สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โปรโตคอลอจิกอนาไลเซอร์

PROTOCOLOGIC ANALYZER

T055113

โดย

นายกฤษดากร ไชยชุมภู

นายชัยกร แซ่เซียว

อาจารย์ที่ปรึกษา

ดร. สุรินทร์ กิตติธรกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่
เลขทะเบียน 55113
วัน,เดือน,ปี - 8 พ.ย. 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2546

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรโตคอลอจิกอนาไลเซอร์

PROTOCOLOGIC ANALYZER

คณะผู้จัดทำ นายกฤษดากร ไชยชุมภู รหัส 43010015

นายชัยกร แซ่เซียว รหัส 43010179

_____ อาจารย์ที่ปรึกษา

(ดร. สุรินทร์ กิตติธรกุล)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลอจิกอนาไลเซอร์

นายกฤษดากร ไชยชุมภู 43010015
นายชัยกร แซ่เซียว 43010179
ดร. สุรินทร์ กิตติธรกุล อาจารย์ที่ปรึกษา
ปีการศึกษา 2546

บทคัดย่อ

โปรโตคอลอจิกอนาไลเซอร์ เป็นโปรแกรมที่ใช้วิเคราะห์สัญญาณซึ่งวัดมาจากสายส่งสัญญาณที่ใช้ในการสื่อสารข้อมูล (Data communication) และวงจรดิจิตอล โปรแกรมโปรโตคอลอจิกอนาไลเซอร์ทำหน้าที่วิเคราะห์สัญญาณที่วัดจากสายส่งสัญญาณ (Protocol analyze) ว่ามีการส่งข้อมูลอะไรบ้าง โดยทำการแยกแยะสัญญาณควบคุมและข้อมูลได้อย่างชัดเจน เช่น สัญญาณควบคุม (Control Word) และสัญญาณตอบรับ (Acknowledge) เป็นต้น นอกจากนี้ โปรแกรมโปรโตคอลอจิกอนาไลเซอร์ มีความสามารถในการวิเคราะห์สัญญาณจากวงจรดิจิตอลทั่วไป เช่น รวมกลุ่มของสัญญาณเพื่อช่วยให้ดูข้อมูลได้ง่ายขึ้น (group signal) ใส่ตัวชี้ตำแหน่งเพื่อให้ง่ายต่อการเปรียบเทียบข้อมูลหลายๆ จุด (place cursor) และค้นหารูปแบบของสัญญาณที่วัดมาได้ (search pattern) เป็นต้น แล้วแสดงผลข้อมูลที่วิเคราะห์ได้ในรูปแบบของกราฟการเปลี่ยนแปลงของระดับสัญญาณ (Waveform) และในรูปแบบเชิงตัวเลขทางตรรกะ เช่น เลขฐาน 2 และเลขฐาน 16 เป็นต้น

โดยภาพรวม โปรแกรมโปรโตคอลอจิกอนาไลเซอร์ สามารถนำไปใช้ในการตรวจสอบหาข้อผิดพลาดของการสื่อสารข้อมูล (ทั้งชนิด Serial และ Parallel) และวงจรดิจิตอลได้ ซึ่งสามารถตรวจสอบว่าอุปกรณ์ที่สร้างขึ้นมาทำงานได้ถูกต้องหรือไม่ อาจใช้ตรวจสอบความถูกต้องของไอซี (IC) ไมโครคอลโทรเลอร์ (Microcontroller) หรืออุปกรณ์ที่สามารถโปรแกรมลงไปได้ เช่น พีแอลดี (Programmable Logic Device : PLD) เอฟพีจีเอ (Field Programmable Gate Array : FPGA) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROTOCOLOGIC ANALYZER

Kridsadakorn Chaichoompu 43010015
Chaiyakorn Saeseow 43010179
Dr. Surin Kittitornkun Advisor
Academic Year 2003

ABSTRACT

Protocologic Analyzer is a signal analysis software for data communication and digital circuit. Protocologic Analyzer software uses for analyzing signals in cable (protocol analysis) to find data that separated from control signals: such as control word and acknowledge, etc. Protocologic Analyzer software also uses for analyzing signals in digital circuit: signals grouping, cursors placing, and pattern searching, etc: and displays in waveform or number format (binary number and hexadecimal number).

Protocologic Analyzer software can verify an error in serial/parallel data communication and can react the same way as your digital circuit does. Protocologic Analyzer software also uses for verifying output of microcontroller if you need to look at signals in the same way your microcontroller does, and it can use for IC (Integrated Circuit), (Programmable Logic Device) and FPGA (Field Programmable Gate Array), too.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลายๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้ปริญญานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์สุรินทร์ กิตติธรกุล อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

และต้องขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดา มารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

กฤษดากร ไชยชุมภู

ชัยกร แซ่เซียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้าที่

บทคัดย่อภาษาไทย I
บทคัดย่อภาษาอังกฤษ II
สารบัญ IV
สารบัญรูปภาพ VII
สารบัญตาราง XI
บทที่ 1 บทนำ 1
1.1 ความสำคัญและที่มา 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 เป้าหมายของโครงงาน 2
1.4 ปัญหาและประโยชน์ของโครงงาน 2
1.5 ขอบเขตของโครงงาน 2
1.5.1 ขอบเขตของส่วนซอฟต์แวร์โดยสรุป 2
1.5.2 ขอบเขตของส่วนฮาร์ดแวร์โดยสรุป 2
1.6 ขั้นตอนการพัฒนา 3
1.6.1 ขั้นตอนการพัฒนาของส่วนซอฟต์แวร์โดยสรุป 3
1.6.2 ขั้นตอนการพัฒนาของส่วนฮาร์ดแวร์โดยสรุป 3
บทที่ 2 เทคโนโลยีพื้นฐานและระบบที่มีอยู่ 4
2.1 การศึกษาและสำรวจในส่วนของซอฟท์แวร์ 4
2.1.1 สำรวจซอร์สโค้ดที่แจกฟรี 4
2.1.2 สำรวจซอฟต์แวร์ที่ขายตามท้องตลาด 6
2.2 การศึกษาและสำรวจในส่วนของฮาร์ดแวร์ 12
2.2.1 บอร์ดทดลองต้นแบบ XS40 Board V1.4 12
2.2.2 การทดสอบบอร์ดทดลอง 17
2.2.3 การเปลี่ยนแปลงสัญญาณนาฬิกา 18
2.2.4 การโปรแกรมข้อมูลลงบนบอร์ดทดลอง 18
2.2.5 รายละเอียดของอุปกรณ์ต่างๆ บนบอร์ดทดลอง 20
2.2.5.1 เอฟพีจีเอ (FPGA : Field Programmable Gate Array) 20
2.2.5.2 คุณสมบัติโดยทั่วไปของเอฟพีจีเอ XC4010E 24
2.2.5.3 เปรียบเทียบ เอฟพีจีเอ กับ ไมโครคอนโทรลเลอร์ 25
2.2.5.4 เอสแรม (SRAM : Static Random-Access Memory) 26
2.2.5.5 Logic block Diagram 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

2.2.5.6 ไมโครคอนโทรลเลอร์ 29

บทที่ 3 ขอบเขตของโครงงาน 31

3.1 ภาพรวมและองค์ประกอบหลักของโครงงาน 31

3.2 คุณลักษณะหลักของโครงงาน 32

3.2.1 ฟังก์ชันหลักสำหรับใช้งานของซอฟต์แวร์ 32

3.2.2 ความสามารถของฮาร์ดแวร์ 33

3.3 ขอบเขตของส่วนซอฟต์แวร์ 33

3.3.1 ฟังก์ชันที่เพิ่มเติมเข้าไปในซอร์สโค้ดต้นแบบ 33

3.3.2 ฟังก์ชันและส่วนของโปรแกรมที่แก้ไขให้ดีขึ้น 34

3.4 ขอบเขตของส่วนฮาร์ดแวร์ 34

บทที่ 4 การออกแบบโครงงาน 35

4.1 โครงสร้างของโครงงาน 35

4.2 การพัฒนาในส่วนซอฟต์แวร์ 36

4.2.1 อินพุต 36

4.2.2 เอาต์พุต 36

4.2.3 เครื่องมือที่ใช้ในการพัฒนา 37

4.2.4 โครงสร้างของซอฟต์แวร์ (Software Design) 38

4.3 การพัฒนาในส่วนฮาร์ดแวร์ 40

4.3.1 การออกแบบบอร์ดทดลองด้วย VHDL 41

4.3.2 องค์ประกอบที่สำคัญของ VHDL 41

4.3.3 รูปแบบการเขียน Architecture ของ VHDL 41

4.3.4 ข้อดีของ VHDL 41

4.3.5 VHDL ช่วยในการออกแบบได้อย่างไรบ้าง 42

4.3.6 การทำงานของโปรโตคอลอจิกอนาไลเซอร์ 46

4.3.7 รายละเอียดของลอจิกอนาไลเซอร์ 47

4.3.8 การออกแบบโปรแกรม 47

4.4 แผนการดำเนินงาน 49

บทที่ 5 รายละเอียดการพัฒนาและผลการทดสอบ 50

5.1 การพัฒนาและการทดสอบในส่วนซอฟต์แวร์ 50

5.1.1 การพัฒนาในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 51

5.1.2 การพัฒนาในส่วนบันทึกข้อมูลลงไฟล์ชนิด Value Change Dump (VCD) 54

5.1.3 การพัฒนาในส่วนแสดงผลและวิเคราะห์ข้อมูล 57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

5.1.4 การพัฒนาในส่วนวิเคราะห์โปรโตคอล 60

5.2 การพัฒนาและการทดสอบในส่วนฮาร์ดแวร์ 62

5.2.1 บอร์ดแปลงพอร์ตอนุกรมเป็น TTL 62

5.2.2 ชิพ MAX232 63

5.2.3 พอร์ตอนุกรม 63

5.2.4 โหมดการติดต่อทางพอร์ตอนุกรม 64

5.2.5 พอร์ตขนาน 65

บทที่ 6 บทวิจารณ์และสรุป 67

6.1 สรุปการดำเนินงาน 67

6.1.1 สรุปการดำเนินงานส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 67

6.1.2 สรุปการดำเนินงานส่วนบันทึกข้อมูลลงไฟล์ชนิด Value Change Dump 67

6.1.3 สรุปการดำเนินงานส่วนแสดงผลและวิเคราะห์ข้อมูล 67

6.1.4 สรุปการดำเนินงานส่วนวิเคราะห์โปรโตคอล 68

6.2 แนวทางในการพัฒนาต่อไป 68

ภาคผนวก ก ซอร์สโค้ด (Source Code) 69

ภาคผนวก ข เอพีไอมาตรฐานที่ใช้ในการพัฒนาโปรแกรม 72

ภาคผนวก ค ตัวอักษรย่อ 80

ภาคผนวก ง คู่มือการใช้งานโปรแกรม 81

ง-1 คู่มือการใช้งานในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 81

ง-2 คู่มือการใช้งานในส่วนส่วนวิเคราะห์โปรโตคอล 82

ง-3 คู่มือการใช้งานในส่วนแสดงผลและวิเคราะห์ข้อมูล 83

บรรณานุกรม 100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้าที่

รูปที่ 2-1 GTKWave Version 1.3.20 บน MS Windows 6
รูปที่ 2-2 La4xxx Logic Analyzer 7
รูปที่ 2-3 TLA Application Software 8
รูปที่ 2-4 USBee LX Logic Analyzer 8
รูปที่ 2-5 WaveFormer Pro 9
รูปที่ 2-6 รายละเอียดต่างๆของบอร์ดทดลอง 12
รูปที่ 2-7 โครงสร้างของบอร์ดทดลอง 14
รูปที่ 2-8 หน้าจอโปรแกรม XSTEST 17
รูปที่ 2-9 หน้าจอโปรแกรม XSTEST ขณะลงข้อมูล 17
รูปที่ 2-10 หน้าจอโปรแกรม XSTEST เมื่อการทำงานเสร็จสมบูรณ์ 17
รูปที่ 2-11 หน้าจอโปรแกรม XSSETCLK 18
รูปที่ 2-12 หน้าจอโปรแกรม XSLOAD 19
รูปที่ 2-13 แผนผังขั้นตอนการโปรแกรมข้อมูลลงบนเอฟพีจีเอ 19
รูปที่ 2-14 ASIC 20
รูปที่ 2-15 สถาปัตยกรรมภายในของ เอฟพีจีเอ 21
รูปที่ 2-16 โครงสร้างส่วน ซีแอลบี 22
รูปที่ 2-17 โครงสร้างส่วน ไอโอบี 23
รูปที่ 2-18 โครงสร้างส่วน อินเตอร์คอนเน็ค 24
รูปที่ 2-19 โครงสร้าขาพินของ ซิพเอสแรม 26
รูปที่ 2-20 สถาปัตยกรรมภายในของ เอสแรม 27
รูปที่ 2-21 รอบเวลาในการอ่านข้อมูล 28
รูปที่ 2-22 รอบเวลาในการเขียนข้อมูล 28
รูปที่ 2-23 โครงสร้างภายใน ขณะเอาต์พุต 28
รูปที่ 2-24 สถาปัตยกรรมภายนอกของ เอสแรม 28
รูปที่ 2-25 โครงสร้างขาของไมโครคอนโทรลเลอร์ 29
รูปที่ 2-26 โครงสร้างภายใน ไมโครคอนโทรลเลอร์ 30
รูปที่ 2-27 สถาปัตยกรรมภายนอกของ ไมโครคอนโทรลเลอร์ 30
รูปที่ 3-1 องค์ประกอบของโปร โตคอลอจิกอนาไลเซอร์ 31
รูปที่ 3-2 ภาพรวมของโปร โตคอลอจิกอนาไลเซอร์ 31
รูปที่ 4-1 ผังการทำงานของระบบ 35
รูปที่ 4-2 ตัวอย่างอินพุตสัญญาณจาก Serial Port (RS-232) 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 4-3 ตัวอย่างของ Output 37
รูปที่ 4-4 ผังการทำงานของโปรแกรม 38
รูปที่ 4-5 การทำงานเมื่อรับข้อมูลสัญญาณมาจากอุปกรณ์วัดผลแล้วนำไปแสดงผล 39
รูปที่ 4-6 การทำงานเมื่อรับคำสั่งจากผู้ใช้แล้วแสดงผลข้อมูล 39
รูปที่ 4-7 ขั้นตอนการโหลดซอร์สโค้ดลงบอร์ดทดลอง 40
รูปที่ 4-8 แสดงขั้นตอน Top-Down Design 42
รูปที่ 4-9 การเลือกไฟล์ VHDL 43
รูปที่ 4-10 แสดงการโหลดโค๊ด VHDL 43
รูปที่ 4-11 แสดงสัญญาณต่างๆในโปรแกรม 44
รูปที่ 4-12 แสดงการทำงานของโปรแกรม 44
รูปที่ 4-13 การแปลงโค๊ด VHDL ให้ได้วงจรในระดับ Gate-level 45
รูปที่ 4-14 การบันทึกโปรแกรมลงบนอุปกรณ์ FPGA 46
รูปที่ 4-15 สายจับสัญญาณ TTL 47
รูปที่ 4-16 แสดงลักษณะการทำงานของอินพุต/เอาต์พุต 48
รูปที่ 4-17 แสดงรายละเอียดชุดบิตควบคุม ที่แอดเดรส "0000000000000000" 48
รูปที่ 4-18 แสดงรายละเอียดชุดบิตควบคุมสัญญาณนาฬิกา ที่แอดเดรส "0000000000000001" 48
รูปที่ 4-19 แสดงรายละเอียดบิตควบคุมจำนวนพอร์ต 48
รูปที่ 5-1 แผนผังส่วนต่างๆ ของโปรแกรมที่พัฒนา 50
รูปที่ 5-2 Use Case Diagram ของส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 51
รูปที่ 5-3 Flow Chart ของส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 52
รูปที่ 5-4 หน้าจอโปรแกรมส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 53
รูปที่ 5-5 การทดสอบโดยการวัดค่าสัญญาณความถี่จากเครื่องกำเนิดความถี่ 53
รูปที่ 5-6 ข้อมูลในบัฟเฟอร์ที่ได้จากการสุ่มจับสัญญาณ 54
รูปที่ 5-7 Use Case Diagram ของส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD 54
รูปที่ 5-8 Flow Chart ของส่วนส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD 55
รูปที่ 5-9 หน้าจอโปรแกรมส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD 55
รูปที่ 5-10 ข้อมูลในไฟล์ชนิด VCD 56
รูปที่ 5-11 Use Case Diagram ของส่วนแสดงผลและวิเคราะห์ข้อมูล 57
รูปที่ 5-12 Flow Chart ของส่วนแสดงผลและวิเคราะห์ข้อมูล 58
รูปที่ 5-13 หน้าจอโปรแกรมส่วนแสดงผลและวิเคราะห์ข้อมูล 59
รูปที่ 5-14 การทดสอบโปรแกรมในส่วนแสดงผลและวิเคราะห์ข้อมูล 59
รูปที่ 5-15 Use Case Diagram ของส่วนวิเคราะห์โปรโตคอล 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ 5-16 Flow Chart ของส่วนวิเคราะห์โปรโตคอล 60
รูปที่ 5-17 หน้าจอโปรแกรมส่วนวิเคราะห์โปรโตคอล 61
รูปที่ 5-18 สัญญาณในสายส่งข้อมูลแบบ RS-232 Serial ที่วัดด้วยออสซิลโลสโคป 61
รูปที่ 5-19 ข้อมูลที่แสดงหลังจากทำการวิเคราะห์โปรโตคอล 62
รูปที่ 5-20 บอร์ดแปลงพอร์ตอนุกรมเป็น TTL (รูปแทน) 62
รูปที่ 5-21 โครงสร้างของบอร์ดแปลงพอร์ตอนุกรมเป็น TTL 63
รูปที่ 5-22 สถาปัตยกรรมภายในของชิพ MAX232 63
รูปที่ 5-23 การส่งข้อมูลแบบอนุกรม 64
รูปที่ 5-24 การทำงานในโหมด 0 64
รูปที่ 5-25 การทำงานในโหมด 1 65
รูปที่ 5-26 แสดงการติดต่อกับพอร์ตขนาน 66
รูปที่ 5-27 แสดงรายละเอียดของ พอร์ตขนาน 66
รูปที่ ง-1 หน้าจอส่วนติดต่อกับอุปกรณ์วัดสัญญาณ 81
รูปที่ ง-2 หน้าจอส่วนวิเคราะห์โปรโตคอล 82
รูปที่ ง-3 หน้าจอส่วนแสดงผลและวิเคราะห์ข้อมูล 83
รูปที่ ง-4 เมนู File 83
รูปที่ ง-5 หน้าจอ Read Save File 84
รูปที่ ง-6 เมนู Edit 85
รูปที่ ง-7 เมนูย่อย Data Format 86
รูปที่ ง-8 หน้าจอ Show-Change All Highlighted 87
รูปที่ ง-9 เมนูย่อย Time Warp 88
รูปที่ ง-10 เมนูย่อย Sort 89
รูปที่ ง-11 เมนู Search 90
รูปที่ ง-12 หน้าจอ Pattern Search 90
รูปที่ ง-13 หน้าจอ Signal Search Regexp 91
รูปที่ ง-14 หน้าจอ Signal Search Hierarchy 92
รูปที่ ง-15 หน้าจอ Signal Search Tree 93
รูปที่ ง-16 เมนู Time 94
รูปที่ ง-17 หน้าจอ Move To Time 94
รูปที่ ง-18 เมนูย่อย Zoom 94
รูปที่ ง-19 หน้าจอ Fetch Size 95
รูปที่ ง-20 เมนู Markers 96

