

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การพัฒนาโฮสต์ยูเอสบีเวอร์ชัน 1.1 บนอุปกรณ์เอฟพีจีเอด้วยภาษาวีเอชดีแอล
USB HOST 1.1 IMPLEMENTATION ON FPGA WITH VHDL



เลขหมู่.....
เลขทะเบียน..... 62324
วัน,เดือน,ปี..... 16 ส.ค. 2549

b..... 1162114x
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโฮสต์ยูเอสบีเวอร์ชัน 1.1 บนอุปกรณ์เอ็ฟพีอีเอด้วยภาษาวีเอชดีแอล
USB HOST 1.1 IMPLEMENTATION ON FPGA WITH VHDL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2548

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโฮสต์ยูเอสบีเวอร์ชัน 1.1 บนอุปกรณ์เอฟพีจีเอด้วยภาษาวีเอชดีแอล

USB HOST 1.1 IMPLEMENTATION ON FPGA WITH VHDL

ผู้จัดทำ

1. นายพิศุทธิ์ มหัดโกมล รหัสนักศึกษา 45010546



อาจารย์ที่ปรึกษา

(เจริญ วงษ์ชุ่มเย็น)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโฮสต์ยูเอสบีเวอร์ชัน 1.1 บนอุปกรณ์เอฟพีจีเอด้วยภาษาวีเอชดีแอล

นายพิศุทธิ์ มหัทธโคมล 45010546
อ. เจริญ วงษ์ขุ่มเย็น อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอรูปแบบการทำงานของของการสื่อสารข้อมูลด้วยบัสยูเอสบีและนำเสนอการพัฒนาส่วนการทำงานของยูเอสบีโฮสต์โดยพัฒนานั้นจะยึดถือตามข้อกำหนดของมาตรฐานของการสื่อสารข้อมูลด้วยยูเอสบีเวอร์ชัน 1.1 ในการพัฒนาโฮสต์จะทำการพัฒนาโดยใช้ภาษาวีเอชดีแอล ซึ่งผลลัพธ์ของโครงการจะได้ไลบรารีของการสื่อสารข้อมูลด้วยยูเอสบี ทำให้สามารถนำไปใช้เป็นส่วนประกอบของการพัฒนาวงจรดิจิทัลด้วยภาษาวีเอชดีแอลที่ต้องการความสามารถในการติดต่อสื่อสารข้อมูลด้วยยูเอสบีโดยเน้นการพัฒนาในส่วนการทำงานของประมวลผลข้อมูลเท่านั้น ซึ่งในการสื่อสารข้อมูลด้วยยูเอสบีนั้น ยังมีความจำเป็นที่จะต้องใช้ไคร์เวอร์ในการสั่งการทำงานของอุปกรณ์ด้วย

ในการทดลองการทำงานของไลบรารีดังกล่าว จะใช้การทดลองบนอุปกรณ์เอฟพีจีเอ ซึ่งใช้อุปกรณ์เอฟพีจีเอเบอร์ XC3s200 เป็นอุปกรณ์ในการทดลองการทำงาน

USB HOST 1.1 IMPLEMENTATION ON FPGA WITH VHDL

Mr.Pisut Mahutkomol 45010001

Mr.Charoen Vongchumyen Advisor

Academic Year 2005

ABSTRACT

This thesis proposes a communication on bus USB (Universal Serial Bus) and development of USB Host which reference from USB 1.1 Standard. The implementation of USB 1.1 Host is using VHDL (Very High Speed Integrated Circuit Hardware Description Language). The product of project is a programmer library of USB communication, so it can be use as a part of other digital circuit development which want to use USB communication. The focus of project is only a data processing, but in USB communication also need a device driver to control the process of the USB Host.

In a testing the processing of USB library, I use a FPGA (Field Programmable Gate Array) device with part number XC3s200 for testing system.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้สำเร็จได้อย่างดี เนื่องจากได้รับคำแนะนำ สนับสนุน และการให้คำปรึกษาเป็นอย่างดี จากอาจารย์เจริญ วงษ์ชุ่มเย็น ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาบัตร จึงต้องขอกราบพระคุณเป็นอย่างสูง รวมทั้งอาจารย์ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ให้การอบรมสั่งสอนวิชาความรู้แก่ข้าพเจ้ามาโดยตลอด

และขอขอบพระคุณอย่างสูงสำหรับบุคคลที่สำคัญที่สุดที่ทำให้ข้าพเจ้าสามารถมาถึงวันนี้ได้ คือบิดามารดา ผู้เป็นที่เคารพรักยิ่งของข้าพเจ้า ซึ่งท่านให้การอบรมสั่งสอน เลี้ยงดู และให้โอกาสในการศึกษาอย่างเต็มที่ จึงขอกราบขอบพระคุณมา ณ ที่นี้

สุดท้ายนี้ ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆ และคอยให้กำลังใจเสมอมา

คุณค่าและประโยชน์อันพึงมาจากปริญญาบัตรฉบับนี้ ข้าพเจ้าขอบแต่ผู้มีพระคุณทุกท่าน

พิศุทธิ์ มหัตโกมล

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	IX
สารบัญรูป.....	X
บทที่ 1 บทนำ.....	1
1.1. ความสำคัญและที่มา.....	1
1.2. วัตถุประสงค์ของโครงการ.....	1
1.3. ประโยชน์ที่คาดว่าจะได้รับ.....	1
1.4. ขอบเขตของโครงการ.....	2
1.5. เนื้อหาของรายงาน.....	2
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	3
2.1. ภาพรวมของการสื่อสารข้อมูลด้วยยูเอสบี.....	3
2.1.1. ส่วนประกอบของระบบการสื่อสารข้อมูลด้วยยูเอสบี.....	3
1) USB Bus Interface Layer.....	3
2) USB Device Layer.....	4
3) Function Layer.....	4
2.1.2. โสตส์ยูเอสบี.....	4
2.1.3. อุปกรณ์ยูเอสบี.....	5
2.1.4. ยับยูเอสบี.....	6
2.1.5. รูปแบบการเชื่อมต่อ.....	7
1) การเชื่อมต่อในระดับกายภาพ.....	7
2) การเชื่อมต่อของการสื่อสารข้อมูล.....	7
3) การเชื่อมต่อของโปรแกรมผู้ใช้และการทำงานของอุปกรณ์.....	8
2.1.6. รูปแบบการสื่อสารข้อมูล.....	9
1) การสื่อสารข้อมูลแบบ Isochronous.....	9

สารบัญ(ต่อ)

	หน้า
2) การสื่อสารข้อมูลแบบ Interrupt.....	9
3) การสื่อสารข้อมูลแบบ Bulk.....	9
4) การสื่อสารข้อมูลแบบ Control.....	9
2.1.7. อินเทอร์เน็ต.....	9
2.1.8. เอนพอยน์.....	10
2.1.9. ไปป์.....	10
2.1.10. วิธีการสื่อสารข้อมูล.....	11
2.2. การทำงานในระดับโปรโตคอล.....	12
2.2.1. บทนำ.....	12
2.2.2. ฟิลด์ข้อมูลต่างๆ ในการสื่อสารข้อมูลยูเอสบี.....	13
1) ฟิลด์ SYNC (Synchronization).....	13
2) ฟิลด์ PID (Packet Identifier).....	13
3) ฟิลด์ ADDR (Address).....	14
4) ฟิลด์ ENDP (Endpoint).....	15
5) ฟิลด์ Frame Number.....	15
6) ฟิลด์ Data.....	15
7) ฟิลด์ CRC.....	16
2.2.3. แฟ้มกเกตต่างๆ ที่ใช้ในการสื่อสารข้อมูลยูเอสบี.....	16
1) แฟ้มกเกต Token.....	17
2) แฟ้มกเกต SOF (Start-of-Frame).....	17
3) แฟ้มกเกต Data.....	17
4) แฟ้มกเกต Handshake.....	18
2.2.4. ทรานแซคชัน.....	19
2.2.5. รูปแบบของการสื่อสารข้อมูล.....	19
1) การสื่อสารข้อมูลแบบ Isochronous.....	19
2) การสื่อสารข้อมูลแบบ Interrupt.....	20
3) การสื่อสารข้อมูลแบบ Bulk.....	20
4) การสื่อสารข้อมูลแบบ Control.....	20

สารบัญ(ต่อ)

	หน้า
2.3. ระดับการทำงานในระดับกายภาพ	21
2.3.1 สายสัญญาณยูเอสบี	21
2.3.2 ตัวเชื่อมต่อสำหรับอุปกรณ์ยูเอสบี	22
1) ตัวเชื่อมต่ออุปกรณ์ยูเอสบีชนิดเอ	22
2) ตัวเชื่อมต่ออุปกรณ์ยูเอสบีชนิดบี	23
2.3.3 ระดับสัญญาณแบบผลต่าง (Differential Signaling)	24
2.3.4 สถานะต่างๆ ของบัสยูเอสบี	24
1) สถานะบัสว่าง (Idle State)	26
2) สถานะบัส Single-Ended 0 (SE0)	26
3) สถานะบัส Differential 0	26
4) สถานะบัส Differential 1	26
5) สถานะบัส J (J State)	27
6) สถานะบัส K (K State)	27
7) สถานะบัสรีเซ็ต (Reset State)	27
8) สถานะบัสกลับสู่การทำงานปกติ (Resume State)	28
9) สถานะบัสเชื่อมต่ออุปกรณ์ (Connect State)	28
10) สถานะบัสสลอคอุปกรณ์ (Disconnect State)	30
11) สถานะบัส SOP (Start-Of-Packet)	31
12) สถานะบัส EOP (End-Of-Packet)	31
2.3.5 การเข้ารหัสและถอดรหัสสัญญาณข้อมูลแบบ NRZI	31
2.3.6 การแทรกบิต	32
2.4. การตั้งค่าการทำงานของอุปกรณ์	33
2.4.1 การรีเซ็ตตัวอุปกรณ์	33
2.4.2 การกำหนดหมายเลขแอดเดรสให้กับตัวอุปกรณ์	33
2.4.3 การเลือกโหมดการทำงานของอุปกรณ์	33
2.5. การตอบสนองต่อคำร้องขอ	34
2.6. เดสคริปเตอร์	36
2.6.1 ดีไวซ์เดสคริปเตอร์	36

สารบัญ(ต่อ)

	หน้า
2.6.2 คอนฟิกูเรชันเดสคริปเตอร์.....	38
2.6.3 อินเตอร์เฟซเดสคริปเตอร์.....	39
2.6.4 เอนพอยน์เดสคริปเตอร์.....	40
2.6.5 สตริงเดสคริปเตอร์.....	41
บทที่ 3 การออกแบบและการพัฒนา.....	42
3.1 การออกแบบระบบโดยรวม.....	42
3.1.1 ส่วนการทำงานของไคร์เวอร์เฉพาะอุปกรณ์.....	43
3.1.2 ส่วนการทำงานของชั้น โปรโตคอล.....	44
3.1.3 ส่วนการทำงานของชั้นการทำงานชั้นกายภาพ.....	44
3.2 การออกแบบส่วนการทำงานชั้นกายภาพ.....	45
3.2.1 ส่วนที่ทำหน้าที่ในการส่งข้อมูล.....	45
3.2.2 ส่วนที่ทำหน้าที่ในการรับข้อมูล.....	47
3.3 การออกแบบส่วนการทำงานในชั้น โปรโตคอล.....	49
3.3.1 การออกแบบส่วนตรวจสอบความถูกต้องของข้อมูล.....	51
3.3.2 การออกแบบส่วนประมวลผลหลัก.....	52
3.4 การออกแบบส่วนไคร์เวอร์.....	54
บทที่ 4 การทดลองและผลการทดลอง.....	55
4.1 ลำดับขั้นการทดลอง.....	55
4.2 การทดลองการทำงานของชั้นกายภาพ.....	55
4.2.1 การทดลองการทำงานของส่วนการเข้ารหัสข้อมูลด้วยการแทรกบิต.....	56
4.2.2 การทดลองการเข้ารหัสข้อมูลแบบ NRZI.....	57
4.2.3 การทดลองการทำงานของส่วนการแปลงข้อมูลสำหรับการส่งออก.....	58
4.2.4 การทดลองการทำงานในการส่งข้อมูล.....	58
4.2.5 การทดลองการถอดรหัสข้อมูลแบบ NRZI.....	59
4.2.6 การทดลองส่วนการถอดรหัสข้อมูลการแทรกบิต.....	60

สารบัญ(ต่อ)

หน้า

4.2.7 การทดลองหาค่าตรวจสอบความถูกต้องของข้อมูล CRC ขนาด 5 บิต.....	62
4.2.8 การทดลองตรวจสอบความผิดพลาดของข้อมูล ด้วยข้อมูลตรวจสอบขนาด 5 บิต.....	63
4.2.9 การทดลองหาค่าตรวจสอบความถูกต้องของข้อมูล CRC ขนาด 16 บิต.....	64
4.2.10 การทดลองตรวจสอบความผิดพลาดของข้อมูล ด้วยข้อมูลตรวจสอบขนาด 16 บิต.....	64
4.3 การทดลองการทำงานของส่วนไคร์เวอร์ร่วมกับการทำงานชั้นโปรโตคอล	66
4.3.1 การทดลองเขียนข้อมูลทรานแซคชัน เดสคริปเตอร์.....	65
4.3.2 การทดลองการทำงานของชั้นโปรโตคอล เมื่อได้รับทรานแซคชันเดสคริปเตอร์.....	67
4.3.3 การทดลองการทำงานของชั้นโปรโตคอล ในการส่งข้อมูลให้กับชั้นการทำงานระดับกายภาพ	68
4.3.4 การทดลองการนำข้อมูลเพื่อก่อกำเนิดจากชั้นโปรโตคอล ไปสร้างสัญญาณแบบผลต่าง.....	68
บทที่ 5 บทวิจารณ์และสรุป	70
5.1 บทสรุป.....	71
5.2 วิจารณ์สิ่งที่ได้จากโครงงาน	70
5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข.....	71
5.4 แนวทางการพัฒนาต่อ.....	72
บรรณานุกรม.....	73
ภาคผนวก	74
ภาคผนวก ก. รูปแสดงผลของการ Synthesis วงจรด้วยโปรแกรม Xilinx Project Navigator	75

สารบัญตาราง

ตารางที่	หน้า
2.1 แสดงชนิดของเพื่อก่เกิดต่างๆ.....	14
2.2 แสดงความยาวสูงสุดของสายสัญญาณที่สามารถใช้ได้.....	22
2.3 แสดงสถานะต่างๆ ของบัลยูเอสบี.....	25
2.4 แสดงโครงสร้างของคำร้องขอ.....	34
2.5 แสดง โครงสร้างข้อมูลของดีไวซ์เคสคริปเตอร์.....	37
2.6 แสดง โครงสร้างข้อมูลของคอนฟิกูเรชันเคสคริปเตอร์.....	38
2.7 แสดง โครงสร้างข้อมูลของอินเตอร์เฟสเคสคริปเตอร์.....	39
2.8 แสดง โครงสร้างข้อมูลของเอนพอยน์เคสคริปเตอร์.....	40
2.9 แสดง โครงสร้างข้อมูลของสตริงเคสคริปเตอร์.....	41
2.10 แสดง โครงสร้างข้อมูลของรหัสภาษา.....	41



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
IX
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 แสดงส่วนประกอบของการสื่อสารข้อมูลด้วยยูเอสบี	3
2.2 แสดงตัวอย่างยูเอสบี โฮสต์	4
2.2 แสดงอุปกรณ์เมาส์ที่เป็นอุปกรณ์ยูเอสบี	5
2.4 แสดงตัวอย่างคีย์บอร์ดที่เป็นอุปกรณ์ยูเอสบี	5
2.5 แสดงพอร์ตออปติคัลและพอร์ตควาน์ตัมของฮับ	6
2.6 แสดงตัวอย่างอุปกรณ์ฮับยูเอสบี	6
2.7 แสดงลักษณะการเชื่อมต่อแบบสตาร์เป็นลำดับชั้น	7
2.8 แสดงลักษณะการเชื่อมต่อแบบ Point-to-Point	8
2.9 แสดงลักษณะการเชื่อมต่อของ โปรแกรมและการทำงานของอุปกรณ์	8
2.10 แสดงอินเตอร์เฟส, เอนพอยน์, ไปป์	10
2.11 แสดงรูปแบบการสื่อสารข้อมูลเป็นเฟรม	11
2.12 แสดงรูปแบบของฟิลด์ PID	13
2.13 แสดงรูปแบบของฟิลด์ ADDR	14
2.14 แสดงรูปแบบของฟิลด์ ENDP	15
2.15 แสดงรูปแบบของฟิลด์ Data	15
2.16 แสดงรูปแบบของแฟ้มเกิด Token	17
2.17 แสดงรูปแบบของแฟ้มเกิด SOF	17
2.18 แสดงรูปแบบของแฟ้มเกิด Data	17
2.19 แสดงรูปแบบของแฟ้มเกิด Handshake	18
2.20 แสดงลักษณะของสายสัญญาณสำหรับอุปกรณ์ความเร็วต่ำ	21
2.21 แสดงลักษณะของสายสัญญาณสำหรับอุปกรณ์ความเร็วเต็มที่	22
2.22 แสดงลักษณะของตัวเชื่อมต่อชนิดเอ แบบตัวเสียบ	23
2.23 แสดงลักษณะของตัวเชื่อมต่อชนิดเอ แบบตัวรับ	23
2.24 แสดงลักษณะของตัวเชื่อมต่อชนิดบี แบบตัวเสียบ	23
2.25 แสดงลักษณะของตัวเชื่อมต่อชนิดบี แบบตัวรับ	24
2.26 แสดงลักษณะสัญญาณสถานะบัสว่าง	26
2.27 แสดงลักษณะสัญญาณของสถานะบัสรีเซต	27
2.28 แสดงลักษณะการต่อตัวต้านพูลอัพของอุปกรณ์ความเร็วเต็มที่	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า	
2.29	แสดงลักษณะของสัญญาณที่เกิดจากการเชื่อมต่ออุปกรณ์ความเร็วเต็มที่	29
2.30	แสดงลักษณะการต่อตัวต้านพูล์พของอุปกรณ์ความเร็วต่ำ	29
2.31	แสดงลักษณะของสัญญาณที่เกิดจากการเชื่อมต่ออุปกรณ์ความเร็วต่ำ	30
2.32	แสดงลักษณะของสัญญาณที่เกิดจากถอดอุปกรณ์ออกจากพอร์ต	30
2.33	แสดงลักษณะสัญญาณของสถานะบัส SOP ของอุปกรณ์ความเร็วเต็มที่	31
2.34	แสดงลักษณะสัญญาณของสถานะบัส SOP ของอุปกรณ์ความเร็วเต็มที่	31
2.35	แสดงตัวอย่างการเข้ารหัสข้อมูล NRZI และการแทรกบิต	32
2.36	แสดงรูปแบบของการอ้างอิงพอยน์ด้วย wIndex	35
2.37	แสดงรูปแบบของการอ้างอิงอินเตอร์เฟสด้วย wIndex	35
3.1	แสดงส่วนประกอบโดยรวมของระบบ	42
3.2	แสดงส่วนประกอบโดยรวมของส่วนการทำงานในชั้นกายภาพ	45
3.3	แสดงส่วนประกอบที่สำคัญของส่วนการเข้ารหัสโดยการแทรกบิต	46
3.4	แสดงขั้นตอนการทำงานของส่วนการเข้ารหัสโดยการแทรกบิต	46
3.5	แสดงส่วนประกอบที่สำคัญของส่วนการแปลงรูปแบบข้อมูลสำหรับการส่งออก	47
3.6	แสดงส่วนประกอบที่สำคัญของส่วนการแปลงข้อมูลสำหรับการรับเข้า	47
3.7	แสดงส่วนประกอบที่สำคัญของส่วนการถอดรหัสข้อมูลการแทรกบิต	48
3.8	แสดงขั้นตอนการทำงานของส่วนการถอดรหัสการแทรกบิต	48
3.9	แสดงรูปแบบข้อมูลของทรานแซกชันเดสคริปเตอร์ สำหรับแพ็กเก็ต SOF	49
3.10	แสดงรูปแบบข้อมูลของทรานแซกชันเดสคริปเตอร์ สำหรับแพ็กเก็ต IN, OUT, SETUP	49
3.11	แสดงรูปแบบข้อมูลของทรานแซกชันเดสคริปเตอร์ สำหรับแพ็กเก็ต DATA	50
3.12	แสดงส่วนประกอบของส่วนการทำงานในชั้นโปรโตคอล	50
3.13	แสดงส่วนประกอบหลักของส่วนการสร้าง และตรวจสอบความถูกต้องของข้อมูล	51
3.14	แสดงส่วนประกอบของส่วนประมวลผลหลัก	52

สารบัญรูป(ต่อ)

รูปที่	หน้า
4.1	แสดงอุปกรณ์เอฟพีจีเอที่ใช้ในการทดลอง..... 55
4.2	แสดงอุปกรณ์ออสซิลอสโคปที่ใช้ในการตรวจวัดสัญญาณในการทดลอง..... 56
4.3	แสดงผลการทดลองการทำงานของส่วนเข้ารหัสข้อมูลด้วยการแทรกบิต..... 56
4.4	แสดงผลการทดลองการทำงานของส่วนการเข้ารหัสข้อมูลแบบ NRZI..... 57
4.5	แสดงผลการทดลองการทำงานของส่วนการแปลงข้อมูลสำหรับการส่งออก 58
4.6	แสดงผลการทดลองการทำงานในการส่งข้อมูล..... 59
4.7	แสดงผลการทดลองของการแปลงรูปแบบข้อมูลสำหรับรับเข้า และการถอดรหัสข้อมูลแบบ NRZI..... 60
4.8	แสดงผลการทดลองการทำงานของส่วนการแปลงข้อมูลในการรับข้อมูลเข้า และการถอดรหัสข้อมูลการแทรกบิต..... 61
4.9	แสดงตัวอย่าง โปรแกรมจำลองการทำงาน Modelsim 63
4.10	แสดงผลการทดลองหาค่าตรวจความถูกต้องของข้อมูลขนาด 5 บิต 63
4.11	แสดงผลการทดลองการตรวจสอบความผิดพลาดของข้อมูล ด้วยข้อมูลตรวจสอบขนาด 5 บิต..... 63
4.12	แสดงผลการทดลองหาค่าตรวจความถูกต้องของข้อมูลขนาด 16 บิต 64
4.13	แสดงผลการทดลองการตรวจสอบความผิดพลาดของข้อมูล ด้วยข้อมูลตรวจสอบขนาด 16 บิต..... 64
4.14	แสดงผลการทดลองโดยการเขียนข้อมูล ABCDH ลงในทรานแซคชัน เคสคริปเตอร์..... 66
4.15	แสดงผลการทดลองการทำงานของชั้นโปรโตคอล เมื่อเขียนข้อมูลทรานแซคชัน เคสคริปเตอร์ด้วยค่า 5100H 67
4.16	แสดงผลการทดลองของการส่งข้อมูลให้ 68
4.17	แสดงผลการทดลองของการสร้างสัญญาณแบบผลต่างของแพ็กเก็ตข้อมูล..... 69

บทที่ 1

บทนำ

1.1. ความสำคัญและที่มาของโครงการ

ในปัจจุบัน อุปกรณ์อิเล็กทรอนิกส์ต่างๆ มักจะมีความสามารถในการสื่อสารข้อมูลด้วยยูเอสบีซี ซึ่งแสดงให้เห็นถึงความแพร่หลายของรูปแบบการสื่อสารข้อมูลในลักษณะนี้ และในขณะเดียวกัน ก็ยังคงมีการศึกษาและพัฒนา รูปแบบการสื่อสารข้อมูลแบบยูเอสบีซีให้มีขีดความสามารถเพิ่มขึ้น ตลอดจนความสะดวกต่อการใช้งาน ดังนั้น จึงมีความเป็นไปได้สูงที่รูปแบบการสื่อสารข้อมูลด้วยยูเอสบีซีนั้น จะเป็นการสื่อสารข้อมูลพื้นฐานที่จะถูกใช้งานอย่างแพร่หลายในอนาคตอันใกล้

แต่อย่างไรก็ตาม อุปกรณ์ที่มีการสื่อสารข้อมูลด้วยยูเอสบีซีส่วนใหญ่ นั้น ถูกออกแบบมาเพื่อใช้สื่อสารข้อมูลกับเครื่องคอมพิวเตอร์เป็นหลัก และเนื่องจากรูปแบบการทำงานของยูเอสบีซีที่จำเป็นจะต้องมีส่วนควบคุมการทำงานจึงจะสามารถทำงานได้ ในขณะที่ส่วนควบคุมการทำงานดังกล่าว มักจะมีอยู่ในเครื่องคอมพิวเตอร์ อุปกรณ์อื่นๆ พบว่ายังมีจำนวนน้อยที่มีความสามารถในการทำงานเป็นตัวควบคุมการสื่อสารข้อมูลด้วยยูเอสบีซี

ด้วยเหตุผลข้างต้น โครงการนี้จึงเสนอการพัฒนา ยูเอสบีซีโฮสต์ซึ่งเป็นตัวควบคุมการทำงานของการสื่อสารข้อมูลด้วยยูเอสบีซีเพื่อขยายขีดความสามารถของอุปกรณ์ยูเอสบีซีที่จะต้องสื่อสารกับเครื่องคอมพิวเตอร์เพียงอย่างเดียว ให้สามารถติดต่อสื่อสารข้อมูลกับอุปกรณ์อื่นๆ ได้ด้วย

1.2. วัตถุประสงค์ของโครงการ

- 1.2.1. เพื่อเพิ่มความสะดวกในการออกแบบวงจรดิจิทัลที่ต้องการความสามารถในการสื่อสารข้อมูลด้วยยูเอสบีซี
- 1.2.2. เพื่อศึกษาการกระบวนทำงานของการสื่อสารข้อมูลด้วยยูเอสบีซี

1.3. ประโยชน์ที่คาดว่าจะได้รับ

- 1.3.1. ได้รับความรู้ ความเข้าใจเกี่ยวกับกระบวนการทำงาน ของยูเอสบีซี
- 1.3.2. ได้รับความรู้ ความเข้าใจเกี่ยวกับการออกแบบวงจรดิจิทัลด้วยวีเอชดีแอล
- 1.3.3. สามารถสร้างไลบรารีของอุปกรณ์ยูเอสบีซีโฮสต์
- 1.3.4. สามารถเป็น โครงการเริ่มต้นสำหรับการพัฒนาต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4. ขอบเขตของโครงการงาน

- 1.4.1. สร้างอุปกรณ์ยูเอสบีไอส์ต์ที่มีลักษณะการสื่อสารข้อมูลและการทำงานครอบคลุมข้อกำหนดของมาตรฐานยูเอสบีเวอร์ชัน 1.1 ในส่วนของการสื่อสารข้อมูล
- 1.4.2. สามารถรองรับการใช้งานกับอุปกรณ์ในระดับความเร็วเต็มที่ โดยมีความเร็วในการสื่อสารข้อมูล 12 เมกะบิตต่อวินาที (Mbps)
- 1.4.3. สามารถสื่อสารข้อมูลกับอุปกรณ์แฟลชไดรฟ์ (Flash Drive) ที่สนับสนุนการทำงานของยูเอสบีเวอร์ชัน 1.1 โดยสามารถอ่านและเขียนข้อมูลต่างๆ ลงบนแฟลชไดรฟ์ได้

1.5. ส่วนประกอบของปริยญาณิพนธ์

บทที่ 1 บทนำ

กล่าวถึงความสำคัญและที่มาของโครงการ, วัตถุประสงค์ของโครงการ, ขอบเขตของโครงการ, วิธีการดำเนินการ, ประโยชน์ที่คาดว่าจะได้รับ และส่วนประกอบของปริยญาณิพนธ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

กล่าวถึงทฤษฎีพื้นฐานที่ใช้ในโครงการ ซึ่งประกอบด้วยทฤษฎีดังนี้ คือ ภาพรวมของการสื่อสารข้อมูลด้วยยูเอสบี, การทำงานในระดับโปรโตคอล, การทำงานในระดับกายภาพ, การตั้งค่าการทำงาน และเคสคริปเตอร์ต่างๆ ของอุปกรณ์ยูเอสบี

บทที่ 3 การออกแบบและการพัฒนา

กล่าวถึงขั้นตอนการออกแบบและการพัฒนาโฮสต์ยูเอสบี โดยอธิบายการออกแบบและพัฒนาออกเป็นขั้นตอน ตั้งแต่การออกแบบระบบโดยรวม, การออกแบบส่วนการทำงานในชั้นกายภาพ, การออกแบบส่วนการทำงานในชั้นโปรโตคอล และการออกแบบไดรเวอร์

บทที่ 4 การทดลองและผลการทดลอง

กล่าวถึงขั้นตอนของการทดลองและผลการทดลองการทำงานของโฮสต์ยูเอสบี โดยแบ่งการทดลองออกเป็นขั้นตอนต่างๆ ได้แก่ การทดลองการทำงานในชั้นกายภาพ, การทดลองการทำงานในชั้นโปรโตคอล, การทดลองการทำงานในชั้นโปรโตคอลร่วมกับไดรเวอร์

บทที่ 5 บทวิจารณ์และสรุป

กล่าวถึงบทสรุปของโครงการ, วิจารณ์สิ่งที่ได้รับจากโครงการ, ปัญหาของการดำเนินงานและการแก้ไข และแนวทางในการพัฒนาต่อ

