read data asynchronously to string	ibrda (ud, rd, cnt)
Read data to file	ibrdf (ud, flname)
Conduct a Parallel Poll	ibrpp (ud, &ppr)
Request/release System Control	ibrsc (ud, v)
Return serial poll byte	ibrsp (ud, &spr)
Request service, set/change serial poll	ibrsv (ud, v)
Change/disable Secondary Address	ibsad (ud, v)
Send Interface Clear for 100 µsec	ibsic (ud)
Set/clear Remote Enable line	ibsre (ud, v)
Abort asynchronous operation	ibstop (ud)
Change/disable time limit	ibtmo (ud, v)
Configure Applications Monitor	ibtrap (mask, mode)
Trigger selected device	ibtrg (ud)
Wait for selected event	ibwait (ud, mask)
Write data from string	ibwrt (ud, wrt, cnt)
Write data asynchronously from string	ibwrta (ud, wrt, cnt)
Write data from file	ibwrtf (ud, flname)

ตาราง 4.1 C language NI-488 calls

4.7.3 **ฟังก์ชันของ NI-488** ประกอบด้วยฟังก์ชันในระดับสูง (high-level functions) และฟังก์ชันในระดับต่ำ (low-level functions) ใช้ในการสื่อสารและควบคุมอุปกรณ์ต่าง ๆ

1) **ฟังก์ชันในระดับสูง** หรือบางทีเรียกว่า **ฟังก์ชันของอุปกรณ์** (device

functions) เป็นฟังก์ชันที่ง่ายต่อการเรียนรู้และการใช้งาน ซึ่งผู้ใช้งานไม่จำเป็นต้องเรียนรู้ถึงพิธีการ (protocol) ของ IEEE-488 bus และรายละเอียดของวิธีการจัดการกับบัส ฟังก์ชันเหล่านี้จะปฏิบัติงานตามลำดับคำสั่งโดยอัตโนมัติ ทำหน้าที่จัดการกับบัสในการอ่านและเขียนข้อมูลกับอุปกรณ์ หรือแสดงสถานภาพของอุปกรณ์

2) ฟังก์ชันในระดับต่ำ หรือฟังก์ชันของบอร์ด (board functions) เป็นฟังก์ชันพื้นฐาน ที่ผู้ใช้งานจะต้องมีความรู้เกี่ยวกับพิธีการของ IEEE-488 bus ฟังก์ชันนี้ทำให้สามารถใช้งานได้อย่างยืดหยุ่น และสามารถอินเตอร์เฟสกับ GPIB-PCII หรือ GPIB-PCIIA ได้โดยตรง ผู้ใช้งานจะต้องจัดการกับบัสเอาเอง

4.7.4 **สถานภาพและข้อผิดพลาด** ได้เก็บไว้ในตัวแปร IBSTA และ IBERR ตามลำดับ ซึ่งทำให้สามารถตรวจสอบสถานภาพปัจจุบัน และข้อผิดพลาดที่เกิดขึ้นได้

1) Status Word (IBSTA) มีความยาว 1 ไบต์ ซึ่งทุกฟังก์ชันจะใช้สำหรับแสดงสถานภาพในการอินเตอร์เฟส แต่ละบิตจะถูกกำหนดโดยฟังก์ชันสำหรับอุปกรณ์ (dev) หรือฟังก์ชันสำหรับผู้ควบคุม (brd) ความหมายของสถานภาพแสดงในตาราง 4.2

Mnemonics	Bit Position	Hex Value	Function Type	Description
ERR	15	8000	dev, brd	GPIB error
TIMO	14	4000	dev, brd	Time limit exceeded
END	13	2000	dev, brd	END or EOS detected
SRQI	12	1000	brd	SRQ interrupt received
RQS	11	800	dev	Device requesting service
CMPL	8	100	dev, brd	I/O completed
LOK	7	80	brd	Lockout State
REM	6	40	brd	Remote State
CIC	5	20	brd	Controller-In-Charge
ATN	4	10	brd	Attention is asserted
TACS	3	8	brd	Talker
LACS	2	4	brd	Listener
DTAS	1	2	brd	Device Trigger State
DCAS	0	1	brd	Device clear State

ตาราง 4.2 Status Word Layout

2) Error Variable (IBERR) เมื่อบิต ERR ใน IBSTA ถูกกำหนดตัวแปรแสดงความผิดพลาด (IBERR) ซึ่งมีความยาว 1 ไบต์ ก็จะเป็นตัวบ่งบอกว่าเกิดความผิดพลาดอะไรขึ้น ดังตารางที่ 4.3

Suggested Mnemonic	Decimal Value	Explanation
EDVR	0	DOS error
ECIC	1	Function requires GPIB board to be CIC
ENOL	2	Write handshake error (e.g., no listener)
EADR	3	GPIB board not addressed correctly
EARG	4	Invalid argument to function call
ESAC	5	GPIB board not System Controller as required
EABO	6	I/O operation aborted
ENEB	7	Non-existent GPIB board
EOIP	10	I/O started before previous operation completed
ECAP	11	No capability for operation
EFSO	12	File system error
EBUS	14	Command error during device call
ESTB	15	Serial Poll status byte lost
ESRQ	16	SRQ stuck in ON position

ตาราง 4.3 Error Codes

4.7.5 การกำหนดเวลาในการปฏิบัติงาน (Time setting)

เวลาในการปฏิบัติงานได้ถูกกำหนดไว้สำหรับฟังก์ชันที่ติดต่อกับอุปกรณ์ เช่น IBRD, IBWRT และ IBCMD ซึ่งฟังก์ชันเหล่านี้จะปฏิบัติงานจนกระทั่งเสร็จ หรือได้รับสัญญาณ TIMO ก็จะหยุดการปฏิบัติงาน การกำหนดเวลาในการปฏิบัติงานสามารถกำหนดโดยโปรแกรม IBCONF.EXE หรือฟังก์ชัน IBTMO ก็ได้ ตารางการกำหนดเวลาแสดงในตาราง 4.4

Code Timeout	Actual Value	Minimum
TNONE	0	disabled *
T10µsec	1	10 µsec
T30µsec	2	30 µsec
T100µsec	3	100 µsec
T300µsec	4	300 µsec
T1msec	5	1 msec
T3msec	6	3 msec
T10msec	7	10 msec
T30msec	8	30 msec
T100msec	9	100 msec
T300msec	10	300 msec
T1sec	11	1 sec
T3sec	12	3 sec
T10sec	13	10 sec
T30sec	14	30 sec
T100sec	15	100 sec
T300sec	16	300 sec
T1000sec	17	1000 sec

* If you select TNONE, no limit will be in effect and I/O operations could proceed indefinitely.

ตาราง 4.4 การกำหนดเวลาในการปฏิบัติงาน

4.7.6 จุดสิ้นสุดของข้อมูล

IEEE-488 ได้กำหนดให้จุดสิ้นสุดของข้อมูลที่ส่งโดยผู้พูดไว้ 2 วิธี คือ ผู้พูดส่งสัญญาณ EOI (End of Identify) เพื่อบ่งบอกให้ผู้ฟังรับรู้ว่า ข้อมูลสิ้นสุดแล้ว และผู้พูดส่งรหัส EOS (End of String) ต่อท้ายข่าวสารที่ส่งไป ซึ่งจะเลือกวิธีไหนขึ้นกับผู้ใช้จะเลือกอย่างใดอย่างหนึ่งโดยใช้ฟังก์ชัน IBEOS และ IBEOT