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

รูปที่ ง-21 เมนู View 97
รูปที่ ง-22 เมนู Trigger 98
รูปที่ ง-23 หน้าจอ Trigger Dialog 98
รูปที่ ง-24 หน้าจอ เมนู Help 99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้าที่

ตารางที่ 2-1 ข้อมูลของโปรเจค Dinotrace 4
ตารางที่ 2-2 ข้อมูลของโปรเจค FreeHDL 5
ตารางที่ 2-3 ข้อมูลของโปรเจค GTKWave 5
ตารางที่ 2-4 ข้อมูลของ vIDE 5
ตารางที่ 2-5 ข้อมูลของ La4xxx Logic Analyzer 7
ตารางที่ 2-6 ข้อมูลของ TLA Application Software 7
ตารางที่ 2-7 ข้อมูลของ USBee LX Logic Analyzer 8
ตารางที่ 2-8 ข้อมูลของ WaveFormer Pro 9
ตารางที่ 2-9 เปรียบเทียบความสามารถของซอฟต์แวร์ 11
ตารางที่ 2-10 รายละเอียดพินต่างๆ ของบอร์ดทดลอง 16
ตารางที่ 2-11 แสดงสถานะการทำงาน 27
ตารางที่ 4-1 การดำเนินงานในภาคเรียนที่ 1 49
ตารางที่ 4-2 การดำเนินงานในภาคเรียนที่ 2 49
ตารางที่ ข-1 แสดงการเติมเต็มบิตด้านซ้ายของเวคเตอร์ 72
ตารางที่ ข-2 แสดงการเขียนค่าในรูปแบบย่อ 73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญและที่มา

เนื่องจากการใช้งานของเครือข่ายอินเตอร์เน็ตเป็นไปอย่างรวดเร็วมาก จึงทำให้ผู้ให้บริการต้องปรับปรุงและขยายเครือข่าย ซึ่งต้องนำเข้าอุปกรณ์จากต่างประเทศเป็นจำนวนมากตาม ส่วนหนึ่งของอุปกรณ์อยู่ในรูปของเครื่องมือวัดชนิดที่เรียกว่า โปรโตคอลอนาไลเซอร์ มีราคาแพงมาก และมีความสามารถสนับสนุนชนิดของโปรโตคอลไม่กี่ชนิด เมื่อโปรโตคอลชนิดนั้นๆ ล้าสมัยไป ตัวเครื่องก็ไม่สามารถอัพเกรด (Upgrade) ได้หรือได้แต่ต้องเพิ่มค่าใช้จ่ายที่สูง การพัฒนาโปรโตคอลอนาไลเซอร์ ที่สามารถปรับเปลี่ยนได้ตามความต้องการจึงเป็นการประหยัดต้นทุนและลดการนำเข้าสินค้าจากต่างประเทศได้

ในขณะเดียวกัน เครื่องมือที่ใช้ตรวจสอบความผิดพลาดของวงจร โดยทั่วไปมักใช้ Logic Probe เพราะว่าราคาถูก แต่ว่า Logic Probe สามารถวัดสัญญาณได้แค่ 1 สัญญาณ ไม่สามารถวัดหลายๆ สัญญาณในเวลาเดียวกันได้ และ Logic Probe ไม่สามารถบอกได้ว่า การเปลี่ยนแปลงของระดับสัญญาณช้าหรือเร็วมากแค่ไหน ซึ่งทำให้เราตรวจสอบความถูกต้องของวงจรได้ยาก อีกทั้งไม่สามารถวัดสัญญาณที่สายส่งสัญญาณได้ว่า ข้อมูลที่ส่งถูกต้องหรือไม่

Protocol Analyzer ทั่วๆไป สามารถวิเคราะห์ข้อมูลในสายส่งสัญญาณทั้งชนิด Serial และ Parallel ได้ สามารถในการวิเคราะห์ได้ว่า สัญญาณข้อมูลที่ส่งนั้นคืออะไรบ้าง และสัญญาณส่วนใดไม่ใช่ส่วนของข้อมูล ขึ้นอยู่กับโปรโตคอลที่ใช้ในการสื่อสารกัน เพราะแต่ละโปรโตคอลมีลักษณะไม่เหมือนกัน เช่น จำนวนบิตในการส่งแต่ละแพ็คเก็ต คำสั่งที่ให้ควบคุม เป็นต้น มีความคล้ายคลึงกับเครื่องมืออีกชนิดหนึ่งที่ใช้ตรวจสอบความผิดพลาดของวงจร ที่เรียกว่า Logic Analyzer ซึ่งมีความสามารถวัดสัญญาณระดับ TTL/CMOS ได้หลายสัญญาณพร้อมกัน และยังสามารถแสดงข้อมูลได้หลายๆ รูปแบบ ทำให้เราค้นหาข้อผิดพลาดได้ง่าย แต่ Logic Analyzer มีราคาสูงมาก เมื่อเทียบกับ Logic Probe

จากสาเหตุดังที่กล่าวมา จึงเล็งเห็นว่า ควรพัฒนาอุปกรณ์ในลักษณะนี้ขึ้นมา เพื่อเป็นต้นแบบในการพัฒนาโปรแกรมที่มีสามารถในการวิเคราะห์ทั้งสายส่งสัญญาณและวงจรดิจิตอลได้ในตัวเดียวกัน และเพื่อให้อุปกรณ์ประเภทนี้มีราคาถูกลงและสามารถนำมาใช้งานได้อย่างแพร่หลาย

1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อสร้างโปรแกรมที่สามารถวิเคราะห์โปรโตคอลพื้นฐานที่ใช้สื่อสารทั้งชนิด Serial และ Parallel ได้

2) เพื่อสร้างโปรแกรมที่สามารถวิเคราะห์สัญญาณทางดิจิตอลได้อย่างมีประสิทธิภาพ

3) เพื่อสร้างโมดูลสำหรับ FPGA โดยใช้เป็นอุปกรณ์ในการวัดสัญญาณ

4) เพื่อสร้างโครงงานที่มีประโยชน์ในการศึกษาและใช้งานทางด้านวิทยาศาสตร์และเทคโนโลยี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เป้าหมายของโครงงาน

1) เพื่อสร้างโปรแกรมที่มีความสามารถวิเคราะห์โปรโตคอลสำหรับการสื่อสารแบบ Serial และ Parallel ได้ว่า ส่วนใดคือข้อมูล ส่วนใดคือลักษณะหรือรูปแบบของโปรโตคอล

2) เพื่อสร้างโปรแกรมที่มีความสามารถวัด วิเคราะห์และตรวจทานความถูกต้อง (Verify) ของสัญญาณดิจิตอล มาใช้งานในการศึกษาหรือผู้ที่สนใจทั่วไป

3) เพื่อสร้างโมดูลใช้กับ FPGA โดยใช้ภาษา VHDL เพื่อเป็นอุปกรณ์วัดสัญญาณ สำหรับส่งข้อมูลให้โปรแกรมทำการวิเคราะห์ต่อไป

4) เพื่อทดแทนซอฟต์แวร์และรวมไปถึงฮาร์ดแวร์ที่มีราคาแพง ซึ่งต้องซื้อมาจากต่างประเทศเป็นส่วนใหญ่

1.4 ปัญหาและประโยชน์ของโครงงาน

เครื่องมือที่ใช้วิเคราะห์โปรโตคอลพื้นฐานที่ใช้สื่อสารทั้งชนิด Serial และ Parallel และ เครื่องมือในการวัดและวิเคราะห์สัญญาณดิจิตอลมีราคาแพงและต้องนำเข้าจากต่างประเทศ ทำให้ขาดดุลการค้าระหว่างประเทศ และเป็นการส่งเสริมให้ประเทศไทยพึ่งพาตนเองได้ทางเทคโนโลยี และในที่สุดลดการนำเข้าสินค้าจากต่างประเทศได้

1.5 ขอบเขตของโครงงาน

1.5.1 ขอบเขตของส่วนซอฟต์แวร์โดยสรุป

โครงงานนี้ใช้การพัฒนาต่อยอดจากโปรแกรมที่ใช้แสดงเวฟฟอร์ม (Waveform Viewer) ดังนั้นต้องแก้ไขซอร์สโค้ดที่ได้มาให้สามารถทำงานได้ตามต้องการ รวมทั้งตัดส่วนที่ไม่เกี่ยวข้องออก และเพิ่มความสามารถใหม่ๆ เข้าไป การทำงานในส่วนซอฟต์แวร์โดยสรุปดังนี้

1) เพิ่มฟังก์ชันเข้าไปในซอร์สโค้ดต้นแบบ
   - ส่วนของโปรแกรมที่จัดการเกี่ยวกับการวิเคราะห์และแสดงผลข้อมูล
   - ส่วนของโปรแกรมที่ติดต่อกับฮาร์ดแวร์

2) แก้ไขฟังก์ชันและส่วนของโปรแกรมให้ดีขึ้น
   - ส่วนของโปรแกรมที่จัดการเกี่ยวกับการปรับแต่งส่วนแสดงผลข้อมูล
   - ส่วนของโปรแกรมที่ติดต่อกับผู้ใช้

3) ตัดฟังก์ชันที่ไม่เกี่ยวข้องออก

1.5.2 ขอบเขตของส่วนฮาร์ดแวร์โดยสรุป

ส่วนฮาร์ดแวร์เป็นการพัฒนาโมดูลที่มีความสามารถในการสุ่มสัญญาณข้อมูล (Sampling) ติดต่อกับหน่วยความจำ และติดต่อกับส่วนที่ใช้ส่งข้อมูลไปยังคอมพิวเตอร์ โดยโมดูลนี้นำไปใช้กับ FPGA และสร้างส่วนที่ติดต่อกับอุปกรณ์ภายนอก เพื่อใช้วัดสัญญาณ การทำงาน ในส่วนฮาร์ดแวร์โดยสรุปดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) สร้างโมดูลที่เป็นส่วนควบคุมและประมวลผล

2) สร้างส่วนที่ติดต่อกับอุปกรณ์ภายนอก ได้แก่ ส่วนที่แปลงสัญญาณในสายส่งข้อมูลเป็นสัญญาณ TTL/CMOS

3) สร้างส่วนที่ติดต่อกับคอมพิวเตอร์

1.6 ขั้นตอนการพัฒนา

1.6.1 ขั้นตอนการพัฒนาของส่วนซอฟต์แวร์โดยสรุป

1) หาซอร์สโค้ดต้นแบบ

2) ศึกษาซอร์สโค้ดต้นแบบ

3) ศึกษาและสำรวจซอฟต์แวร์ในท้องตลาด

4) กำหนดขอบเขตและความสามารถของซอฟต์แวร์

5) กำหนดอินพุตและเอาต์พุต

6) วางโครงสร้างและการทำงานของซอฟต์แวร์

7) หาเครื่องมือเพื่อใช้ในการพัฒนา

8) พัฒนาซอฟท์แวร์

9) ทดสอบและแก้ไขข้อผิดพลาด

1.6.2 ขั้นตอนการพัฒนาของส่วนฮาร์ดแวร์โดยสรุป

1) ศึกษาทฤษฎีและหลักการที่เกี่ยวข้อง

2) ศึกษาและสำรวจผลิตภัณฑ์ในท้องตลาด

3) กำหนดขอบเขตและความสามารถของฮาร์ดแวร์

4) กำหนดอินพุตและเอาต์พุต

5) ออกแบบโครงสร้างของฮาร์ดแวร์

6) ศึกษาบอร์ดที่ใช้พัฒนา

7) หาเครื่องมือที่ใช้ในการพัฒนา

8) พัฒนาฮาร์ดแวร์

9) ทดสอบและแก้ไขข้อผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
เทคโนโลยีพื้นฐานและระบบที่มีอยู่

2.1 การศึกษาและสำรวจในส่วนของซอฟท์แวร์

2.1.1 สำรวจซอร์สโค้ดที่แจกฟรี

1) หาซอร์สโค้ดที่แจกฟรีเพื่อนำมาเป็นต้นแบบแล้วพัฒนาต่อยอดให้เป็นโปรแกรมที่สมบูรณ์แบบตามที่ต้องการ โดยได้ทำการสำรวจโปรเจคต่างๆ ดังนี้

- Dinotrace
- FreeHDL Project
- GTKWave
- vIDE

2) ทำการศึกษาแต่ละโปรเจคโดยได้ผลดังนี้

ชื่อโปรเจค	Dinotrace
เวอร์ชัน	9.1k
ภาษาที่ใช้พัฒนา	C
เทคโนโลยีหรือ API ที่ใช้	X-11 library, Emacs
ความสมบูรณ์ของซอร์สโค้ด	สมบูรณ์
เอกสารประกอบซอร์สโค้ด	ไม่มี
ระบบปฏิบัติการ	Unix, Linux, MS Windows (ใช้ร่วมกับโปรแกรม Cygwin)
ชนิดของไฟล์ที่รองรับ	Verilog Value Change Dumps (VCD), ASCII
แหล่งที่มาของซอร์สโค้ด	http://www.veripool.com/dinotrace

ตารางที่ 2-1 ข้อมูลของโปรเจค Dinotrace

ชื่อโปรเจค	FreeHDL
เวอร์ชัน	-
ภาษาที่ใช้พัฒนา	C
เทคโนโลยีหรือ API ที่ใช้	VHDL Analyzer and Utility Library (VAUL)
ความสมบูรณ์ของซอร์สโค้ด	มีความสมบูรณ์ แต่มีขนาดใหญ่ เพราะประกอบด้วยหลายโปรเจคย่อย
เอกสารประกอบซอร์สโค้ด	ไม่มี
ระบบปฏิบัติการ	Unix, Linux

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของไฟล์ที่รองรับ	-
แหล่งที่มาของซอร์สโค้ด	http://www.freehdl.seul.org

ตารางที่ 2-2 ข้อมูลของโปรเจค FreeHDL

ชื่อโปรเจค	GTKWave
เวอร์ชัน	1.3.20
ภาษาที่ใช้พัฒนา	C
เทคโนโลยีหรือ API ที่ใช้	GTK library
ความสมบูรณ์ของซอร์สโค้ด	สมบูรณ์
เอกสารประกอบซอร์สโค้ด	ไม่มี
ระบบปฏิบัติการ	Unix, Linux, MS Windows (ใช้ร่วมกับ Library ของ GTK, GDK, GLib ที่ถูกแปลงเป็นไฟล์ .dll)
ชนิดของไฟล์ที่รองรับ	Verilog Value Change Dumps (Verilog VCD), interLaced eXtensible Trace (LXT)
แหล่งที่มาของซอร์สโค้ด	http://www.cs.man.ac.uk/apt/tools/gtkwave

ตารางที่ 2-3 ข้อมูลของโปรเจค GTKWave

ชื่อโปรเจค	vIDE (Verilog IDE)
เวอร์ชัน	0.1b
ภาษาที่ใช้พัฒนา	Java
เทคโนโลยีหรือ API ที่ใช้	Java™ Technology, Java1.2 compatible VM
ความสมบูรณ์ของซอร์สโค้ด	ไม่สมบูรณ์ ซอร์สโค้ดคอมไพล์ไม่ผ่าน
เอกสารประกอบซอร์สโค้ด	มีเอกสารประกอบ
ระบบปฏิบัติการ	ทุกระบบปฏิบัติการ โดยผ่าน Java Virtual Machine
ชนิดของไฟล์ที่รองรับ	-
แหล่งที่มาของซอร์สโค้ด	http://vlogide.sourceforge.net