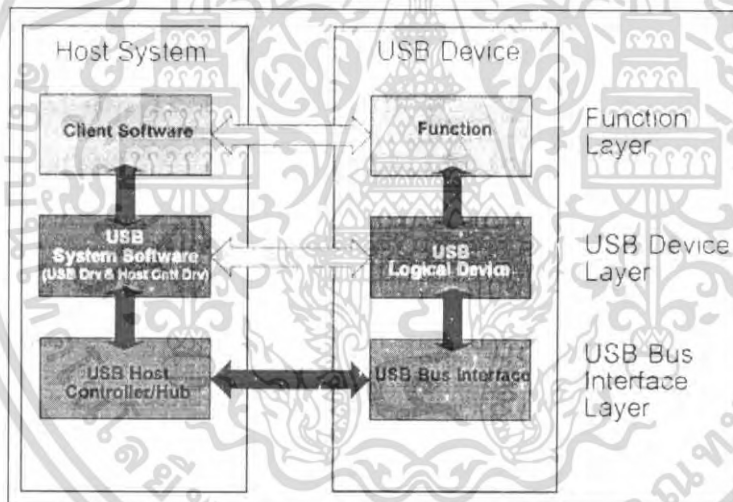
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1. ภาพรวมของการสื่อสารข้อมูลด้วยยูเอสบี

ในส่วนนี้ จะแสดงให้เห็นภาพรวมของการสื่อสารข้อมูลด้วยยูเอสบี โดยจะเป็นการแสดงการทำงานของแต่ละส่วนบนระบบการสื่อสารข้อมูลด้วยยูเอสบี

2.1.1. ส่วนประกอบของระบบการสื่อสารข้อมูลด้วยยูเอสบี

ส่วนประกอบของระบบการสื่อสารข้อมูลด้วยยูเอสบีประกอบด้วยการทำงานของ 2 ฝ่าย ได้แก่ ฝ่ายที่เป็นโฮสต์ (USB Host) และอุปกรณ์ยูเอสบี (USB Device) ซึ่งมีการเชื่อมต่อสื่อสารด้วย USB Interconnect ดังรูปที่ 2.1 แสดงส่วนประกอบของระบบการสื่อสารข้อมูลด้วยยูเอสบี



รูปที่ 2.1 แสดงส่วนประกอบของการสื่อสารข้อมูลด้วยยูเอสบี

1) USB Bus Interface Layer

มีหน้าที่รับผิดชอบการทำงานในระดับกายภาพซึ่งได้แก่ การเชื่อมต่อระหว่างโฮสต์และอุปกรณ์ยูเอสบี, การควบคุมและจัดการลักษณะของสัญญาณไฟฟ้า, การสื่อสารข้อมูลในรูปแบบสัญญาณไฟฟ้า โดยที่การสื่อสารระหว่างโฮสต์และอุปกรณ์ยูเอสบี ในลักษณะทางกายภาพนั้น จะใช้สื่อกลางการสื่อสารด้วยสายสัญญาณยูเอสบี (USB Cable)

2) USB Device Layer

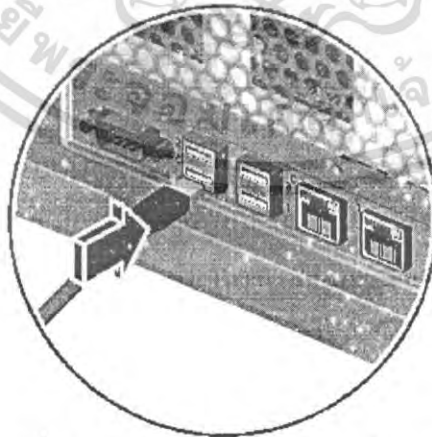
มีหน้าที่ในการจัดการการสื่อสารข้อมูล โดยจะเป็นการติดต่อสื่อสารระหว่างโปรแกรมระบบ (ไดรเวอร์ต่างๆ ของทางฝั่งโฮสต์) และ USB Logical Device ของทางฝั่งอุปกรณ์ยูเอสบี ทั้งนี้ เนื่องจากการทำงานของ อุปกรณ์ยูเอสบี อาจจะมีหน้าที่การทำงานได้มากกว่า 1 หน้าที่ ดังนั้น USB Logical Device จะเป็นตัวแทนของแต่ละหน้าที่การทำงาน ของ อุปกรณ์ยูเอสบี ตัวนั้นๆ

3) Function Layer

เป็นการสื่อสารในมุมมองผู้ใช้งาน ซึ่งจะเป็นการติดต่อสื่อสารข้อมูลระหว่างโปรแกรมของผู้ใช้งาน (ทางฝั่งโฮสต์) และส่วนที่เป็นการทำงาน (Function) ของอุปกรณ์ยูเอสบี

2.1.2. ยูเอสบีโฮสต์ (USB Host)

เป็นส่วนประกอบที่จะมีเพียงตัวเดียวเท่านั้นในระบบการสื่อสารข้อมูลด้วยยูเอสบีและเป็นส่วนประกอบที่ทำหน้าที่ในการควบคุมการทำงานของบัสและคอยควบคุมลำดับการสื่อสารข้อมูลทั้งหมดในระบบ และเพื่อเป็นการขยายความสามารถในการรองรับการเชื่อมต่อของอุปกรณ์ยูเอสบี จึงมีการรวมความสามารถของฮับ (Hub) ไว้ในตัวโฮสต์ซึ่งจะเรียกฮับที่ถูกรวมเข้ากับตัวโฮสต์ ว่า รุทฮับ (Root Hub) ตัวอย่างของอุปกรณ์ยูเอสบีเช่น คอมพิวเตอร์ (เฉพาะส่วนที่รับผิดชอบการสื่อสารข้อมูลด้วยยูเอสบี ดังแสดงดังรูปที่ 2.2

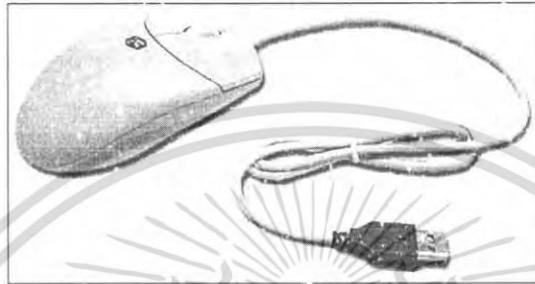


รูปที่ 2.2 แสดงตัวอย่างของอุปกรณ์ยูเอสบี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3. อุปกรณ์ยูเอสบี (USB Device)

เป็นส่วนประกอบที่เป็นลักษณะของตัวอุปกรณ์ยูเอสบีต่างๆที่ถูกออกแบบให้มีหน้าที่การทำงานเฉพาะ และมีการสื่อสารข้อมูลด้วยการใช้รูปแบบยูเอสบีซึ่งในการที่จะสื่อสารข้อมูลได้นั้น จำเป็นที่จะต้องมีส่วนที่เป็นโฮสต์คอยควบคุมให้ทำงานให้เป็นไปตามรูปแบบการสื่อสารที่ถูกต้อง ตัวอย่างของ อุปกรณ์ยูเอสบี เช่น เมาส์, คีย์บอร์ด รูปที่ 2.3 และ 2.4 แสดงตัวอย่างอุปกรณ์ยูเอสบี



รูปที่ 2.3 แสดงอุปกรณ์เมาส์ที่เป็นอุปกรณ์ยูเอสบี

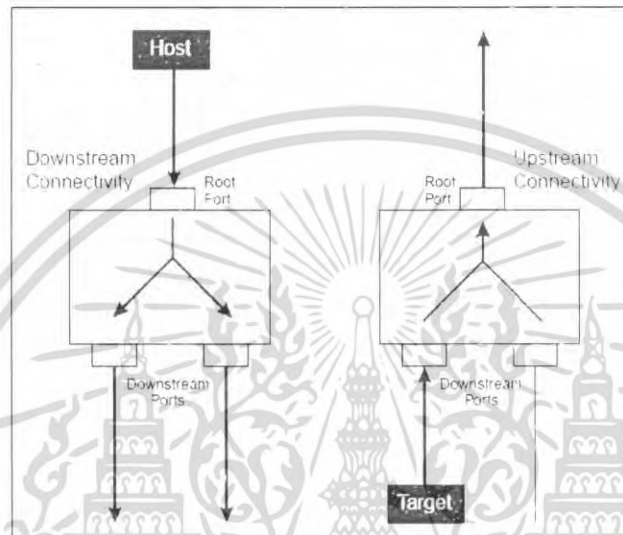


รูปที่ 2.4 แสดงตัวอย่างคีย์บอร์ดที่เป็นอุปกรณ์ยูเอสบี

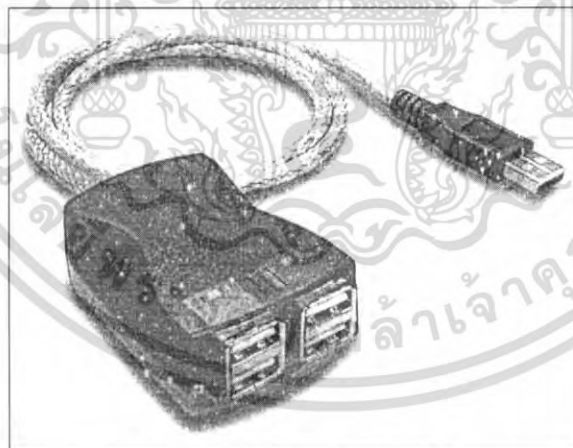
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4. ยูเอสบีฮับ (USB Hub)

หน้าที่ของฮับจะใช้ในการขยายจำนวนพอร์ตของการเชื่อมต่อกับ อุปกรณ์ยูเอสบี ซึ่งการทำงานของฮับนั้น จะทำการส่งต่อข้อมูลที่รับจากทางพอร์ตอัปสตรีม (Upstream) ซึ่งเป็นพอร์ตที่ต่อมาจากโฮสต์ ออกไปยังพอร์ตดาวนสตรีม (Downstream) ซึ่งเป็นพอร์ตที่ต่อไปยังอุปกรณ์ ตัวอย่างการทำงานของฮับแสดงดังรูปที่ 2.5 และ 2.6



รูปที่ 2.5 แสดงพอร์ตอัปสตรีมและพอร์ตดาวนสตรีมของฮับ



รูปที่ 2.6 แสดงตัวอย่างอุปกรณ์ฮับ

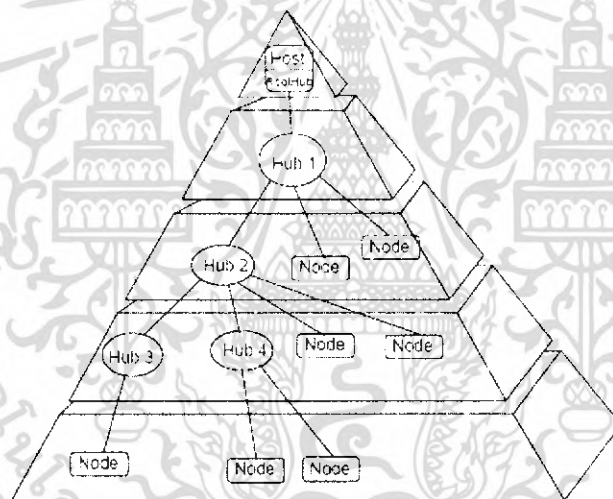
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5. รูปแบบการเชื่อมต่อ (Topology)

การเชื่อมต่ออุปกรณ์ต่างๆ ในระบบการสื่อสารยุคสมัยนั้น สามารถแบ่งออกได้หลายลักษณะ ขึ้นอยู่กับระดับของการพิจารณาว่าจะพิจารณาการเชื่อมต่อในระดับใด ซึ่งได้แก่

1) การเชื่อมต่อในระดับกายภาพ (Physical Bus Topology)

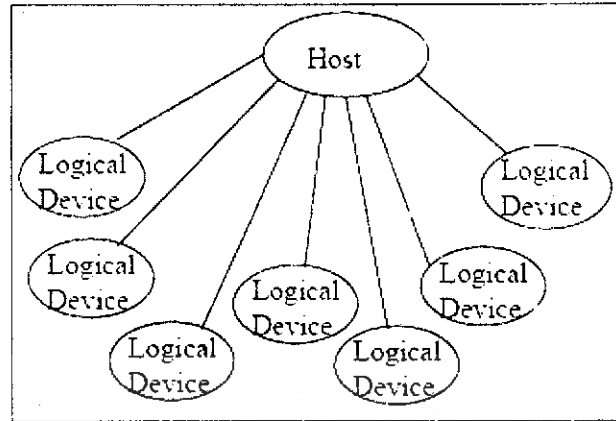
การเชื่อมต่อในระดับกายภาพจะเป็นการเชื่อมต่อของสายสัญญาณยุคสมัยโดยจะเริ่มต้นจากรูทฮับภายในโฮสต์ ซึ่งจะต่อจากพอร์ตนั้นของรูทฮับไปยังอุปกรณ์ต่างๆ ซึ่งอาจจะเป็น อุปกรณ์ยุคสมัย หรืออาจจะเป็นฮับ ตัวอื่นๆ ก็ได้ ด้วยรูปแบบการต่อเชื่อมอุปกรณ์ในลักษณะนี้ จึงเป็นรูปแบบการเชื่อมต่อแบบสตาร์ (Star) ที่มีลักษณะเป็นลำดับชั้น โดยมีโฮสต์เป็นจุดรวมสัญญาณ แล้วกระจายไปตามพอร์ตต่างๆ ของรูทฮับ รูปที่ 2.7 แสดงลักษณะการเชื่อมต่อแบบสตาร์โดยแสดงเป็นลำดับชั้น



รูปที่ 2.7 แสดงลักษณะการเชื่อมต่อแบบสตาร์เป็นลำดับชั้น

2) การเชื่อมต่อของการสื่อสารข้อมูล (Logical Bus Topology)

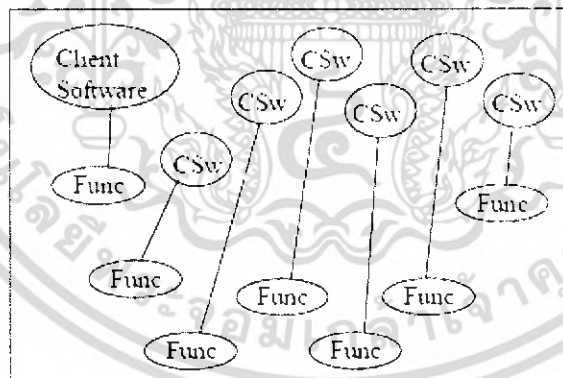
การเชื่อมต่อของการสื่อสารข้อมูลจะพิจารณาจากลักษณะของข้อมูลที่มีการสื่อสารภายในบัส ซึ่งเป็นการสื่อสารข้อมูลระหว่างโฮสต์และ Logical Device โดยในการสื่อสารข้อมูลนั้นจะเป็นการส่งจากโฮสต์ไปยังอุปกรณ์ที่มีการกำหนดโดยจะมีอุปกรณ์เพียงตัวเดียวเท่านั้นที่จะทำการสื่อสารข้อมูลกับโฮสต์ในระยะเวลาหนึ่งๆ ลักษณะการเชื่อมต่อนั้น จะเป็นการเชื่อมต่อจากโฮสต์ไปยัง Logical Device ทุกๆ ตัว จึงสามารถพิจารณาการเชื่อมต่อของการสื่อสารข้อมูลเป็นแบบสตาร์ ซึ่งแสดงตัวอย่างดังรูปที่ 2.8



รูปที่ 2.8 แสดงลักษณะการเชื่อมต่อแบบสตาร์

3) การเชื่อมต่อของโปรแกรมผู้ใช้และการทำงานของอุปกรณ์ยูเอสบี

ในส่วนของโปรแกรมผู้ใช้จะทำการสื่อสารข้อมูลกับส่วนที่เป็นการทำงานของอุปกรณ์ในแต่ละหน้าที่การทำงาน (ดังที่กล่าวมาแล้วในหัวข้อ 2.1.1 ส่วนประกอบของการสื่อสารข้อมูลด้วย ยูเอสบี) ซึ่งในแต่ละหน้าที่การทำงานของอุปกรณ์นั้นจะถูกแทนด้วย Logical Device และมีการเชื่อมต่อกับแต่ละโปรแกรมของบนโฮสต์ จึงสามารถพิจารณาการเชื่อมต่อของโปรแกรมและการทำงานของอุปกรณ์ได้เป็นแบบ Point-to-Point ดังตัวอย่างตามรูปที่ 2.9



รูปที่ 2.9 แสดงลักษณะการเชื่อมต่อของโปรแกรมและการทำงานของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6. รูปแบบการสื่อสารข้อมูล

การสื่อสารข้อมูลมีหลายรูปแบบ ขึ้นอยู่กับลักษณะการทำงานของอุปกรณ์และความต้องการของโปรแกรมผู้ใช้ ซึ่งรูปแบบการสื่อสารข้อมูลต่างๆ มีดังนี้

1) การสื่อสารข้อมูลแบบ Isochronous (Isochronous Transfer)

การสื่อสารข้อมูลแบบ Isochronous เป็นการสื่อสารข้อมูลที่มีอัตราเร็วของข้อมูลคงที่ แต่ไม่มีการรับประกันความถูกต้องของข้อมูล

2) การสื่อสารข้อมูลแบบ Interrupt (Interrupt Transfer)

การสื่อสารข้อมูลแบบ Interrupt มีลักษณะการทำงานโดยตัวโฮสต์จะคอยวนรับข้อมูลเป็นระยะๆ ตามเวลาที่กำหนดไว้อย่างสม่ำเสมอ และยังมีรับประกันความถูกต้องของข้อมูลด้วย

3) การสื่อสารข้อมูลแบบ Bulk (Bulk Transfer)

การสื่อสารข้อมูลแบบ Bulk เป็นการสื่อสารข้อมูลที่มีอัตราเร็วของการสื่อสารข้อมูลมีค่าไม่คงที่ โดยขึ้นอยู่กับปริมาณข้อมูลภายในบัฟและมีการรับประกันความถูกต้องของข้อมูลเช่นกัน

4) การสื่อสารข้อมูลแบบ Control (Control Transfer)

การสื่อสารแบบ Control ไม่ได้ถูกใช้สำหรับการสื่อสารข้อมูล แต่จะใช้เพื่อส่งคำสั่งควบคุมให้ตัวอุปกรณ์ยูเอสบี และมีการตรวจสอบความถูกต้องก่อนที่จะทำการประมวลคำสั่งนั้นๆ เพื่อความถูกต้องในการทำงาน

2.1.7. อินเทอร์เฟซ

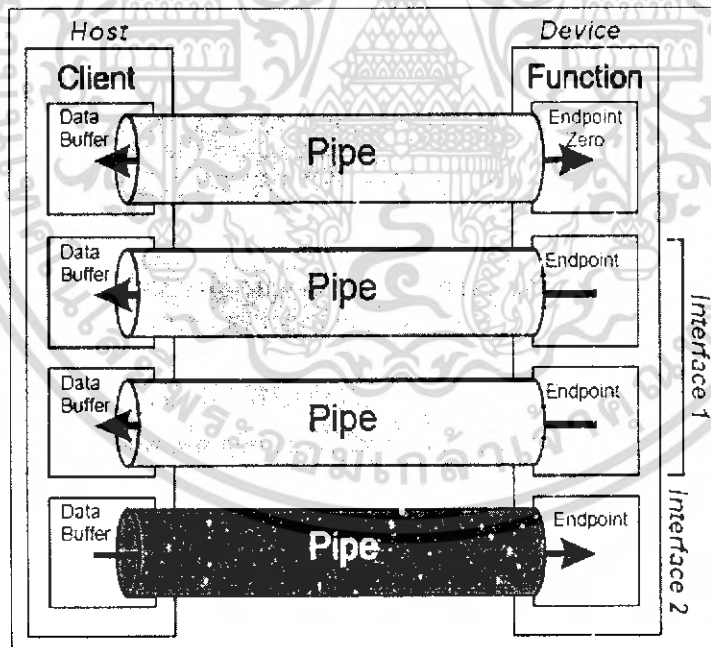
เนื่องจากอุปกรณ์ยูเอสบีแต่ละอุปกรณ์นั้น สามารถมีหน้าที่การทำงานที่มากกว่า 1 หน้าที่การทำงาน ซึ่งในแต่ละการทำงาน จะแสดงด้วยแต่ละ Logical Device และในการติดต่อกับแต่ละ Logical Device นั้น จะใช้การรูปแบบของอินเทอร์เฟซในการติดต่อ ซึ่งในแต่ละอินเทอร์เฟซก็จะหมายถึงแต่ละ Logical Device นั้นเอง ตัวอย่างการติดต่อ Logical Device โดยใช้อินเทอร์เฟซ แสดงดังรูปที่ 2.10

2.1.8. เอนพอยน์

ในการทำงานของแต่ละ Logical Device นั้นอาจมีความจำเป็นที่จะต้องมีช่องทางการสื่อสารมากกว่า 1 ช่องทาง ซึ่งในแต่ละช่องทางการสื่อสารก็จะมีรูปแบบการสื่อสารข้อมูลที่แตกต่างกัน แล้วแต่ความต้องการของ Logical Device นั้นๆ โดยในแต่ละช่องทางการสื่อสาร ก็จะแสดงด้วย 1 เอนพอยน์และในทุกๆ อุปกรณ์จะต้องมีเอนพอยน์หมายเลข 0 สำหรับการควบคุมการทำงานของอุปกรณ์ ตัวอย่างการใช้เอนพอยน์ติดต่อกับอินเตอร์เฟซ แสดงดังรูปที่ 2.10

2.1.9. ไปป์

ในการติดต่อสื่อสารกับอินเตอร์เฟซ ของตัวอุปกรณ์ด้วยการใช้เอนพอยน์นั้น จะใช้ไปป์ในการแสดงการเชื่อมต่อระหว่างโปรแกรมและเอนพอยน์ ซึ่งจำนวนของไปป์ในการติดต่อสื่อสารกับอุปกรณ์จะขึ้นอยู่กับจำนวนของเอนพอยน์ของอุปกรณ์นั้นๆ และในการเชื่อมต่อกับอุปกรณ์ใดๆ จะต้องมี Default Control Pipe ที่ติดต่อสื่อสารกับเอนพอยน์หมายเลข 0 ของอุปกรณ์ เพื่อใช้เป็นช่องทางการสื่อสารคำสั่งควบคุมการทำงานของอุปกรณ์ ตัวอย่างการสร้างการเชื่อมต่อด้วยไปป์ แสดงดังรูปที่ 2.10

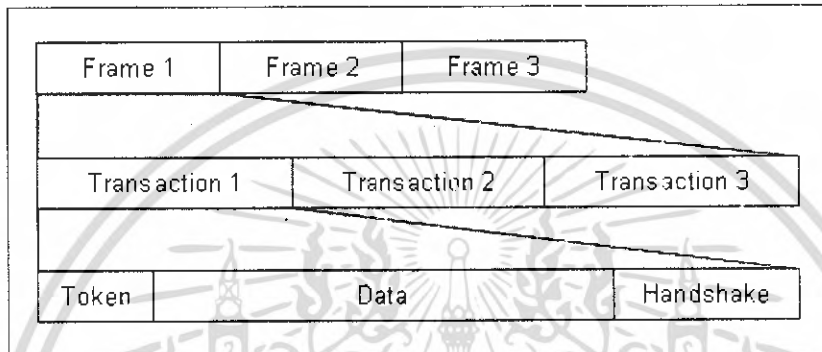


รูปที่ 2.10 แสดงอินเตอร์เฟซ, เอนพอยน์, ไปป์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.10. วิธีการสื่อสารข้อมูล

รูปแบบของการสื่อสารข้อมูลจะมีลักษณะเป็นเฟรมข้อมูล โดยโฮสต์จะต้องสร้างเฟรมข้อมูลในทุกๆ 1 มิลลิวินาที โดยที่ในแต่ละเฟรมนั้นจะประกอบด้วยหลายช่วงของการสื่อสารข้อมูลหรือทรานแซคชัน (Transaction) และในแต่ละทรานแซคชันก็จะประกอบขึ้นมาจากหลายๆ แพ็กเก็ต โดยขึ้นอยู่กับประเภทของทรานแซคชันนั้นๆ ว่าจะต้องประกอบด้วยแพ็กเก็ต ชนิดใดบ้าง ซึ่งรูปแบบของแต่ละแพ็กเก็ตจะกล่าวในภายหลัง ตัวอย่างการสื่อสารข้อมูลแสดงดังรูปที่ 2.11



รูปที่ 2.11 แสดงรูปแบบการสื่อสารข้อมูลเป็นเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2. การทำงานในระดับโปรโตคอล (Protocol Layer)

2.2.1. บทนำ

ในการสื่อสารข้อมูลด้วยยูเอสบีซีจะมีลักษณะการสื่อสารข้อมูลในลักษณะเป็นเฟรม ซึ่งในแต่ละเฟรม ก็ประกอบไปด้วยหลายช่วงการสื่อสารข้อมูลหรือทรานแซคชัน โดยที่แต่ละทรานแซคชันอาจจะมีรูปแบบการสื่อสารข้อมูลที่แตกต่างกัน ซึ่งทรานแซคชันรูปแบบที่ต่างกัน ก็จะมีส่วนประกอบภายในที่แตกต่างกันเช่นกัน โดยที่ส่วนประกอบของทรานแซคชันจะประกอบจากแพ็กเก็ต หลายๆ แพ็กเก็ตรวมกันขึ้นอยู่กับชนิดของทรานแซคชัน โดยที่ในแต่ละประเภทของแพ็กเก็ตจะมีหน้าที่แตกต่างกัน ชนิดของแพ็กเก็ตต่างๆ มีดังนี้

- แพ็กเก็ต **Token** ใช้สำหรับการบอกกับระบบว่าโฮสต์ต้องการสื่อสารเฟรมกับอุปกรณ์ตัวใดบนบัส
- แพ็กเก็ต **SOF** ใช้เป็นสัญลักษณ์ของจุดเริ่มต้นของเฟรมข้อมูลในแต่ละเฟรมข้อมูล
- แพ็กเก็ต **Data** ใช้สำหรับบรรจุข้อมูลที่ต้องการสื่อสารกับอุปกรณ์ต่างๆ
- แพ็กเก็ต **Handshake** ใช้สำหรับการรายงานผลของการสื่อสารข้อมูลในแต่ละทรานแซคชัน

ส่วนประกอบของแพ็กเก็ตต่างๆ ดังกล่าวนั้น จะเกิดจากการนำฟิลด์ข้อมูลต่างๆ มาประกอบเข้าด้วยกัน ซึ่งฟิลด์ข้อมูลก็จะมีหลายรูปแบบ ขึ้นอยู่กับหน้าที่ของแต่ละฟิลด์ โดยชนิดของฟิลด์ต่างๆ นั้นมีดังต่อไปนี้

- ฟิลด์ **SYNC** เป็นส่วนที่ใช้โดยตัวอุปกรณ์ในการจับจังหวะของการสื่อสารข้อมูลให้สัมพันธ์กับสัญญาณนาฬิกาของตัวอุปกรณ์เอง
- ฟิลด์ **PID** เป็นตัวบอกตัวอุปกรณ์ว่าแพ็กเก็ตที่ตัวอุปกรณ์ได้รับนั้นเป็นแพ็กเก็ตประเภทใด
- ฟิลด์ **ADDR** เป็นส่วนที่ทำการระบุตัวอุปกรณ์ว่าจะให้อุปกรณ์ตัวใดบนบัสทำงานตอบสนองในเฟรมข้อมูลนั้นๆ
- ฟิลด์ **ENDP** เป็นส่วนที่ใช้แจ้งเอนพอยน์ของตัวอุปกรณ์ว่าต้องการติดต่อกับ เอนพอยน์ใด
- ฟิลด์ **Frame Number** ใช้สำหรับการอ้างอิงหมายเลขของเฟรมข้อมูล
- ฟิลด์ **Data** เป็นส่วนที่ใช้สำหรับบรรจุข้อมูลที่ต้องการสื่อสาร
- ฟิลด์ **CRC Field** ใช้สำหรับการตรวจสอบความถูกต้องของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2. ฟิลด์ข้อมูลต่างๆ ในการสื่อสารข้อมูลยูเอสบี

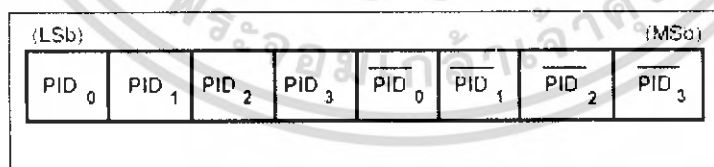
ในโปรโตคอลการสื่อสารยูเอสบีได้มีการกำหนดลักษณะโครงสร้างของข้อมูลที่ใช้ในการสื่อสาร โดยส่วนย่อยที่สุดของโครงสร้างข้อมูลนั้น คือฟิลด์ข้อมูลต่างๆ ซึ่งจะถูกใช้แตกต่างกันไปตามลักษณะหน้าที่ของฟิลด์ข้อมูลนั้นๆ ในการแสดงรูปแบบโครงสร้างข้อมูลของฟิลด์ข้อมูลต่างๆ จะเริ่มต้นจาก LSB (Least Significant Bit) ทางซ้ายและสิ้นสุดด้วย MSB (Most Significant Bit) ที่ทางขวา ชนิดของฟิลด์ข้อมูลต่างๆ ได้แก่

1) ฟิลด์ SYNC (Synchronization)

ฟิลด์ SYNC เป็นส่วนที่ใช้โดยตัวอุปกรณ์ในการจับจังหวะของการสื่อสารข้อมูล ให้สัมพันธ์กับสัญญาณนาฬิกาของตัวอุปกรณ์เอง ข้อมูลในฟิลด์ SYNC จะประกอบด้วยข้อมูล 8 บิต ได้แก่ "00000001" ทำให้เมื่อผ่านการทำ NRZI แล้วจะได้ลักษณะสัญญาณที่สลับไปมาระหว่างสถานะบิต Differential 0 และ Differential 1 ทำให้ตัวอุปกรณ์สามารถจับจังหวะการเปลี่ยนแปลงของบิตข้อมูลได้

2) ฟิลด์ PID (Packet Identifier)