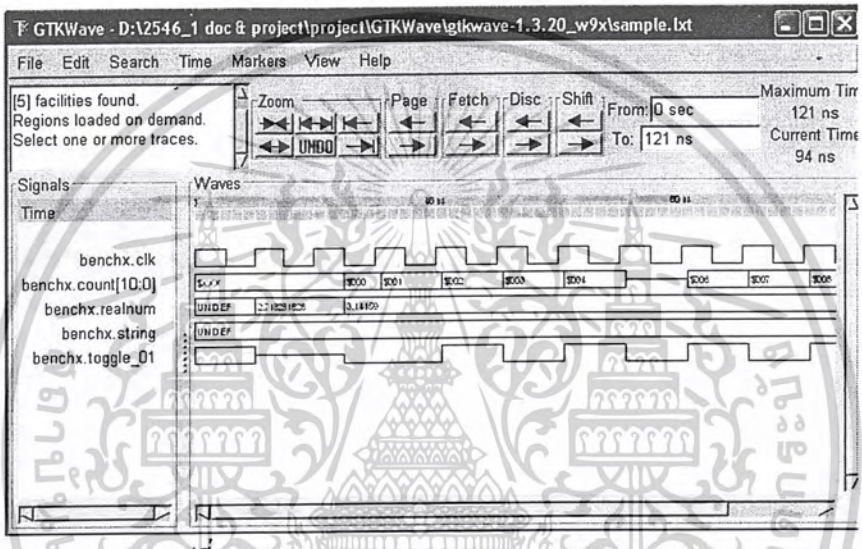
ตารางที่ 2-4 ข้อมูลของ vIDE

3) สรุปและเลือกซอร์สโค้ดต้นแบบ

- เลือกโปรเจค GTKWave เป็นต้นแบบ เพราะซอร์สโค้ดสมบูรณ์ สามารถคอมไพล์และใช้งานโปรแกรมได้ อีกทั้งไลบรารีใช้ มีการแปลงเป็นไฟล์ .dll ทำให้ง่ายในการคอมไพล์ซอร์สโค้ดให้ใช้งานบนระบบปฏิบัติการวินโดวส์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่เลือกโปรเจค Dinotrace เป็นต้นแบบ เพราะมีบางส่วนของโปรแกรมแกรมที่ต้องอาศัยโปรแกรมอื่นในการทำงาน ซึ่งทำให้ยากในการศึกษาเพราะต้องศึกษาการทำงานในส่วนที่ติดต่อกับโปรแกรมอื่นด้วย
- ไม่เลือกโปรเจค FreeHDL เป็นต้นแบบ เพราะเป็นโปรเจคที่ใหญ่ ซึ่งทำให้ยากในการศึกษาซอร์สโค้ด อีกทั้งโปรเจคนี้ยังประกอบด้วยโปรเจคย่อยๆ อีกหลายโปรเจคซึ่งไม่เกี่ยวข้องกับโครงงานที่ทำ
- ไม่เลือกโปรเจค vIDE เป็นต้นแบบ เพราะซอร์ดโค้ดที่ได้มาคอมไพล์ใช้งานไม่ได้ เนื่องจากเขียนโปรแกรมผิด อีกทั้งยังเป็นเวอร์ชันให้ทดลองให้ใช้งานอยู่ (เวอร์ชัน 0.1 beta)



รูปที่ 2-1 GTKWave Version 1.3.20 บน MS Windows

2.1.2 สำรวจซอฟต์แวร์ที่ขายตามท้องตลาด

1) ทำการค้นหาซอฟต์แวร์ และได้ศึกษาซอฟต์แวร์ดังนี้

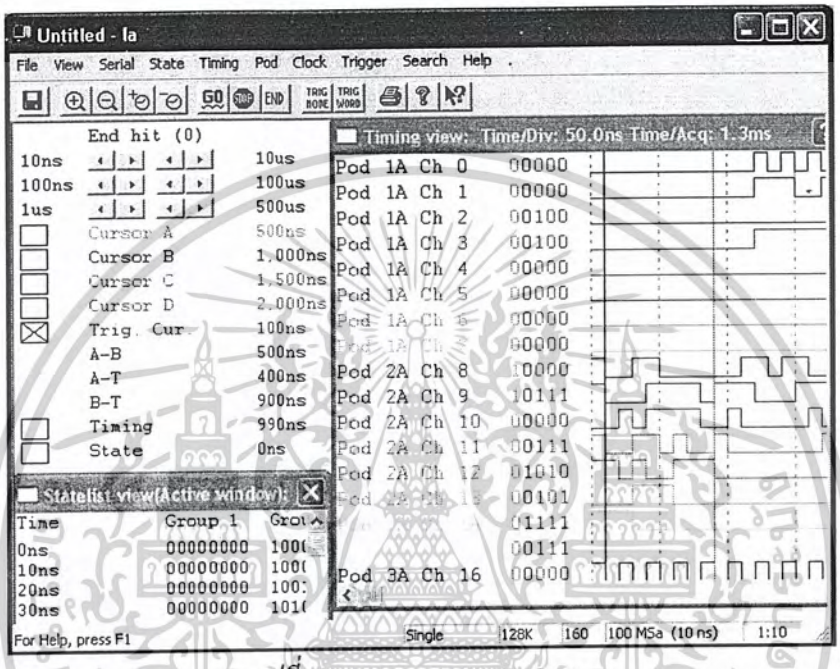
- La4xxx Logic Analyzer
- TLA Application Software
- USBee LX Logic Analyzer
- WaveFormer Pro

2) ข้อมูลและรูปภาพของแต่ละซอฟต์แวร์ที่ได้ทำงานสำรวจ มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อซอฟต์แวร์	La4xxx Logic Analyzer
เวอร์ชัน	1.60b
ระบบปฏิบัติการ	MS Windows
บริษัท	Link Instruments, Inc.
แหล่งที่มาของซอฟร์แวร์	www.linkinstruments.com

ตารางที่ 2-5 ข้อมูลของ La4xxx Logic Analyzer

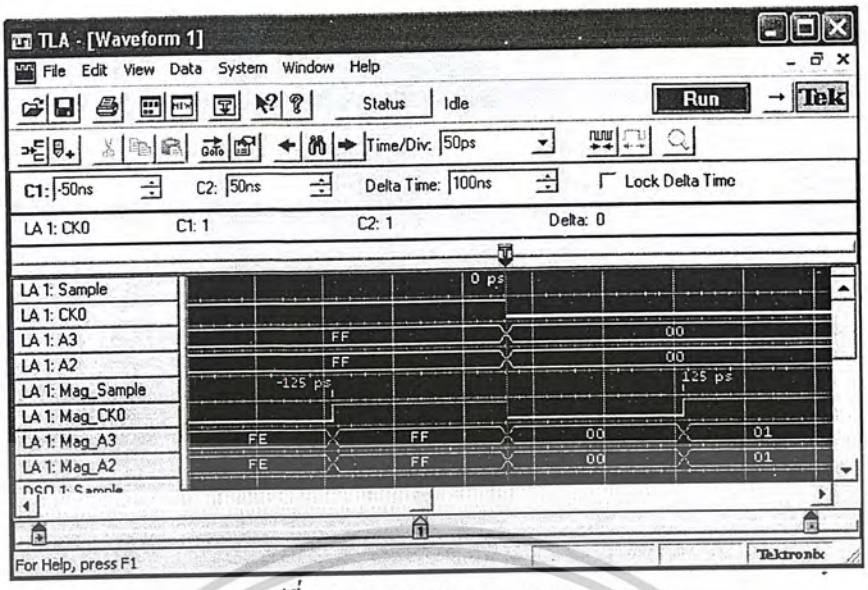


รูปที่ 2-2 La4xxx Logic Analyzer

ชื่อซอฟต์แวร์	TLA Application Software
เวอร์ชัน	4.2
ระบบปฏิบัติการ	MS Windows
บริษัท	Tektronix, Inc.
แหล่งที่มาของซอฟร์แวร์	www.tektronix.com

ตารางที่ 2-6 ข้อมูลของ TLA Application Software

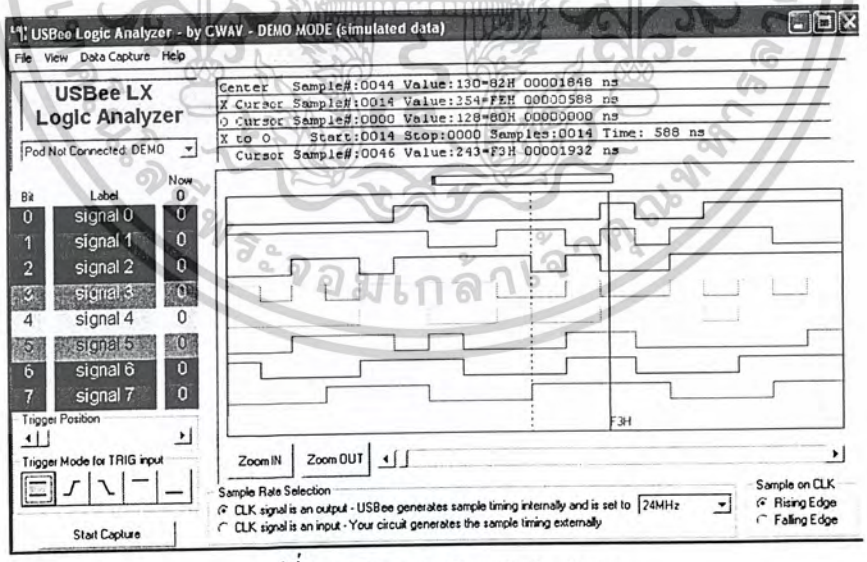
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-3 TLA Application Software

ชื่อซอฟต์แวร์	USBee LX Logic Analyzer
เวอร์ชัน	1.7.0
ระบบปฏิบัติการ	MS Windows
บริษัท	CWAV
แหล่งที่มาของซอฟร์แวร์	www.cwav.com

ตารางที่ 2-7 ข้อมูลของ USBee LX Logic Analyzer

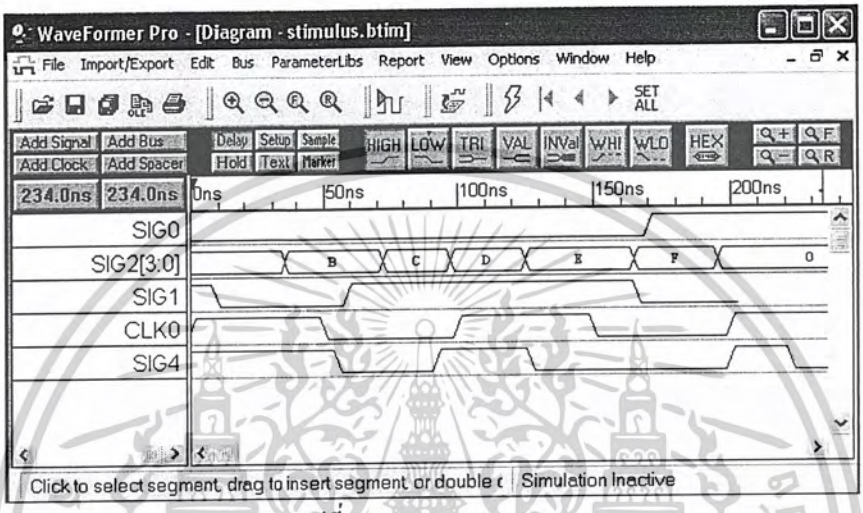


รูปที่ 2-4 USBee LX Logic Analyzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อซอฟต์แวร์	WaveFormer Pro
เวอร์ชัน	9.0
ระบบปฏิบัติการ	MS Windows
บริษัท	SynaptiCAD, Inc.
แหล่งที่มาของซอฟร์แวร์	www.syncad.com

ตารางที่ 2-8 ข้อมูลของ WaveFormer Pro



รูปที่ 2-5 WaveFormer Pro

3) ศึกษาซอฟต์แวร์ต่างๆ ว่ามีความสามารถอะไรบ้าง แล้วเปรียบเทียบความสามารถของแต่ละซอฟต์แวร์และซอฟต์แวร์ต้นแบบ เพื่อใช้เป็นแนวทางในการพัฒนาโปรแกรมที่จะสร้างขึ้น โดยทำการเปรียบเทียบโปรแกรมดังต่อนี้

- La4xxx Logic Analyzer (LA4xxx)
- TLA Application Software (TLA)
- USBee LX Logic Analyzer (USBee)
- WaveFormer Pro (WaveFormer)
- GTKwave

ความสามารถของโปรแกรม	LA4xxx	TLA	USBee	WaveFormer	GTKwave
เปิดข้อมูลที่บันทึกไว้ (Load file)	✓	✓	✓	✓	✓
บันทึกข้อมูล (Save File)	✓	✓	✓	✓	✓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิมพ์ข้อมูล (Print)	✓	✓	✓	✓	✓
ปรับแต่งชื่อและสีของแต่ละสัญญาณที่แสดงผล (Edit name and color for each channel)	✓		เฉพาะชื่อ	เฉพาะชื่อ	เฉพาะชื่อ
แสดงผลในรูปแบบของเวลาหรือลำดับครั้งของการสุ่มสัญญาณ (Display as Time or Samples)	✓	✓	✓		
จัดกลุ่มของสัญญาณที่แสดงผล (Group define for channels)	✓	✓		✓	✓
แสดงผลแบบตัวเลข (View as number)	✓	✓	✓		
ปรับแต่งรูปแบบการแสดงผลแบบตัวเลข (Number list setup for display)	✓	✓			
ปรับแต่งตารางการเปลี่ยนแปลงสเตจ (Edit state translation table)	✓				
ย้ายตำแหน่งหน้าจอแสดงผลไปที่เคอร์เซอร์ (Number list to cursor)	✓				
ย้ายเคอร์เซอร์มาที่หน้าจอแสดงผล (Cursor to Number list)	✓				
แสดงผลแบบเวฟฟอร์ม (View as waveform)	✓	✓	✓	✓	✓
แสดงค่าข้อมูล ณ ตำแหน่งของเคอร์เซอร์ (View cursor values)	✓		✓	✓	✓
ปรับแต่งรูปแบบการแสดงผลแบบเวฟฟอร์ม (Waveform setup for display)	✓	✓			
ย่อหรือขยายเวฟฟอร์ม (Waveform zoom)	✓	✓	✓	✓	✓
ย้ายตำแหน่งหน้าจอแสดงผลไปที่เคอร์เซอร์ (Waveform to cursor)	✓				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ย้ายเคอร์เซอร์มาที่หน้าจอแสดงผล (Cursor to waveform)	✓				
ค้นหาแพ็ทเทิร์ทที่ต้องการ (Search pattern)	✓	✓		✓	✓
ปรับแต่งลักษณะของหน้าจอแสดงผล (Edit display style)	✓				✓
เพิ่มหรือลดสัญญาณที่หน้าจอแสดงผล (Insert or Delete channel)	✓	✓		✓	✓
เพิ่มเคอร์เซอร์เข้าไปที่หน้าจอแสดงผล (Add cursor)	มี 4 เส้น	✓	มี 1 เส้น	มี 1 เส้น	มี 1 เส้น
เรียงลำดับชื่อของสัญญาณที่หน้าจอแสดงผล (Sorting view)					✓
ส่วนของโปรแกรมที่ติดต่อกับฮาร์ดแวร์					
ปรับแต่งความเร็วของสัญญาณนาฬิกา (Clock setup)	✓				
สั่งให้เริ่มจับสัญญาณจากวงจร (Triger start)	✓	✓	✓	✓	
สั่งให้หยุดจับสัญญาณจากวงจร (Triger stop)	✓	✓	✓	✓	

ตารางที่ 2-9 เปรียบเทียบความสามารถของซอฟต์แวร์

4) สรุปผลศึกษาและการเปรียบเทียบซอฟต์แวร์ต่างๆ

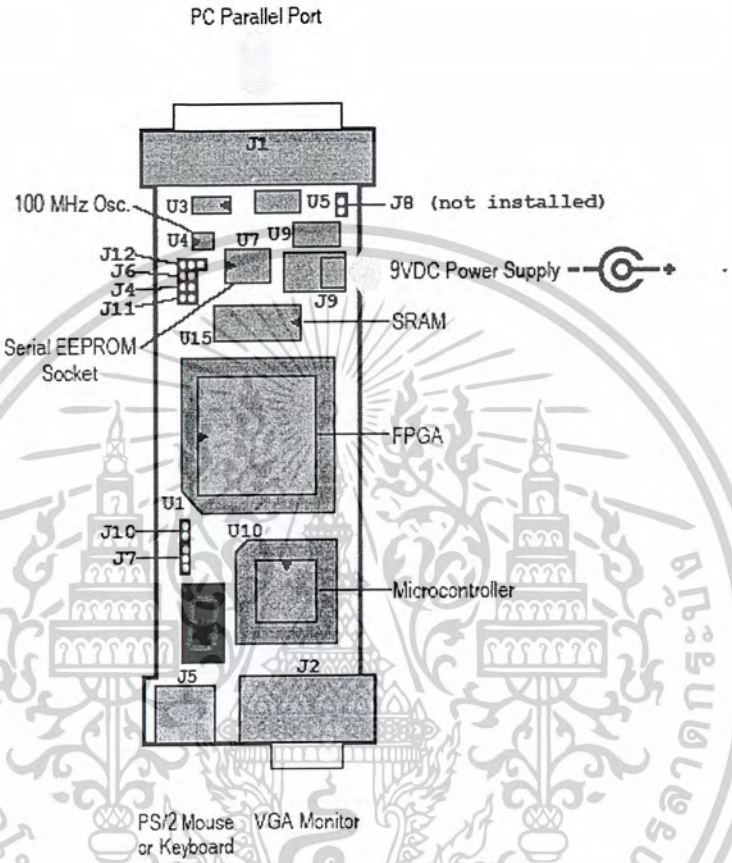
- ทำให้ทราบถึงความสามารถและฟังก์ชันต่างๆ ของซอฟต์แวร์ที่ศึกษา
- ทำให้ทราบว่าโปรแกรมใช้วัดและวิเคราะห์สัญญาณทางดิจิตอล ควรมีสามารถและฟังก์ชันอะไรบ้าง
- ทำให้ทราบถึงความสามารถและฟังก์ชันของซอฟต์แวร์ต้นแบบที่ยังขาด และควรได้รับการพัฒนาให้สมบูรณ์มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การศึกษาและสำรวจในส่วนของฮาร์ดแวร์

2.2.1 บอร์ดทดลองต้นแบบ XS40 Board V1.4

เป็นของบริษัท เอ็คซ เอนจิเนีย ซอฟต์แวร์ ซิสเท็ม จำกัด ประเทศสหรัฐอเมริกา



รูปที่ 2-6 รายละเอียดต่างๆของบอร์ดทดลอง

ติดต่อกับคอมพิวเตอร์ทาง พอร์ตขนาน ใช้สายสัญญาณ ดีบี-25 (DB-25) หัวต่อ เจ 1 (J1) และสามารถแสดงผลทางหน้าจอมอนิเตอร์ ทางหัวต่อ เจ 2 (J2) ขนาด 15 พิน รับอินพุทได้จาก คีย์บอร์ด หรือ เมาส์ ทางหัวต่อเจ 5 (J5) พีเอส 2 (PS/2) มีรูปแบบการตั้งค่า จั้มเปอร์ (Jumper) ดังนี้

- เจ 4 (J4) สำหรับโปรแกรมข้อมูลลง บอร์ดทดลองทางพอร์ตขนาน เอาจั้มเปอร์ (Jumper) ออก สำหรับการรับข้อมูลจาก อีอีพรอม (EEPROM) ยู 7 (U7) บนบอร์ดทดลอง

- เจ 6 (J6) สำหรับโปรแกรมข้อมูลลงอีอีพรอม (EEPROM) ยู 7 (U7) เอาจั้มเปอร์ (Jumper) ออก สำหรับการใช้งานตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เจ 7 (J7) ใส่พิน 1-2 สำหรับเก็บค่าโปรแกรมของ ไมโครคอนโทรลเลอร์ 8031 บน เอสแรม (SRAM) ยู 8 (U8) ภายนอก ขนาด 32 กิโลไบท์ (KByte) ใส่พิน 2-3 สำหรับเก็บค่าโปรแกรมของ ไมโครคอนโทรลเลอร์ 8031 ไว้ภายในตัว ไมโครคอนโทรลเลอร์เอง

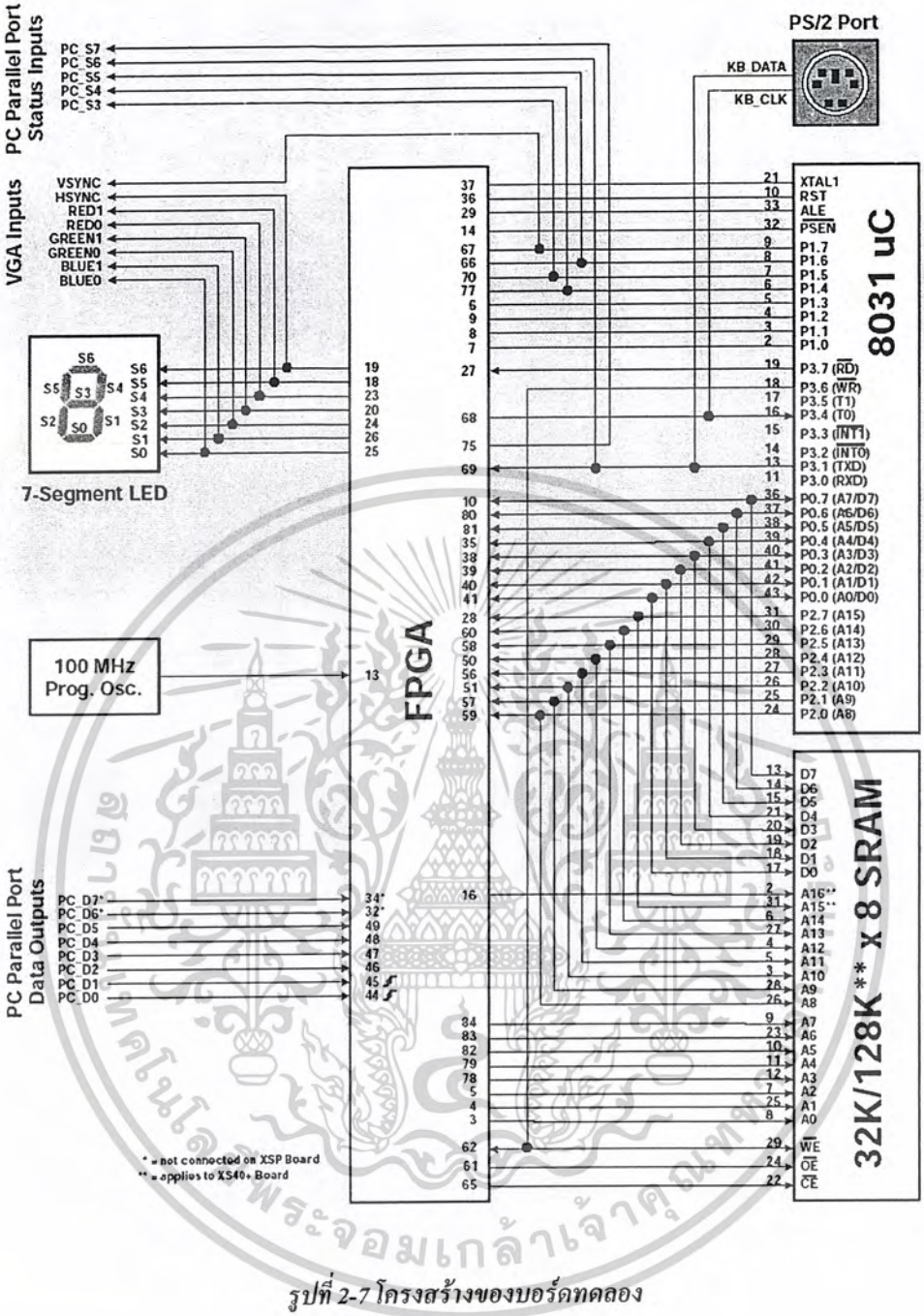
- เจ 8 (J8) สำหรับโปรแกรมข้อมูลลง บอร์ดทดลองโดยใช้ เอฟพีจีเอ 3.3V XC4000XL เอาจั้มเปอร์ (Jumper) ออกสำหรับโปรแกรมข้อมูลลง บอร์ดทดลองโดยใช้ เอฟพีจีเอ 5V XC4000E

- เจ 10 (J10) สำหรับบอร์ดทดลองรับการตั้งค่าจากอีอีพรอม (EEPROM) เอาจั้มเปอร์ (Jumper) ออกสำหรับ บอร์ดทดลองรับการตั้งค่าจากคอมพิวเตอร์ทางพอร์ตขนาน

- เจ 11 (J11) สำหรับ บอร์ดทดลองรับการตั้งค่าจากคอมพิวเตอร์ทางพอร์ตขนาน เอาจั้มเปอร์ (Jumper) ออกสำหรับบอร์ดทดลองรับการตั้งค่าจาก อีอีพรอม

- เจ 12 (J12) ใส่พิน 1-2 สำหรับการใช้งานสัญญาณนาฬิกาที่ ออสซิเลเตอร์ (oscillator) สร้างตามปกติ ใส่พิน 2-3 สำหรับตั้งค่าการใช้งานสัญญาณนาฬิกาที่ ออสซิเลเตอร์ สร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-7 โครงสร้างของบอร์ดทดลอง

XS40 พิน	ต่อกับ พิน	รายละเอียด
25	S0	ขานี้ต่อกับ LED (S0-S6) และแสดงเส้นสี ด้านขวางสำหรับ VGA
26	S1	
24	S2	
20	S3	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

23	S4	
18	S5	
19	S6	
13	CLK	ขาเข้าของความถี่ 100 เม็กกะเฮิร์ท
44	PC D0	ต่อกับเอาต์พุต ไปพอร์ตขนานของ คอมพิวเตอร์
45	PC D1	
46	PC D2	
47	PC D3	
48	PC D4	
49	PC D5	
32	PC D6	
34	PC D7	
37	XTAL 1	คล็อคขาเข้า ของไมโครคอนโทรลเลอร์
36	RST	รีเซ็ทขาเข้า ของไมโครคอนโทรลเลอร์
29	ALEB	แอดเดรสของไมโครคอนโทรลเลอร์ เริ่มทำงาน
14	PSENB	การเก็บโปรแกรมของไมโครคอนโทรลเลอร์ เริ่มทำงาน
7	P1.0	พินชุดนี้ ต่อกับพอร์ต 1 ของไมโครคอนโทรลเลอร์ บางพินควบคุมสถานะอินพุต ของพอร์ตขนาน พิน 67 สร้างสัญญาณแนวตั้งของ VGA
8	P1.1	
9	P1.2	
6	P1.3	
77	P1.4 PC S4	
70	P1.5 PC S3	
66	P1.6 PC S5	
67	P1.7 VSYNCB	
69	P3.1(TXD) PC S6	ต่อกับพอร์ต 3 ของไมโครคอนโทรลเลอร์ พิน 62 ควบคุมการเขียนข้อมูลของ ไมโครคอนโทรลเลอร์ และ เอสแรม พิน 69 ควบคุมสถานะของพอร์ตขนาน และ ข้อมูลของ PS/2 พิน 68 เป็นสัญญาณนาฬิกาของ PS/2
68	P3.4(T0) PS/2 CLK	
62	P3.6(WRB) WEB	
27	P3.7(RDB)	
41	P0.0(AD0).D0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

40	P0.1(AD1).D1	ต่อกับพอร์ต 0 ของไมโครคอนโทรลเลอร์ ต่อกับอินพุตข้อมูลของเอสแรม
39	P0.2(AD2).D2	
38	P0.3(AD3).D3	
35	P0.4(AD4).D4	
81	P0.5(AD5).D5	
80	P0.6(AD6).D6	
10	P0.7(AD7).D7	
59	P2.0(A8).A8	ต่อกับพอร์ต 2 ของไมโครคอนโทรลเลอร์ ซึ่งเป็นแอดเดรส ส่วนบน และต่อกับแอดเดรสส่วนบนของ เอสแรม
57	P2.0(A9).A9	
51	P2.0(A10).A10	
56	P2.0(A11).A11	
50	P2.0(A12).A12	
58	P2.0(A13).A13	
60	P2.0(A14).A14	
28	P2.0(A15).A15	
16	A16	
3	A0	ต่อกับแอดเดรสส่วนล่างของ เอสแรม
4	A1	
5	A2	
78	A3	
79	A4	
82	A5	
83	A6	
84	A7	
61	OEB	ควบคุมสถานะเอาต์พุตของ เอสแรม
65	CEB	ควบคุมสถานะชิพของ เอสแรม
75	PC S7	ควบคุมสถานะอินพุตของ พอร์ตขนาน

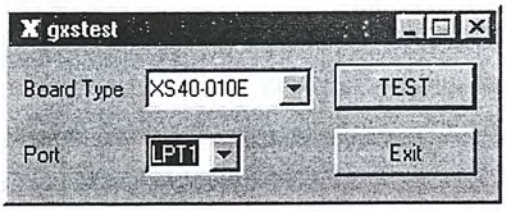
ตารางที่ 2-10 รายละเอียดพินต่างๆ ของบอร์ดทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.2 การทดสอบบอร์ดทดลอง

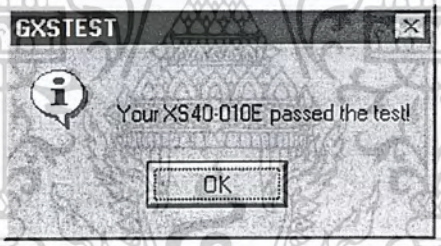
ใช้โปรแกรมทดสอบบอร์ดทดลอง XSTEST



รูปที่ 2-8 หน้าจอโปรแกรม XSTEST

เลือกรุ่นของบอร์ดทดลองที่ต้องการ แล้วกดปุ่ม Test

รูปที่ 2-9 หน้าจอโปรแกรม XSTEST ขณะลงข้อมูล



รูปที่ 2-10 หน้าจอโปรแกรม XSTEST เมื่อการทำงานเสร็จสมบูรณ์

ขั้นตอนการทดสอบบอร์ดทดลอง เริ่มจากโหลดข้อมูลลงไปที่ หน่วยความจำแล้วไมโครคอนโทรลเลอร์ จะประมวลผลโปรแกรม ซึ่งใช้ระยะเวลาทั้งหมด 15 วินาที ถ้าผลการทดสอบสมบูรณ์ แอลอีดี (LED) จะแสดงผล "0" แต่ถ้าผลการทดสอบบกพร่อง แอลอีดี (LED) จะแสดงผล "E" ซึ่งอาจมีปัญหาดังนี้

- แหล่งจ่ายไฟฟ้า กระแสตรงขนาด 9 โวลต์ อาจมีปัญหา
- ตรวจสอบหน้าสัมผัสของแหล่งจ่ายไฟฟ้า กับบอร์ดทดลอง
- ตรวจสอบจั้มเปอร์ ว่าอยู่ที่การตั้งค่าปกติหรือปล่าว
- ตรวจสอบว่าสายสัญญาณได้ต่อระหว่างบอร์ดทดลองกับ คอมพิวเตอร์ทางพอร์ตขนานหรือไม่

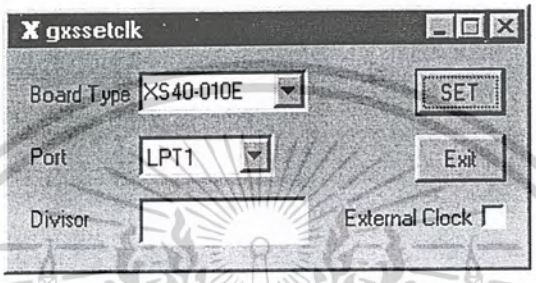
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
55113

- ตรวจสอบให้แน่ใจว่าพอร์ตขนานเป็นแบบ อีซีพี (ECP)โดยปกติทั่วๆ ไปสามารถเป็นได้ทั้ง เอสพีพี (SPP), อีพีพี (EPP), อีซีพี (ECP), หรือ ไบไดเร็กชั่น (bidirectional)

2.2.3 การเปลี่ยนแปลงสัญญาณนาฬิกา

บอร์ดทดลองจะมีความถี่หลักขนาด 100 MHz ซึ่งสามารถหารความถี่ด้วย 1, 2, ..., 2052 จะได้ความถี่ตั้งแต่ 100 เม็กกะเฮิร์ท (MHz), 50 เม็กกะเฮิร์ท (MHz), ..., 48.7 กิโลเฮริท์ (KHz) โดยจะส่งความถี่ไปที่ เอฟพีจีเอ เป็นสัญญาณนาฬิกา

ใช้โปรแกรมตั้งค่าสัญญาณนาฬิกาบอร์ดทดลอง XSSETCLK



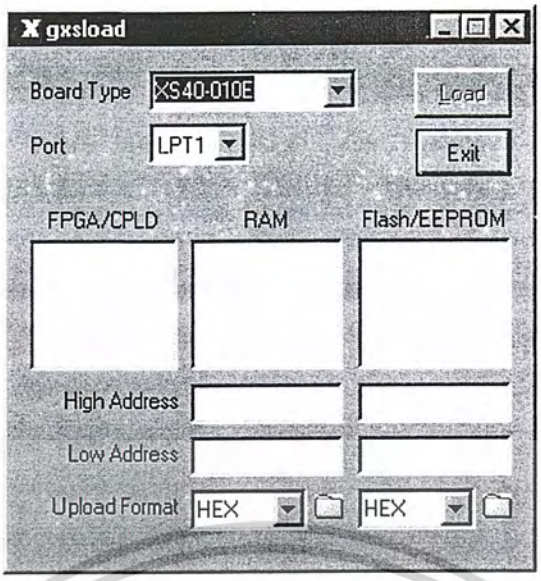
รูปที่ 2-11 หน้าจอโปรแกรม XSSETCLK

2.2.4 การโปรแกรมข้อมูลลงบนบอร์ดทดลอง

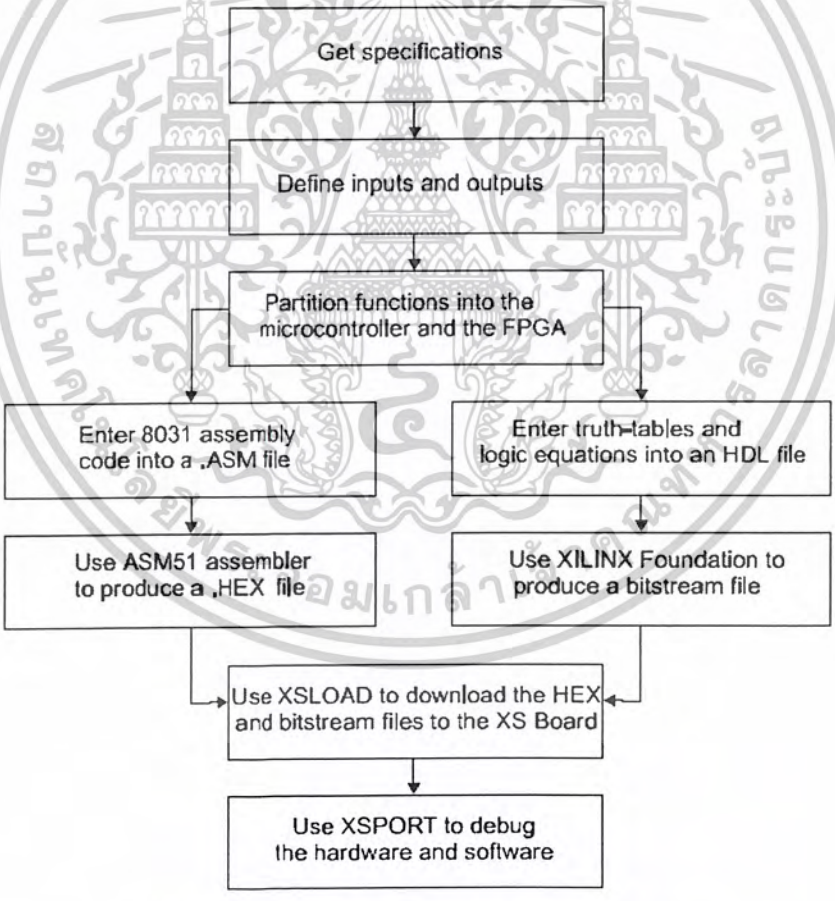
การโปรแกรมข้อมูลจากคอมพิวเตอร์ ลงบนบอร์ดทดลองจะเก็บข้อมูลลงบน อีอีพรอม โดยสามารถโปรแกรมข้อมูลที่เป็น บิตสตรีมไฟล์ (Bitstream file) โปรแกรมข้อมูลเฮ็คซ์ไฟล์ (HEX file) ลงบน เอสแรม และเฮ็คซ์ไฟล์ (HEX file) สามารถเก็บในไมโครคอนโทรลเลอร์ โดยได้รับการแก้ไขโค้ดจาก โปรแกรม ASM51 assembler โดยในบอร์ดทดลองรุ่นนี้สามารถเก็บ บิตสตรีม (Bitstream) ได้ 178,144 และมี อีอีพรอม รุ่น XC17256E

ใช้โปรแกรมโหลดข้อมูลลงบนบอร์ดทดลอง XSLOAD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-12 หน้าจอโปรแกรม XSLOAD



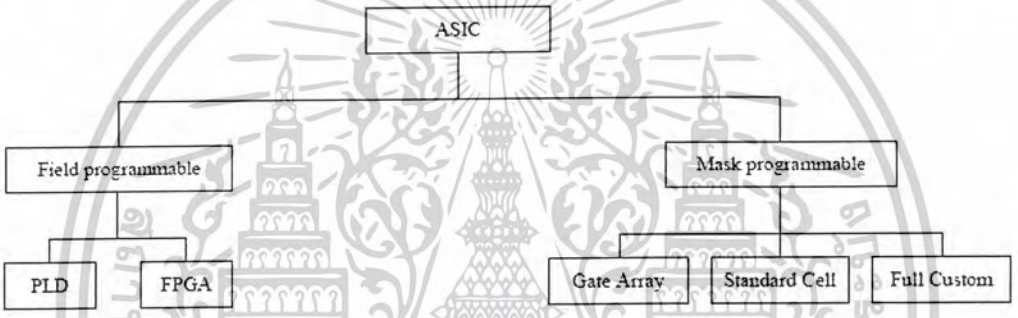
รูปที่ 2-13 แผนผังขั้นตอนการโปรแกรมข้อมูลลงบนเอฟพีจีเอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 รายละเอียดของอุปกรณ์ต่างๆ บนบอร์ดทดลอง