ฟิลด์ PID เป็นตัวบอกตัวอุปกรณ์ที่ได้รับแพ็กเก็ตว่าแพ็กเก็ตที่ได้รับนั้นเป็นแพ็กเก็ตประเภทใด ซึ่งฟิลด์ข้อมูลนี้จะประกอบด้วยข้อมูล 8 บิต โดยจะมีข้อมูลจำนวน 4 บิตที่ระบุชนิดของแพ็กเก็ตและข้อมูลอีก 4 บิตสำหรับการตรวจสอบความถูกต้องของข้อมูล โดยที่ข้อมูลที่ใช้ในการตรวจสอบความถูกต้องนั้น จะได้มาจากการทำ 1's Complement ของข้อมูล 4 บิตในส่วนที่ใช้ระบุชนิดของแพ็กเก็ต โดยรูปแบบของฟิลด์ PID แสดงดังรูปที่ 2.12 และชนิดของแพ็กเก็ตต่างๆ ดังตารางที่ 2.1



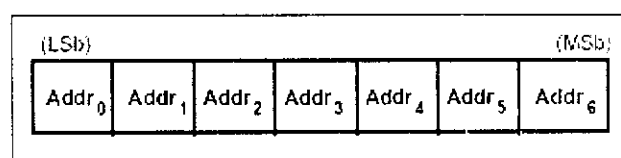
รูปที่ 2.12 แสดงรูปแบบของฟิลด์ PID

ตารางที่ 2.1 แสดงชนิดของแพ็กเก็ตต่างๆ

ประเภท แพ็กเก็ต	ชนิดของ ทรานแซกชัน	ค่าของ PID(3:0)	ความหมายและหน้าที่ ของแพ็กเก็ต
Token	OUT	0001B	ทรานแซกชันสำหรับการส่งข้อมูลจากโฮสต์ไปยังอุปกรณ์
	IN	1001B	ทรานแซกชันสำหรับการส่งข้อมูลจากอุปกรณ์ไปยังโฮสต์
	SOF	0101B	จุดเริ่มต้นของเฟรมข้อมูล
	SETUP	1101B	ทรานแซกชันสำหรับการส่งคำสั่ง ใช้ใน Default Control Pipe
Data	DATA0	0011B	แพ็กเก็ตข้อมูลชนิด DATA0
	DATA1	1011B	แพ็กเก็ตข้อมูลชนิด DATA1
Handshake	ACK	0010B	ข้อมูลที่ได้รับ ไม่มีความผิดพลาด
	NAK	1010B	ข้อมูลที่ได้รับมีความผิดพลาด
	STALL	1110B	เอนพอยน์หยุดทำงานหรือไม่สนับสนุนคำสั่ง
Special	PRE	1100B	ใช้สำหรับการสื่อสารกับอุปกรณ์ความเร็วต่ำ

3) ฟิลด์ ADDR (Address)

ฟิลด์ ADDR เป็นส่วนที่ทำการระบุตัวอุปกรณ์ว่าจะให้อุปกรณ์ตัวใดบนบัสทำงานตอบสนองในเฟรมข้อมูลนั้นๆ ซึ่งแต่ละอุปกรณ์บนบัสจะถูกกำหนดหมายเลขแอดเดรส (Address) ที่ไม่ซ้ำกันในระบบ ส่วนอุปกรณ์ตัวอื่นๆ ที่มีหมายเลขแอดเดรสที่ไม่ตรงกับที่ระบุอยู่ในฟิลด์ ADDR ก็จะไม่ตอบสนองต่อการทำงานเฟรมนั้นๆ ฟิลด์ ADDR ประกอบด้วยข้อมูลจำนวน 7 บิต ซึ่งสามารถอ้างอุปกรณ์ได้ตั้งแต่ 1-127 ส่วนค่าแอดเดรส 0 นั้น ถูกใช้สำหรับเป็นค่า Default Address ของอุปกรณ์ทุกตัว เพื่อใช้ในตอนเริ่มต้นการทำงาน และห้ามกำหนดหมายเลขนี้ให้กับอุปกรณ์ใดๆ ซึ่งรูปแบบของฟิลด์ ADDR แสดงดังรูปที่ 2.13

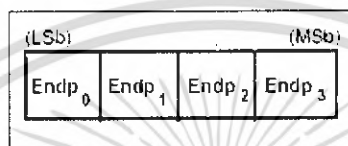


รูปที่ 2.13 แสดงรูปแบบของฟิลด์ ADDR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ฟิลด์ ENDP (Endpoint)

ฟิลด์ ENDP เป็นส่วนที่ใช้อ้างอิงเอนพอย์นของตัวอุปกรณ์ว่าต้องการติดต่อกับเอนพอย์นใดในการสร้างไปป์สื่อสารข้อมูล โดยที่อุปกรณ์ทุกตัวจะต้องมี Default Control Pipe ที่เอนพอย์นหมายเลข 0 (Default Endpoint) และแต่ละอุปกรณ์นั้นอาจจะมีมากกว่า 1 เอนพอย์น โดยที่อุปกรณ์ความเร็วต่ำจะสามารถมีจำนวนเอนพอย์นได้มากที่สุด 3 เอนพอย์น และสำหรับอุปกรณ์ความเร็วเต็มที่สามารมีจำนวนเอนพอย์นได้ 16 เอนพอย์น ส่วนประกอบของฟิลด์ ENDP จะประกอบด้วยข้อมูลจำนวน 4 บิต ดังรูปที่ 2.14



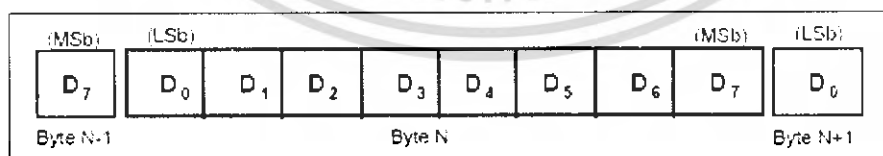
รูปที่ 2.14 แสดงรูปแบบของฟิลด์ ENDP

5) ฟิลด์ Frame Number

ฟิลด์ Frame Number ใช้สำหรับการอ้างอิงของเฟรมข้อมูล โดยใช้หมายเลขเฟรม โดยที่ค่าของหมายเลขเฟรมจะเพิ่มขึ้นทีละ 1 ในทุกๆ เฟรมข้อมูล รูปแบบของฟิลด์ Frame Number จะมีขนาด 11 บิต ซึ่งจะมีค่าตั้งแต่ 0x000 ถึง 0x7FF และเมื่อข้อมูลมีค่าถึง 0x7FF แล้วก็จะเริ่มต้นที่ 0x000 ในเฟรมถัดไป

6) ฟิลด์ Data

ฟิลด์ Data เป็นส่วนที่ใช้สำหรับบรรจุข้อมูลที่ต้องการสื่อสาร สามารถมีขนาดได้ตั้งแต่ 0-1023 ไบต์ โดยที่ขนาดของแพ็กเกจนั้นจะขึ้นอยู่กับปริมาณข้อมูลและชนิดของการสื่อสารข้อมูล รูปแบบของฟิลด์ Data แสดงดังรูปที่ 2.15



รูปที่ 2.15 แสดงรูปแบบของฟิลด์ Data

7) ฟیلด์ CRC

ฟیلด์ CRC ใช้สำหรับการตรวจสอบความถูกต้องของข้อมูล ซึ่งจะถูกคำนวณก่อนที่จะทำการแทรกบิต (Bit Stuffing) และการเข้ารหัส NRZI รูปแบบของฟیلด์ CRC จะมี 2 รูปแบบ ขึ้นอยู่กับข้อมูลที่น่ามาคำนวณค่า CRC ซึ่งรูปแบบของ CRC ได้แก่ CRC สำหรับแฟ้มเกิด Token และแฟ้มเกิด Data

- *Token CRC*

Token CRC มีขนาด 5 บิต ซึ่งครอบคลุมข้อมูลในส่วนฟیلด์ ADDR และฟیلด์ ENDP ของแฟ้มเกิด Token และครอบคลุมข้อมูลในส่วนฟیلด์ Frame Number ในแฟ้มเกิด SOF ในการคำนวณ Token CRC จะใช้รูปแบบของ Polynomial คือ $G(X) = X^5 + X^2 + 1$ และมีค่า Residual เท่ากับ $(01100)_2$

- *Data CRC*

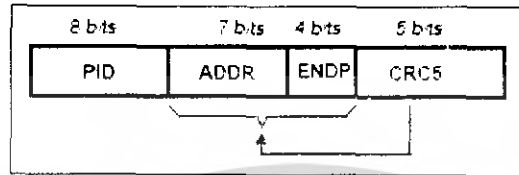
Data CRC มีขนาด 16 บิต ซึ่งครอบคลุมข้อมูลในส่วนฟیلด์ Data เท่านั้น การคำนวณ Data CRC จะใช้ Polynomial คือ $G(X) = X^{16} + X^{15} + X^2 + 1$ และมีค่า Residual เท่ากับ $(1000000000001101)_2$

2.2.3. แฟ้มเกิดต่างๆ ที่ใช้ในการสื่อสารข้อมูลยูเอสบี

แฟ้มเกิดเกิดจากการนำฟیلด์ข้อมูลต่างๆ มารวมกัน ซึ่งในแต่ละชนิดของแฟ้มเกิดก็จะเกิดจากการประกอบจากฟیلด์ข้อมูลที่แตกต่างกัน และแต่ละแฟ้มเกิดก็จะมีหน้าที่ที่แตกต่างกันไป แล้วแต่แฟ้มเกิดและทุกๆ แฟ้มเกิดที่จะมีการส่งจาก โฮสต์จะต้องมีการเพิ่มส่วนหัวแฟ้มเกิดด้วย ฟیلด์ SYNC เสมอ ชนิดของแฟ้มเกิดต่างๆ ได้แก่

1) แพ็กเก็ต Token

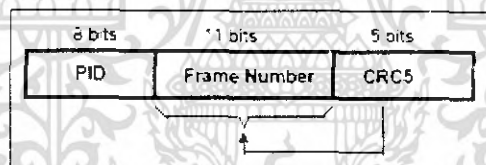
แพ็กเก็ต Token จะประกอบด้วยฟิลด์ PID, ฟิลด์ ADDR, ฟิลด์ ENDP และฟิลด์ Token CRC ดังรูปที่ 2.16 โดยแพ็กเก็ต Token จะใช้สำหรับการบอกกับระบบว่าโฮสต์ต้องการสื่อสาร:พร้อมข้อมูลกับอุปกรณ์ใด



รูปที่ 2.16 แสดงรูปแบบของแพ็กเก็ต Token

2) แพ็กเก็ต SOF (Start-of-Frame)

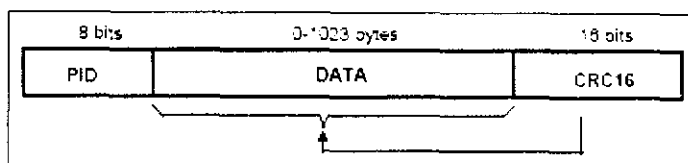
แพ็กเก็ต SOF จะประกอบด้วยฟิลด์ PID, ฟิลด์ Frame Number และฟิลด์ Token CRC ดังรูปที่ 2.17 ในการทำงานปกติแพ็กเก็ต SOF จะถูกส่งในทุกๆ 1 มิลลิวินาที เพื่อเป็นสัญลักษณ์ของจุดเริ่มต้นของเฟรมข้อมูลในแต่ละเฟรมข้อมูลและข้อมูลในฟิลด์ PID จะ เป็นค่าที่ระบุประเภทของแพ็กเก็ตว่าเป็นแพ็กเก็ต SOF



รูปที่ 2.17 แสดงรูปแบบของแพ็กเก็ต SOF

3) แพ็กเก็ต Data

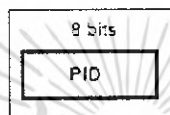
แพ็กเก็ต Data จะประกอบด้วยฟิลด์ PID, ฟิลด์ Data และฟิลด์ Data CRC ดังรูปที่ 2.18 โดยที่รูปแบบของแพ็กเก็ต Data จะมี 2 รูปแบบซึ่งระบุโดย PID โดยจะมีรูปแบบ DATA0 และ DATA1 ทั้งนี้เพื่อช่วยให้เกิดการสื่อสารข้อมูลใหม่ในกรณีที่เกิดความ ผิดพลาดของการสื่อสารข้อมูลได้



รูปที่ 2.18 แสดงรูปแบบของแพ็กเก็ต Data

4) แพ็กเก็ต Handshake

แพ็กเก็ต Handshake จะประกอบด้วยฟิลด์ PID เท่านั้น ใช้สำหรับการรายงานผลของการสื่อสารข้อมูลในแต่ละทรานแซกชันว่าสามารถสื่อสารข้อมูลได้อย่างถูกต้องหรือไม่, การยอมรับหรือปฏิเสธคำสั่งต่างๆ โดยในการส่งแพ็กเก็ต Handshake จะถูกส่งในช่วง Handshake Phase ของแต่ละทรานแซกชัน โดยที่ทรานแซกชันนั้นจะต้องสื่อสารในรูปแบบการสื่อสารที่มีการตรวจสอบความถูกต้องของข้อมูลเท่านั้น รูปแบบของแพ็กเก็ต Handshake แสดงดังรูป 2.19 และนอกจากนี้แพ็กเก็ต Handshake จะมี 3 ชนิดโดยระบุชนิดของแพ็กเก็ต Handshake นั้นๆ โดยฟิลด์ PID ซึ่ง ได้แก่ ACK, NACK และ STALL



รูปที่ 2.19 แสดงตัวอย่างแพ็กเก็ต Handshake

- **ACK**

ใช้สำหรับการบอกว่าสามารถรับข้อมูลได้และข้อมูลมีความถูกต้อง ACK จะถูกส่งจากตัวอุปกรณ์ไปยังโฮสต์ในทรานแซกชัน OUT และถูกส่งจากโฮสต์ไปยังอุปกรณ์ในกรณีที่เป็นทรานแซกชัน IN

- **NAK**

ใช้สำหรับการบอกว่าไม่สามารถรับข้อมูลได้หรือข้อมูลไม่ถูกต้องในทรานแซกชัน OUT ซึ่งจะถูกส่งใน Handshake Phase และการใช้บอกว่าไม่มีข้อมูลที่ต้องการส่งในทรานแซกชัน IN ในการสื่อสารข้อมูลแบบ Interrupt ซึ่งจะถูกส่งใน Data Phase โดยที่ NAK จะใช้สำหรับตัวอุปกรณ์ส่งไปยังโฮสต์เท่านั้น

- **STALL**

ใช้สำหรับให้ตัวอุปกรณ์ส่งไปยังโฮสต์เท่านั้น โดยจะใช้ใน 2 กรณี ได้แก่

เมื่อตัวอุปกรณ์หยุดทำงานเมื่อได้รับข้อมูลใดๆ ก็จะไม่ตอบสนองต่อข้อมูลนั้น และจะทำการส่งแพ็กเก็ต STALL เพื่อป้องกันให้รู้ว่าอุปกรณ์นั้นหยุดการทำงาน ซึ่งการที่จะให้อุปกรณ์เริ่มต้นทำงานนั้น ตัวโฮสต์จะต้องใช้คำสั่งสั่งให้ตัวอุปกรณ์ทำงาน จึงจะเข้าสู่การทำงานปกติ การใช้ STALL ในกรณีนี้ จะทำการส่งใน Handshake Phase ของทรานแซกชัน OUT

ใช้สำหรับทรานแซคชัน SETUP จะใช้เมื่อตัวอุปกรณ์ได้รับคำสั่งจากโฮสต์แต่ไม่สามารถทำงานตามคำสั่งนั้นได้ หรือใช้ในกรณีที่ตัวอุปกรณ์ไม่เข้าใจคำสั่งดังกล่าว โดยจะทำการส่ง STALL ใน Data Phase หรือ Status Phase ของทรานแซคชันและเมื่อเกิดเหตุการณ์ในกรณีนี้แล้ว ตัวอุปกรณ์จะไม่ทำงานจนกว่าจะได้รับทรานแซคชัน SETUP ถัดไป

2.2.4. ทรานแซคชัน

ทรานแซคชัน คือการนำแพ็คเกจแต่ละชนิดมาทำจัดกลุ่มเพื่อให้เกิดการสื่อสารข้อมูลกับตัวอุปกรณ์ โดยที่แต่ละทรานแซคชันก็จะเป็นสื่อสารข้อมูลกับตัวอุปกรณ์ในแต่ละครั้ง

2.2.5. รูปแบบของการสื่อสารข้อมูล

การสื่อสารข้อมูลของยูเอสบีซีนั้นมีหลายรูปแบบซึ่งขึ้นอยู่กับลักษณะการใช้งานที่แตกต่างกันออกไป ซึ่งแต่ละเอนพอยน์ของตัวอุปกรณ์นั้นจะมีรูปแบบการสื่อสารข้อมูลเพียงรูปแบบเดียวเท่านั้น แต่ตัวอุปกรณ์สามารถมีเอนพอยน์ได้หลายเอนพอยน์ โดยแต่ละเอนพอยน์ก็อาจจะมีรูปแบบการสื่อสารข้อมูลที่แตกต่างกันได้ ขึ้นอยู่กับความต้องการของอุปกรณ์นั้นๆ ว่าต้องการรูปแบบการสื่อสารข้อมูลแบบใดบ้าง รูปแบบการสื่อสารข้อมูลของยูเอสบีซีมี 4 รูปแบบ ได้แก่

1) การสื่อสารข้อมูลแบบ Isochronous

การสื่อสารข้อมูลแบบ Isochronous เป็นการสื่อสารข้อมูลที่มีอัตราเร็วของข้อมูลคงที่ โดยจะมีการรับหรือส่งข้อมูลในทุกๆ เฟรมข้อมูล ดังนั้นในการสื่อสารข้อมูลแบบ Isochronous จึงมีความจำเป็นที่จะต้องทำการจองพื้นที่ของทุกเฟรมด้วย และเนื่องจากความต้องการในการสื่อสารข้อมูลด้วยอัตราเร็วคงที่ จึงไม่สามารถส่งข้อมูลซ้ำในกรณีที่ข้อมูลมีความผิดพลาดได้ จึงทำให้การสื่อสารข้อมูลแบบนี้ไม่มีการรับประกันความถูกต้องของข้อมูล

2) การสื่อสารข้อมูลแบบ Interrupt

การสื่อสารข้อมูลแบบ Interrupt มีลักษณะการทำงานโดยโฮสต์จะคอยวนถามข้อมูลเป็นระยะๆ ตามช่วงเวลาที่กำหนดไว้อย่างสม่ำเสมอ หรือเรียกว่า โพลลิง (Polling) โดยสามารถกำหนดระยะเวลาได้ตั้งแต่ ทุกๆ 1 เฟรมจนถึงทุกๆ 255 เฟรมต่อการวนถาม 1 ครั้ง (สำหรับอุปกรณ์ความเร็วต่ำ สามารถกำหนดระยะเวลาสั้นที่สุดคือทุกๆ 10 เฟรมต่อการวนถาม 1 ครั้ง) เมื่อตัวอุปกรณ์รับการวนอ่านข้อมูลจากโฮสต์ ตัวอุปกรณ์ก็จะส่งข้อมูลลงบน Data Phase ในทรานแซกชันแต่ถ้าตัวอุปกรณ์ไม่มีข้อมูลที่ต้องการส่ง ก็จะทำการส่ง NAK แทนข้อมูล ขนาดของข้อมูลที่มากที่สุดที่สามารถส่งได้ในแต่ละครั้งคือ 64 ไบต์ สำหรับอุปกรณ์ความเร็วเต็มที่และ 8 ไบต์สำหรับอุปกรณ์ความเร็วต่ำ แต่ถ้าข้อมูลที่มีขนาดใหญ่กว่าขนาดที่กำหนด ก็จะต้องแบ่งออกเป็นส่วนๆ แล้วทยอยส่ง โดยการส่งข้อมูลที่มีขนาดข้อมูลเท่ากับขนาดที่กำหนดจะทำให้โฮสต์เข้าใจว่าข้อมูลที่ได้รับยังไม่ครบ การสื่อสารข้อมูลแบบ Interrupt มีการรับประกันความถูกต้องของข้อมูล โดยการส่งข้อมูลที่ผิดพลาดในทรานแซกชันถัดไป

3) การสื่อสารข้อมูลแบบ Bulk

การสื่อสารข้อมูลแบบ Bulk เป็นการสื่อสารข้อมูลที่อัตราเร็วของการสื่อสารข้อมูลมีค่าไม่คงที่ โดยขึ้นอยู่กับปริมาณข้อมูลภายในบัฟ ซึ่งการที่จะสามารถสื่อสารข้อมูลในรูปแบบนี้ได้ จะต้องมีส่วนที่ว่างในเฟรมเหลือสำหรับการสื่อสารข้อมูล หากไม่มีพื้นที่เหลือ ก็จะต้องรอจนกว่าจะสามารถส่งข้อมูลได้ ขนาดของข้อมูลที่ใหญ่ที่สุดในการสื่อสารข้อมูลแบบ Bulk คือ 8, 16, 32 และ 64 ไบต์และมีการรับประกันความถูกต้องของข้อมูลเช่นกัน

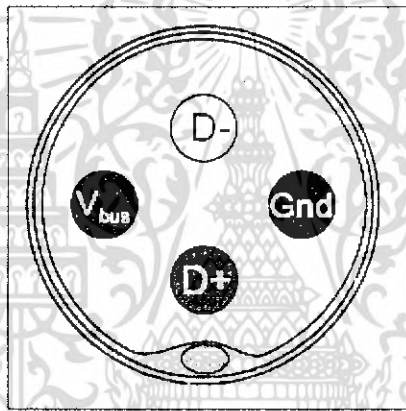
4) การสื่อสารข้อมูลแบบ Control

การสื่อสารข้อมูลแบบ Control ใช้สำหรับการสื่อสารคำสั่งควบคุมต่างๆ ระหว่างโฮสต์กับอุปกรณ์ ซึ่งในการสื่อสารแบบ Control นั้น จะแบ่งช่วงการสื่อสารออกเป็น 3 ส่วนคือ Setup Stage, Status Stage และ Data Stage โดยที่ Data Stage นั้นอาจจะมีหรือไม่มีก็ได้ ขึ้นอยู่กับชนิดของคำสั่งที่ต้องการใช้งานในส่วนของ Data Stage หรือไม่ และในส่วนของ Status Stage ใช้สำหรับการรายงานผลการทำงานต่างๆ ของคำสั่งที่ได้รับว่าสามารถดำเนินการตามคำสั่งได้หรือไม่หรืออย่างไร

2.3. ระดับการทำงานในระดับกายภาพ (Physical Layer)

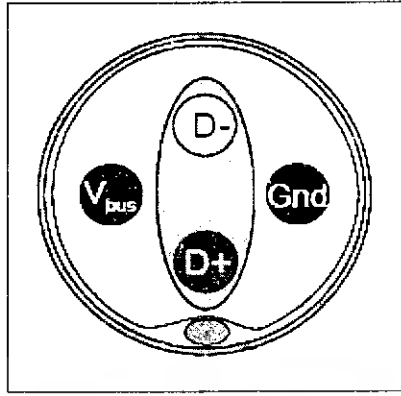
2.3.1. สายสัญญาณยูเอสบี

สายสัญญาณยูเอสบีของอุปกรณ์ความเร็วต่ำและความเร็วสูงจะแตกต่างกัน โดยที่สายสัญญาณสำหรับอุปกรณ์ความเร็วต่ำจะกอบด้วยฉนวนพีวีซี (PVC) หุ้มด้านนอกและชีลด์ (Shield) ด้านในและอาจจะมีชีลด์ ด้านนอกหรือไม่มีก็ได้ ภายในมีสายสัญญาณจำนวน 2 เส้น (D+ และ D-) ที่ไม่มีการบิดเกลียว, สายจ่ายพลังงานไฟฟ้า 2 เส้น (V_{bus} และ Ground) และสายที่ใช้สำหรับการคายกระแสไฟฟ้า (Drain Wire) อีกจำนวน 1 เส้น ลักษณะภายในของสายสัญญาณสำหรับอุปกรณ์ความเร็วเต็มที่แสดงดังรูปที่ 2.20 และความยาวสูงสุดของสายสัญญาณที่สามารถใช้ได้ แสดงดังตารางที่ 2.1



รูปที่ 2.20 แสดงลักษณะของสายสัญญาณสำหรับอุปกรณ์ความเร็วต่ำ

สำหรับสายสัญญาณความเร็วสูงจะมีลักษณะคล้ายกับสายสัญญาณของอุปกรณ์ความเร็วต่ำ โดยจะประกอบด้วยสายสัญญาณ 4 เส้นและสายคายกระแสไฟฟ้าอีก 1 เส้น แต่มีลักษณะแตกต่างกันที่ สายสัญญาณสำหรับอุปกรณ์ความเร็วเต็มที่ต้องมีชีลด์ด้านนอกด้วย และสายสัญญาณจำนวน 2 เส้น (D+ และ D-) จะมีการบิดเกลียว ลักษณะของสายสัญญาณสำหรับอุปกรณ์ความเร็วเต็มที่แสดงดังรูปที่ 2.21 และความยาวสูงสุดของสายสัญญาณที่สามารถใช้ได้ แสดงดังตารางที่ 2.1



รูปที่ 2.21 แสดงลักษณะของสายสัญญาณสำหรับอุปกรณ์ความเร็วเต็มที่

ตารางที่ 2.2 แสดงความยาวสูงสุดของสายสัญญาณที่สามารถใช้ได้

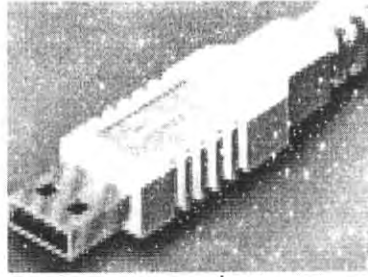
ความล่าช้าของสัญญาณ (นาโนวินาทีต่อเมตร)	ความยาวสายสัญญาณที่ใช้ได้ (เมตร)
9.0	3.3
8.0	3.7
7.0	4.3
6.5	4.6

2.3.2. ตัวเชื่อมต่อสำหรับอุปกรณ์ยูเอสบี

ตัวเชื่อมต่อสำหรับอุปกรณ์ยูเอสบีมี 2 ลักษณะ ได้แก่ ตัวเสียบ (Plug) และตัวรับ (Receptacle) โดยที่รูปแบบของสายสัญญาณ จะถูกกำหนดไว้ 2 รูปแบบ ซึ่งแต่ละรูปแบบจะถูกใช้ในลักษณะการทำงานที่แตกต่างกัน รูปแบบของตัวเชื่อมต่ออุปกรณ์ยูเอสบีได้แก่

1) ตัวเชื่อมต่ออุปกรณ์ยูเอสบีชนิดเอ (A)

ตัวเชื่อมต่ออุปกรณ์ยูเอสบีชนิดเอ ที่เป็นตัวเสียบนั้น จะใช้เป็นพอร์ต์ฮับสตรีมสำหรับตัวอุปกรณ์ ที่จะใช้เชื่อมต่ออุปกรณ์นั้นไปยังโฮสต์ซึ่งเป็นตัวเชื่อมต่อชนิดเอแบบที่เป็นตัวรับ ดังนั้น อุปกรณ์ยูเอสบีทุกตัว จะต้องมีส่วนเสียบชนิดเอ เป็นตัวเชื่อมต่อกับโฮสต์ โดยลักษณะของตัวเชื่อมต่อยูเอสบีชนิดเอ แบบตัวเสียบแสดงดังรูปที่ 2.22 และแบบตัวรับแสดงดังรูปที่ 2.23



รูปที่ 2.22 แสดงลักษณะของตัวเชื่อมต่อชนิดเอ แบบตัวเสียบ

รูปที่ 2.23 แสดงลักษณะของตัวเชื่อมต่อชนิดเอ แบบตัวรับ

2) ตัวเชื่อมต่ออุปกรณ์ยูเอสบี ชนิดบี

ตัวเชื่อมต่ออุปกรณ์ยูเอสบีชนิดบี ใช้สำหรับอุปกรณ์ยูเอสบีที่ต้องใช้สายสัญญาณในการเชื่อมต่อกับโฮสต์ โดยที่ตัวเชื่อมต่อชนิดบีแบบตัวรับ จะอยู่ติดกับตัวอุปกรณ์ ซึ่งจะต้องเชื่อมต่อกับตัวเชื่อมต่อชนิดบีแบบตัวเสียบ ที่ปลายของสายสัญญาณด้านหนึ่ง ซึ่งลักษณะของตัวเชื่อมต่อยูเอสบีชนิดบี แบบตัวเสียบแสดงดังรูปที่ 2.24 และแบบตัวรับแสดงดังรูปที่ 2.25



รูปที่ 2.24 แสดงลักษณะของตัวเชื่อมต่อชนิดบี แบบตัวเสียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 แสดงลักษณะของตัวเชื่อมต่อชนิดบี แบบตัวรับ

2.3.3. ระดับสัญญาณแบบผลต่าง (Differential Signaling)

การสื่อสารข้อมูลด้วยยูเอสบีเอ็น จะใช้การส่งข้อมูลโดยใช้สัญญาณทางไฟฟ้า ซึ่งจะมีการใช้รูปแบบการส่งสัญญาณแบบผลต่าง การแปลความหมายของสัญญาณนั้นจะใช้ระดับความแตกต่างของสัญญาณในการพิจารณา

2.3.4. สถานะต่างๆ ของบัสยูเอสบี

ในการควบคุมทำงานต่างๆ ของอุปกรณ์ยูเอสบีจะใช้การส่งคำสั่งควบคุมผ่านทาง Default Control Pipe เป็นหลัก แต่อย่างไรก็ตาม ยังมีการควบคุมการทำงานบางอย่างที่จะใช้ระดับแรงดันรูปแบบของสัญญาณไฟฟ้าในการควบคุมการทำงานของอุปกรณ์ ซึ่งจะกำหนดเป็นสถานะบัสต่างๆ ดังตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