2.2.5.1 เอฟพีจีเอ (FPGA : Field Programmable Gate Array)

ความก้าวหน้าของอุตสาหกรรมอิเล็คทรอนิกส์ปัจจุบัน ทำให้เกิดการพัฒนาความสามารถ ของอุปกรณ์ต่างๆ มากมายซึ่งทำให้เกิดการลดค่าใช้จ่าย การสิ้นเปลืองพลังงานและขนาด ในขณะเดียวกันก็มีการเพิ่มประสิทธิภาพ และระดับความเชื่อถือได้ของวงจรรวมที่สูงขึ้นเห็นได้ชัดจาก เทคโนโลยีไมโครโปรเซสเซอร์และหน่วยความจำปัจจุบัน ทุกๆ ครั้งที่มีการพัฒนาขึ้นทำให้เกิดช่องว่างวงจรรวมและไอซีมาตรฐานมากขึ้น ในการพัฒนาเพิ่มความหนาแน่นและจำนวน ฟังก์ชั่นลอจิกที่เหมาะสม นักออกแบบอุปกรณ์ทางด้านดิจิตอล ได้พิจารณาถึงการผลิตให้ขนาดมากๆ และการผลิตวงจรรวม เอเอสไอซี (ASIC : Application Specific Integrated Circuit) ซึ่งวงจรรวม จะแบ่งตามการสร้าง ออกเป็น 2 กลุ่ม คือ ฟิลด์โปรแกรมเมเบิล (Field programmable) และ มาร์คโปรแกรมเมเบิล (Mask programmable)



รูปที่ 2-14 ASIC

อุปกรณ์วงจรรวม ASIC แบบฟิลด์โปรแกรมเมเบิล มีอยู่มากมายหลายชนิด แต่มีลักษณะการสร้างหรือกำหนดการทำงานของวงจรที่เหมือนกัน กล่าวคือ ผู้ใช้งานสามารถออกแบบและสร้างวงจรที่ต้องการใช้ลงในตัวอุปกรณ์ได้เองโดยไม่ต้องไปโรงงานเพื่อผลิต โดยเฉพาะ อย่างยิ่งในปัจจุบันนี้มีเครื่องมือที่ใช้ ช่วยในการออกแบบ และสร้างวงจร่วมกับ ไมโครคอมพิวเตอร์ที่มีความสามารถสูงในการพัฒนาตั้งแต่ขั้นการออกแบบ การจำลองการทำงาน จนถึงจัดสร้างวงจรลงในอุปกรณ์ รวมทั้งอุปกรณ์ ฟิลด์โปรแกรมเมเบิล เหล่านี้ สามารถหาซื้อได้ง่ายทำให้การสร้างวงจรอิเล็กทรอนิกส์จนถึงระบบไมโครโพรเซสเซอร์หันมาใช้อุปกรณ์จำพวกนี้ เป็นอุปกรณ์ประกอบในวงจรแทนอุปกรณ์ย่อยๆ แยกชิ้น (Discrete component) มาก

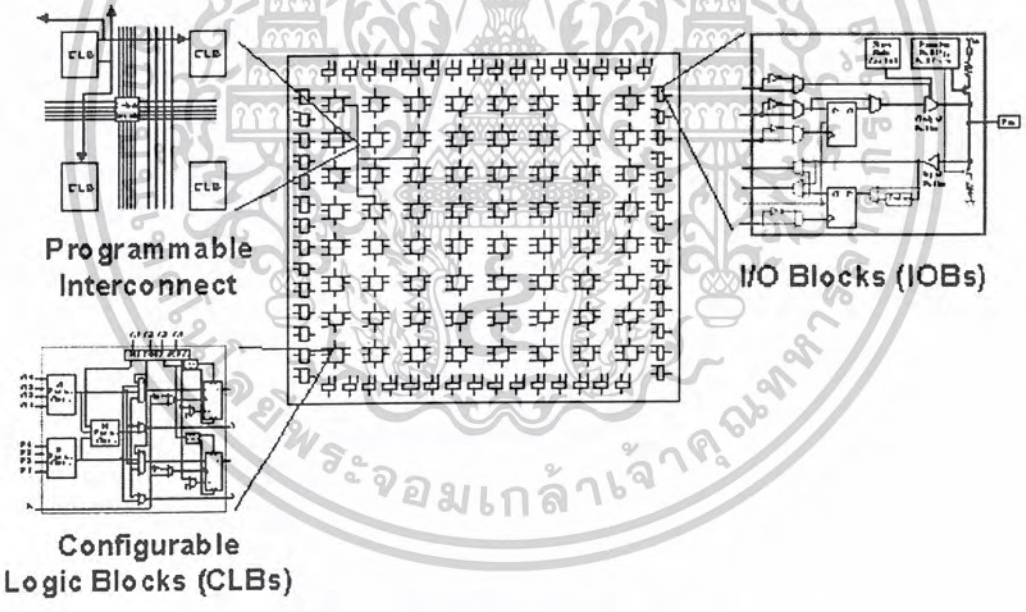
เอฟพีจีเอ เป็นอุปกรณ์ที่ถูกพัฒนาต่อจากอุปกรณ์แอลซีเอของบริษัทไซลิงซ์ (XILINX Inc.) โดยมีประสิทธิภาพการทำงานและมีปริมาณความหนาแน่นของเกตสูง สามารถจะกำหนดฟังก์ชั่นการทำงานได้ความต้องการของผู้ใช้โดยผ่าน การโปรแกรมเอฟพีจีเอได้รวบรวมข้อดี ทั้งหมดของการทำคัสตัมวีแอลเอสไอ(Custom VLSI) มารวมไว้ทั้งหมดได้แก่ การออกแบบการผลิต, ลดเวลาที่จะส่งตัวผลิตภัณฑ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกตลาด ซึ่งเป็นประโยชน์ต่อการผลิตวงจรเป็นอย่างมาก นักออกแบบเพียงกำหนดฟังก์ชั่นการทำงานของวงจร ดังนั้นการออกแบบวงจรโดยใช้เอฟพีจีเอ สามารถออกแบบและทดสอบภายในเวลาเพียง 2-3 วัน เท่านั้น ตรงกันข้ามกับการออกแบบโดยใช้เกตอาเรย์ ซึ่งใช้เวลาหลายสัปดาห์การเปลี่ยนแปลงแก้ไขแบบก็เช่นเดียวกัน จากประโยชน์ของเอฟพีจีเอ ดังกล่าวมา ทำให้เกิดการประหยัดค่าใช้จ่ายเป็นอย่างมาก เพราะได้ความเสี่ยงในการที่จะต้องแก้ไขตัววงจร การเลื่อนเวลาการออกผลิตภัณฑ์ ลดค่าเอ็นอาร์อี (NRE: Nonrecurring Engineering Cost) ลงไปด้วย

เอฟพีจีเอ เป็นชิบที่มีสนามวงจรขนาดใหญ่อยู่ภายใน ที่เราสามารถนำมาใช้ และออกแบบวงจรต่างๆได้ ตามที่เราต้องการโดยมีโปรแกรมสำเร็จรูปที่ใช้ในการออกแบบอยู่แล้ว มีความเร็วสูง มีสถาปัตยกรรมการออกแบบคล้าย ซีพีแอลดี (CPLD :Complex Programmable Logic Device) แต่มีส่วนประกอบที่ซับซ้อน และมีประสิทธิภาพมากกว่า ในการออกแบบวงจรสามารถทำได้ง่าย ทั้งในการเชื่อมต่อและการแก้ไข ดังนั้น เอฟพีจีเอ จึงเหมาะสมสำหรับการออกแบบวงจรเป็นอย่างมาก สถาปัตยกรรมภายในของเอฟพีจีเอ แบ่งเป็น 3 ส่วน คือ

- CLB (Configuration Logic Block)
- IOB (Input Output Block)
- Interconnect



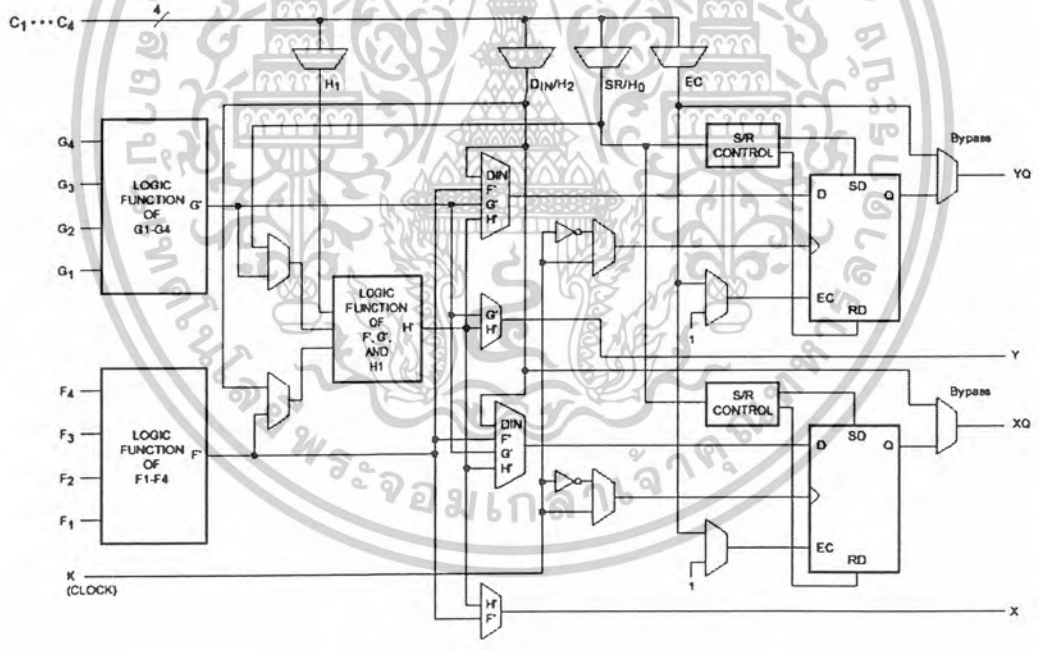
รูปที่ 2-15 สถาปัตยกรรมภายในของ เอฟพีจีเอ

ซึ่งภายในมีสถาปัตยกรรมของ เอฟพีจีเอ จะมีลักษณะเป็นตารางของลอจิกบล็อก (Logic block) และล้อมรอบไปด้วยบล็อกการเชื่อมต่อของไอโอ (I/O Interface block) การเชื่อมต่อระหว่างซีแอลบี (CLB: Configuration Logic Block) และไอโอบี (IOB: Input Output Block) ทำได้โดยผ่านช่องที่ว่างพาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านระหว่างแถว (Row) และคอลัมน์ (Column) มีการทำงาน เหมือนกันไมโครโพรเซสเซอร์ ตัวแอลซีเอจะทำงานได้ ต้องใช้ Program driven logic device หน้าที่ของซีแอลบีและไอโอบีแต่ละตัว การเชื่อมต่อภายใน (Interconnection) ถูกกำหนดไว้ในโปรแกรมคอนฟิกคูเรชัน (Configuration program) หรือเก็บไว้ใน อีพรอม (EPROM) ภายในแอลซีเอ(LCA : Logic Cell Array) โปรแกรมจะถูกโหลดเข้าสู่แอลซีเอเมื่อมีการจ่ายไฟ (Power-up) โดยทางคำสั่ง (Command) ซึ่งเป็นส่วนหนึ่งของการเริ่มต้นระบบ (System initialization) ประสิทธิภาพของแอลซีเอกำหนดโดย ความเร็วของลอจิกส่วนประกอบหน่วยความจำและการโปรแกรมการเชื่อมต่อต่างๆ ความเร็วของอัตราของระบบสัญญาณนาฬิกา (System clock rate) ถูกกำหนดด้วย ทอกเกิลฟลิปฟลอปสำหรับการประยุกต์ใช้โดยทั่วไปจะอยู่ที่ประมาณ 1/3 ถึง 1/2 ค่าสูงสุดของทอกเกิลเกต (Maximum toggle gate)

ซีแอลบี (CLB: Configuration Logic Block)ภายใน LCA(Logic Cell Array) คือเมทริกซ์ของ ซีแอลบี แต่ละตัวประกอบด้วยหน่วยของคอมเบนชั่นลอจิกที่สามารถโปรแกรมได้ (Programmable combinationlogic) และส่วนของเรจิสเตอร์เก็บข้อมูล (Storage register) ส่วนของวงจรคอมเบนชั่นลอจิกสามารถใช้สร้างวงจรทางด้านฟังก์ชั่นบูลีนของอินพุต ส่วนเรจิสเตอร์รับค่าจากส่วนคอมบิเนชั่นหรือโดยตรงจากเอาต์พุตของซีแอลบี สามารถขับวงจรคอมบิเนชั่นลอจิกโดยตรงผ่านเส้นทางเดินย้อนกลับ (Feedback path)

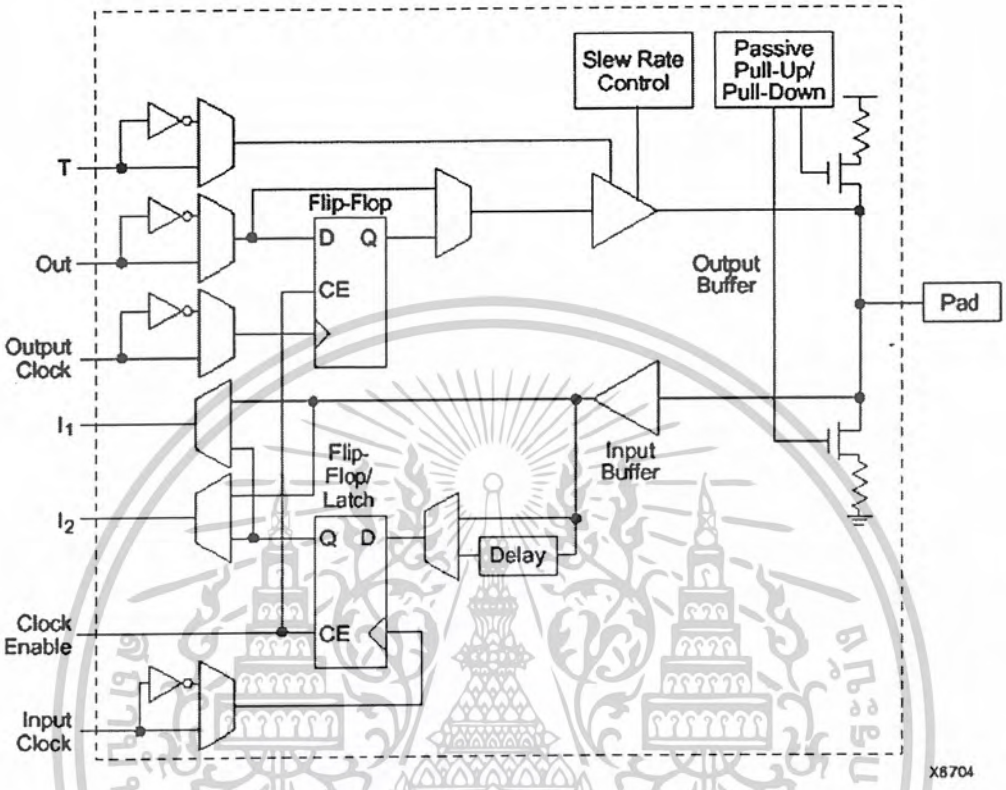


รูปที่ 2-16 โครงสร้างส่วน ซีแอลบี

ไอโอบี (IOB: Input Output Block)เป็นส่วนติดต่อกับวงจรภายนอกของแอลซีเอสร้างมาจากส่วนของอุปกรณ์อินพุต/เอาต์พุตที่สามารถโปรแกรมได้ (Programmable Input/Output device) แต่ละตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

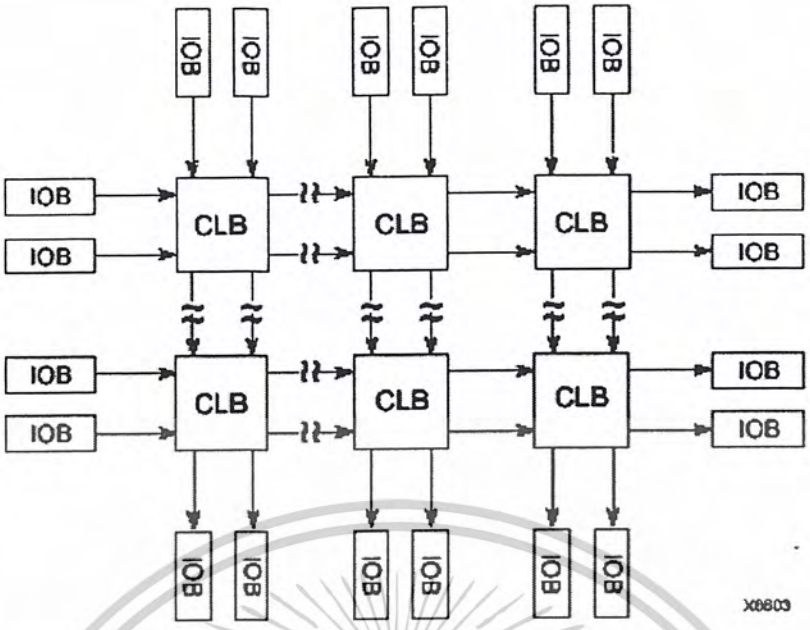
สามารถโปรแกรมได้อย่างอิสระโดยจะให้เป็นอินพุต/เอาต์พุตแบบ 3 สถานะหรือไอโอแบบสองทิศทางก็ได้ โดยอินพุตสามารถโปรแกรมให้รู้จักทั้งระดับสัญญาณทีทีแอลและซีมอสเทรดโฮล ของไอโอบี แต่ละตัวมีฟลิปฟลอปสามารถใช้เป็นบัฟเฟอร์สำหรับอินพุตและเอาต์พุต



รูปที่ 2-17 โครงสร้างส่วน ไอโอบี

อินเตอร์คอนเน็ค (Interconnect)ความยืดหยุ่นของการใช้แอลซีเอมาทำเป็นอุปกรณ์ขึ้นอยู่กับการโปรแกรม ทรัพยากรต่างๆที่อยู่ภายในเข้าด้วยกันการที่จะควบคุมการเชื่อมต่อระหว่างจุดสองจุดภายในชิปเหมือนกับ เกตอาเรย์ทั่วๆไปการเชื่อมต่อภายในแอลซีเอประกอบด้วยเน็ตเวริ์ค 2 ทิศทางคือทางแถว และคอลัมน์ซึ่งวางอยู่ระหว่าง CLB programmable switch จะทำการเชื่อมต่ออินพุต และเอาต์พุตของไอโอบีและซีแอลบี ที่จุดต่อร่วมระหว่างแถวกับคอลัมน์สามารถสลับสัญญาณจาก เส้นทางไปยังส่วนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-18 โครงสร้างส่วน อินเตอร์คอนเน็ค

2.2.5.2 คุณสมบัติโดยทั่วไปของเอฟพีจีเอ XC4010E

- มีฟลิปฟลอปเป็นจำนวน 1,120
- ในการผลิตฟังก์ชั่นของการทำงานมีความยืดหยุ่นสูง
- มีจำนวนเกตภายในจำนวน 7,000 – 20,000 เกต
- มี 160 อินพุต/เอาต์พุท
- เพิ่มความสามารถของรีจิสเตอร์ และอินพุต/เอาต์พุท
- มีค่าแฟนเอาต์ (fan-out) สูง
- มีบัสภายใน 3 สถานะ
- ทำงานกับสัญญาณทีทีแอล และซีมอส
- มีออสซิเลเตอร์แอมปลิฟลายเออร์ภายใน
- มีแรมภายในความสูง
- ใช้กับงานที่ต้องการความเร็วสูงได้
- มี ไวด์เอ็จดีโค๊ดเดอร์ (Wide edge decoder)
- เส้นทางการเชื่อมต่อ (Interconnect line) เป็นแบบลำดับชั้น
- มีการกระจายกำลังงานของสัญญาณต่ำ
- มีลอจิกบล็อค และไอโอบล็อคที่สามารถโปรแกรมได้
- มีอินเตอร์คอนเน็ค และ ไวด์ดีโค๊ดเดอร์ (Wide decoder) ที่โปรแกรมได้
- มีลอจิก และอินเตอร์คอนเน็คที่มีความเร็วสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ใช้กำลังงานต่ำ
- รองรับมาตราฐาน IEEE 1149.1 ในการทำ เบานด์คารี่สแกนลอจิก (boundary-scan logic)
- สามารถโปรแกรมค่า เอาท์พุตซลูเรท (output slew rate) ได้
- สามารถโปรแกรมให้อินพุตมีลักษณะพูลอัพ (Pull-up) หรือ พูลดาวน์ (Pull-down) รีจิสเตอร์ได้
- ให้กระแสเอาต์พุตได้ตั้งแต่ 12-24 มิลลิแอมป์
- ไม่จำกัดจำนวนครั้งในการโปรแกรมซ้ำ
- แรมภายในประมาณ 13 กิโลบิต

แรมในเอฟพีจีเอตระกูล XC4000แอลซีเอทำงานโดยใช้ตารางการค้นหา (Look-up table) ซึ่งจะทำการเก็บตารางที่ว่านี้ในสแตติกแรม ซึ่งจะถูกเขียนในระหว่างการโปรแกรมโครงแบบลงบนแอลซีเอ และจะถูกอ่านในการโอเปอ-เรชั่น ดังนั้นแรมภายในจึงควรถูกรวมไว้ในการออกแบบของผู้ใช้ด้วยหน้าที่ของแรมในเอฟพีจีเอตระกูล XC4000 มีหน้าที่คล้ายแรมโดยทั่วๆ ไป เช่น เอฟไอเอฟโอ (FIFO: First In First Out) แอลไอเอฟโอ (LIFO: Last In First Out) เรจิสเตอร์ไฟล์รวมทั้งแอปพลิเคชั่นบางอย่างอย่างเช่น เรจิสเตอร์เลื่อนข้อมูล (Shift register) แรมของเอฟพีจีเอตระกูล XC4000 มีความเร็วสูงเสมือนเอสแรม (SRAM) จึงไม่จำเป็นต้องคำนึงถึงเวลาหน่วงของการเชื่อมต่อ (Interconnection delay) โดยใน XC4010E จะมีแรมภายในประมาณ 13 กิโลบิต

2.2.5.3 เปรียบเทียบ เอฟพีจีเอ กับ ไมโครคอนโทรลเลอร์

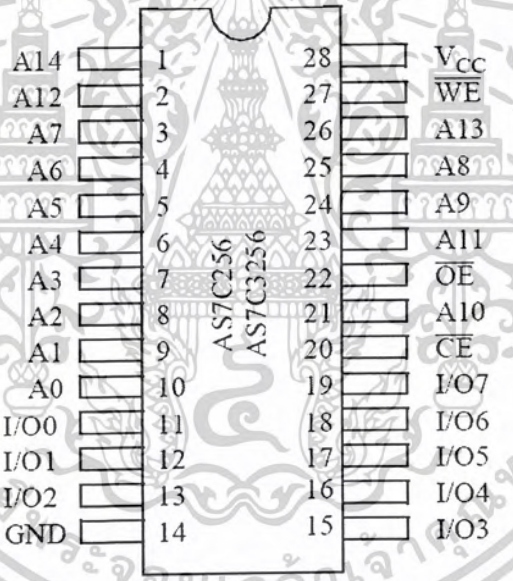
ผู้ที่ใช้งานอิเล็กทรอนิกส์ทั่วไปอาจมีความคุ้นเคยกับไมโครคอนโทรลเลอร์มากกว่า เอฟพีจีเอ ซึ่งไมโครคอนโทรลเลอร์ นั้นเป็นชิปประเภท เจเนรอลโพโพสดีไวด์ (general purpose devices) หมายความว่า สถาปัตยกรรมรวมถึงชุดคำสั่งทั้งหมดของชิปได้ถูกสร้างไว้อย่างถาวร ผู้ใช้ไม่สามารถเปลี่ยนแปลงได้ หากแต่มีหน้าที่เรียงร้อยชุดคำสั่งต่างๆ ที่มีอยู่ เกิดเป็นชุดลำดับของคำสั่งที่สามารถสั่งงานชิป ให้ปฏิบัติหน้าที่ให้เป็นไปตามประสงค์ ข้อจำกัดทางด้านสถาปัตยกรรม และชุดคำสั่งทำให้ไมโครคอนโทรลเลอร์ไม่ใช่คำตอบของงาน ทั้งหมดที่มีอยู่ ตัวอย่างเช่น ไมโครคอนโทรลเลอร์ที่ทำงานด้วยความเร็ว 20 เม็กกะเฮิร์ท ย่อมไม่สามารถใช้กับงานที่ต้องการความเร็วมากกว่านั้นซึ่งอาจเป็นคุณสมบัติที่จำเป็นมากสำหรับงานบางประเภท เช่นงานด้านสื่อสารโทรคมนาคมได้ ข้อจำกัดเหล่านี้จะไม่พบใน เอฟพีจีเอ เพราะผู้ใช้สามารถออกแบบสถาปัตยกรรมทั้งหมดได้เอง และยังสามารถทำงานแบบขนานที่สัญญาณนาฬิกามากกว่า 100 เม็กกะเฮริท์ได้อย่างมีประสิทธิภาพ นอกจากนั้น เอฟพีจีเอ ยังสามารถรวมเอาวงจรดิจิตอลอื่นๆ ที่อยู่ภายนอกชิปมาอยู่รวมกันบนชิปเอฟพีจีเอ ชิปเดียวได้ด้วย ดังนั้น เอฟพีจีเอ จึงเป็นได้มากกว่าทางเลือกหนึ่งของไมโครคอนโทรลเลอร์ แต่เป็นคำตอบที่ดีกว่าในงานออกแบบทางด้านดิจิตอล (digital design)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.4 เอสแรม (SRAM : Static Random-Access Memory)

เอสแรมที่ใช้งานบนบอร์ดทดลองนี้เป็นของ บริษัท แอลไลแอ็นซ์ เซมิคอนดักเตอร์ จำกัด เป็นเอสแรม ประเภทซีมอส (CMOS) ความสามารถสูงซึ่งสามารถรับข้อมูลได้เร็วที่แรงดันต่ำโดยมีรายละเอียดดังนี้

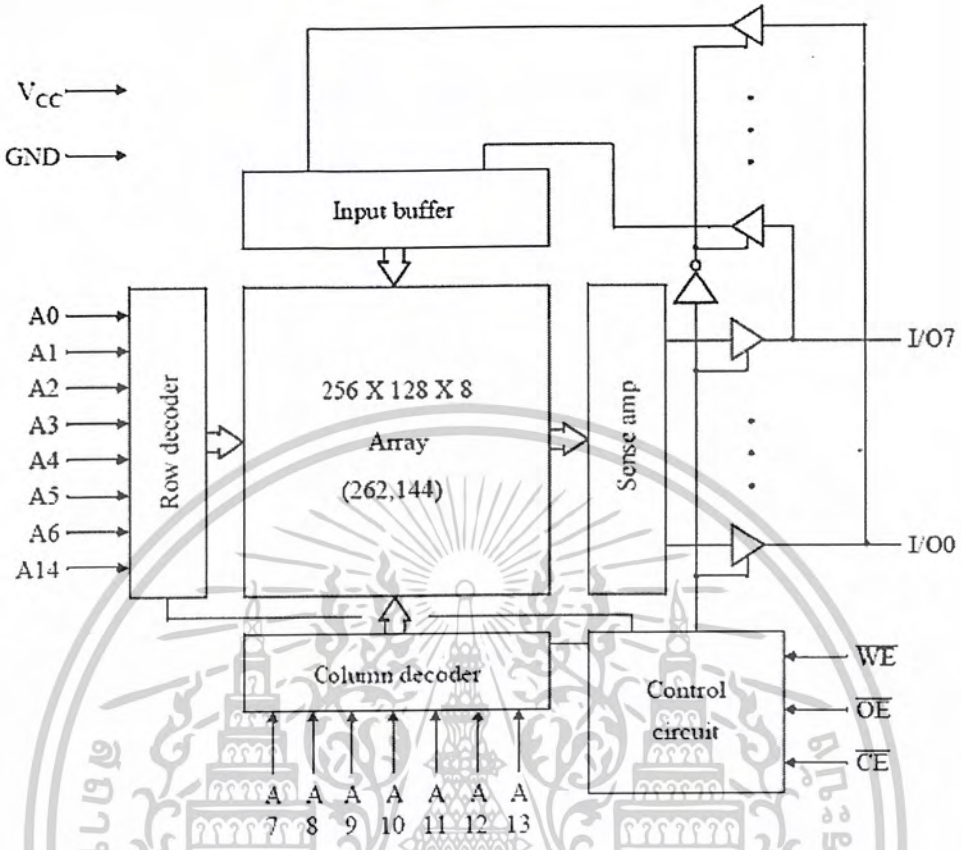
- รุ่น AS7C256
- ทำงานที่แรงดัน 5 โวลต์
- มีความจุข้อมูลประมาณ 262,144 บิต
- ทำงานที่ความเร็วประมาณ 15 นาโนเซ็ค ในการเข้าถึงที่อยู่ของข้อมูล
- ทำงานที่ความเร็วประมาณ 6 นาโนเซ็ค ในการเอาต์พุต
- ใช้กำลังไฟฟ้าต่ำเพียง 660 มิลลิวัตต์ ในการทำงาน
- ใช้กำลังไฟฟ้าต่ำเพียง 22 มิลลิวัตต์ ในการรอทำงาน
- ใช้ไฟเลี้ยง 2 โวลต์ ในการรักษาข้อมูล
- รับสัญญาณอินพุต และเอาต์พุต เป็น TTL



รูปที่ 2-19 โครงสร้างขาพินของ ชิพเอสแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.5 Logic block Diagram



รูปที่ 2-20 สถาปัตยกรรมภายในของ เอสแรม

มีการทำงานดังนี้ ถ้าขา CE เป็น "1" จะเป็นการเริ่มรอการทำงาน จะเริ่มทำงานเมื่อ ขา CE เป็น "0" โดยจะเป็นสถานะการเขียนข้อมูลเมื่อ ขา WE เป็น "0" และจะเป็นสถานะการอ่านข้อมูลเมื่อ ขา WE เป็น "1" OE เป็น "0" ดังตาราง

Truth table

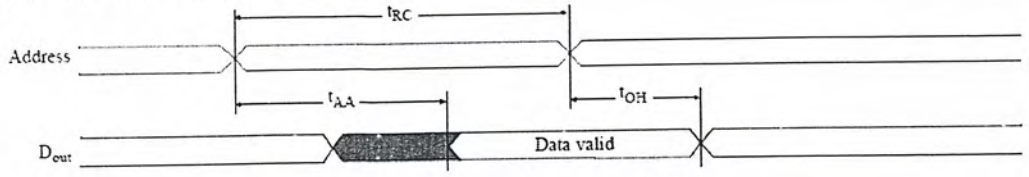
$\overline{CE}$	$\overline{WE}$	$\overline{OE}$	Data	Mode
H	X	X	High Z	Standby (I_{SB}, I_{SB1})
L	H	H	High Z	Output disable (I_{CC})
L	H	L	D_{OUT}	Read (I_{CC})
L	L	X	D_{IN}	Write (I_{CC})

Key: X = Don't care, L = Low, H = High

ตารางที่ 2-11 แสดงสถานะการทำงาน

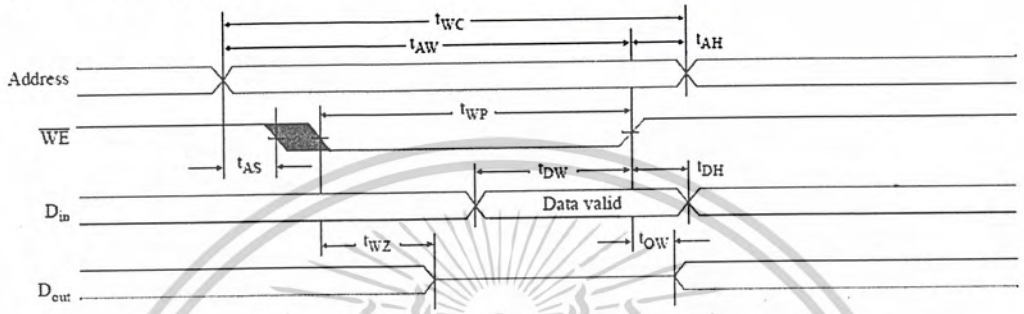
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read waveform 1 (address controlled)[3,6,7,9]

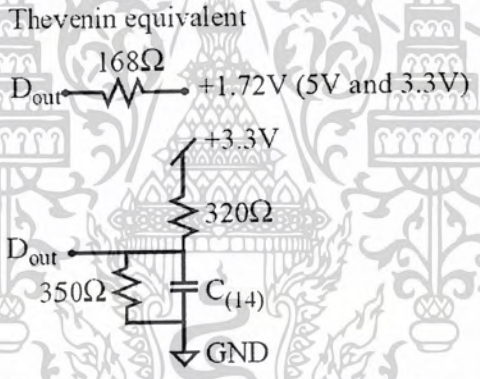


รูปที่ 2-21 รอบเวลาในการอ่านข้อมูล

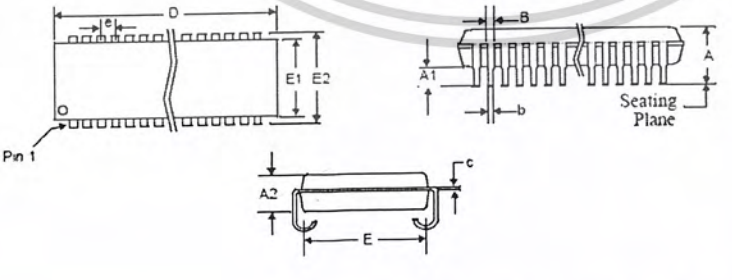
Write waveform 1 ($\overline{WE}$ controlled)[10,11]



รูปที่ 2-22 รอบเวลาในการเขียนข้อมูล



รูปที่ 2-23 โครงสร้างภายใน ขณะเอาต์พุต



	28-pin SOJ	
	Min	Max
A	-	0.140
A1	0.025	-
A2	0.095	0.105
B	0.028 TYP	
b	0.018 TYP	
c	0.010 TYP	
D	-	0.730
E	0.245	0.255
E1	0.295	0.305
E2	0.327	0.347
e	0.050 BSC	

รูปที่ 2-24 สถาปัตยกรรมภายนอกของ เอสแรม

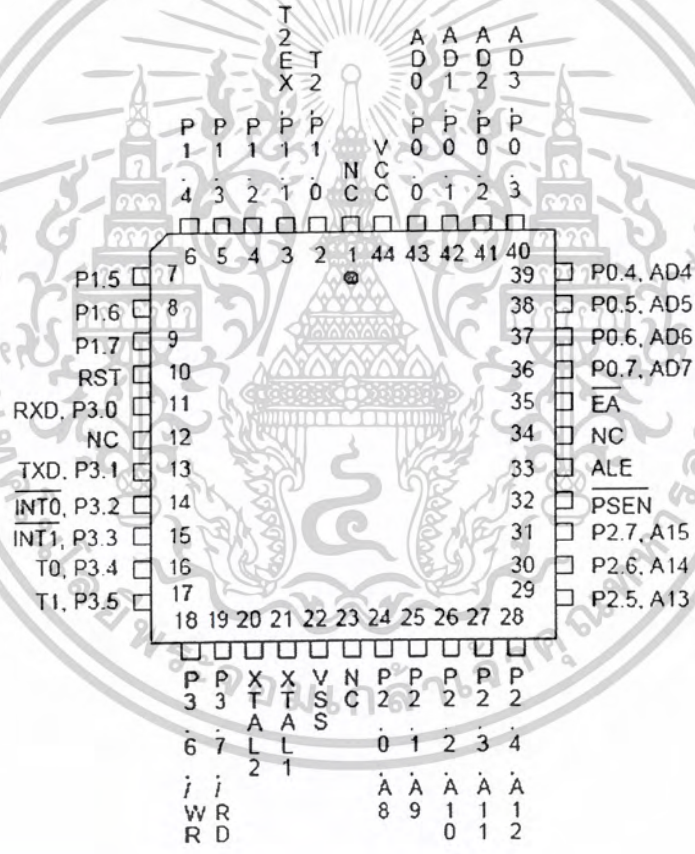
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5.6 ไมโครคอนโทรลเลอร์