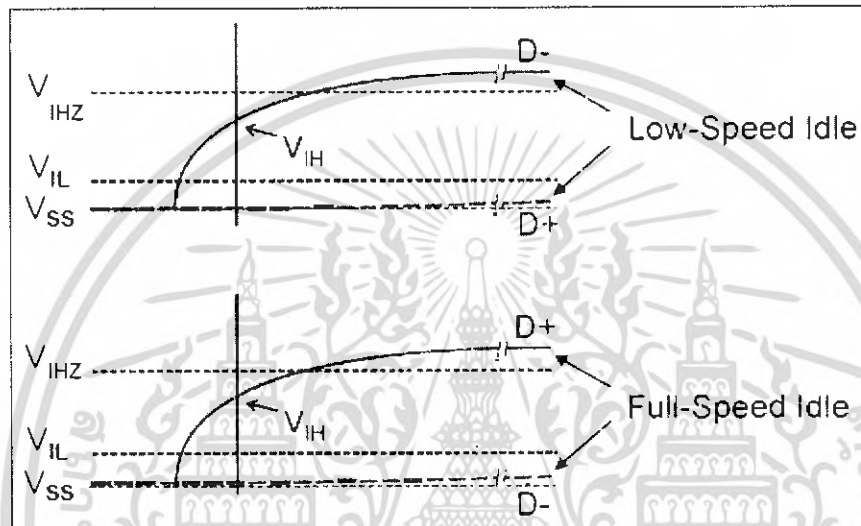
ตารางที่ 2.3 แสดงสถานะต่างๆ ของบัส USB

สถานะบัส	ลักษณะและระดับสัญญาณ		
	ที่อุปกรณ์ที่สร้าง สัญญาณ	ที่อุปกรณ์ปลายทาง	
		ความต้องการ	ยอมรับได้
ว่าง (Idle)			
ความเร็วต่ำ	-	$D+ > V_{IHZ} (min)$ และ $D+ < V_{IL} (max)$	$(D+) - (D-) > 200mV$
ความเร็วสูง	-	$D+ > V_{IHZ} (min)$ และ $D- < V_{IL} (max)$	$(D-) - (D+) > 200mV$
SE0	$D+ และ D- < V_{OL} (max)$	$D+ และ D- < V_{IL} (max)$	$D+ และ D- < V_{IH} (min)$
Differential "0"	$D- > V_{OH} (min)$ และ $D+ < V_{OL} (max)$	$(D-) - (D+) > 200mV$ และ $D- > V_{IH} (min)$	$(D-) - (D+) > 200mV$
Differential "1"	$D+ > V_{OH} (min)$ และ $D- < V_{OL} (max)$	$(D+) - (D-) > 200mV$ และ $D+ > V_{IH} (min)$	$(D+) - (D-) > 200mV$
J State			
ความเร็วต่ำ	Differential "1"		Differential "1"
ความเร็วสูง	Differential "0"		Differential "0"
K State			
ความเร็วต่ำ	Differential "0"		Differential "0"
ความเร็วสูง	Differential "1"		Differential "1"
Reset	$D+ และ D- < V_{OL} (max)$ เป็นเวลา $\geq 10 ms$	$D+ และ D- < V_{IL} (max)$ เป็นเวลา $\geq 10 ms$	$D+ and D- < V_{IL} (max)$ เป็นเวลา $\geq 2.5 \mu s$
Resume	สถานะบัส K		สถานะบัส K
Connect	-	Idle เป็นเวลา $\geq 2 ms$	Idle เป็นเวลา $\geq 2.5 \mu s$
Disconnect	-	SE0 เป็นเวลา $> 2.5 \mu s$	
SOP	เปลี่ยนสถานะจาก Idle เป็น K state		
EOP	$SE0 \geq 1 \text{ bit time}$ และตามด้วย J state for 1 bit time		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) สถานะบัสว่าง (Idle State)

สถานะบัสว่าง เป็นสถานะที่ใช้ในกรณีที่ไม่มีข้อมูลที่ต้องการสื่อสารบนบัสยูเอสบีลักษณะของสัญญาณนั้น จะขึ้นอยู่กับชนิดของอุปกรณ์ว่าเป็นอุปกรณ์ความเร็วต่ำหรืออุปกรณ์ความเร็วเต็มที่จะมีลักษณะสัญญาณที่ตรงกันข้ามกัน โดยถ้าเป็นอุปกรณ์ความเร็วเต็มที่จะมีลักษณะสัญญาณ D+ มีค่ามากกว่าหรือเท่ากับ V_{IHZ} และ D- มีค่าน้อยกว่าหรือเท่ากับ V_{IL} ลักษณะสัญญาณของสถานะบัสว่างแสดงดังรูปที่ 2.26



รูปที่ 2.26 แสดงลักษณะสัญญาณสถานะบัสว่าง

2) สถานะบัส Single-Ended 0 (SE0)

สถานะบัส SE0 เป็นสถานะของบัสที่นำไปใช้ในการอ้างอิงของสถานะบัสอื่นๆ ซึ่งในสถานะบัส SE0 นี้ จะมีลักษณะสัญญาณคือ สัญญาณ D+ และ D- จะต้องน้อยกว่า V_{OL} (สูงสุด)

3) สถานะบัส Differential 0

สถานะบัส Differential 0 มีลักษณะสัญญาณ D- ที่มากกว่า V_{OH} (ต่ำสุด) และสัญญาณ D+ ที่น้อยกว่า V_{OL} (สูงสุด)

4) สถานะบัส Differential 1

สถานะบัส Differential 1 มีลักษณะสัญญาณ D+ ที่มากกว่า V_{OH} (ต่ำสุด) และสัญญาณ D- ที่น้อยกว่า V_{OL} (สูงสุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) สถานะบัส J (J State)

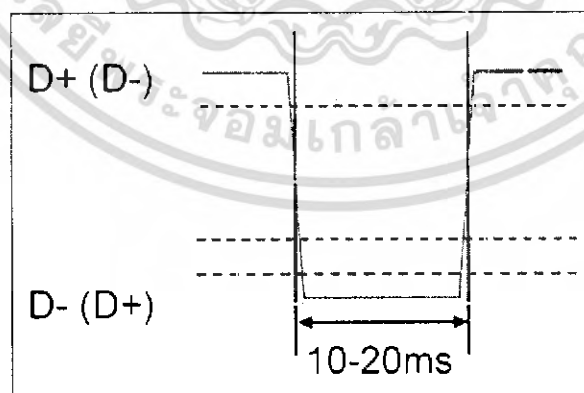
ลักษณะสัญญาณของสถานะบัส J จะแตกต่างกันระหว่างอุปกรณ์ความเร็วต่ำและอุปกรณ์ความเร็วเต็มที่ โดยที่อุปกรณ์ความเร็วต่ำความเร็วต่ำจะมีลักษณะสัญญาณเป็นแบบ Differential 0 ในขณะที่อุปกรณ์ความเร็วเต็มที่จะเป็นลักษณะสัญญาณแบบ Differential 1 และสถานะบัส J นั้นจะเป็นตัวแทนสำหรับบิตข้อมูล '1' จำนวน 1 บิต

6) สถานะบัส K (K State)

ลักษณะสัญญาณของสถานะบัส K สำหรับอุปกรณ์ความเร็วต่ำจะมีลักษณะเป็น Differential 1 และสำหรับอุปกรณ์ความเร็วเต็มที่จะมีลักษณะเป็นแบบ Differential 0 และสถานะบัส K นั้นจะเป็นตัวแทนสำหรับบิตข้อมูล '0' จำนวน 1 บิต

7) สถานะบัสรีเซ็ต (Reset State)

สถานะบัสรีเซ็ตจะทำให้ตัวอุปกรณ์มีการเริ่มต้นการทำงานใหม่ ซึ่งจะถูกใช้ในหลายๆ ครั้งที่มีการตรวจพบการเชื่อมต่อของอุปกรณ์เข้ากับพอร์ต เพื่อให้อุปกรณ์อยู่ในสภาวะเริ่มต้นและพร้อมที่จะรับคำสั่งจากโฮสต์ ลักษณะของสัญญาณในสถานะบัสรีเซ็ตจะต้องทำให้บัสอยู่ในสถานะ SEO และคงสถานะนั้นไว้เป็นเวลาตั้งแต่ 10 มิลลิวินาทีขึ้นไป สำหรับการต่อเชื่อมกับอุปกรณ์ยูเอสบี โดยตรง แต่ถ้าเป็นการต่อเชื่อมกับฮับ จะคงสถานะบัส SEO ตั้งแต่ 50 มิลลิวินาทีขึ้นไป ทั้งนี้เพื่อให้สามารถมั่นใจว่าตัวอุปกรณ์ได้ทำงานในกระบวนการรีเซ็ตจนเสร็จสมบูรณ์แล้ว ลักษณะสัญญาณของสถานะบัสรีเซ็ต แสดงดังรูปที่ 2.27



รูปที่ 2.27 แสดงลักษณะสัญญาณของสถานะบัสรีเซ็ต

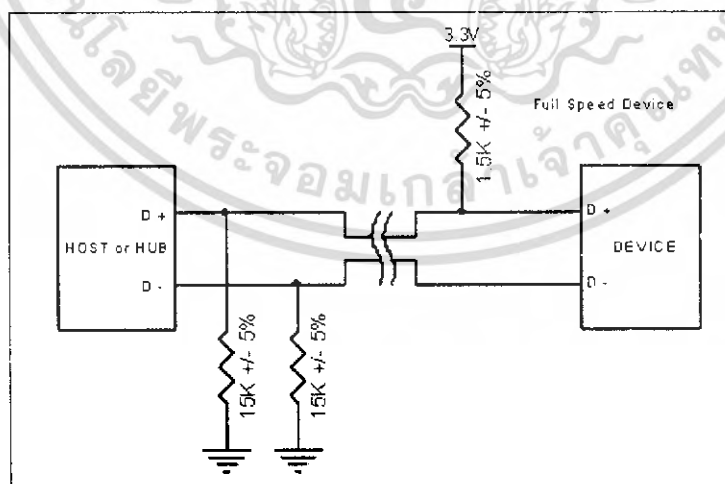
8) สถานะบัลกลับสู่การทำงานปกติ (Resume State)

การกลับสู่สถานะการทำงานปกติ จะถูกใช้สำหรับอุปกรณ์ที่อยู่ในสถานะหยุดทำงานชั่วคราวให้กลับมาสู่สถานะการทำงานปกติ ลักษณะสัญญาณสำหรับการกลับสู่การทำงานปกติ จะต้องให้บัสอยู่ในสถานะ K อย่างน้อย 20 มิลลิวินาที โดยปกติแล้วการสร้างสถานะการกลับสู่การทำงานปกติ จะถูกสร้างโดยโฮสต์แต่สำหรับบางอุปกรณ์ที่สนับสนุนการทำงานแบบ Remote Wakeup ก็สามารถที่จะสร้างสถานะบัลกลับสู่การทำงานปกติได้เช่นกัน

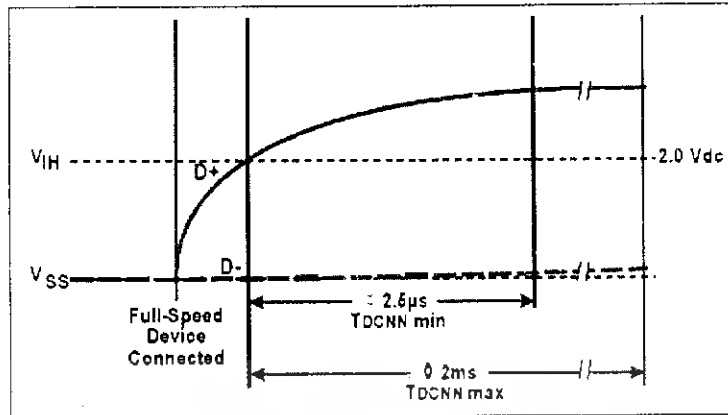
9) สถานะบัลเชื่อมต่ออุปกรณ์ (Connect State)

การตรวจสอบการเชื่อมต่อและความเร็วของตัวอุปกรณ์ จะมีวิธีในการตรวจสอบ โดยการนำตัวต้านทานมาพูลดาวน์ (Pull-Down) สายสัญญาณ D+ และ D- เข้ากับกราวด์ ดังนั้นแรงดันของสายสัญญาณทั้ง 2 เส้นขณะที่ไม่มีอุปกรณ์เชื่อมต่อจะมีระดับแรงดันเท่ากับกราวด์ ซึ่งก็คือ 0 โวลต์

สำหรับอุปกรณ์ความเร็วเต็มที่จะมีการต่อตัวต้านทานพูลอัพ (Pull-Up) กับแหล่งจ่ายแรงดันไฟฟ้ากับตัวอุปกรณ์เข้ากับสายสัญญาณ D+ ซึ่งแสดงดังรูปที่ 2.28 ซึ่งเมื่อเชื่อมต่อเข้ากับพอร์ตยูเอสบีแล้วจะทำให้สายสัญญาณ D+ มีระดับแรงดันที่สูงขึ้นและจะทำให้บัสอยู่ในสถานะบัลสว่าง (ของอุปกรณ์ความเร็วเต็มที่) ซึ่งถ้ายังคงสถานะบัลสว่างเป็นเวลาตั้งแต่ 2 มิลลิวินาทีขึ้นไป ก็จะทำให้โฮสต์สามารถตรวจสอบการเชื่อมต่อและความเร็วของอุปกรณ์ได้ ลักษณะสัญญาณของการเชื่อมต่ออุปกรณ์ความเร็วเต็มที่ แสดงดังรูปที่ 2.29

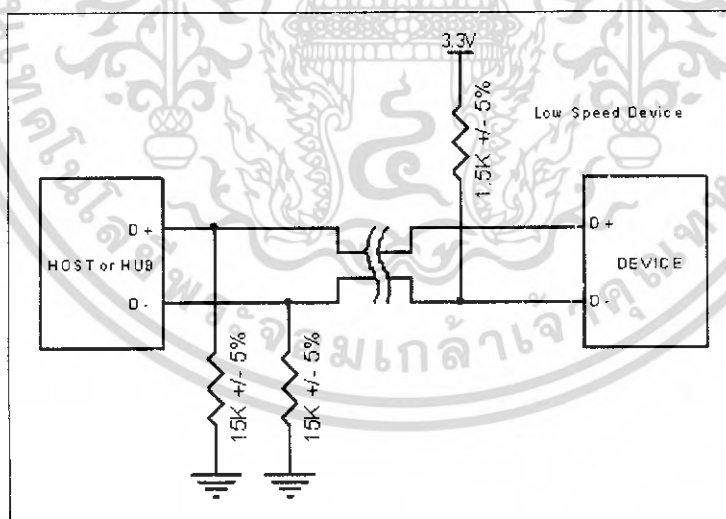


รูปที่ 2.28 แสดงลักษณะการต่อตัวต้านทานพูลอัพของอุปกรณ์ความเร็วเต็มที่



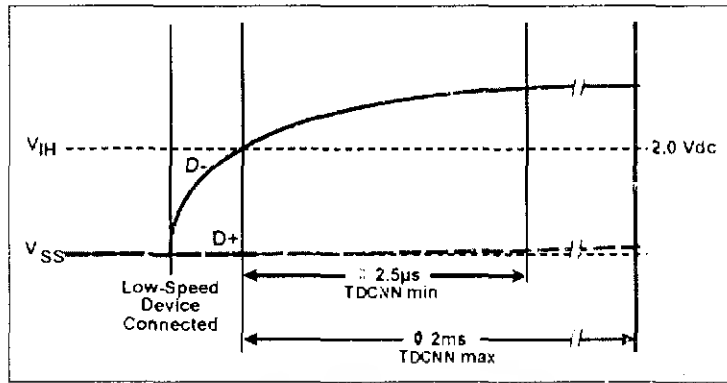
รูปที่ 2.29 แสดงลักษณะของสัญญาณที่เกิดจากการเชื่อมต่ออุปกรณ์ความเร็วเต็มที่

สำหรับอุปกรณ์ความเร็วต่ำจะทำให้หลักการเช่นเดียวกับอุปกรณ์ความเร็วเต็มที่ แต่แตกต่างกันตรงที่อุปกรณ์ความเร็วต่ำจะทำการต่อตัวต้านทานพูลอัพไว้กับสายสัญญาณ D- แทน ซึ่งแสดงดังรูปที่ 2.30 ซึ่งเมื่อทำการเชื่อมต่ออุปกรณ์เข้ากับพอร์ตแล้ว จะทำให้สายสัญญาณ D- มีระดับแรงดันที่สูงขึ้นและจะทำให้บัสอยู่ในสถานะบัสว่าง (ของอุปกรณ์ความเร็วต่ำ) ซึ่งถ้ายังคงสถานะบัสว่างเป็นเวลาดังแต่ 2 มิลลิวินาทีขึ้นไป ก็จะทำให้โฮสต์สามารถตรวจสอบการเชื่อมต่อและความเร็วของอุปกรณ์ได้ ลักษณะสัญญาณของการเชื่อมต่ออุปกรณ์ความเร็วต่ำ แสดงดังรูปที่ 2.31



รูปที่ 2.30 แสดงลักษณะการต่อตัวต้านทานพูลอัพ ของอุปกรณ์ความเร็วต่ำ

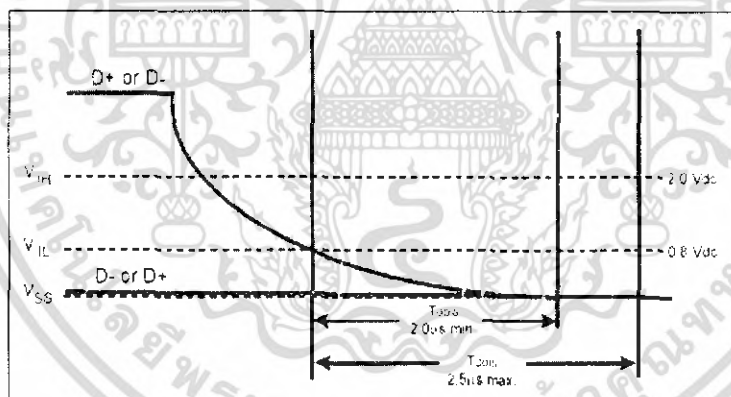
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 แสดงลักษณะของสัญญาณที่เกิดจากการเชื่อมต่ออุปกรณ์ความเร็วต่ำ

10) สถานะบัสถอดอุปกรณ์ (Disconnect State)

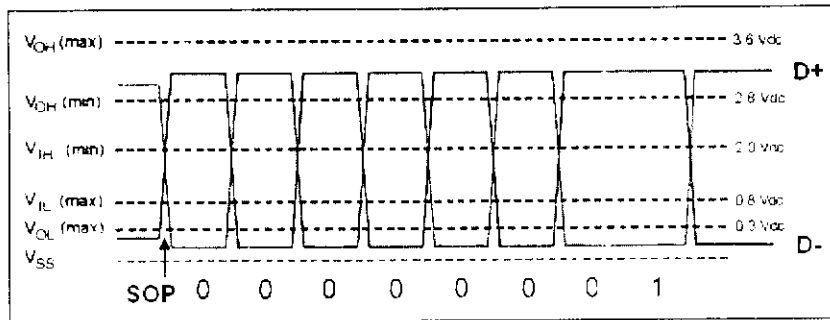
เมื่อมีการถอดต่ออุปกรณ์ออกจากพอร์ต จะมีผลทำให้ระดับแรงดันของสายสัญญาณ D+ (ในกรณีเป็นอุปกรณ์ความเร็วเต็มที่) และ D- (ในกรณีอุปกรณ์ความเร็วต่ำ) มีระดับแรงดันที่ต่ำลง จนทำให้บัสอยู่ในสถานะ SEO ซึ่งถ้าบัสอยู่ในสถานะนี้เป็นเวลาตั้งแต่ 2.5 ไมโครวินาทีขึ้นไป จะทำให้โฮสต์สามารถตรวจว่ามีการถอดอุปกรณ์ออกจากพอร์ต ลักษณะของสัญญาณที่เกิดจากการถอดอุปกรณ์ แสดงดังรูปที่ 2.32



รูปที่ 2.32 แสดงลักษณะของสัญญาณที่เกิดจากการถอดอุปกรณ์ออกจากพอร์ต

11) สถานะบัส SOP (Start-Of-Packet)

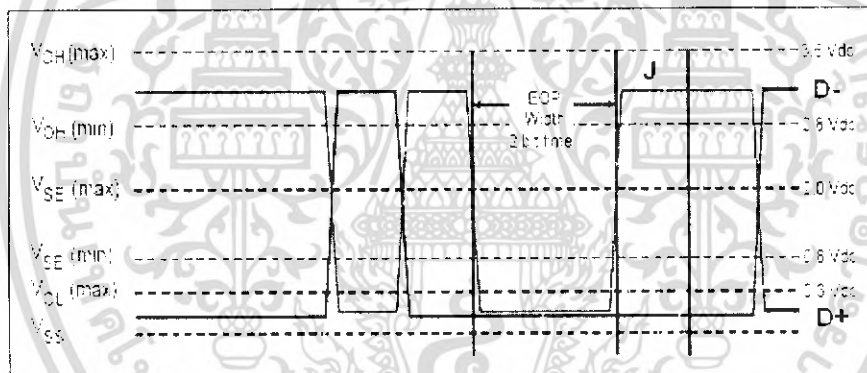
สถานะบัส SOP ใช้สำหรับการบอกให้ตัวอุปกรณ์ที่ต่ออยู่บนบัสให้รู้ว่าการเริ่มต้นการส่งข้อมูลแพ็กเก็ตใหม่ โดยสถานะบัส SOP จะเกิดขึ้นจากการข้อมูลบิตแรกของฟิลด์ SYNC ของทุกๆ แพ็กเก็ต (กล่าวภายหลังในหัวข้อชนิดของแพ็กเก็ต) โดยลักษณะของสัญญาณนั้นจะมีการเปลี่ยนสถานะของบัสจากสถานะบัสว่างเป็นสถานะ K ตัวอย่างของสถานะบัส SOP ของอุปกรณ์ความเร็วเต็มที่แสดงดังรูปที่ 2.33



รูปที่ 2.33 แสดงลักษณะสัญญาณของสถานะบิต SOP ของอุปกรณ์ความเร็วเต็มที

12) สถานะบิต EOP (End-Of-Packet)

สถานะบิต EOP ใช้สำหรับการบอกอุปกรณ์ว่าสิ้นสุดข้อมูลของแพ็กเก็ตแล้ว โดยลักษณะของสัญญาณของบิตจะต้องอยู่ในสถานะ SE0 เป็นเวลาของข้อมูลจำนวน 2 บิต และตามด้วยสถานะ J อีกเป็นเวลา 1 บิต ลักษณะสัญญาณของสถานะบิต EOP ของอุปกรณ์ความเร็วเต็มที แสดงดังรูปที่ 2.34



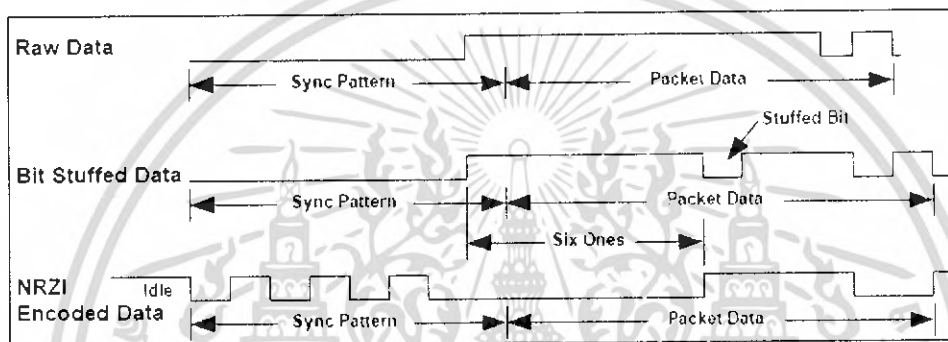
รูปที่ 2.34 แสดงลักษณะสัญญาณของสถานะบิต EOP ของอุปกรณ์ความเร็วเต็มที

2.3.5. การเข้ารหัสและถอดรหัสสัญญาณข้อมูลแบบ NRZI

ในการส่งข้อมูลของยูเอสบีนั้น ไม่ได้ทำการส่งตัวข้อมูลนั้นอย่างแท้จริง แต่เป็นการนำข้อมูลที่ต้องการส่งมาเข้ารหัสสัญญาณ เพื่อให้สามารถส่งผ่านไปยังสายสัญญาณได้ (Line Coding) ซึ่งในข้อกำหนดของยูเอสบีเวอร์ชัน 1.1 การเข้ารหัสสัญญาณข้อมูลจะใช้รูปแบบของ NRZI ในการเข้ารหัส และในทางกลับกัน เมื่อมีการรับข้อมูลจากสายสัญญาณ ก็จะต้องมีการถอดรหัสสัญญาณนั้นให้กลับมาเป็นข้อมูลเดิม ตัวอย่างการเข้ารหัสสัญญาณข้อมูลด้วย NRZI แสดงดังรูปที่ 2.35

2.3.6. การแทรกบิต (Bit Stuffing)

ก่อนที่ข้อมูลจะถูกเข้ารหัสด้วย NRZI จะต้องผ่านขั้นตอนการแทรกบิตก่อน โดยการทำงานของ การแทรกบิตนั้น จะทำการตรวจสอบหาข้อมูล 1 ในข้อมูลเดิม ว่ามีความต่อเนื่องกันของข้อมูล 1 ที่มากกว่า 5 ตัวหรือไม่ ถ้ามีการตรวจพบว่าข้อมูลเดิมมีข้อมูล 1 ต่อเนื่องกันมากกว่า 5 ตัว จะมีการ แทรกบิตข้อมูล 0 ระหว่างข้อมูล 1 ตัวที่ 5 และ 6 และในทุกๆ ครั้งที่มีการตรวจพบข้อมูล 1 ที่ มากกว่า 5 ตัว ก็จะมีการแทรกข้อมูล 0 ลงไปทุกครั้ง ทั้งนี้ เพื่อการป้องกันการเกิดปัญหาการ สูญเสียจังหวะการสื่อสารข้อมูล (Loss Synchronization) ตัวอย่างการทำการแทรกข้อมูล แสดงดัง รูปที่ 2.35



รูปที่ 2.35 แสดงตัวอย่างการเข้ารหัสข้อมูล NRZI และการแทรกบิต

2.4. การตั้งค่าการทำงาน (Configuration)

ในการทำงานของอุปกรณ์ยูเอสบีซีผู้ใช้งานสามารถจะเชื่อมต่ออุปกรณ์หรือถอดอุปกรณ์ออกในขณะใดๆ ก็ได้ ดังนั้น จึงเป็นหน้าที่ของโฮสต์ที่จะต้องคอยตรวจสอบสถานะของบัสเป็นระยะๆ เพื่อให้สามารถทราบได้ว่ามีการเชื่อมต่อหรือถอดอุปกรณ์จากระบบ ในกระบวนการทำงานของโฮสต์เมื่อมีการตรวจพบอุปกรณ์ จะต้องมีการตั้งค่าการทำงานให้กับอุปกรณ์นั้นๆ ซึ่งประกอบด้วยหลายขั้นตอน ได้แก่

2.4.1. การรีเซ็ตตัวอุปกรณ์

ในขั้นตอนแรก จะต้องมีการรีเซ็ตอุปกรณ์เพื่อให้อุปกรณ์มีการทำงานในสภาวะเริ่มต้น ซึ่งจะทำให้หมายเลขแอดเดรสของอุปกรณ์เป็นหมายเลข 0 ซึ่งเป็น Default Address ของอุปกรณ์ทุกๆ ตัว ในการรีเซ็ตอุปกรณ์นั้น โฮสต์จะใช้คำสั่ง SET_PORT_RESET_FEATURE ของฮับ เพื่อให้ฮับทำการสร้างสัญญาณสถานะบัสรีเซ็ต เป็นการสั่งให้ตัวอุปกรณ์รีเซ็ต และเมื่ออุปกรณ์รีเซ็ตเรียบร้อยแล้ว จะมีการอ่านค่าดีไวซ์เดสคริปเตอร์ (Device Descriptor) เพื่อทำการตรวจสอบความต้องการทรัพยากรของอุปกรณ์ (ดีไวซ์เดสคริปเตอร์จะกล่าวถึงภายหลัง)

2.4.2. การกำหนดหมายเลขแอดเดรสให้กับตัวอุปกรณ์

ในการกำหนดหมายเลขแอดเดรส ให้กับตัวอุปกรณ์จะใช้คำสั่ง SET_ADDRESS โดยหมายเลขแอดเดรสที่กำหนดนั้น จะต้องไม่ซ้ำกับอุปกรณ์อื่นๆ ในระบบ

2.4.3. การเลือกโหมดการทำงานของอุปกรณ์

หลังจากที่มีการกำหนดหมายเลขแอดเดรสให้กับตัวอุปกรณ์แล้ว จะต้องกำหนดโหมดการทำงานของอุปกรณ์ว่าจะให้อุปกรณ์ทำงานในโหมดใด ซึ่งในบางอุปกรณ์อาจจะมีโหมดการทำงานเพียงโหมดเดียว และบางอุปกรณ์อาจจะมีโหมดการทำงานที่มากกว่า 1 โหมดการทำงาน ซึ่งจะสามารถทราบได้จากดีไวซ์เดสคริปเตอร์และในการเลือกโหมดการทำงาน จะใช้คำสั่ง SET_CONFIGURATION ในการเลือกโหมดการทำงาน

2.5. การตอบสนองต่อคำร้องขอ (Request Response)

อุปกรณ์ยูเอสบีทุกๆ ตัวจะต้องมีการตอบสนองต่อคำร้องขอของโฮสต์บน Default Control Pipe ด้วยรูปแบบการสื่อสารแบบ Control Transfer โดยคำร้องขอจะถูกบรรจุอยู่ใน SETUP Packet ที่มีขนาดคงที่ 8 ไบต์ และเมื่อโฮสต์ ต้องการร้องขอข้อมูลการตั้งค่าจากตัวอุปกรณ์ยูเอสบีจะต้องทำการส่งคำร้องขอไปยังตัวอุปกรณ์ยูเอสบีที่มีหมายเลขแอดเดรสและเอนพอยน์ ที่ต้องการ รูปแบบและรายละเอียดของคำร้องขอ แสดงดังตารางที่ 2.4

ตารางที่ 2.4 แสดงโครงสร้างของคำร้องขอ

ตำแหน่ง	ชื่อฟิลด์ข้อมูล	ขนาด	ความหมายข้อมูล
0	bmRequestType	1	ลักษณะของคำร้องขอ โดย D7: 0 = โฮสต์ไปยังอุปกรณ์ 1 = อุปกรณ์ไปยังโฮสต์ D6...5: รูปแบบ 0 = มาตรฐาน 1 = คำสั่งเฉพาะคลาสอุปกรณ์ 2 = คำสั่งเฉพาะผู้ผลิต 3 = ไม่ใช้งาน D4...0: ผู้รับ 0 = ตัวอุปกรณ์ 1 = อินเตอร์เฟส 2 = เอนพอยน์ 3 = อื่นๆ 4...31 = ไม่ใช้งาน
1	bRequest	1	ระบุคำร้องขอ
2	wValue	2	ค่าของข้อมูลต่างๆ
4	wIndex	2	ค่าอินเด็กซ์หรือออฟเซต
6	wLength	2	จำนวนไบต์ของข้อมูลที่ต้องการรับหรือส่ง

- **bmRequestType**

เป็นส่วนที่ระบุรายละเอียดและลักษณะของคำร้องขอ โดยส่วนสำคัญคือ การบอกทิศทางของข้อมูลในกรณีที่คำร้องขอนั้นๆ จำเป็นต้องมีข้อมูลเพิ่มเติม ซึ่งจะใช้บิตที่ 7 เป็นส่วนในการบอกทิศทาง แต่อย่างไรก็ตาม ถ้า wLength ที่ใช้บอกความยาวของข้อมูลเป็น 0 ข้อมูลในบิตที่ 7 ดังกล่าว ก็จะถูกละทิ้ง

- **bRequest**

เป็นส่วนที่ระบุชนิดของคำร้องขอนั้นๆ ว่าเป็นการร้องขอสิ่งใดจากตัวอุปกรณ์ยูเอสบี ซึ่งกลุ่มชนิดของคำร้องขอจะแตกต่างกันไปตามแต่ละลักษณะและรายละเอียดของชนิดของคำร้องขอนั้นๆ

- *wValue*

ใช้สำหรับการส่งค่าพารามิเตอร์ต่างๆ ในการส่งคำร้องขอจากโฮสต์ไปยังอุปกรณ์ยูเอสบีซึ่งขึ้นอยู่กับชนิดของคำร้องขอต่างๆ ว่าคำร้องขอชนิดใด ต้องการพารามิเตอร์อะไรบ้าง

- *wIndex*

ใช้สำหรับการส่งค่าพารามิเตอร์ต่างๆ ในการส่งคำร้องขอจากโฮสต์ไปยังอุปกรณ์ยูเอสบีซึ่งขึ้นอยู่กับชนิดของคำร้องขอต่างๆ ว่าคำร้องขอชนิดใด ต้องการพารามิเตอร์อะไรบ้าง

แต่สำหรับ *wIndex* จะถูกใช้สำหรับการอ้างถึงหมายเลขเอนพอยน์และหมายเลขอินเตอร์เฟซ ด้วยรูปแบบของ *wIndex* สำหรับการอ้างถึงเอนพอยน์ จะแสดงดังรูปที่ 2.36 และรูปแบบของการอ้างถึงอินเตอร์เฟซแสดงดังรูปที่ 2.37

D7	D6	D5	D4	D3	D2	D1	D0
Direction				Reserved (Reset to zero)			
Endpoint Number				Endpoint Number			
D15	D14	D13	D12	D11	D10	D9	D8
Reserved (Reset to zero)							

รูปที่ 2.36 แสดงรูปแบบของการอ้างเอนพอยน์ด้วย *wIndex*

D7	D6	D5	D4	D3	D2	D1	D0
Interface Number							
D15	D14	D13	D12	D11	D10	D9	D8
Reserved (Reset to zero)							

รูปที่ 2.37 แสดงรูปแบบของการอ้างอินเตอร์เฟซด้วย *wIndex*

- *wLength*

ใช้สำหรับการกำหนดความยาวของข้อมูลที่จำเป็นต้องใช้เพิ่มเติมสำหรับคำร้องขอ โดยตัวอุปกรณ์ยูเอสบีจะต้องส่งข้อมูลที่มีความยาวไม่เกินจากที่กำหนดไว้ ถ้า *wLength* มีค่าเป็น 0 ก็จะหมายถึง ไม่มีข้อมูลที่ต้องการเพิ่มเติมสำหรับคำร้องขอนั้นๆ

2.6. เดสคริปเตอร์ (Descriptors)

เดสคริปเตอร์ คือส่วนที่อธิบายการทำงานต่างๆ ของตัวอุปกรณ์ยูเอสบี โดยเดสคริปเตอร์จะมีหลายประเภท โดยแต่ละประเภทรัน ก็จะใช้สำหรับการอธิบายการทำงานของอุปกรณ์ในส่วนต่างๆ กัน ซึ่งได้แก่

2.6.1. ดีไวซ์เดสคริปเตอร์ (Device Descriptor)

ใช้สำหรับเก็บข้อมูลต่างๆ ไปของตัวอุปกรณ์ตัวนั้นๆ ซึ่งจะมีข้อมูลเกี่ยวกับการตั้งค่าสำหรับตัวอุปกรณ์ โดยที่ตัวอุปกรณ์ยูเอสบีนั้นจะมีดีไวซ์เดสคริปเตอร์ได้เพียงตัวเดียวเท่านั้น รูปแบบของดีไวซ์เดสคริปเตอร์ แสดงดังตารางที่ 2.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 แสดงโครงสร้างข้อมูลของดีไวซ์เดสคริปเตอร์

ตำแหน่ง	ชื่อ Field ข้อมูล	ขนาด	ความหมายข้อมูล
0	bLength	1	ขนาดของเดสคริปเตอร์
1	bDescriptorType	1	ค่าคงที่ที่บอกว่าเป็นดีไวซ์เดสคริปเตอร์
2	bcdUSB	2	ค่าของเวอร์ชันของยูเอสบีที่ตัวอุปกรณ์สนับสนุน
4	bDeviceClass	1	คลาสของอุปกรณ์นั้นๆ
5	bDeviceSubClass	1	คลาสย่อยของอุปกรณ์นั้นๆ
6	bDeviceProtocol	1	โปรโตคอลของอุปกรณ์นั้นๆ
7	bMaxPacketSize0	1	ขนาดของข้อมูลที่สามารถใช้ได้ในแต่ละแพ็กเก็ตสำหรับเอนพอยน์หมายเลข 0
8	idVendor	2	หมายเลขผู้ผลิตอุปกรณ์นั้นๆ
10	idProduct	2	หมายเลขของอุปกรณ์นั้นๆ
12	bcdDevice	2	เวอร์ชันของอุปกรณ์นั้นๆ
14	iManufacturer	1	อินเด็กซ์ของสตริงเดสคริปเตอร์ ที่อธิบายข้อมูลของผู้ผลิต
15	iProduct	1	อินเด็กซ์ของสตริงเดสคริปเตอร์ ที่อธิบายข้อมูลของอุปกรณ์
16	iSerialNumber	1	อินเด็กซ์ของสตริงเดสคริปเตอร์ ที่อธิบายหมายเลขประจำตัวของอุปกรณ์
17	NumConfigurations	1	จำนวนของโหมดการทำงานของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2. คอนฟิกูเรชันเดสคริปเตอร์ (Configuration Descriptor)

ใช้สำหรับเก็บข้อมูลเกี่ยวกับการตั้งค่าของอุปกรณ์ในการทำงานโหมดต่างๆ ว่าในแต่ละโหมดการทำงานของอุปกรณ์จะต้องมีการตั้งค่าอย่างไรบ้าง ซึ่งในการเลือกที่จะให้ตัวอุปกรณ์ทำงานในโหมดนั้น ก็คือการเลือกคอนฟิกูเรชัน (Configuration) นั้นเอง โดยที่การเลือกคอนฟิกูเรชันนั้น สามารถเลือกได้โดยใช้คำสั่ง SetConfiguration โดยการอ้างอิงถึงตัวคอนฟิกูเรชันเดสคริปเตอร์ตัวใดนั้น สามารถอ้างอิงได้จากค่าในส่วนของ bConfigurationValue ของคอนฟิกูเรชันเดสคริปเตอร์ตัวนั้นๆ ในตัวอุปกรณ์ยูเอสบี สามารถมีคอนฟิกูเรชันเดสคริปเตอร์ได้มากกว่า 1 ตัว ซึ่งขึ้นอยู่กับจำนวนโหมดการทำงานของตัวอุปกรณ์ยูเอสบี รูปแบบของคอนฟิกูเรชันเดสคริปเตอร์ แสดงดังตารางที่ 2.6

ตารางที่ 2.6 แสดงโครงสร้างข้อมูลของคอนฟิกูเรชันเดสคริปเตอร์

ตำแหน่ง	ชื่อ Field ข้อมูล	ขนาด	ความหมายข้อมูล
0	bLength	1	ขนาดของเดสคริปเตอร์
1	bDescriptorType	1	ค่าคงที่ที่บอกว่าเป็นคอนฟิกูเรชันเดสคริปเตอร์
2	wTotalLength	2	ขนาดของข้อมูลที่จะส่งคืนจากการร้องขอข้อมูลเดสคริปเตอร์นี้
4	bNumInterfaces	1	จำนวนอินเตอร์เฟซของคอนฟิกูเรชันนั้นๆ
5	bConfigurationValue	1	ค่าของคอนฟิกูเรชันนี้ ซึ่งใช้อ้างถึงเดสคริปเตอร์นี้โดยโฮสต์
6	iConfiguration	1	อินเด็กซ์ของสตริงเดสคริปเตอร์ที่อธิบายข้อมูลของคอนฟิกูเรชัน
7	bmAttributes	1	ความสามารถเพิ่มเติมของอุปกรณ์ในคอนฟิกูเรชันนั้นๆ
8	MaxPower	1	ค่ากระแสไฟฟ้าสูงสุดที่สามารถรับได้

2.6.3. อินเทอร์เฟซเดสคริปเตอร์ (Interface Descriptor)

ในแต่ละคอนฟิกูเรชันเดสคริปเตอร์ จะประกอบด้วยอินเทอร์เฟซเดสคริปเตอร์ตั้งแต่ 1 ตัวขึ้นไป ซึ่งเป็นการกำหนดอินเทอร์เฟซ ที่มีการใช้งานในโหมดการทำงานของคอนฟิกูเรชันนั้นๆ ซึ่งในแต่ละอินเทอร์เฟซ ก็จะมีหน้าที่การทำงานที่แตกต่างกัน โดยที่จะมีอินเทอร์เฟซเดสคริปเตอร์ที่คอยเป็นตัวกำหนดการทำงานของแต่ละอินเทอร์เฟซ และนอกจากนี้ ยังมี Alternate Interface ที่จะถูกเรียกใช้แทนอินเทอร์เฟซเดสคริปเตอร์ในกรณีที่โฮสต์ไม่สามารถทำงานตามการตั้งค่าของอินเทอร์เฟซเดสคริปเตอร์ได้ ซึ่ง Alternate Interface นั้นจะเป็นการเปลี่ยนแปลงการตั้งค่าของอินเทอร์เฟซเดสคริปเตอร์ เพื่อที่จะช่วยให้ตัวอุปกรณ์ยูเอสบีสามารถที่จะติดต่อสื่อสารข้อมูลโฮสต์ได้ โดยในแต่ละอินเทอร์เฟซเดสคริปเตอร์ จะประกอบด้วยเอนพอยน์ตเดสคริปเตอร์ ตั้งแต่ 1 ตัวขึ้นไปหรืออาจจะไม่มีเลขก็ได้ ขึ้นอยู่กับอินเทอร์เฟซ นั้นๆ ว่าต้องการเอนพอยน์จำนวนเท่าใดในการทำงานตามหน้าที่ของ อินเทอร์เฟซ นั้นๆ และกลุ่มของเอนพอยน์ที่ถูกใช้ในอินเทอร์เฟซ จะไม่สามารถใช้ร่วมกับ อินเทอร์เฟซ อื่นๆ ที่อยู่ภายใต้โหมดการทำงาน (คอนฟิกูเรชัน) เดียวกันได้ แต่สามารถใช้ร่วมกับ อินเทอร์เฟซ อื่นๆ ที่อยู่คนละโหมดการทำงานและใช้ร่วมกับ Alternate Interface ของอินเทอร์เฟซเดสคริปเตอร์นั้นๆ ได้ โดยรูปแบบของอินเทอร์เฟซเดสคริปเตอร์แสดงดังตารางที่ 2.7

ตารางที่ 2.7 แสดงโครงสร้างข้อมูลของอินเทอร์เฟซเดสคริปเตอร์

ตำแหน่ง	ชื่อฟิลด์ข้อมูล	ขนาด	ความหมายข้อมูล
0	bLength	1	ขนาดของเดสคริปเตอร์
1	bDescriptorType	1	ค่าคงที่ที่บอกว่าเป็นอินเทอร์เฟซเดสคริปเตอร์
2	bInterfaceNumber	1	หมายเลขของอินเทอร์เฟซ
3	bAlternateSetting	1	การเลือกอินเทอร์เฟซ อื่นๆ ในการใช้แทนอินเทอร์เฟซนั้นๆ
4	bNumEndpoints	1	จำนวนของเอนพอยน์ของอินเทอร์เฟซนั้นๆ
5	bInterfaceClass	1	คลาสของอินเทอร์เฟซ
6	bInterfaceSubClass	1	ซับคลาสของอินเทอร์เฟซ
7	bInterfaceProtocol	1	โปรโตคอลของอินเทอร์เฟซ
8	iInterface	1	อินเด็กซ์ของสตริงเดสคริปเตอร์ที่อธิบายข้อมูลของอินเทอร์เฟซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.4. เอนพอยน์เตสคริปเตอร์ (Endpoint Descriptor)

ใช้สำหรับเก็บข้อมูลของเอนพอยน์ที่เอนพอยน์เตสคริปเตอร์ตัวนั้นๆ อ้างถึง โดยจะมีข้อมูลในส่วนที่ตัวยูเอสบีโฮสต้องการนำไปทำการประมวล เช่น ความต้องการแบนด์วิท (Bandwidth) ของตัวอุปกรณ์ยูเอสบี, รูปแบบการสื่อสารข้อมูล แต่อย่างไรก็ตาม จะไม่มีอินเตอร์เฟส ใดๆ ที่จะมีเอนพอยน์เตสคริปเตอร์ของเอนพอยน์ 0 เนื่องจากเอนพอยน์ 0 เป็นเอนพอยน์ที่ถูกตั้งค่าเริ่มต้นให้เป็น Control Endpoint ซึ่งใช้สำหรับการสื่อสาร Control Transfer เท่านั้น รูปแบบของเอนพอยน์เตสคริปเตอร์แสดงตารางที่ 2.8

ตารางที่ 2.8 แสดงโครงสร้างข้อมูลของ Endpoint Descriptor

ตำแหน่ง	ชื่อฟิลด์ข้อมูล	ขนาด	ความหมายข้อมูล
0	bLength	1	ขนาดของเตสคริปเตอร์
1	bDescriptorType	1	ค่าคงที่ที่บอกว่าเป็นเอนพอยน์เตสคริปเตอร์
2	EndpointAddress	1	แอดเดรสของเอนพอยน์
3	bmAttributes	1	การเลือกอินเตอร์เฟส อื่นๆ ในการใช้แทนอินเตอร์เฟส นั้นๆ
4	wMaxPacketSize	2	ขนาดของข้อมูลที่สามารถใช้ได้ในแต่ละแพ็คเกจ
6	bInterval	1	ช่วงเวลาของการรวนถามข้อมูลแต่ละครั้ง สำหรับการสื่อสารข้อมูลแบบ Interrupt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.5. สตริงเดสคริปเตอร์ (String Descriptor)

เป็นเดสคริปเตอร์ที่ไม่เกี่ยวข้องกับการทำงานของตัวอุปกรณ์ยูเอสบีแต่เป็นเดสคริปเตอร์ที่ใช้สำหรับการอธิบายรายละเอียดของเดสคริปเตอร์ตัวอื่นๆ ในสตริงเดสคริปเตอร์จะใช้รูปแบบการเก็บข้อมูลเป็นแบบยูนิโค้ด (UNICODE) จึงทำให้สตริงเดสคริปเตอร์สามารถสนับสนุนข้อมูลได้หลายภาษา ดังนั้นในการใช้สตริงเดสคริปเตอร์จึงจำเป็นที่จะต้องมีการระบุภาษาที่ใช้งานด้วย โดยระบุด้วยรหัสภาษา (LANGID) รูปแบบของ สตริงเดสคริปเตอร์แสดงดังตารางที่ 2.9 และตารางที่ 2.10

ตารางที่ 2.9 แสดงโครงสร้างข้อมูลของสตริงเดสคริปเตอร์

ตำแหน่ง	ชื่อฟิลด์ข้อมูล	ขนาด	ความหมายข้อมูล
0	bLength	1	ขนาดของเดสคริปเตอร์
1	bDescriptorType	1	ค่าคงที่ที่บอกว่าเป็นสตริงเดสคริปเตอร์
2	bString	N	ข้อความสตริงของสตริงเดสคริปเตอร์

ตารางที่ 2.10 แสดงโครงสร้างข้อมูลของรหัสภาษา

ตำแหน่ง	ชื่อ Field ข้อมูล	ขนาด	ความหมายข้อมูล
0	bLength	1	ขนาดของเดสคริปเตอร์
1	bDescriptorType	1	ค่าคงที่ที่บอกว่าเป็นสตริงเดสคริปเตอร์
2	wLANG[0]	1	รหัสภาษาลำดับที่ 0
...
N	wLANG[x]	2	รหัสภาษาลำดับที่ x

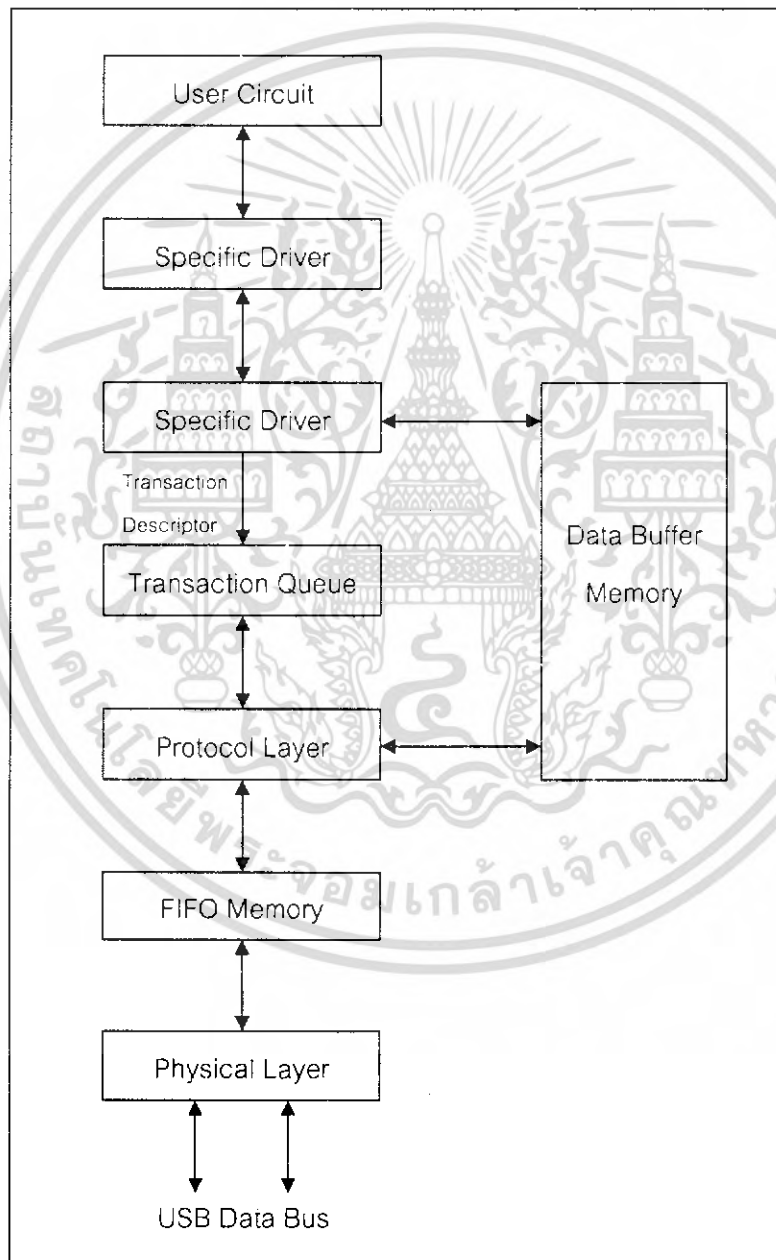
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการพัฒนา

3.1 การออกแบบระบบโดยรวม

การออกแบบระบบโดยรวม จะมีการแบ่งหน้าที่การทำงานของระบบออกเป็นส่วนๆ โดยในแต่ละส่วนมีการทำงานที่แตกต่างกันโดยชัดเจน ซึ่งได้แก่ส่วนการทำงานต่างๆ ดังรูปที่ 3.1



รูปที่ 3.1 แสดงส่วนประกอบโดยรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 ส่วนการทำงานของไคร์เวอร์เฉพาะอุปกรณ์

เป็นส่วนการทำงานที่ขึ้นอยู่กับแต่ละอุปกรณ์ โดยในแต่ละอุปกรณ์ก็จะมีขั้นตอนการทำงานของไคร์เวอร์ที่แตกต่างกันไป ซึ่งไคร์เวอร์นั้น จะเป็นตัวกลางในการติดต่อสื่อสารกับส่วนที่เป็นอุปกรณ์ที่ทำหน้าที่เป็นโฮสต์ โดยการทำงานนั้น จะมีการรับคำสั่งจากส่วนวงจรมือถือว่า ต้องการรับหรือส่งข้อมูลกับอุปกรณ์ตัวใด แล้วจะทำการสร้างทรานแซคชันเดสคริปเตอร์ (Transaction Descriptor) และเขียนลงในคิวของทรานแซคชันเดสคริปเตอร์ ซึ่งข้อมูลทรานแซคชันเดสคริปเตอร์จะถูกส่วนการทำงานชั้น โปรโตคอลทำการอ่านไปประมวลผลต่อไปตามลำดับการเขียนของทรานแซคชันเดสคริปเตอร์ หน้าที่ต่างๆ ของไคร์เวอร์ ได้แก่

1) การอ่านค่าเดสคริปเตอร์ต่างๆ ของตัวอุปกรณ์ยูเอสบี

การอ่านค่าเดสคริปเตอร์ต่างๆ จากอุปกรณ์เพื่อทำการตรวจสอบค่ารายละเอียดของอุปกรณ์ว่ามีคุณสมบัติอะไรบ้าง เช่น จำนวนเอนพอยน์ รูปแบบการสื่อสารข้อมูลของเอนพอยน์

2) การปรับตั้งค่าการทำงานต่างๆ ของตัวอุปกรณ์

การปรับตั้งค่าการทำงานต่างๆ ให้กับอุปกรณ์ จะต้องทำเป็นอันดับแรกในการติดต่อสื่อสารกับอุปกรณ์ยูเอสบี โดยการตัดสินใจตั้งค่าการทำงานต่างๆ ของอุปกรณ์จะนำข้อมูลเดสคริปเตอร์มาทำการตัดสินใจ

3) การรับและส่งข้อมูลระหว่างตัวอุปกรณ์กับวงจรมือถือ

ส่วนของวงจรมือถือจะทำการร้องขอการส่งหรือรับข้อมูลมายังไคร์เวอร์ ก่อนที่ไคร์เวอร์จะทำการสร้างทรานแซคชันเดสคริปเตอร์ส่งให้ชั้นการทำงานระดับโปรโตคอล เพื่อให้เกิดการรับส่งข้อมูลได้

3.1.2 ส่วนการทำงานของชั้นโปรโตคอล

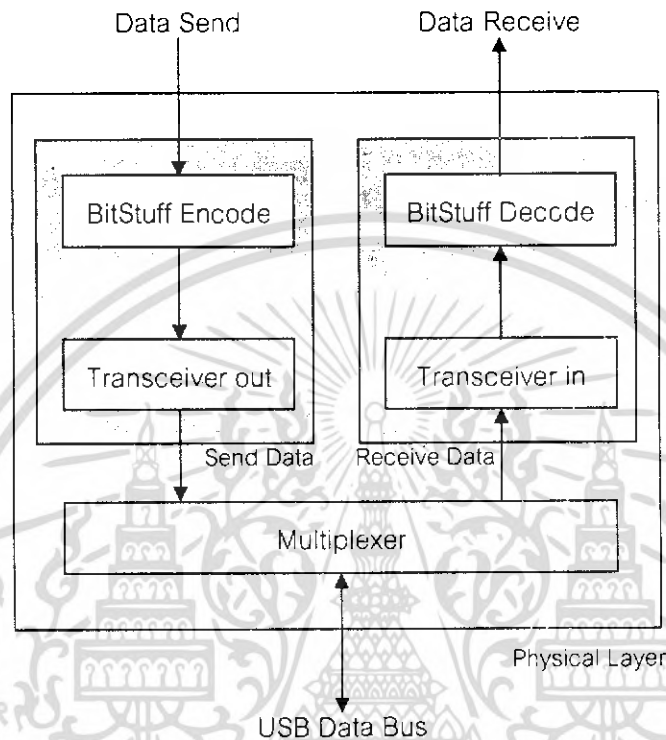
เป็นส่วนการทำงานที่สำคัญมาก ซึ่งเป็นส่วนที่จะทำหน้าที่ควบคุมการทำงานต่างๆ ของทั้งระบบคือการทำงานของชั้นโปรโตคอล โดยจะมีการตรวจสอบทรานแซคชันแคสเคดรีปเคอร์ ว่ามีการร้องขอมาจากส่วนของใครเวอร์หรือไม่ ถ้ามีการร้องขอจากใครเวอร์ ชั้นโปรโตคอลจะต้องทำการสร้างแพ็กเก็ตข้อมูลต่างๆ เพื่อส่งให้กับตัวอุปกรณ์ตัวที่ใครเวอร์ต้องการ นอกจากนี้ ยังจะต้องคอยตรวจสอบความถูกต้องของข้อมูลที่มีการสื่อสารระหว่างวงจรผู้ใช้กับตัวอุปกรณ์ โดยใช้การสร้างข้อมูลสำหรับการตรวจสอบความถูกต้องของข้อมูลตามหลักการทางคณิตศาสตร์ ซึ่งในการสื่อสารข้อมูลแบบยูเอสบีนั้นมีการใช้ข้อมูลตรวจสอบดังกล่าวแบบ CRC (Cyclic Redundancy Checks) และในการสื่อสารข้อมูลนั้น ข้อมูลต่างๆ ที่จะมีการรับส่ง ถูกอ่านและเขียนจากหน่วยความจำบัฟเฟอร์ข้อมูล และในชั้นโปรโตคอลจะทำการสร้างแพ็กเก็ตข้อมูลและเขียนลงในหน่วยความจำ FIFO (First-In First-Out Memory) และรอให้ส่วนการทำงานชั้นกายภาพ ทำการส่งข้อมูลไปยังตัวอุปกรณ์ โดยใช้รูปแบบของสัญญาณแบบผลต่าง และในกรณีที่ต้องการตรวจสอบส่วนของผู้ใช้ที่ต้องการที่จะรับข้อมูลจากตัวอุปกรณ์ การทำงานจะเริ่มจากการอ่านข้อมูลจากหน่วยความจำ FIFO แล้วทำการแยกข้อมูลออกจากส่วนที่เป็นแพ็กเก็ตข้อมูลและเฟรมข้อมูล

3.1.3 ส่วนการทำงานของชั้นการทำงานชั้นกายภาพ

เป็นส่วนการทำงานที่รับผิดชอบเกี่ยวกับสัญญาณทางไฟฟ้าและข้อมูลระดับบิต เมื่อจะทำการส่งข้อมูล จะทำการอ่านข้อมูลจากหน่วยความจำ FIFO แล้วทำการประมวลผลข้อมูลต่างๆ ก่อนที่จะส่งสัญญาณทางไฟฟ้าสู่สายสัญญาณยูเอสบีและเมื่อต้องการรับข้อมูล ก็จะนำสัญญาณไฟฟ้าจากสายสัญญาณยูเอสบีมาทำการประมวลผลก่อนนำข้อมูลไปเขียนลงในหน่วยความจำ FIFO

3.2 การออกแบบส่วนการทำงานชั้นกายภาพ

การออกแบบส่วนการทำงานชั้นกายภาพ จะแบ่งออกเป็น 2 ส่วนตามทิศทางของข้อมูล ซึ่งได้แก่ส่วนที่ทำหน้าที่ส่งข้อมูลและส่วนที่ทำหน้าที่รับข้อมูล