เป็นแบบ 8 บิตไมโครคอนโทรลเลอร์ มาตรฐาน 80C32 ของ บริษัท วินบ็อนด์ อิเลคทรอนิคส์ จำกัด มีรายละเอียดดังนี้

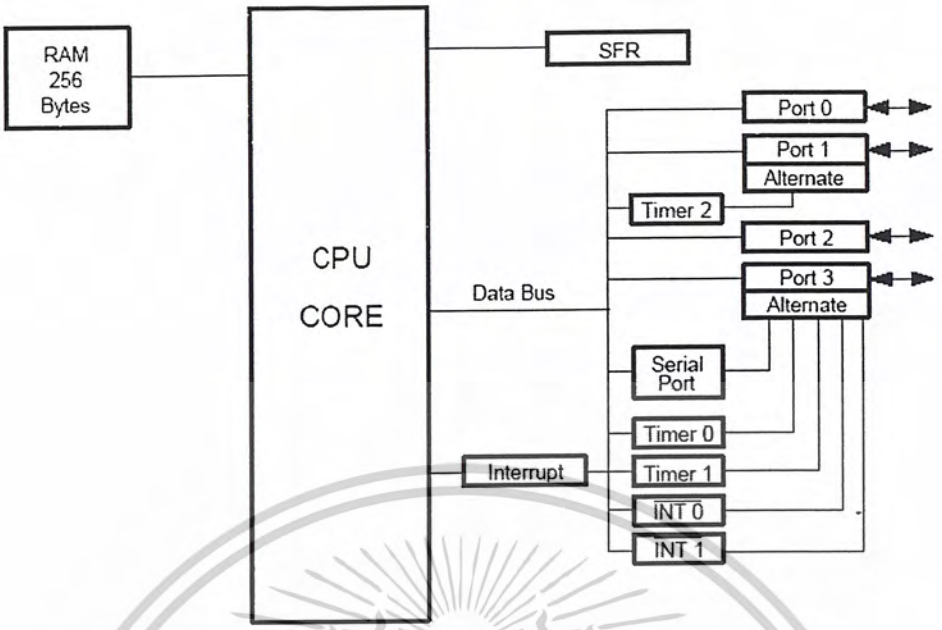
- เป็นการสื่อสาร 2 ทางแบบขนาน ขนาด 8 บิต ซีมอส (CMOS) มี 4 ชุด
- วงจรจับเวลา/นับ ขนาด !6 บิต 3 ชุด
- ความเร็ว 40 เม็กกะเฮิร์ท
- มีหน่วยความจำขนาด 256 ไบท์ บนชิพ
- มีพื้นที่ในการเก็บโปรแกรม ขนาด 64 กิโลไบท์
- มีพื้นที่ในการเก็บข้อมูล ขนาด 64 กิโลไบท์
- ซีเรียล พอร์ต (Serial port) ติดต่อแบบ ฟลูดูเฟล็ก (full duplex)

ไมโครคอนโทรลเลอร์ 44-Pin PLCC (W78C32CP)

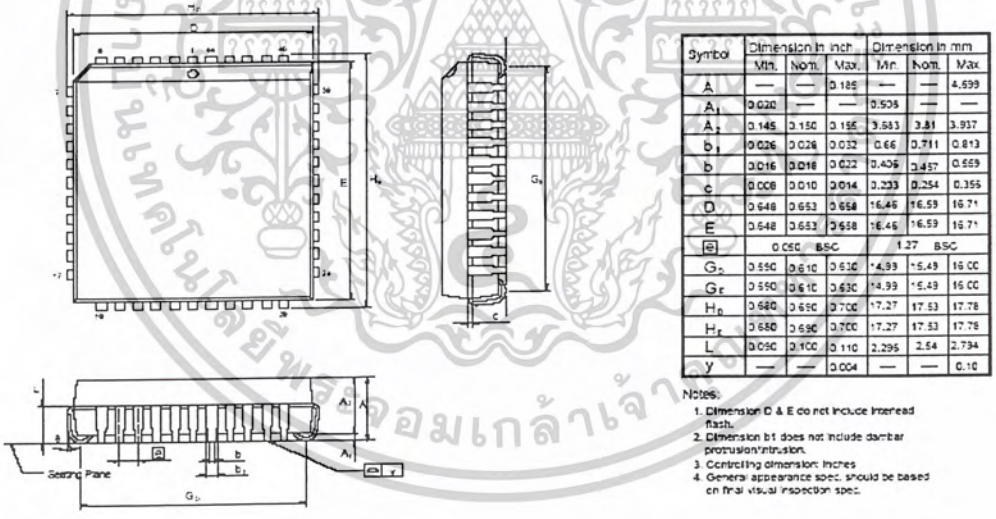


รูปที่ 2-25 โครงสร้างขาของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-26 โครงสร้างภายใน ไมโครคอนโทรลเลอร์



Symbol	Dimension in inch			Dimension in mm		
	Min.	Nom.	Max.	Min.	Nom.	Max.
A	—	—	0.185	—	—	4.699
A_1	0.020	—	—	0.508	—	—
A_2	0.145	0.150	0.155	3.683	3.81	3.937
b_1	0.026	0.028	0.032	0.66	0.711	0.813
b	0.016	0.018	0.022	0.406	0.457	0.559
c	0.008	0.010	0.014	0.203	0.254	0.356
D	0.648	0.653	0.658	16.46	16.59	16.71
E	0.648	0.653	0.658	16.46	16.59	16.71
e	0.050 BSC			1.27 BSC		
G_D	0.590	0.610	0.630	14.99	15.49	16.00
G_E	0.590	0.610	0.630	14.99	15.49	16.00
H_D	0.680	0.690	0.700	17.27	17.53	17.78
H_E	0.680	0.690	0.700	17.27	17.53	17.78
L	0.090	0.100	0.110	2.295	2.54	2.794
y	—	—	0.004	—	—	0.10

Notes:
1. Dimension D & E do not include interlead flash.
2. Dimension b1 does not include dambar protrusion/intrusion.
3. Controlling dimension: Inches
4. General appearance spec. should be based on final visual inspection spec.

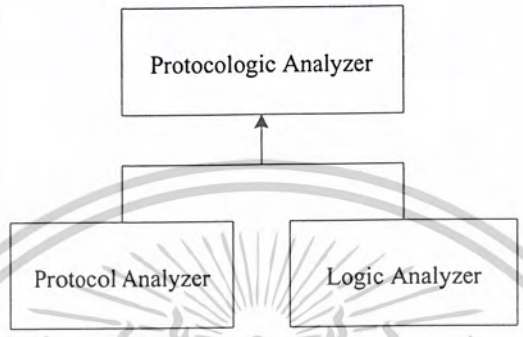
รูปที่ 2-27 สถาปัตยกรรมภายนอกของ ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขอบเขตของโครงงาน

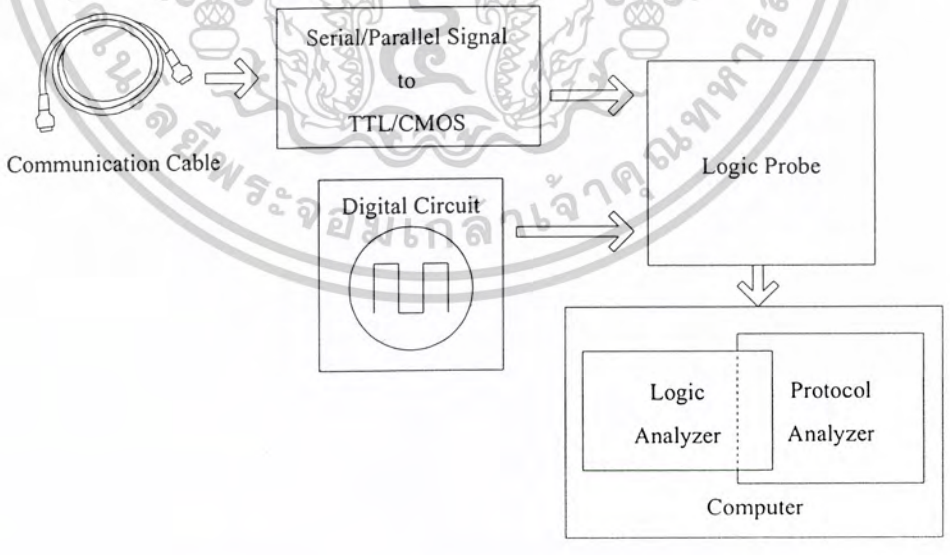
3.1 ภาพรวมและองค์ประกอบหลักของโครงงาน

โครงงานโปรโตคอลอจิกอนาไลเซอร์ ประกอบด้วย 2 ส่วนคือ โปรโตคอลอนาไลเซอร์ และลอจิกอนาไลเซอร์ ดังรูป



รูปที่ 3-1 องค์ประกอบของโปรโตคอลอจิกอนาไลเซอร์

ลอจิกอนาไลเซอร์ทำหน้าที่วิเคราะห์สัญญาณที่เป็น TTL/CMOS ซึ่งวัดสัญญาณมาจากวงจรดิจิตอล โดยผ่านลอจิกโปร์บ ส่วนโปรโตคอลอนาไลเซอร์ทำหน้าที่วิเคราะห์สัญญาณของการสื่อสาร ซึ่งวัดสัญญาณมาจากสายที่ส่งข้อมูล โดยต้องทำการแปลงสัญญาณให้เป็น TTL/CMOS ก่อน แล้วทำการสุ่มสัญญาณ (Sampling) โดยผ่านลอจิกโปร์บเช่นเดียวกับลอจิกอนาไลเซอร์



รูปที่ 3-2 ภาพรวมของโปรโตคอลอจิกอนาไลเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 คุณลักษณะหลักของโครงงาน

3.2.1 ฟังก์ชันหลักสำหรับใช้งานของซอฟต์แวร์

ฟังก์ชันหลักสำหรับใช้งาน ทั้งหมดมีดังนี้

ส่วนของโปรแกรมที่จัดการเกี่ยวกับไฟล์

1) เปิดข้อมูลที่บันทึกไว้ (Load file)

2) บันทึกข้อมูล (Save File)

3) พิมพ์ข้อมูล (Print)

ส่วนของโปรแกรมที่จัดการเกี่ยวกับการแสดงผลข้อมูล

1) วิเคราะห์โปรโตคอล (Analyze protocol) ในระดับ Physical Layer ได้หลายชนิด โดยในโครงงานนี้จะสนับสนุนเฉพาะ USB Protocol แต่สามารถเพิ่มเติมได้หลายชนิดในอนาคต ด้วยการใส่ Protocol Definition ชนิดที่ต้องการเข้าไป

2) ปรับแต่งชื่อและสีของแต่ละสัญญาณที่แสดงผล (Edit name and color for each channel)

3) แสดงผลในรูปแบบของเวลาหรือลำดับครั้งของการสุ่มสัญญาณ (Display as Time or Samples)

4) จัดกลุ่มของสัญญาณที่แสดงผล (Group define for channels)

5) แสดงผลแบบตัวเลข (View as number)

6) ปรับแต่งรูปแบบการแสดงผลแบบตัวเลข (Number list setup for display)

7) ย้ายตำแหน่งหน้าจอแสดงผลไปที่เคอร์เซอร์ (Number list to cursor)

8) แสดงผลแบบเวฟฟอร์ม (View as waveform)

9) แสดงค่าข้อมูล ณ ตำแหน่งของเคอร์เซอร์ (View cursor values)

10) ปรับแต่งรูปแบบการแสดงผลแบบเวฟฟอร์ม (Waveform setup for display)

11) ย่อหรือขยายเวฟฟอร์ม (Waveform zoom)

12) ย้ายตำแหน่งหน้าจอแสดงผลไปที่เคอร์เซอร์ (Waveform to cursor)

13) ค้นหาแพ็ทเทิร์ทที่ต้องการ (Search pattern)

14) ปรับแต่งลักษณะของหน้าจอแสดงผล (Edit display style)

15) เพิ่มหรือลดสัญญาณที่หน้าจอแสดงผล (Insert or Delete channel)

16) เพิ่มเคอร์เซอร์เข้าไปที่หน้าจอแสดงผล (Add cursor)

17) เรียงลำดับชื่อของสัญญาณที่หน้าจอแสดงผล (Sorting view)

ส่วนของโปรแกรมที่ติดต่อกับฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ปรับแต่งความเร็วของสัญญาณนาฬิกา (Clock setup)

2) สั่งให้เริ่มจับสัญญาณจากวงจร (Triger start)

3) สั่งให้หยุดจับสัญญาณจากวงจร (Triger stop)

3.2.2 ความสามารถของฮาร์ดแวร์

ความสามารถของฮาร์ดแวร์มีดังนี้

1) แปลงสัญญาณจากสัญญาณในสายส่งข้อมูล (Physical Layer Protocol Signal : Serial/Parallel Signal) ให้เป็นสัญญาณแบบ TTL/CMOS

2) สุ่มสัญญาณข้อมูล (Sampling)

3) ส่งสัญญาณข้อมูลไปยังคอมพิวเตอร์ผ่านพอร์ต Parallel หรือ USB

4) รับคำสั่งจากซอฟต์แวร์เพื่อเริ่มทำงาน หรือหยุดทำงาน

3.3 ขอบเขตของส่วนซอฟต์แวร์

จากผลการศึกษาและสำรวจ สามารถกำหนดขอบเขตของโปรแกรมที่พัฒนา โดยแยกตามประเภทดังนี้

3.3.1 ฟังก์ชันที่เพิ่มเติมเข้าไปในซอร์สโค้ดต้นแบบ

ฟังก์ชันที่ต้องเพิ่มเข้าไปในซอร์สโค้ดต้นแบบ มีดังนี้

ส่วนของโปรแกรมที่จัดการเกี่ยวกับการแสดงผลข้อมูล

1) วิเคราะห์โปรโตคอล (Analyze protocol) ในระดับ Physical Layer ได้หลายชนิด โดยในโครงงานนี้จะสนับสนุนเฉพาะ USB Protocol แต่สามารถเพิ่มเติมได้หลายชนิดในอนาคต ด้วยการใส่ Protocol Definition ชนิดที่ต้องการเข้าไป

2) แสดงผลในรูปแบบของเวลาหรือลำดับครั้งของการสุ่มสัญญาณ (Display as Time or Samples)

3) จัดกลุ่มของสัญญาณที่แสดงผล (Group define for channels)

4) แสดงผลแบบตัวเลข (View as number)

5) ปรับแต่งรูปแบบการแสดงผลแบบตัวเลข (Number list setup for display)

6) ปรับแต่งตารางการเปลี่ยนแปลงสเตจ (Edit state translation table)

7) ย้ายตำแหน่งหน้าจอแสดงผลไปที่เคอร์เซอร์ (Number list to cursor)

8) ย้ายเคอร์เซอร์มาที่หน้าจอแสดงผล (Cursor to Number list)

9) ปรับแต่งรูปแบบการแสดงผลแบบเวฟฟอร์ม (Waveform setup for display)

10) ย้ายตำแหน่งหน้าจอแสดงผลไปที่เคอร์เซอร์ (Waveform to cursor)

11) ย้ายเคอร์เซอร์มาที่หน้าจอแสดงผล (Cursor to waveform)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมที่ติดต่อกับฮาร์ดแวร์

1) ปรับแต่งความเร็วของสัญญาณนาฬิกา (Clock setup)

2) สั่งให้เริ่มจับสัญญาณจากวงจร (Trigger start)

3) สั่งให้หยุดจับสัญญาณจากวงจร (Trigger stop)

3.3.2 ฟังก์ชันและส่วนของโปรแกรมที่แก้ไขให้ดีขึ้น

ฟังก์ชันและส่วนของโปรแกรมที่แก้ไขให้ดีขึ้น มีดังนี้

ส่วนของโปรแกรมที่จัดการเกี่ยวกับการแสดงผลข้อมูล

1) ปรับแต่งชื่อและสีของแต่ละสัญญาณที่แสดงผล (Edit name and color for each channel)

2) เพิ่มเคอร์เซอร์เข้าไปที่หน้าจอแสดงผล (Add cursor)

ส่วนของโปรแกรมที่ติดต่อกับผู้ใช้

1) เมนู

2) ทูลบาร์

3) ส่วนแสดงข้อมูล

3.4 ขอบเขตของส่วนฮาร์ดแวร์

โครงงานนี้เป็นการสร้างโมดูล สำหรับโหลดลง FPGA เพื่อให้ทำหน้าที่เป็นส่วนประมวลผล ในการรับข้อมูลสัญญาณที่ทำการวัด พัฒนาโดยใช้บอร์ดสำหรับทดลองและสร้างโปรแกรมควบคุมบอร์ดให้ทำงานตามต้องการ โดยมีรายละเอียดดังต่อไปนี้

1) สามารถรับข้อมูลได้ 32 ช่องวัดสัญญาณ

2) สามารถรับข้อมูลจากสายเคเบิล แล้วแปลงให้เป็นสัญญาณ TTL/CMOS

3) ใช้ USB Port หรือ Parallel Port ในการส่งถ่ายข้อมูลกับคอมพิวเตอร์

4) ใช้ FPGA ความเร็วขั้นต่ำ 100 MHz ในการประมวลผล (ขึ้นอยู่กับความรุ่นของ FPGA บนบอร์ดทดลอง)