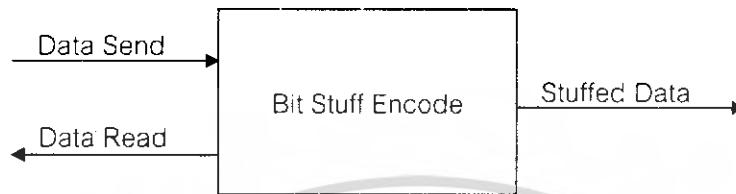


รูปที่ 3.2 แสดงส่วนประกอบโดยรวมของส่วนการทำงานในชั้นกายภาพ

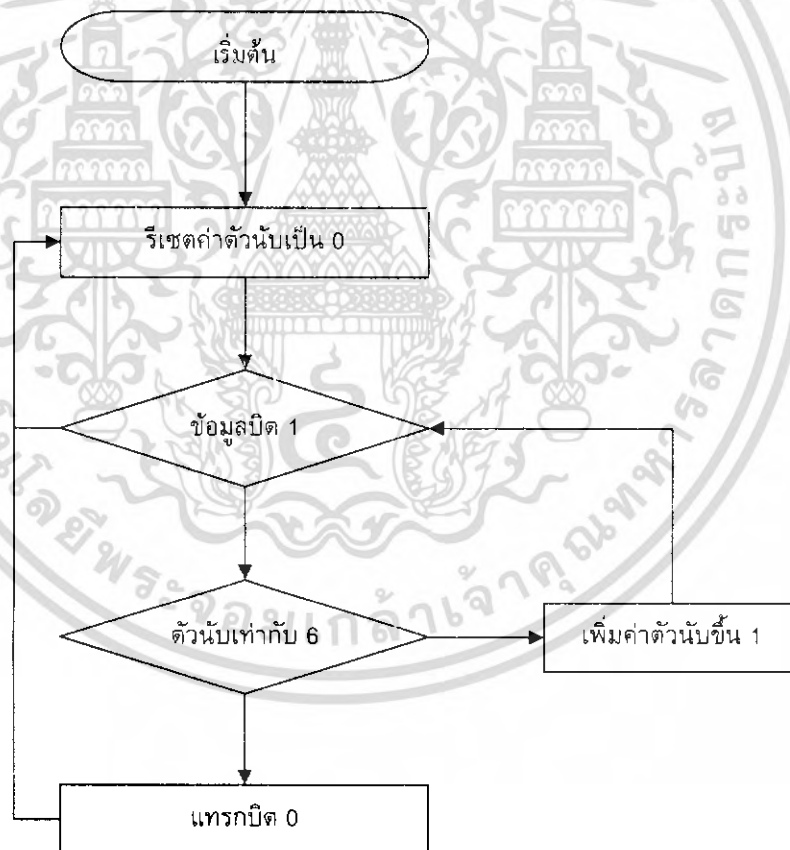
3.2.1 ส่วนที่ทำหน้าที่ในการส่งข้อมูล

ส่วนที่ทำหน้าที่ในการส่งข้อมูล มีหน้าที่นำข้อมูลที่ได้รับจากชั้นการทำงานระดับโปรโตคอลมาสู่กระบวนการการแทรกบิต (Bit Stuffing) โดยในการส่งข้อมูล จะเป็นการนำบิตข้อมูล '0' มาแทรกในข้อมูลเมื่อตรวจพบว่ามีข้อมูลบิต '1' ติดต่อกันจำนวน 6 บิต ดังนั้น ในรูปที่ 3.2 จึงเรียกส่วนแทรกบิตดังกล่าวเป็นส่วนการเข้ารหัสโดยการแทรกบิต (Bit Stuff Encode) และเมื่อข้อมูลได้ผ่านการบวนการแทรกบิตแล้ว จะต้องทำการเข้ารหัสสัญญาณเข้ามุลแบบ NRZI อีกครั้งและทำการแปลงข้อมูลดังกล่าวเป็นสัญญาณทางไฟฟ้าตามรูปแบบของสัญญาณยูเอสบีซึ่งเป็นที่ทำงานของส่วนแปลงรูปแบบข้อมูลสำหรับส่งออก (Transceiver Out) ผลลัพธ์จากการทำงานของส่วนการทำงานดังกล่าว จะได้รูปแบบสัญญาณแบบผลต่าง (Differential Signal) ซึ่งเป็นสัญญาณ D+ และ D-

ส่วนการเข้ารหัสโดยการแทรกบิตนั้น มีส่วนประกอบที่สำคัญดังรูปที่ 3.3 ซึ่งตัวเข้ารหัสข้อมูลจะส่งสัญญาณ Data Read เพื่อทำการอ่านข้อมูลจากสัญญาณ Data Send เข้าไปยังตัวเข้ารหัส และทำการเข้ารหัสและส่งผลลัพธ์ออกมาทางสัญญาณ Stuffed Data ขั้นตอนการทำงานของส่วนการเข้ารหัสด้วยการแทรกบิตแสดงดังรูปที่ 3.4



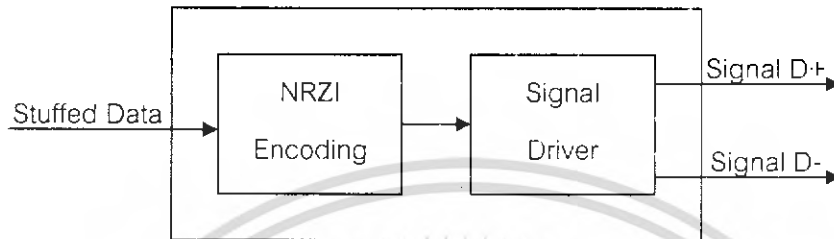
รูปที่ 3.3 แสดงส่วนประกอบที่สำคัญของส่วนการเข้ารหัสโดยการแทรกบิต



รูปที่ 3.4 แสดงขั้นตอนการทำงานของส่วนการเข้ารหัสโดยการแทรกบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของส่วนการแปลงรูปแบบข้อมูลสำหรับการส่งออก จะทำการรับข้อมูลที่ต้องแปลงจากสัญญาณ Stuffed Data ก่อนจะเข้าสู่กระบวนการเข้ารหัสสัญญาณ NRZI และทำการแปลงข้อมูลให้เป็นสัญญาณแบบผลต่าง ส่วนประกอบที่สำคัญของส่วนการแปลงข้อมูลสำหรับการส่งออกแสดงดังรูปที่ 3.5

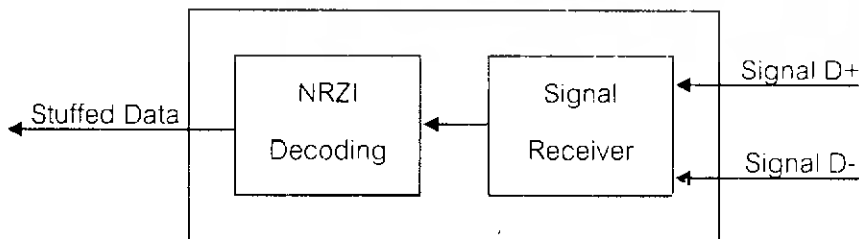


รูปที่ 3.5 แสดงส่วนประกอบที่สำคัญของส่วนการแปลงรูปแบบข้อมูลสำหรับการส่งออก

3.2.2 ส่วนที่ทำหน้าที่ในการรับข้อมูล

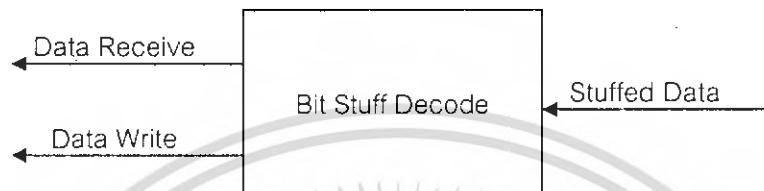
ส่วนที่ทำหน้าที่ในการรับข้อมูล มีหน้าที่ในการรับข้อมูลที่เป็นรูปแบบของสัญญาณแบบผลต่างที่ได้รับจากสัญญาณ D+ และ D- หลังจากนั้นจะทำการแปลงสัญญาณที่ได้รับเป็นข้อมูลแบบดิจิทัล หลังจากนั้น ข้อมูลที่ถูกแปลงแล้วจะถูกถอดรหัสสัญญาณที่เข้ารหัสด้วยรูปแบบ NRZI ซึ่งหน้าที่ดังกล่าวจะเป็นหน้าที่ของส่วนการแปลงรูปแบบข้อมูลสำหรับการรับข้อมูลเข้า (Transceiver In) และหลังจากนั้น ข้อมูลจะถูกส่งต่อไปให้กับส่วนถอดรหัสข้อมูลที่ถูกรักษา โดยการนำบิตที่ถูกแทรกออกจากส่วนของข้อมูล ซึ่งเป็นหน้าที่ของส่วนการถอดรหัสข้อมูลที่ถูกรักษา (Bit Stuff Decode)

การทำงานของส่วนการแปลงรูปแบบข้อมูลสำหรับการรับข้อมูลเข้า จะทำการรับสัญญาณแบบผลต่างมาจากสัญญาณ D+ และ D- และทำการแปลงข้อมูลก่อนส่งให้ส่วนการถอดรหัส NRZI ซึ่งจะได้ข้อมูลที่จะส่งให้กับส่วนการถอดรหัสการแทรกบิตทางสัญญาณ Stuffed Data ส่วนประกอบที่สำคัญของส่วนการแปลงข้อมูลสำหรับรับเข้า แสดงดังรูปที่ 3.6

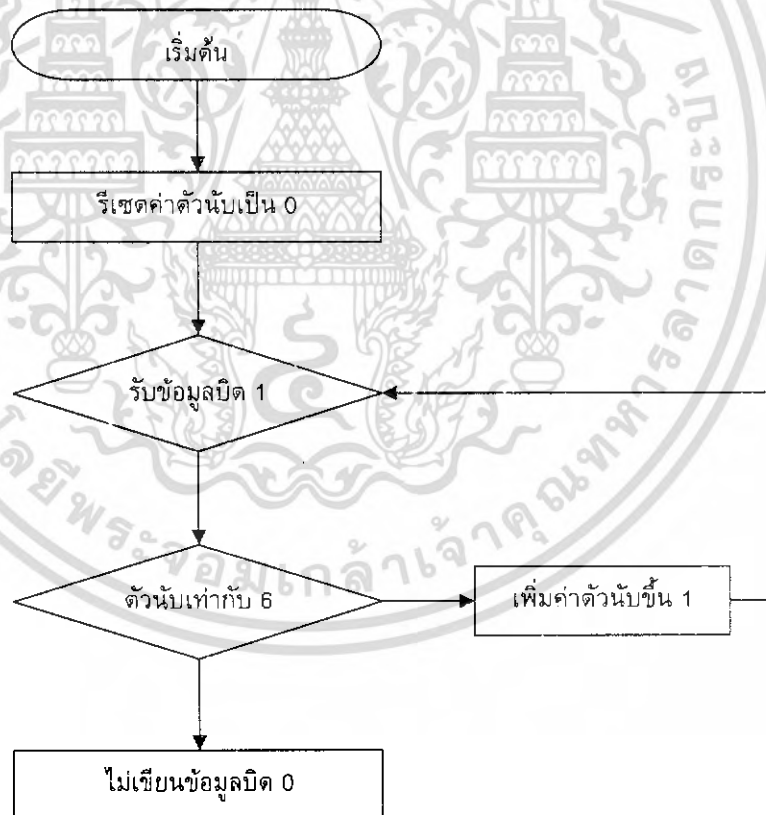


รูปที่ 3.6 แสดงส่วนประกอบที่สำคัญของส่วนการแปลงข้อมูลสำหรับการรับเข้า

การทำงานของส่วนการถอดรหัสข้อมูลการแทรกบิตจะทำการรับข้อมูลจากสัญญาณ Stuffed Data แล้วทำการถอดรหัสข้อมูลโดยการนำบิตข้อมูลที่ถูกรวมออกจากข้อมูล ซึ่งจะได้ผลลัพธ์ข้อมูลจากสัญญาณ Data Receive และสัญญาณ Data Write จะทำการเขียนข้อมูลผ่านการถอดรหัสแล้ว ดังรูปที่ 3.7 และขั้นตอนการทำงานของส่วนการถอดรหัสการแทรกบิตแสดงดังรูปที่ 3.8



รูปที่ 3.7 แสดงส่วนประกอบที่สำคัญของส่วนการถอดรหัสข้อมูลการแทรกบิต



รูปที่ 3.8 แสดงขั้นตอนการทำงานของส่วนการถอดรหัสการแทรกบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบส่วนการทำงานในชั้นโปรโตคอล

การทำงานของชั้น โปรโตคอลจะมีหน้าที่ที่สำคัญคือการสร้างแพ็กเก็ตข้อมูลในรูปแบบต่างๆ ซึ่งมีรูปแบบของข้อมูลและหน้าที่การทำงานที่แตกต่างกันไป ในชั้นการทำงานระดับโปรโตคอลจะทำงานตามคำสั่งของไดรเวอร์ โดยที่ไดรเวอร์จะทำการสั่งให้ชั้น โปรโตคอลทำงานผ่านทางทรานแซคชันเดสคริปเตอร์

รูปแบบข้อมูลของทรานแซคชันเดสคริปเตอร์จะแตกต่างกันไป ตามชนิดของแพ็กเก็ตข้อมูล ซึ่งมีรูปแบบต่างๆ คือ

- 1) รูปแบบทรานแซคชันเดสคริปเตอร์สำหรับแพ็กเก็ต SOF (Start-Of-Frame) แสดงดังรูปที่ 3.9 ซึ่งประกอบด้วย ข้อมูลหมายเลขประจำแพ็กเก็ต จำนวน 4 บิต, หมายเลขเฟรมจำนวน 11 บิต

7	6	5	4	3	2	1	0
PID3 - PID0				FrameNo10 - FrameNo7			
FrameNo6 - FrameNo0							0

รูปที่ 3.9 แสดงรูปแบบข้อมูลของทรานแซคชันเดสคริปเตอร์
สำหรับแพ็กเก็ต SOF

- 2) รูปแบบทรานแซคชันเดสคริปเตอร์สำหรับแพ็กเก็ต IN, OUT, SETUP แสดงดังรูปที่ 3.10 ซึ่งประกอบด้วย ข้อมูลหมายเลขประจำแพ็กเก็ต จำนวน 4 บิต, ตำแหน่งแอดเดรสของอุปกรณ์จำนวน 7 บิต, ตำแหน่งเอนพอยน์แอดเดรส จำนวน 4 บิต

7	6	5	4	3	2	1	0
PID3 - PID0				Addr6 - Addr3			
Addr2 - Addr 0			Endp3 - Endp 0			0	

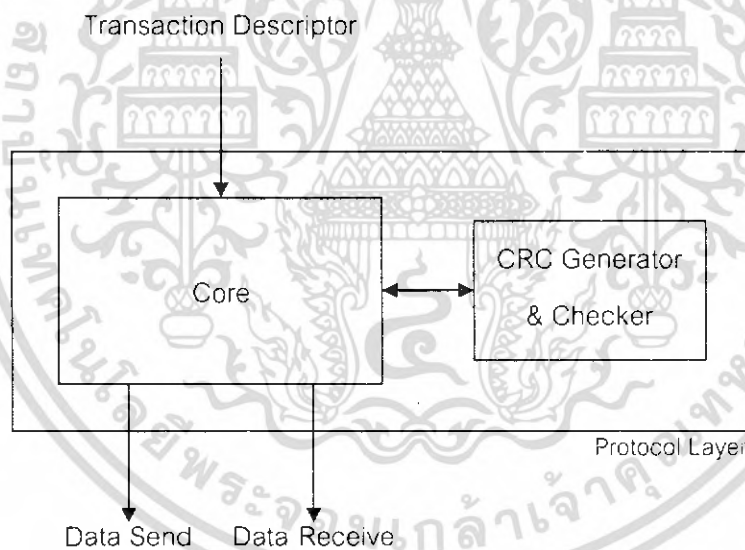
รูปที่ 3.10 แสดงรูปแบบข้อมูลของทรานแซคชันเดสคริปเตอร์
สำหรับแพ็กเก็ต IN, OUT, SETUP

- 3) รูปแบบทรานแซกชันเดสคริปเตอร์สำหรับแพ็คเกจข้อมูล แสดงดังรูปที่ 3.11 ซึ่งประกอบด้วย ข้อมูลหมายเลขประจำแพ็คเกจ จำนวน 4 บิต และขนาดของข้อมูล จำนวน 4 บิต

7	6	5	4	3	2	1	0
PID3 - PID0				DataSize3 - DataSize0			
0							

รูปที่ 3.11 แสดงรูปแบบข้อมูลของทรานแซกชันเดสคริปเตอร์
สำหรับแพ็คเกจข้อมูล

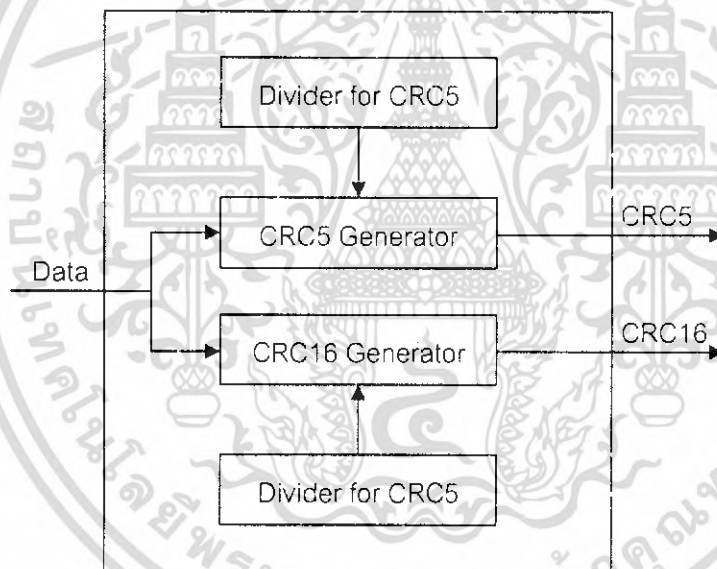
ส่วนประกอบหลักของส่วนการทำงานชั้นโปรโตคอลนั้น ประกอบด้วย ส่วนประมวลผลหลัก (Core) และส่วนที่ทำหน้าที่สร้างและตรวจสอบความถูกต้องของข้อมูลด้วย (CRC Generator, CRC Checker) ซึ่งแสดงดังรูปที่ 3.12



รูปที่ 3.12 แสดงส่วนประกอบของส่วนการทำงานในชั้นโปรโตคอล

3.3.1 การออกแบบส่วนตรวจสอบความถูกต้องของข้อมูล

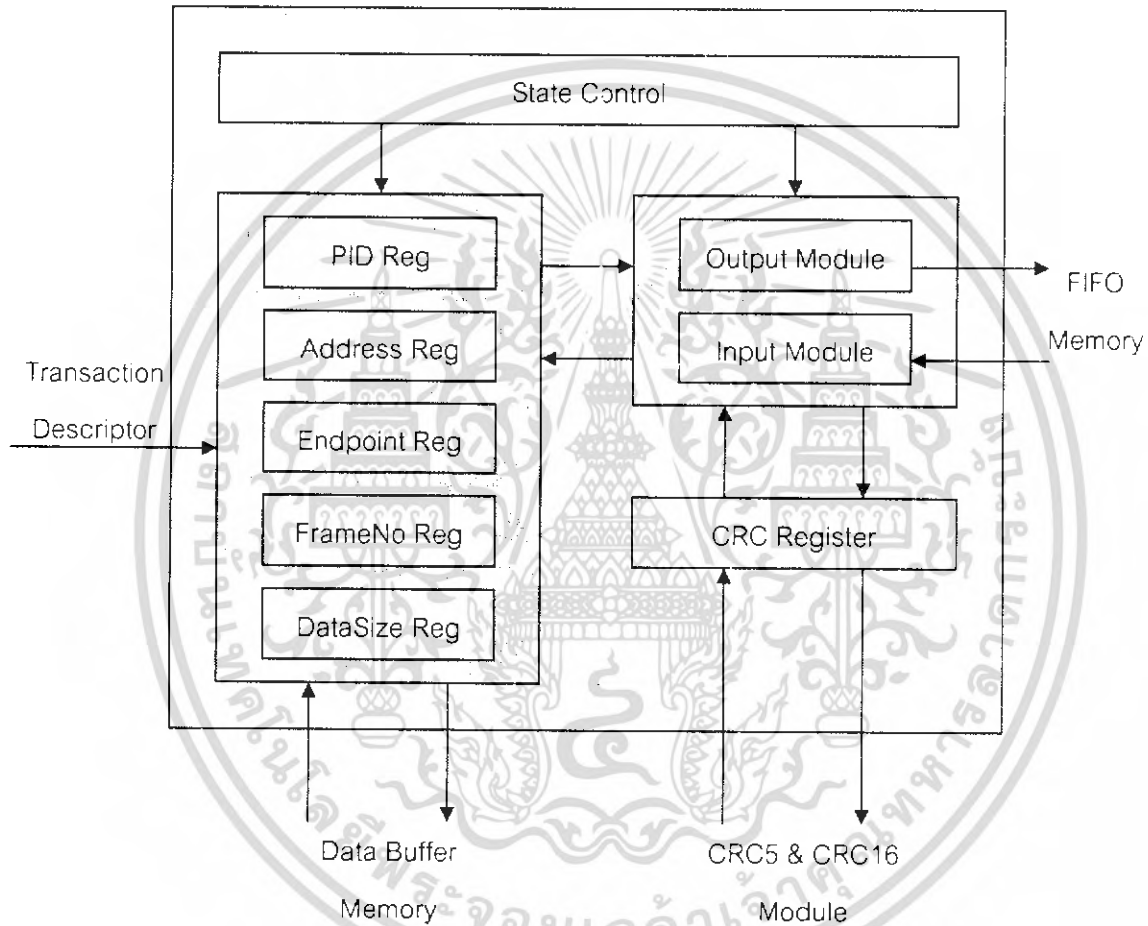
ในการตรวจสอบความถูกต้องของข้อมูลที่ใช้ในการสื่อสารข้อมูลแบบยูเอสบี นั้นมี 2 ประเภทคือ การใช้ CRC ขนาด 5 บิต สำหรับการตรวจสอบข้อมูลหมายเลขเฟรม, ข้อมูลตำแหน่งแอดเดรสของอุปกรณ์, ข้อมูลตำแหน่งเอนพอยน์ของอุปกรณ์ และ CRC ขนาด 16 บิตสำหรับการตรวจสอบความถูกต้องของข้อมูล (ข้อมูลจริงที่ต้องการสื่อสาร) โดยการทำงานของส่วนการสร้างและตรวจสอบความถูกต้องของข้อมูลนั้น จะทำการรับข้อมูลจากสัญญาณข้อมูล และทำการสร้างข้อมูล CRC โดยใช้ Polynomial ในการคำนวณ CRC ขนาด 5 บิตและ CRC ขนาด 16 บิตใน Divider for CRC5 และ Divider for CRC16 ตามลำดับ โดยการสร้างข้อมูล CRC นั้น จะทำการสร้างข้อมูล CRC ทั้ง 2 รูปแบบพร้อมๆ กัน คือ CRC5 ซึ่งส่งผลลัพธ์ทางสัญญาณ CRC5 และ CRC16 ซึ่งส่งผลลัพธ์ทางสัญญาณ CRC16 ดังนั้น ในการอ่านข้อมูล CRC ตัวประมวลผลหลักจะต้องทำการอ่านค่า CRC ให้ถูกต้องตามชนิดของแก็กเก็ต ส่วนประกอบหลักของส่วนการสร้างและตรวจสอบความถูกต้องของข้อมูลจะแสดงดังรูปที่ 3.13



รูปที่ 3.13 แสดงส่วนประกอบหลักของส่วนการสร้าง
และตรวจสอบความถูกต้องของข้อมูล

3.3.2 การออกแบบส่วนประมวลผลหลัก

ส่วนประมวลผลหลัก จะทำหน้าที่สร้างแพ็กเก็ตข้อมูลต่างๆ และเขียนลงหน่วยความจำ FIFO ซึ่งแต่ละแพ็กเก็ตข้อมูลต่างๆ ก็จะมีขนาดแตกต่างกันไป ในการตัดสินใจว่าจะทำการสร้างแพ็กเก็ตข้อมูลแบบใดนั้น จะถูกตัดสินใจจากค่า PID จาก แพ็กเก็ตข้อมูลก่อนหน้าและปัจจุบัน ส่วน State Control จะเป็นการกำหนดขั้นตอนการทำงานต่างๆ ให้ถูกต้อง ส่วนประกอบต่างๆ ของส่วนประมวลผลหลัก แสดงดังรูปที่ 3.14



รูปที่ 3.14 แสดงส่วนประกอบของส่วนประมวลผลหลัก

ข้อมูลในการสร้างแพ็คเกจข้อมูลนั้น จะถูกนำมาจากทรานแซคชันเดสริปเตอร์ ซึ่งไคร์เวอร์จะระบุค่าต่างๆ ที่ใช้ในการสื่อสารข้อมูล ซึ่งได้แก่

- 1) หมายเลขประจำแพ็คเกจ (PID) ใช้สำหรับการบ่งบอกชนิดของแพ็คเกจข้อมูล ว่าเป็นแพ็คเกจข้อมูลชนิดใด เพื่อให้ส่วนการทำงานชั้นโปรโตคอลสามารถสร้างแพ็คเกจข้อมูลได้อย่างถูกต้อง
- 2) หมายเลขเฟรม (Frame Number) ใช้สำหรับการระบุหมายเลขเฟรม ซึ่งจะถูกส่งไปพร้อมๆ กับแพ็คเกจชนิด SOF (Start-of-frame Packet) ใช้สำหรับการตรวจสอบลำดับความถูกต้องของข้อมูลในบางอุปกรณ์
- 3) ตำแหน่งแอดเดรสของอุปกรณ์ (Device Address) ใช้สำหรับการอ้างถึงอุปกรณ์แต่ละตัว ซึ่งจะมีค่าตำแหน่งแอดเดรสที่ไม่เหมือนกัน
- 4) ตำแหน่งแอดเดรสของเอนพอยน์ (Endpoint Address) ใช้สำหรับการอ้างถึงตำแหน่งเอนพอยน์ของอุปกรณ์แต่ละตัว
- 5) ขนาดของข้อมูล (DataSize) ใช้สำหรับการบอกขนาดของข้อมูลในกรณีที่เป็นแพ็คเกจชนิดแพ็คเกจข้อมูล

CRC Register ใช้สำหรับการเก็บข้อมูลการตรวจสอบความถูกต้องในกรณีที่ทำกรส่งข้อมูล และใช้สำหรับการตรวจสอบความถูกต้องในกรณีที่ได้รับข้อมูล

Output Module มีหน้าที่ข้อมูลที่จะทำการเขียนลงในหน่วยความจำ FIFO ซึ่งจะต้องมีการจัดลำดับข้อมูลอย่างถูกต้อง ดังนั้น จึงใช้ State Control มาเป็นส่วนช่วยในการทำการเรียงลำดับข้อมูลที่จะทำการเขียน

Input Module มีหน้าที่รับข้อมูลโดยทำการอ่านจากหน่วยความจำ FIFO ซึ่งในการอ่านข้อมูลนั้น จะถูกกำหนดโดยไคร์เวอร์ ว่าต้องการให้อ่านข้อมูลเมื่อใด และต้องการอ่านเป็นจำนวนเท่าไร และข้อมูลที่ได้อ่านจากหน่วยความจำ FIFO เมื่อตรวจสอบแล้วไม่มีข้อผิดพลาดก็จะนำไปเขียนไว้ในส่วนของ Data Buffer Memory

3.4 การออกแบบส่วนไคร์เวอร์

หน้าที่การทำงานของไคร์เวอร์เป็นส่วนที่ทำหน้าที่ในการร้องขอข้อมูลรายละเอียดของอุปกรณ์ การตั้งค่าการทำงานต่างๆ ให้กับอุปกรณ์ การกำหนดลำดับการสื่อสารข้อมูลของอุปกรณ์ ซึ่งในการทำงานลักษณะดังกล่าว มีการทำงานที่มีลักษณะเป็นลำดับ ในการออกแบบจึงออกแบบส่วนของไคร์เวอร์เป็นอุปกรณ์สเตตแมชชีน (State Machine) ซึ่งจะมีการทำงานในลักษณะเป็นลำดับต่อกันไป โดยในเปลี่ยนแต่ละลำดับการทำงาน ก็จะต้องมีการประมวลผลต่างๆ ซึ่งแตกต่างกันไปตามแต่ละประเภทของอุปกรณ์ ดังนั้น การออกแบบส่วนการทำงานของไคร์เวอร์จึงไม่ตายตัว แต่ขึ้นกับการนำไปใช้ ว่าต้องการใช้กับอุปกรณ์ใด และต้องการสื่อสารข้อมูลรวมถึงการตั้งค่าการทำงานอย่างไรบ้าง

แต่อย่างก็ตาม ขั้นตอนโดยทั่วไปสำหรับการติดต่อสื่อสารข้อมูลกับอุปกรณ์ยูเอสบี ก็มีขั้นตอนดังนี้

- 1) การตั้งค่าแอดเดรสของอุปกรณ์
- 2) การเลือกค่าคอนฟิกูเรชันของตัวอุปกรณ์
- 3) การเลือกอินเทอร์เฟซสำหรับติดต่อสื่อสารกับอุปกรณ์
- 4) การสื่อสารข้อมูล

บทที่ 4

การทดลองและผลการทดลอง

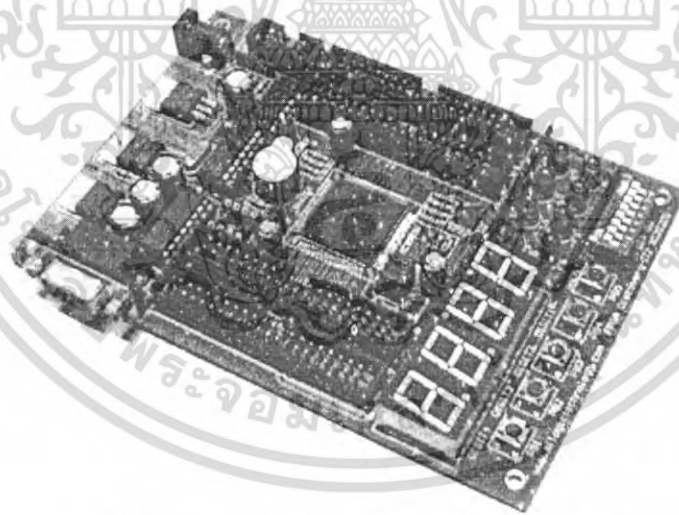
4.1 ลำดับขั้นการทดลอง

การทดลองจะแบ่งออกเป็นหลายขั้นตอน ตามลำดับขั้นของการออกแบบ ซึ่งได้แก่

- 1) การทดลองการทำงานของชั้นกายภาพ
- 2) การทดลองการทำงานของชั้นโปรโตคอล
- 3) การทดลองการทำงานของชั้นเครือข่าย

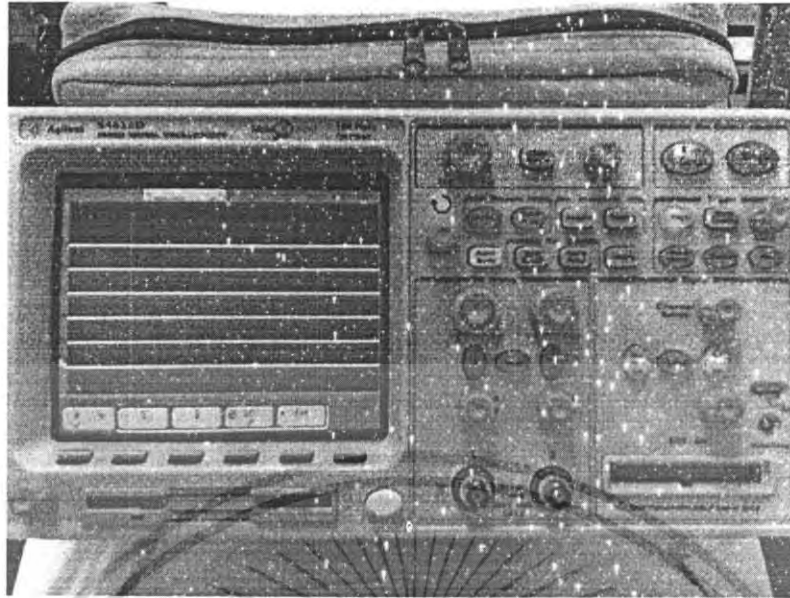
4.2 การทดลองการทำงานของชั้นกายภาพ

การทดลองการทำงานของส่วนการทำงานในชั้นกายภาพ จะใช้อุปกรณ์เอฟพีจีเอเป็นอุปกรณ์สำหรับการทดลองซึ่งแสดงดังรูปที่ 4.1 และใช้ออสซิลอสโคปสำหรับการตรวจวัดผลการทดลอง ซึ่งแสดงดังรูปที่ 4.2



รูปที่ 4.1 แสดงอุปกรณ์เอฟพีจีเอที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงอุปกรณ์ออสซิลโคปที่ใช้ในการตรวจวัดสัญญาณในการทดลอง

โดยในการทดลองการทำงานของส่วนการทำงานในชั้นกายภาพนั้น มีลำดับขั้นการทดลองต่างๆ ดังต่อไปนี้

4.2.1 การทดลองการทำงานของส่วนการเข้ารหัสข้อมูลด้วยการแทรกบิต

การทดลองการทำงานของส่วนการเข้ารหัสข้อมูลด้วยการแทรกบิต สามารถทดลองได้โดยการส่งข้อมูลบิต '0' และบิต '1' ด้วยความยาวต่างๆ กัน และดูผลการทำงานของอุปกรณ์ว่าสามารถทำการแปลงข้อมูลได้อย่างถูกต้องหรือไม่ โดยผลการทดลองจะแสดงดังรูปที่ 4.3



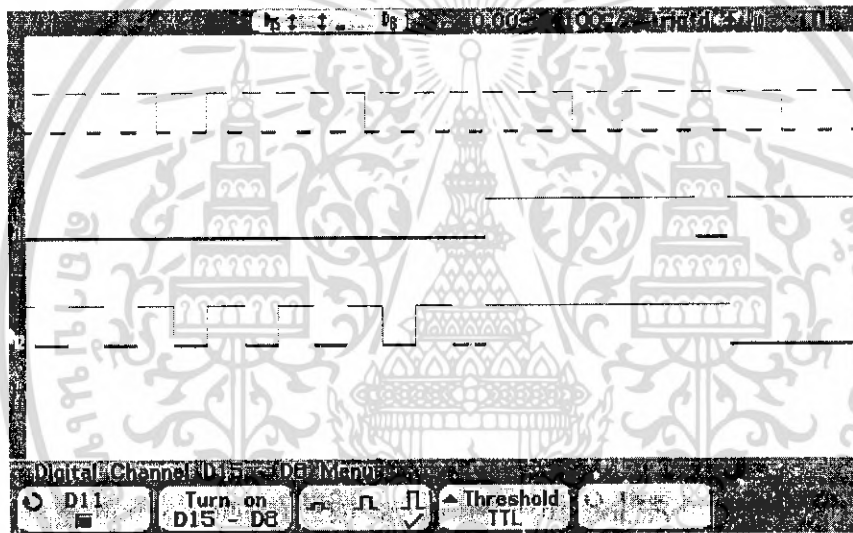
รูปที่ 4.3 แสดงผลการทดลองการทำงานของส่วนเข้ารหัสข้อมูลด้วยการแทรกบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองพบว่า เมื่อมีข้อมูลบิต '0' ติดต่อกัน ตัวอุปกรณ์ก็จะไม่ทำการเปลี่ยนแปลงข้อมูลใดๆ และส่งต่อข้อมูลดังกล่าว แต่เมื่อมีข้อมูลบิต '1' ติดกันเป็นจำนวน 6 บิต พบว่าบิตที่ 7 ของข้อมูลนั้น จะถูกแทรกด้วยข้อมูล '0'

4.2.2 การทดลองการเข้ารหัสข้อมูลแบบ NRZI

การทดลองการเข้ารหัสข้อมูลแบบ NRZI ทดลองโดยการส่งข้อมูลบิต '0' และบิต '1' ต่างๆ เข้าไปในตัวอุปกรณ์ ผลลัพธ์ของข้อมูลที่ได้นั้น จะต้องมีการสลับเปลี่ยนค่า (Toggle) เป็นค่าตรงกันข้าม เมื่อตรวจพบข้อมูลเข้าเป็นข้อมูลบิต '0' และจะคงค่าข้อมูลไว้ที่ค่าเดิม เมื่อตรวจพบเข้าเป็นข้อมูลบิต '1' ผลการทดลองการทำงานในส่วนการเข้ารหัสแบบ NRZI แสดงดังรูปที่ 4.4

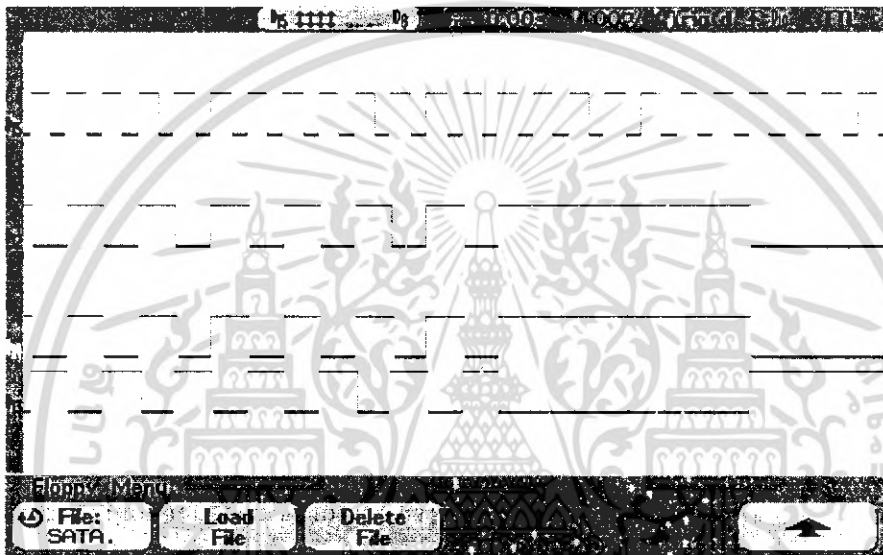


รูปที่ 4.4 แสดงผลการทดลองการทำงานของการเข้ารหัสข้อมูลแบบ NRZI

จากผลการทดลองการทำงานการเข้ารหัสข้อมูลแบบ NRZI พบว่า เมื่อมีการตรวจพบข้อมูลบิต '1' ผลลัพธ์ของข้อมูลที่ได้อาจจะไม่มีการเปลี่ยนแปลง และเมื่อมีการตรวจพบข้อมูลบิต '0' จะมีการเปลี่ยนแปลงของผลลัพธ์ของข้อมูล

4.2.3 การทดสอบการทำงานของส่วนการแปลงข้อมูลสำหรับการส่งออก

การทดสอบการทำงานของส่วนการแปลงข้อมูลสำหรับการส่งออก ทดลองโดยการส่งข้อมูลเข้าไปในอุปกรณ์ ผลลัพธ์ของสัญญาณจะต้องเป็นสัญญาณแบบผลต่าง โดยถ้าข้อมูลเข้าเป็นข้อมูลบิต '0' ผลลัพธ์ของสัญญาณจะได้ สัญญาณ D+ มีค่าเป็น '1' และสัญญาณ D- มีค่าเป็นบิต '0' (ในกรณีที่ใช้กับอุปกรณ์ความเร็วต่ำ สัญญาณ D+ และ D- จะมีลักษณะตรงกันข้าม) และในทางกลับกัน ถ้าข้อมูลเข้าเป็นข้อมูลบิต '1' ผลลัพธ์ของสัญญาณ D+ จะต้องเป็น '0' และสัญญาณ D- จะเป็น '1' ผลการทดสอบการทำงานของส่วนการแปลงข้อมูลสำหรับการส่งออก แสดงดังรูปที่ 4.5



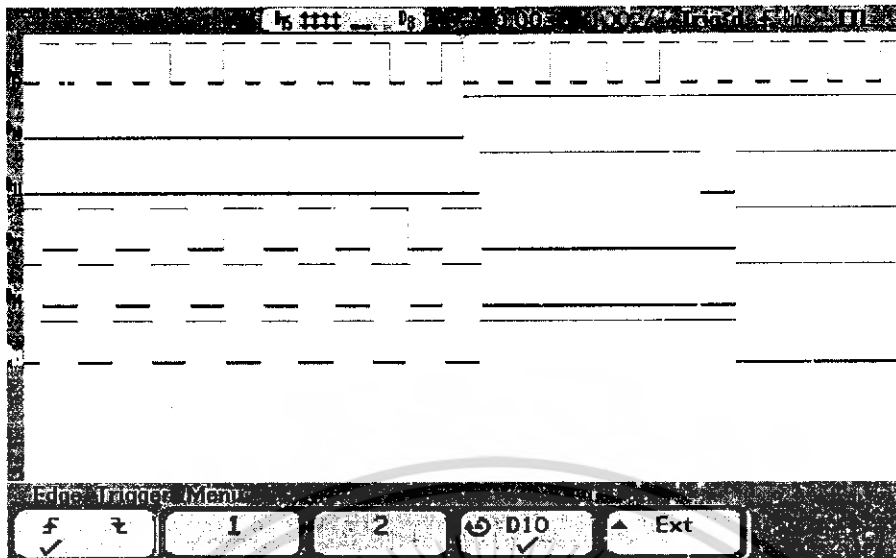
รูปที่ 4.5 แสดงผลการทดสอบการทำงานของส่วนการแปลงข้อมูลสำหรับการส่งออก

จากการทดสอบพบว่า เมื่อมีข้อมูลเป็นข้อมูลบิต '0' จะได้ผลลัพธ์ของสัญญาณ D+ ที่มีค่าเป็น '0' และสัญญาณ D- มีค่าเป็น '1' และในทางกลับกัน เมื่อมีว่ามีข้อมูลบิต '1' ทำให้สัญญาณ D+ มีค่าเป็น '1' และสัญญาณ D- เป็น '0'

4.2.4 การทดสอบการทำงานในการส่งข้อมูล

การทดสอบการทำงานในการส่งข้อมูลจะเป็นการนำส่วนการทำงานต่างๆ มาทดสอบการทำงาน โดยพร้อมๆ กัน ซึ่งในแต่ละส่วนการทำงานก็จะมี การส่งต่อข้อมูลไปยังส่วนการทำงานถัดไป ซึ่งตามลำดับของการส่งข้อมูลนั้น จะเริ่มจากการเข้ารหัสด้วยการแทรกบิต, การเข้ารหัสสัญญาณแบบ NRZI และการแปลงรูปแบบข้อมูลเป็นสัญญาณแบบผลต่าง ซึ่งผลการทดลองแสดงดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

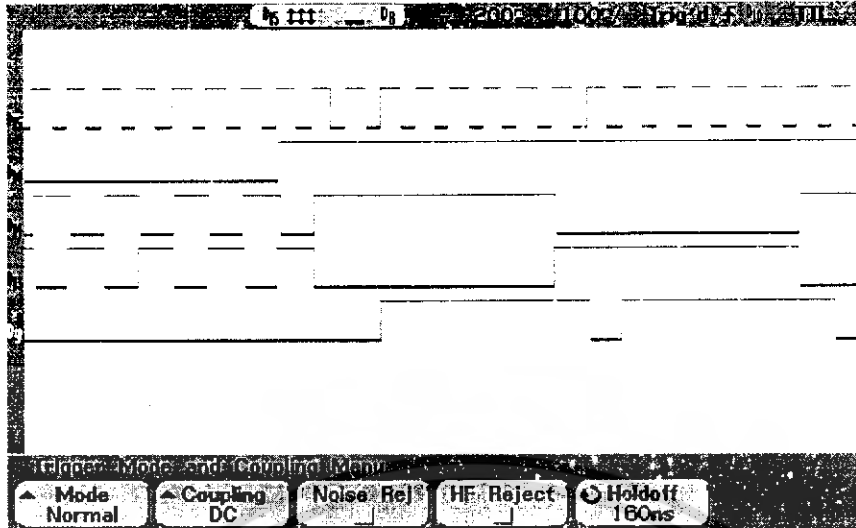


รูปที่ 4.6 แสดงผลการทดลองการทำงานในการส่งข้อมูล

จากการทดลองพบว่า ข้อมูลที่ต้องการจะส่งนั้น จะถูกทำการเข้ารหัสด้วยการแทรกบิตอย่างถูกต้อง และจึงผ่านการเข้ารหัสข้อมูลแบบ NRZI ก่อนที่จะทำการแปลงรูปแบบข้อมูลให้อยู่ในลักษณะของสัญญาณแบบผลต่าง ซึ่งพบว่าสามารถทำงานได้อย่างถูกต้อง

4.2.5 การทดลองการถอดรหัสข้อมูลแบบ NRZI

การทดลองการทำงานของส่วนการถอดรหัสข้อมูลแบบ NRZI จะเป็นการทดลองการทำงานของส่วนการทำงาน 2 ส่วนคือ การแปลงรูปแบบข้อมูลจากสัญญาณแบบผลต่างและผ่านการถอดรหัสข้อมูลแบบ NRZI ซึ่งวิธีในการทดลองทำได้โดยการส่งข้อมูลให้ส่วนการแปลงข้อมูลสำหรับการส่ง (ซึ่งทดลองในขั้นตอนที่ผ่านมาแล้ว และมีการทำงานอย่างถูกต้อง) จะได้ผลลัพธ์ของสัญญาณเป็นสัญญาณแบบผลต่าง แล้วจึงนำสัญญาณผลต่างนั้น มาใช้เป็นสัญญาณขาเข้าของส่วนการแปลงข้อมูลในการรับข้อมูลเข้า ผลลัพธ์ของการทำงานในการถอดรหัสข้อมูลแบบ NRZI จะได้ข้อมูลบิต '1' ถ้าพบว่าข้อมูลที่ผ่านการแปลงจากสัญญาณแบบผลต่างไม่มีการเปลี่ยนแปลง และได้ข้อมูลบิต '0' ถ้าพบการเปลี่ยนแปลงของข้อมูลที่ผ่านการแปลงสัญญาณแบบผลต่าง ผลการทดลองของการแปลงรูปแบบข้อมูลสำหรับรับเข้าและการถอดรหัสข้อมูลแบบ NRZI แสดงดังรูปที่ 4.7

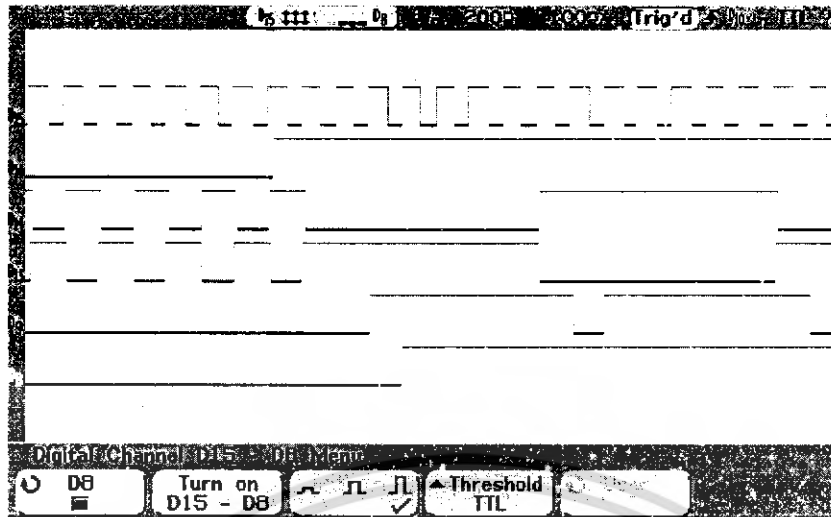


รูปที่ 4.7 แสดงผลการทดลองของการแปลงรูปแบบข้อมูลสำหรับรับเข้า
และการถอดรหัสข้อมูลแบบ NRZI

จากการทดลองการรับข้อมูล พบว่าเมื่อมีการรับสัญญาณแบบผลต่างแล้ว สามารถที่จะแปลความหมายของสัญญาณและทำการถอดรหัสข้อมูลแบบ NRZI และ ได้ผลลัพธ์ของข้อมูลที่ถูกต้อง

4.2.6 การทดลองส่วนการถอดรหัสข้อมูลการแทรกบิต

การทดลองการทำงานในการถอดรหัสข้อมูลการแทรกบิต จะทดลอง โดยการนำข้อมูลที่ผ่านการเข้ารหัสด้วยการแทรกบิต และการแปลงรูปแบบข้อมูลเป็นสัญญาณแบบผลต่างแล้ว มาเป็นข้อมูลเข้าในส่วนของการแปลงข้อมูลสำหรับการรับข้อมูลเข้า, ส่วนของการทำงานการถอดรหัสข้อมูลการแทรกบิต ผลลัพธ์ของข้อมูลที่ได้นั้น จะต้องเหมือนกับข้อมูลก่อนการเข้ารหัสด้วยการแทรกบิต โดยข้อมูลบิต '0' ที่ตามหลังข้อมูล '1' ที่ติดกันจำนวน 6 บิตจะถือว่าเป็นบิตแทรก ซึ่งจะต้องไม่รับข้อมูลบิต '0' นั้น ผลของการทดลองการทำงานการถอดรหัสข้อมูลการแทรกบิต แสดงดังรูปที่ 4.8



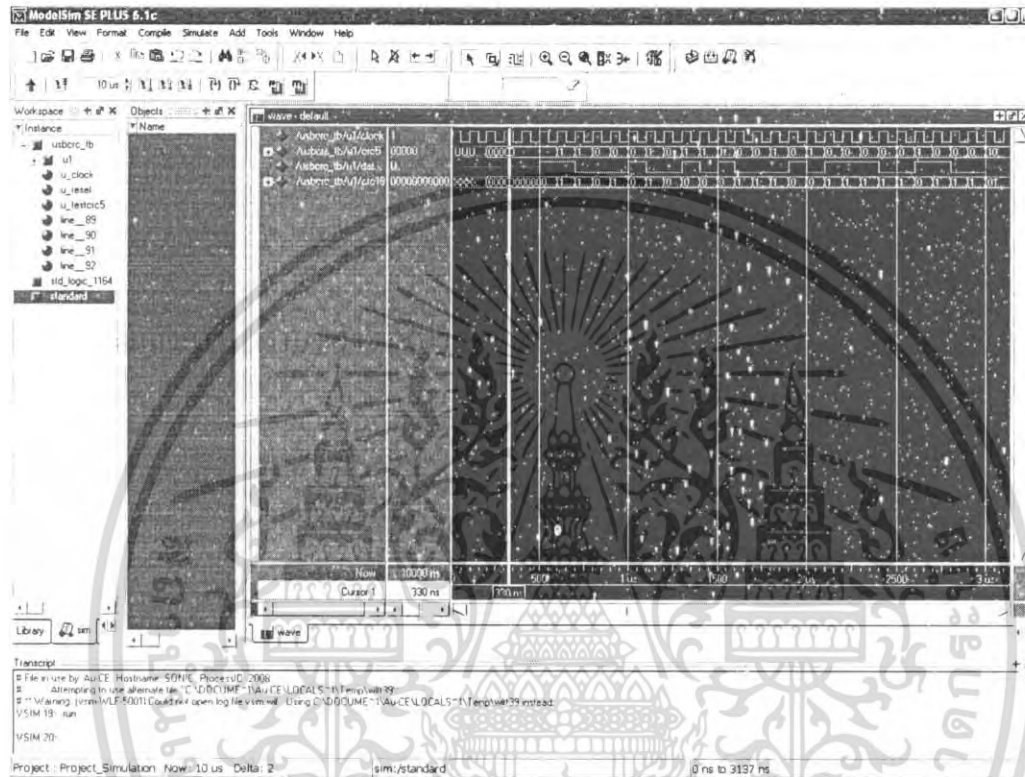
รูปที่ 4.8 แสดงผลการทดลองการทำงานของการทำงานของการแปลงข้อมูลในการรับข้อมูลเข้า และการถอดรหัสข้อมูลการแทรกบิต

จากการทดลองพบว่า เมื่อมีการรับข้อมูลจากส่วนการถอดรหัสข้อมูลแบบ NRZI แล้ว ผลลัพธ์ของข้อมูลที่ผ่านการถอดรหัสข้อมูลการแทรกบิตแล้ว ได้ลักษณะข้อมูลที่เหมือนกับข้อมูลที่ ถูกส่งมาอย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดลองการทำงานของชั้นโปรโตคอล

การทดลองการทำงานในชั้นโปรโตคอลจะใช้เครื่องมือในการทดสอบการทำงานคือโปรแกรมจำลองวงจรภาษาวีเอชดีแอล ชื่อโปรแกรม ModelSim พัฒนาโดยบริษัท Mentor Graphic โดยโปรแกรมที่นำมาใช้ทดลองจะเป็นเวอร์ชัน SE 6.1c ตัวอย่างโปรแกรมแสดงดังรูปที่ 4.9



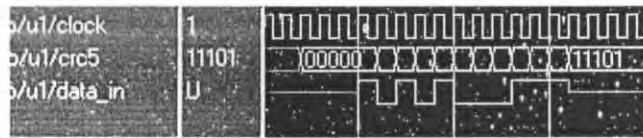
รูปที่ 4.9 แสดงตัวอย่างโปรแกรมจำลองการทำงาน Modelsim

4.3.1 การทดลองหาค่าตรวจสอบความถูกต้องของข้อมูล CRC ขนาด 5 บิต

การทดลองหาค่าตรวจสอบความถูกต้องของข้อมูล CRC ขนาด 5 บิต สามารถทดลองได้ โดยการส่งข้อมูลเข้าทีละ 1 บิต และเมื่อส่งข้อมูลครบทุกบิตแล้ว ข้อมูลผลลัพธ์คือค่าในการตรวจสอบความถูกต้องของข้อมูล ซึ่งในการทดลองหาค่าตรวจสอบความถูกต้องของข้อมูลขนาด 5 บิต จะใช้ข้อมูลตัวอย่างคือ “10101000111” ในการทดลอง ค่าตรวจสอบความถูกต้องของข้อมูลจะมีค่าคือ “10111” อ้างอิงจากเอกสาร “Cyclic Redundancy Checks in USB”

<http://www.usb.org/developers/whitepapers/crcdes.pdf> ซึ่งได้ผลการทดลองแสดงดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 แสดงผลการทดลองหาค่าตรวจสอบความถูกต้องของข้อมูลขนาด 5 บิต

จากการทดลองหาค่าข้อมูลตรวจสอบความถูกต้องขนาด 5 บิต ของข้อมูล “10101000111” ได้ผลลัพธ์คือ “10111” (ในรูปแสดงโดยเรียงตำแหน่งบิตสวนทางกัน) ซึ่งเป็นผลลัพธ์ที่ถูกต้อง แสดงให้เห็นว่า ส่วนการหาค่าข้อมูลตรวจสอบความถูกต้องขนาด 5 บิต มีการทำงานที่ถูกต้อง

4.3.2 การทดลองตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 5 บิต

การทดลองการทำงานของารตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 5 บิต (CRC5) ทดลองได้โดยการนำข้อมูลที่ต่อท้ายด้วยข้อมูลตรวจสอบ ส่งให้กับส่วนการทำงานในการตรวจสอบความผิดพลาดของข้อมูล โดยในการทดลอง จะส่งข้อมูล “10101000111” ที่เป็นข้อมูลและต่อท้ายด้วยข้อมูลตรวจสอบคือ “10111” ดังนั้น ข้อมูลที่จะถูกส่งเข้าไปตรวจสอบคือ “1010100011110111” ซึ่งถ้าข้อมูลที่ส่งเข้าไปนั้นไม่มีความผิดพลาด ผลของการตรวจสอบ (CRC Residual) จะได้เท่ากับ “01100” ซึ่งผลของการทดลองในการตรวจสอบความผิดพลาดแสดงดังรูปที่ 4.11

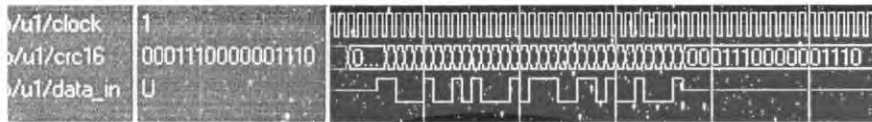


รูปที่ 4.11 แสดงผลการทดลองการตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 5 บิต

จากการทดลองการตรวจสอบความถูกต้องของข้อมูล “10101000111” ที่มีข้อมูลตรวจสอบคือ “10111” พบว่าผลลัพธ์ของการตรวจสอบคือ “01100” (ในรูปแสดงโดยเรียงตำแหน่งบิตสวนทางกัน) ซึ่งหมายความว่าข้อมูลดังกล่าวนี้ไม่มีความผิดพลาดของข้อมูล และแสดงให้เห็นว่าส่วนการทำงานตรวจสอบความผิดพลาดของข้อมูลขนาด 5 บิตนั้น มีการทำงานที่ถูกต้อง

4.3.3 การทดลองหาค่าตรวจสอบความถูกต้องของข้อมูล CRC ขนาด 16 บิต

การทดลองหาค่าตรวจสอบความถูกต้องของข้อมูล CRC ขนาด 16 บิต สามารถทดลองเหมือนกับการทดลองหาค่าตรวจสอบความถูกต้องของข้อมูลขนาด 5 บิตได้ โดยจะใช้ข้อมูลตัวอย่างคือ “11000100101000101110011010010001” ในการทดลอง ค่าตรวจสอบความถูกต้องของข้อมูลจะมีค่าคือ “0111000000111000” ซึ่งได้ผลการทดลองแสดงดังรูปที่ 4.12

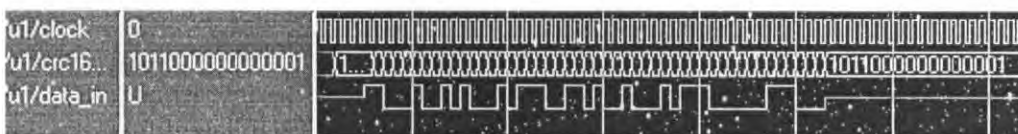


รูปที่ 4.12 แสดงผลการทดลองหาค่าตรวจสอบความถูกต้องของข้อมูลขนาด 16 บิต

จากการทดลองหาค่าข้อมูลตรวจสอบความถูกต้องขนาด 16 บิต ของข้อมูล “11000100101000101110011010010001” ได้ผลลัพธ์คือ “0111000000111000” (ในรูปแสดงโดยเรียงตำแหน่งบิตสวนทางกัน) ซึ่งเป็นผลลัพธ์ที่ถูกต้อง แสดงให้เห็นว่า ส่วนการหาค่าข้อมูลตรวจสอบความถูกต้องขนาด 16 บิต มีการทำงานที่ถูกต้อง

4.3.4 การทดลองตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 16 บิต

การทดลองการทำงานของตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 16 บิต (CRC16) ทดลองได้เช่นเดียวกับการทดลองการตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 5 บิต โดยในการทดลอง จะทำการส่งข้อมูลตัวอย่าง “11000100101000101110011010010001” และต่อท้ายด้วยข้อมูลตรวจสอบคือ “0111000000111000” ดังนั้น ข้อมูลที่จะถูกส่งเข้าไปตรวจสอบคือ “110001001010001011100110100100010111000000111000” ซึ่งถ้าข้อมูลที่ถูส่งเข้าไบนั้นไม่มีความผิดพลาด ผลของการตรวจสอบ (CRC Residual) จะได้เท่ากับ “1000000000001101” ซึ่งผลของการทดลองในการตรวจสอบความผิดพลาดแสดงดังรูปที่ 4.13



รูปที่ 4.13 แสดงผลการทดลองการตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 16 บิต

จากการทดลองการตรวจสอบความถูกต้องด้วยข้อมูลตรวจสอบขนาด 16 บิต ของข้อมูล “11000100101000101110011010010001” ที่มีข้อมูลตรวจสอบคือ “0111000000111000” พบว่า ผลลัพธ์ของการตรวจสอบคือ “1000000000001101” (ในรูปแสดงโดยเรียงตำแหน่งบิตสวนทางกัน) ซึ่งหมายความว่าข้อมูลดังกล่าวนั้น ไม่มีความผิดพลาดของข้อมูล และแสดงให้เห็นว่าส่วนการทำงานตรวจสอบความผิดพลาดของข้อมูลขนาด 16 บิตนั้น มีการทำงานที่ถูกต้อง

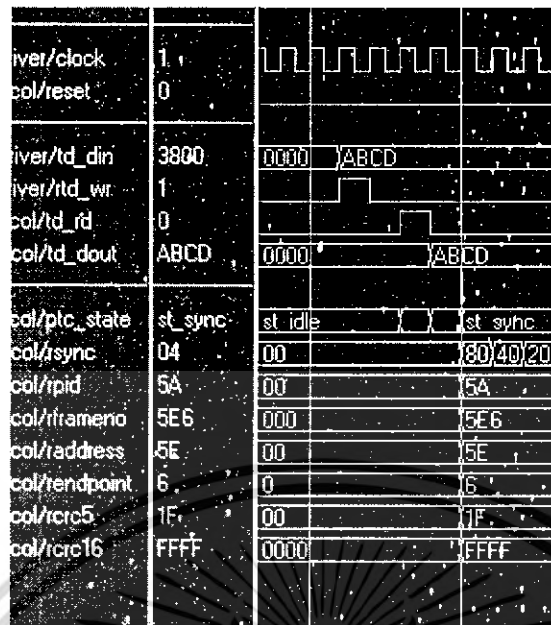
4.4 การทดลองการทำงานของส่วนไคร์เวอร์ร่วมกับการทำงานชั้นโปรโตคอล

ในการทดลองการทำงานของส่วนการทำงานชั้น โปรโตคอล มีความจำเป็นที่จะต้องทดลองร่วมกับส่วนของไคร์เวอร์ เพื่อให้ไคร์เวอร์ทำการส่งงานต่างๆ ให้ชั้น โปรโตคอลทำงานตามที่ต้องการ ดังนั้น ในการทดลองขั้นต่อไปจึงต้องใช้ส่วนการทำงานของไคร์เวอร์ทดลองการทำงานร่วมกับส่วนการทำงานในชั้นโปรโตคอล

4.4.1 การทดลองเขียนข้อมูลทรานแซคชัน เดสคริปเตอร์

เนื่องจากหน้าที่หลักของไคร์เวอร์ในส่วนที่ต้องทำงานร่วมกับการทำงานในชั้นโปรโตคอล คือ การเขียนทรานแซคชัน เดสคริปเตอร์ลงใน กิวของทรานแซคชัน เดสคริปเตอร์ ดังนั้น ในการทดลองขั้นตอนต่อไป จึงเป็นการทดลองในการให้ส่วนไคร์เวอร์ทำการเขียนข้อมูลทรานแซคชัน เดสคริปเตอร์ลงกิว

ในการทดลอง จะทำการทดลองโดยให้ไคร์เวอร์ทำการเขียนข้อมูลทรานแซคชัน เดสคริปเตอร์ด้วยค่าข้อมูล ABCDH ซึ่งได้ผลการทดลองดังรูปที่ 4.14



รูปที่ 4.14 แสดงผลการทดลอง โดยการเขียนข้อมูล ABCDH ลงในทรานแซคชัน เดสคริปเตอร์

จากการทดลอง จะเห็นได้ว่า เมื่อไคร้เวอร์ทำการเขียนข้อมูลลงในทรานแซคชัน เดสคริปเตอร์ ด้วยค่า ABCDH ที่ส่งมาจากสัญญาณ id_din และส่งสัญญาณให้เขียนข้อมูล id_wr แล้ว การทำงานของชั้น โปรโตคอลจะทำการส่งสัญญาณอ่านข้อมูล td_rd มาอ่านข้อมูลทรานแซคชัน เดสคริปเตอร์ แล้วจึงนำไปแปลความหมายตามตำแหน่งบิตข้อมูลต่างๆ โดยบิตข้อมูลทั้งหมดคือ “1010101111001101” ซึ่งทำให้การแปลความหมายของค่าต่างๆ ได้ดังนี้

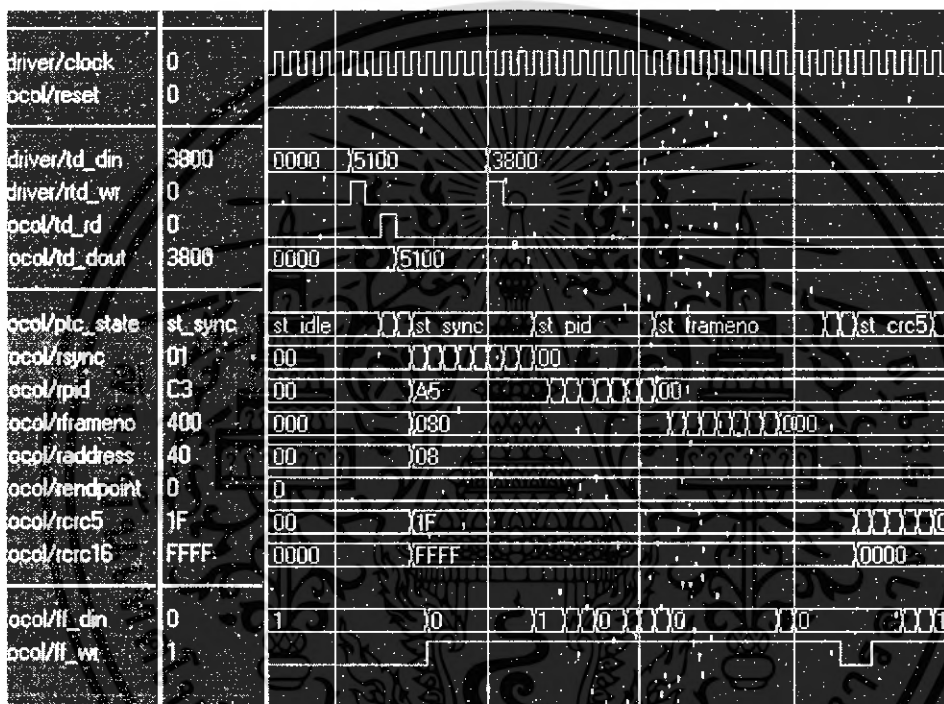
- ส่วนข้อมูลซิงโครไนซ์ มีค่าคงที่คือ 80H
- หมายเลขประจำแพ็คเกจ มีค่าข้อมูลคือ “1010” โดยจะนำไปไว้ยังตำแหน่ง 4 บิตล่างและนำค่าคอมพลิเมนต์ไว้ที่ตำแหน่ง 4 บิตบน ทำให้มีค่าข้อมูลคือ 5AH
- หมายเลขเฟรม มีค่าข้อมูลคือ “10111100110” หรือ 5E6H
- ตำแหน่งแอดเดรส มีค่าข้อมูลคือ “101110” หรือ 5EH
- ตำแหน่งเอนพอยน์ทน์ มีค่าข้อมูลคือ “0110” หรือ 6H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 การทดสอบการทำงานของชั้นโปรโตคอลเมื่อได้รับทรานแซกชันเดสคริปเตอร์

เมื่อส่วนการทำงานในชั้นโปรโตคอลได้รับข้อมูลทรานแซกชันเดสคริปเตอร์แล้ว จะต้องทำการอ่านค่า และสร้างแพ็กเก็ตข้อมูลตามแต่ละรูปแบบของแพ็กเก็ต

ในการทดลอง จะทดลองโดยการเขียนทรานแซกชันเดสคริปเตอร์ด้วยข้อมูล 5100H ซึ่งเป็นการระบุข้อมูลหมายเลขประจำแพ็กเก็ตคือ "0101" ซึ่งคือแพ็กเก็ตชนิด SOF และมีค่าหมายเลขเฟรมเป็น 400H และมีผลการทดลองดังรูปที่ 4.15



รูปที่ 4.15 แสดงผลการทดสอบการทำงานของชั้นโปรโตคอล
เมื่อเขียนข้อมูลทรานแซกชันเดสคริปเตอร์ด้วยค่า 5100H

จากการทดลอง การทำงานของชั้นโปรโตคอลมีลำดับการสร้างการสร้างแพ็กเก็ตที่ถูกต้อง ถูกต้อง โดยดูจากลำดับการทำงานในสัญญาณ ptc_state ที่จะเริ่มต้นด้วยการสร้างฟิลด์ข้อมูล SYNC และตามด้วยฟิลด์ข้อมูล PID, FrameNumber และ CRC5 ตามลำดับ ซึ่งในการสร้างแพ็กเก็ตนั้นก็คือ การเขียนข้อมูลแพ็กเก็ตลงในหน่วยความจำ FIFO ผ่านสัญญาณ ff_din และควบคุมการเขียนข้อมูล โดยสัญญาณ ff_wr

4.4.3 การทดลองการทำงานของชั้นโปรโตคอลในการส่งข้อมูลให้กับชั้นการทำงานระดับกายภาพ

เมื่อส่วนของชั้นโปรโตคอลทำการสร้างแพ็กเก็ตข้อมูลแล้ว จะมีการส่งข้อมูลแพ็กเก็ตดังกล่าวให้กับส่วนการทำงานในระดับกายภาพ เพื่อสร้างเป็นสัญญาณส่งไปยังตัวอุปกรณ์ยูเอสบี การทดลองจะเป็นการทดลองต่อจากขั้นตอนที่แล้ว คือการเขียนข้อมูลทรานแซกชันเดสคริปเตอร์ด้วยค่าข้อมูล 5100H โดยจะสังเกตผลการทดลองเมื่อชั้นโปรโตคอลมีการสร้างแพ็กเก็ตข้อมูลและเขียนลงบนหน่วยความจำ FIFO แล้ว โดยผลการทดลองที่ได้จะเป็นการแสดงผลการอ่านข้อมูลจากหน่วยความจำ FIFO เพื่อให้ชั้นการทำงานระดับกายภาพนำไปสร้างเป็นสัญญาณผลต่าง โดยผลการทดลองแสดงดังรูปที่ 4.16

col/ff_din	No Data	1							
col/ff_wr	No Data	1							
col/data_state	No Data	0	1	0	1	0	1	0	1
col/ff_rd	No Data	0	1	0	1	0	1	0	1
col/ff_dout	No Data	0	1	0	1	0	1	0	1

รูปที่ 4.16 แสดงผลการทดลองของการส่งข้อมูลให้

จากการทดลอง พบว่า เมื่อมีการเขียนข้อมูลแพ็กเก็ตข้อมูลลงในหน่วยความจำ FIFO แล้ว สัญญาณในการอ่านข้อมูล ff_rd จะถูกส่งเพื่อทำการอ่านข้อมูลแพ็กเก็ตออกจากหน่วยความจำ โดยจะได้รับข้อมูลแพ็กเก็ตทางสัญญาณ ff_dout

4.4.4 การทดลองการนำข้อมูลแพ็กเก็ตจากชั้นโปรโตคอลไปสร้างสัญญาณแบบผลต่าง

หลังจากที่มีการส่งแพ็กเก็ตข้อมูลจากชั้นโปรโตคอลมายังชั้นกายภาพแล้ว ในชั้นการทำงานระดับกายภาพจะต้องมีการนำข้อมูลไปทำการสร้างสัญญาณแบบผลต่าง ซึ่งการทดลองจะเป็นการทดลองต่อเนื่องจากขั้นตอนที่แล้ว คือเมื่อชั้นไดร์เวอร์ทำการเขียนข้อมูลทรานแซกชันเดสคริปเตอร์ด้วยค่าข้อมูล 5100H แล้ว ผลลัพธ์ของการทดลองจะได้เป็นสัญญาณแบบผลต่างของข้อมูลแพ็กเก็ตที่ถูกสร้างขึ้นโดยชั้นโปรโตคอลได้หรือไม่ โดยผลการทดลองแสดงดังรูปที่ 4.17

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทสรุป

สรุปผลการดำเนินงานในการทำโครงการ พบว่า การทำงานของโฮสต์ยูเอสบีที่ผ่านทดสอบโดยการจำลอง (Simulate) แล้วสามารถทำงานได้อย่างถูกต้องนั้น มีดังต่อไปนี้

- 1) การเข้ารหัสและการถอดรหัสข้อมูลด้วยการแทรกบิต
- 2) การเข้ารหัสและการถอดรหัสข้อมูลแบบ NRZI
- 3) การแปลงข้อมูลให้อยู่ในรูปแบบของสัญญาณแบบผลต่าง
- 4) การแปลงสัญญาณแบบผลต่างเป็นข้อมูล
- 5) การทำงานของการสร้างข้อมูลตรวจสอบความผิดพลาดขนาด 5 บิตและ 16 บิต
- 6) การตรวจสอบความผิดพลาดของข้อมูลด้วยข้อมูลตรวจสอบขนาด 5 บิตและ 16 บิต
- 7) การทดลองการทำงานของไมโครเวร์ในการเขียนทรานแซกชันเดสคริปเตอร์
- 8) การทำงานของชั้นโปรโตคอลในการอ่านข้อมูลทรานแซกชันเดสคริปเตอร์
- 9) การประมวลผลทรานแซกชันเดสคริปเตอร์ เพื่อสร้างแพ็กเก็ตข้อมูล
- 10) การส่งข้อมูลไปยังส่วนการทำงานชั้นกายภาพ
- 11) การส่งข้อมูลและรับข้อมูลของชั้นกายภาพ

แต่อย่างไรก็ตาม พบว่า เมื่อทำการทดลองการทำงานกับอุปกรณ์ยูเอสบีแล้ว ตัวอุปกรณ์นั้นไม่มีการตอบสนองต่อคำสั่งที่โฮสต์ส่งไปให้ ทำให้ไม่สามารถทำการพัฒนาการทำงานของโฮสต์ยูเอสบีต่อไปได้ ทั้งนี้ สาเหตุของความผิดพลาดนั้น อาจเกิดได้หลายสาเหตุ เช่น การไม่สัมพันธ์กันทางด้านเวลา, การเกิดความล่าช้าของสัญญาณ, การถูกรบกวนจากสัญญาณภายนอก, ลักษณะเฉพาะทางอิเล็กทรอนิกส์ ซึ่งควรจะถูกรตรวจสอบและแก้ไข สำหรับการพัฒนาต่อไป และจากสาเหตุข้างต้น จึงส่งผลให้ผลจากการดำเนินงานของโครงการนั้น ไม่สามารถบรรลุจุดประสงค์และขอบเขตของโครงการได้

5.2 วิจารณ์สิ่งที่ได้จากโครงการ

ไอบริารีของการติดต่อสื่อสารข้อมูลด้วยยูเอสบีนั้น ยังมีส่วนที่ยังพัฒนาได้ไม่เสร็จสมบูรณ์ ซึ่งได้แก่ส่วนในการรับข้อมูลเข้ามาจากอุปกรณ์ยูเอสบี, และการปรับตั้งการทำงานโดยคำนึงถึงผลของเวลาให้เหมาะสมกับการสื่อสารข้อมูล

ในการทดลองการทำงานต่างๆ ของโครงการ พบว่าสามารถทดลองการทำงานจากการนำไปจำลองด้วยโปรแกรมจำลองการทำงานที่ได้ผลการทดลองที่ถูกต้อง แต่เมื่อทำการอิมพลีเมนต์ (Implement) วงจรที่ทำการออกแบบเข้าสู่อุปกรณ์เอพพิเจีเอ ในการใช้งานจริงนั้น มีสิ่งที่มีผลต่อการทำงานเพิ่มขึ้นคือ การล่าช้าของสัญญาณ, การผิดเพี้ยนของสัญญาณ, การไม่เข้าจังหวะในการสื่อสารข้อมูล, การได้รับสัญญาณรบกวน ซึ่งจากผลกระทบดังกล่าวนี้ ส่งผลให้การทดลองการทำงานกับอุปกรณ์จริงไม่เกิดผลลัพธ์ในการทดลองตามที่ต้องการ เนื่องจากตัวอุปกรณ์นั้น ไม่มีการตอบสนองต่อคำสั่งใดๆ มีเพียงการตอบสนองต่อสัญญาณเท่านั้น

5.3 ปัญหาอุปสรรคและแนวทางในการแก้ไข

ปัญหาของการพัฒนาโครงการนั้น มีปัญหาหลักอยู่ที่การไม่ตอบสนองต่อคำสั่งของอุปกรณ์ ซึ่งส่งผลให้การพัฒนาและการทดลองในขั้นตอนอื่นๆ ไม่สามารถที่จะดำเนินการต่อไปได้ ทั้งนี้ เนื่องจากการพัฒนาและการทดลองในขั้นตอนต่อไปนั้น มีความจำเป็นที่จะต้องให้ตัวอุปกรณ์มีการตอบสนองต่อคำสั่ง เช่น การทดลองการส่งคำสั่งในการปรับตั้งค่า ซึ่งเมื่ออุปกรณ์ไม่ตอบสนองคำสั่ง จึงไม่สามารถทดลองได้ว่าสามารถส่งคำสั่งได้อย่างถูกต้องและมีผลการทำงานที่สำเร็จหรือไม่ หรือการทดลองการรับข้อมูลจากอุปกรณ์ ก็ไม่สามารถทดลองได้เช่นกัน เนื่องจากการรับข้อมูลจากตัวอุปกรณ์นั้น จะต้องส่งแพ็คเกจข้อมูลที่ระบุทิศทางของข้อมูลให้ส่งจากอุปกรณ์ไปยังโฮสต์ ซึ่งปัญหาต่างๆ ในการพัฒนาและแนวทางการแก้ไข สามารถสรุปได้ดังต่อไปนี้

- 1) ในการส่งข้อมูลไปให้กับตัวอุปกรณ์ แล้วตัวอุปกรณ์ไม่ตอบสนองต่อคำสั่ง ทั้งนี้ เนื่องจากการพัฒนานั้น ไม่สามารถติดตามผลการทำงานของการสื่อสารได้ ยังไม่สามารถสรุปได้อย่างชัดเจนว่าสาเหตุของการไม่ตอบสนองต่อคำสั่งนั้น เกิดจากสาเหตุใดในการแก้ไวนั้น ควรจะใช้อุปกรณ์ยูเอสบี ที่ใช้ทดลองเป็นอุปกรณ์ประเภทไมโครคอนโทรลเลอร์ก่อน โดยให้ตัวไมโครคอนโทรลเลอร์ทำการรายงานสถานะและผลการประมวลผลคำสั่งที่ได้รับ ทำให้สามารถทราบถึงสาเหตุของความผิดพลาดและแก้ไขให้ตรงสาเหตุของปัญหาได้
- 2) ในส่วนของการรับข้อมูลนั้น ยังไม่ได้ทำการทดลองกับอุปกรณ์จริง สาเหตุเนื่องจากการที่อุปกรณ์ไม่ตอบสนองต่อคำสั่ง ทำให้ไม่สามารถสั่งให้ตัวอุปกรณ์ทำการส่งข้อมูลออกมาจากตัวอุปกรณ์ได้ จึงไม่สามารถทดลองการทำงานในส่วนของการรับข้อมูลได้

- 3) แต่อย่างไรก็ตาม ถ้าอุปกรณ์มีการส่งข้อมูลออกมาแล้ว ปัญหาที่คาดว่าจะพบในการรับข้อมูลคือ การสื่อสารข้อมูลที่ไม่เข้าจังหวะกัน (Loss Synchronize) ส่งผลให้การรับข้อมูลมีการผิดเพี้ยนไป และทำให้มีการตีความหมายข้อมูลที่ผิดพลาด ดังนั้น ในการแก้ไขปัญหาดังกล่าว สามารถทำได้โดยการทำการเข้าจังหวะข้อมูลด้วยสัญญาณนาฬิกาภายในก่อน (Clock Synchronization) โดยอาศัยส่วนข้อมูล SYNC ที่ถูกส่งมาจากตัวอุปกรณ์เพื่อกำหนดจังหวะการรับข้อมูลของโฮสต์ให้เข้าจังหวะกับตัวอุปกรณ์ ทำให้ได้รับข้อมูลที่ถูกต้อง
- 4) การทำงานของไมโครโพรเซสเซอร์ที่มีลักษณะการทำงานแบบเป็นลำดับขั้น ซึ่งในลักษณะการทำงานดังกล่าวนี้ สามารถที่จะทำการพัฒนาโดยใช้อุปกรณ์ประเภทไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ได้ จะมีความง่ายต่อการพัฒนามากกว่าการใช้อุปกรณ์ FPGA ในการสร้างส่วนของไมโครโพรเซสเซอร์

5.4 แนวทางการพัฒนาต่อ

- 1) การพัฒนาในส่วนของการส่งข้อมูล ให้ตัวอุปกรณ์มีการตอบสนองต่อคำสั่งต่างๆ โดยพิจารณาเรื่องเกี่ยวกับเวลาและรูปแบบของสัญญาณให้ถูกต้อง
- 2) การพัฒนาในส่วนของการรับข้อมูล ให้สามารถทำการรับข้อมูลได้อย่างถูกต้อง
- 3) เมื่อมีการทำงานภายในที่ถูกต้องแล้ว การพัฒนาขั้นต่อไปคือการพัฒนาการทำงานของโฮสต์ให้มีลักษณะการทำงานกับอุปกรณ์ยูเอสบี แบบความเร็วสูงได้ (HS : High Speed Device) โดยมีลักษณะการทำงานตามมาตรฐานของยูเอสบี เวอร์ชัน 2.0
- 4) การออกแบบโครงสร้างของไมโครโพรเซสเซอร์ให้มีลักษณะเป็นโครงสร้างแบบหลายชั้น (MultiLayer) เพื่อรองรับการทำงานของหลายๆ อุปกรณ์บนสายสัญญาณบีเอสยูเอสบีเส้นเดียว โดยที่ไมโครโพรเซสเซอร์ชั้นล่างสุด จะเป็นส่วนที่ทำการส่งข้อมูลให้กับส่วนของโปรโตคอล และไมโครโพรเซสเซอร์ชั้นบนจะมีหลายตัว ซึ่งแต่ละตัว จะทำการติดต่อกับอุปกรณ์แต่ละอุปกรณ์
- 5) การพัฒนาการประมวลผลข้อมูลของคำสั่งเฉพาะเจาะจงของอุปกรณ์ในแต่ละคลาสของอุปกรณ์

บรรณานุกรม

- [1] Beyond Logic. "USB in a Nutshell – Making sense of the USB standard." [Online]. Available : <http://www.beyondlogic.org/usbnutshell/usb1.htm>.
- [2] Don Anderson 1997. USB System Architecture, Addison-Wesley Developers Press
- [3] Intel. "Universal Host Controller Interface (UHCI) Design Guide." [Online]. Available : <http://developer.intel.com/technology/usb/uhci11d.htm>. 1996.
- [4] USB.org. "Universal Serial Bus Revision 1.1 specification." [Online]. Available : <http://www.usb.org/developers/docs/usbspec.zip>. 1998
- [5] USB.org. "Cyclic Redundancy Checks in USB." [Online]. Available : <http://www.usb.org/developers/whitepapers/credes.pdf>.

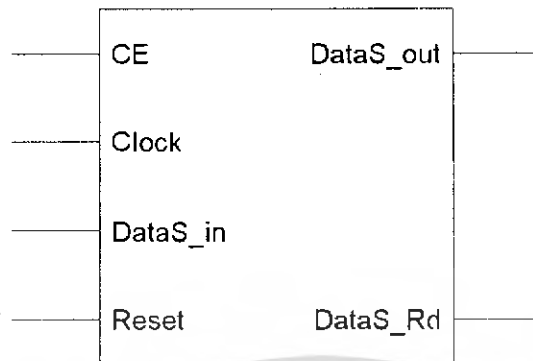
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



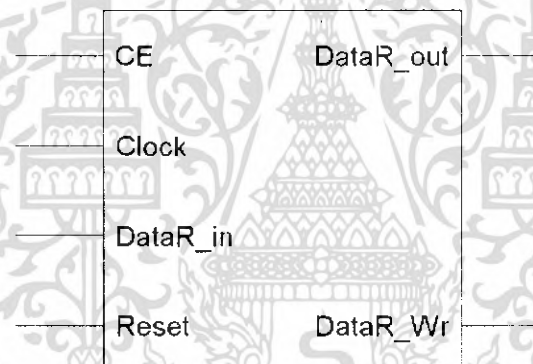
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

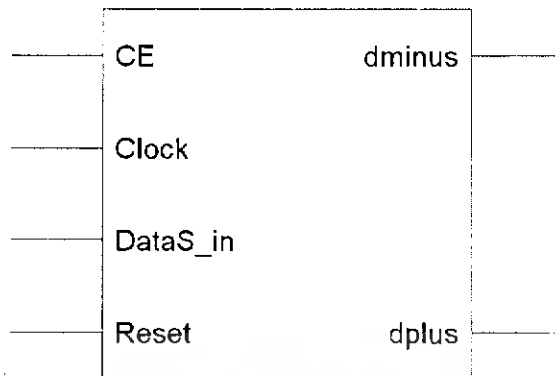


รูปแสดงรายละเอียดของส่วนเข้ารหัสข้อมูลด้วยการแทรกบิต

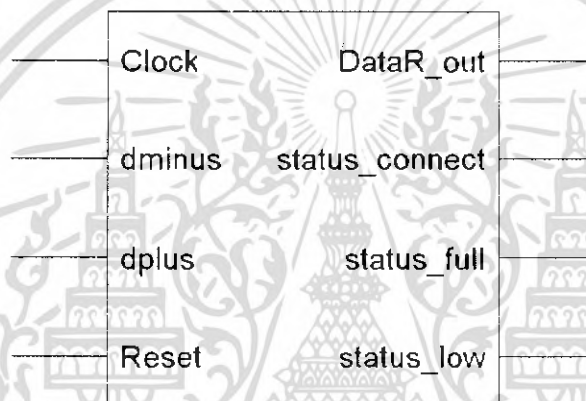


รูปแสดงรายละเอียดของส่วนถอดรหัสข้อมูลด้วยการแทรกบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

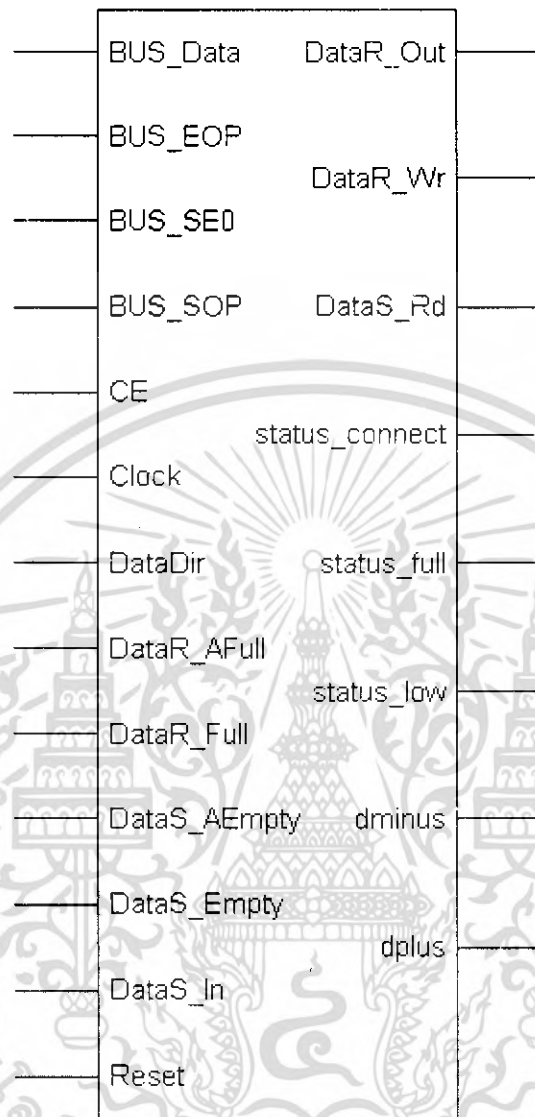


รูปแสดงรายละเอียดของส่วนการเปลี่ยนรูปแบบข้อมูลสำหรับส่งออก



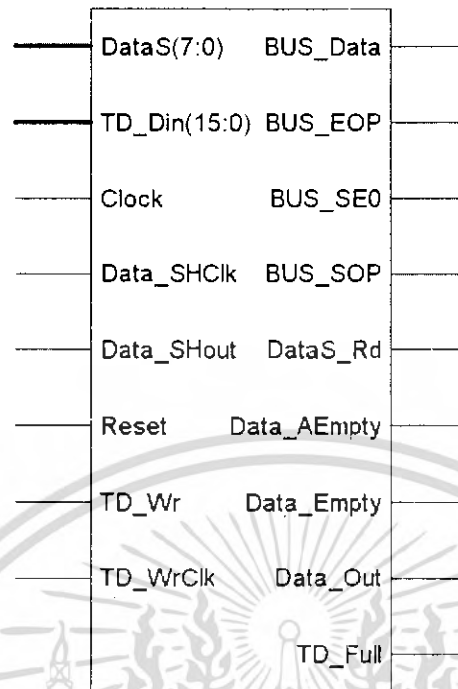
รูปแสดงรายละเอียดของส่วนการเปลี่ยนรูปแบบข้อมูลสำหรับรับเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงรายละเอียดของส่วนการทำงานชั้นกายภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงรายละเอียดของส่วนการทำงานชั้น โปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้