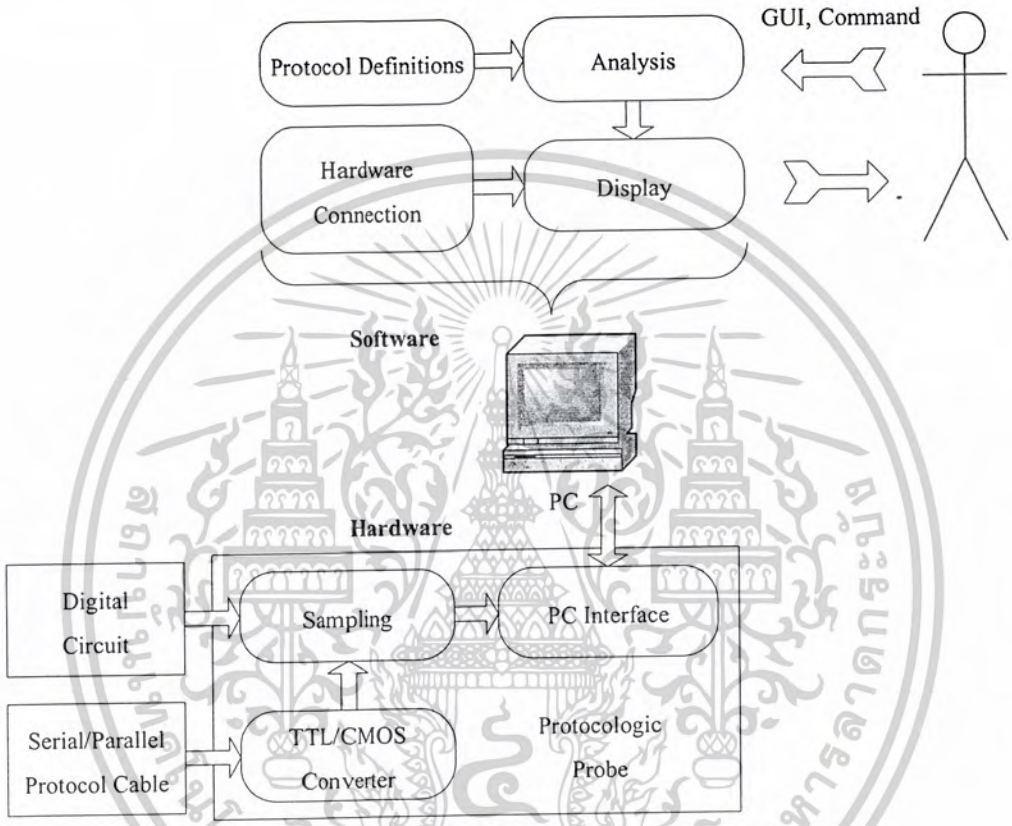
5) ลักษณะการทำงานของ Logic Analyzer จะเริ่มที่การจับสัญญาณจากอุปกรณ์ แล้วนำไปเก็บไว้ที่ หน่วยความจำ ก่อนจะนำส่งเข้าคอมพิวเตอร์ทางพอร์ตที่กำหนดแล้วจึงแสดงผลให้กับผู้ใช้ทราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การออกแบบโครงงาน

4.1 โครงสร้างของโครงงาน

โครงงานนี้ประกอบด้วยส่วนที่เกี่ยวข้องอยู่ 2 ส่วน คือ ส่วนของซอฟท์แวร์และฮาร์ดแวร์



รูปที่ 4-1 ผังการทำงานของระบบ

ส่วนของซอฟท์แวร์

โปรแกรมที่พัฒนาเป็นโปรแกรมที่สามารถได้ทั้งบนระบบปฏิบัติการ Linux และ MS Windows ซึ่งสามารถแบ่งได้เป็น 4 ส่วน คือ ส่วนนิยามโปรโตคอล (Protocol Definition) ส่วนติดต่อกับฮาร์ดแวร์ ส่วนวิเคราะห์ข้อมูล และส่วนแสดงผล

1) ส่วนนิยามโปรโตคอล ทำหน้าที่นิยามว่าสัญญาณที่วัดมาได้ใช้การส่งข้อมูลเป็นแบบใด เช่น ใช้การส่งแบบ USB แบบ Parallel หรือแบบ Ethernet เป็นต้น เพื่อใช้ในการ Analyze Protocol

2) ส่วนติดต่อกับฮาร์ดแวร์ ทำหน้าที่สั่งให้อุปกรณ์ทำงาน เช่น เริ่มและหยุดรับสัญญาณข้อมูลเข้าจากวงจรที่วัดสัญญาณ เป็นต้น และยังมีหน้าที่รับข้อมูลจากอุปกรณ์เข้ามายังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ส่วนวิเคราะห์ข้อมูล ทำหน้าที่ประมวลผลข้อมูลตามที่ผู้ใช้ต้องการ เช่น ผู้ใช้ต้องการที่ค้นหาข้อมูล โปรแกรมจะทำการค้นหาข้อมูลในส่วนที่ต้องการให้

4) ส่วนแสดงผล ทำหน้าที่แสดงข้อมูลที่รับมาจากอุปกรณ์ในรูปแบบของกราฟ (Waveform) หรือตัวเลข นอกจากนี้ยังแสดงผลตามที่ผู้ใช้ต้องการวิเคราะห์ข้อมูล

ส่วนของฮาร์ดแวร์

อุปกรณ์ทำหน้าที่ส่งข้อมูลให้โปรแกรมสามารถวิเคราะห์ข้อมูลได้ โดยประกอบด้วย 3 ส่วน คือ ส่วนแปลงสัญญาณเป็น TTL/CMOS ส่วนสุ่มสัญญาณจากวงจรดิจิตอล และส่วนเชื่อมต่อกับคอมพิวเตอร์

1) ส่วนแปลงสัญญาณเป็น TTL/CMOS ทำหน้าที่แปลงสัญญาณที่วัดจากสายส่งสัญญาณให้เป็นสัญญาณแบบ TTL หรือ CMOS เพื่อให้ส่วนสุ่มสัญญาณสามารถอ่านค่าสัญญาณได้

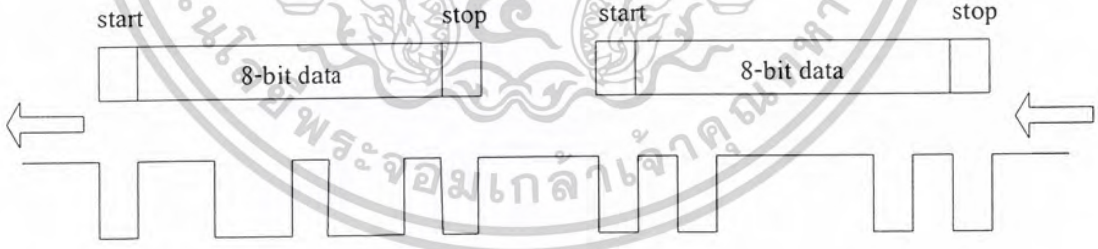
2) ส่วนสุ่มสัญญาณจากวงจรดิจิตอล ทำหน้าที่สุ่มสัญญาณในช่วงความถี่ที่กำหนดเข้ามาเก็บไว้เพื่อให้ได้สัญญาณสำหรับวัดและวิเคราะห์ต่อไป

3) ส่วนเชื่อมต่อกับคอมพิวเตอร์ ทำหน้าที่ส่งสัญญาณที่สุ่มได้ไปยังคอมพิวเตอร์ ผ่านทางพอร์ต USB หรือ Parallel

4.2 การออกแบบในส่วนซอฟต์แวร์

4.2.1 อินพุต

- สัญญาณข้อมูลที่ติดต่อกันในสายส่งสัญญาณ
- สัญญาณข้อมูลที่ส่งมาจากอุปกรณ์วัดสัญญาณ
- รับข้อมูลที่ผู้ใช้ต้องการให้โปรแกรมประมวลผลหรือแสดง เช่น Zoom, Search, Add signal เป็นต้น

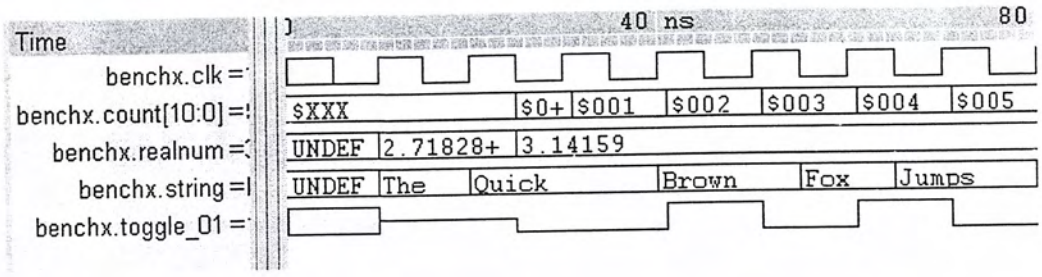


รูปที่ 4-2 ตัวอย่างอินพุตสัญญาณจาก Serial Port (RS-232)

4.2.2 เอาต์พุต

- แสดงผลข้อมูลรูปแบบต่างๆ ตามที่ผู้ใช้ต้องการ เช่น แสดงในรูปของกราฟ (Waveform) แสดงในรูปของตัวเลข ข้อมูลที่ผู้ใช้ค้นหา หรือข้อมูลที่ได้จากการวิเคราะห์สัญญาณสื่อสาร เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-3 ตัวอย่างของ Output

4.2.3 เครื่องมือที่ใช้ในการพัฒนา

1) GTK+ เป็น toolkit ที่ใช้สร้าง Graphic User Interfaces (GUI)

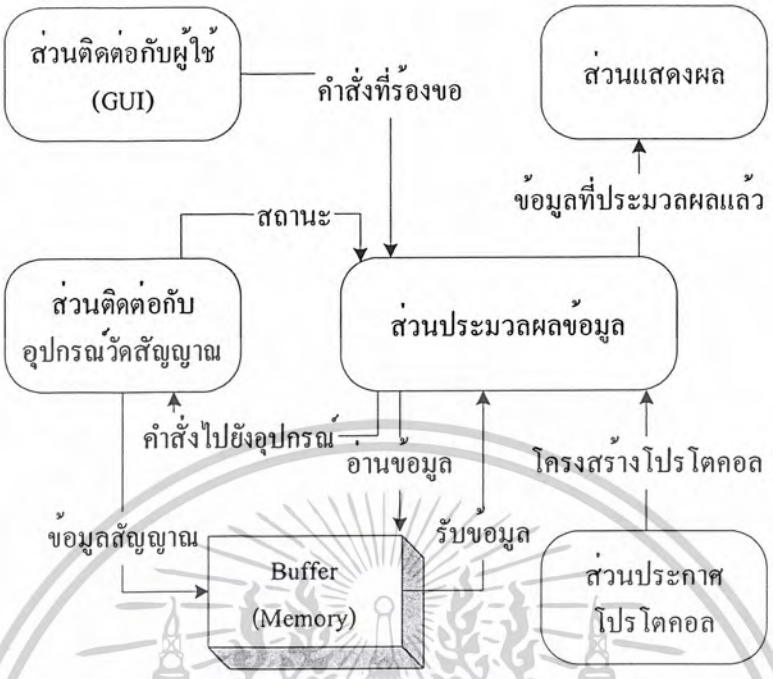
2) gcc เป็น compiler ที่ใช้คอมไพล์ source code ที่เขียนด้วยภาษา C

3) Glade เป็น tool ที่ช่วยสร้าง GUI ให้ง่ายขึ้น

4) Anjuta เป็น source code editor สำหรับเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 โครงสร้างของซอฟต์แวร์ (Software Design)



รูปที่ 4-4 ผังการทำงานของโปรแกรม

- ส่วนติดต่อกับผู้ใช้ ทำหน้าที่รับคำสั่งจากผู้ใช้แล้วส่งให้โปรแกรมประมวลผลข้อมูลทำงานเพื่อให้ได้ข้อมูลที่ต้องการ เช่น ต้องการค้นหาข้อมูล ต้องการขยายกราฟ (Waveform) ต้องการเพิ่มช่องสัญญาณที่แสดงผล เป็นต้น

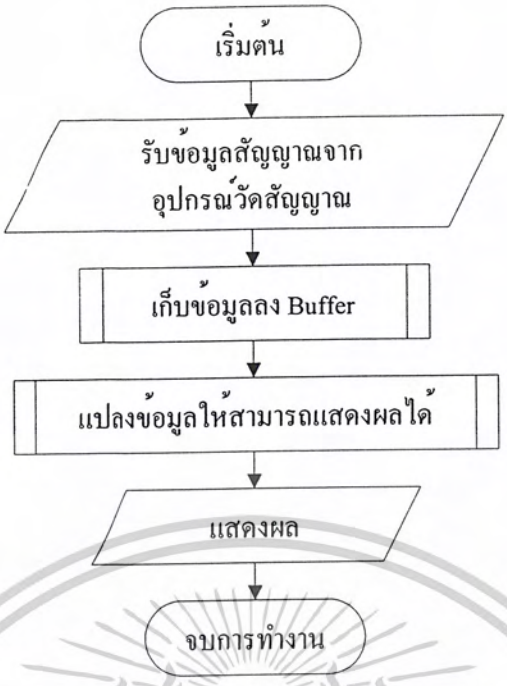
- ส่วนติดต่อกับอุปกรณ์วัดสัญญาณ ทำหน้าที่สั่งงานให้อุปกรณ์ทำงาน และรับข้อมูลสัญญาณที่วัดได้มาเก็บไว้ที่ buffer หรือหน่วยความจำ

- ส่วนนิยามโปรโตคอล ทำหน้าที่นิยามโครงสร้างของโปรโตคอลที่ใช้ในการสื่อสารของสัญญาณที่วัด และส่งโครงสร้างของโปรโตคอลให้กับส่วนประมวลผลข้อมูล เมื่อมีการทำ Protocol Analyze

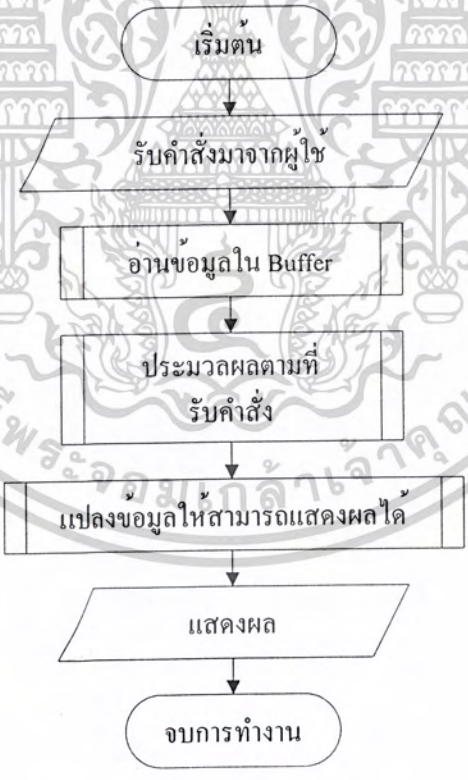
- ส่วนประมวลผลข้อมูล ทำหน้าที่วิเคราะห์ข้อมูล ค้นหาข้อมูล หรือจัดเรียงข้อมูล ซึ่งเป็นการทำให้ได้มาซึ่งข้อมูลที่ต้องการด้วยวิธีการต่างๆ แล้วส่งให้ส่วนแสดงผล

- ส่วนแสดงผล ทำหน้าที่แปลงข้อมูลดิบให้สามารถแสดงผลในรูปแบบต่างๆ ได้ เช่น Waveform เลขฐาน 2 และเลขฐาน 16 เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 การทำงานเมื่อรับข้อมูลสัญญาณมาจากอุปกรณ์วัดผลแล้วนำไปแสดงผล



รูปที่ 4-6 การทำงานเมื่อรับคำสั่งจากผู้ใช้แล้วแสดงผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

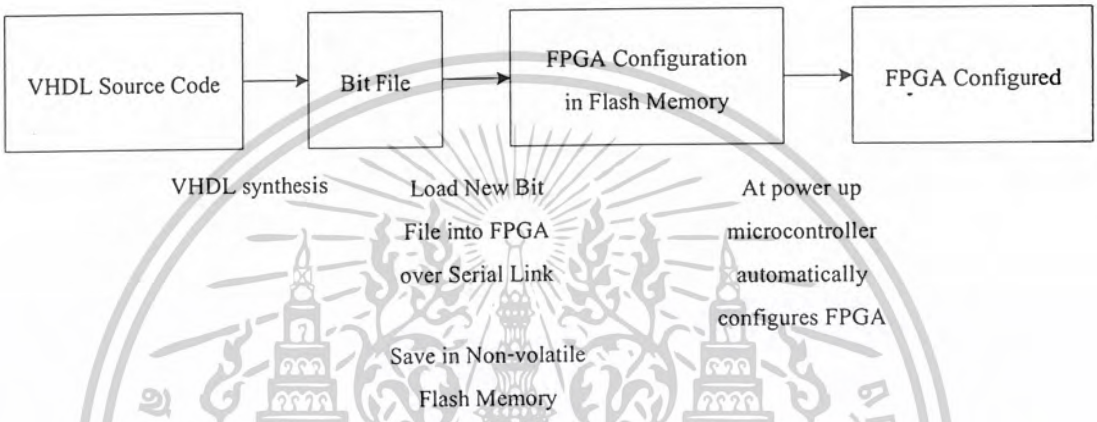
4.3 การออกแบบในส่วนฮาร์ดแวร์

1) ศึกษาทฤษฎีเกี่ยวกับ Logic Analyzer ในส่วนต่างๆ คือ

- ศึกษาทฤษฎีพื้นฐานของ Logic Analyzer
- ศึกษาการออกแบบโครงสร้าง Logic Analyzer ซึ่งก็มีเรื่องหลัก ๆ อยู่ 4 เรื่องด้วยกัน คือ ส่วนประมวลผลหลัก (FPGA), ส่วนติดต่อหน่วยความจำ (Ram), ส่วนรับสัญญาณ และส่วนติดต่อกับคอมพิวเตอร์

2) ออกแบบโครงสร้างของฮาร์ดแวร์ ให้ได้ตรงตามที่ต้องการ

3) ทดสอบการทำงานของ Logic Analyzer



รูปที่ 4-7 ขั้นตอนการโหลดซอร์สโค้ดลงบอร์ดทดลอง

4) ทำการเพิ่มรายละเอียดในจุดต่างๆ จนได้ฮาร์ดแวร์ที่สมบูรณ์

4.3.1 การออกแบบบอร์ดทดลองด้วย VHDL

VHDL เป็นภาษาระดับสูงที่ใช้ในการออกแบบระบบ และ วงจรลอจิก VHDL เป็นภาษาที่สามารถออกแบบวงจรได้ในระดับต่างๆ จะประกอบไปด้วยโครงสร้างต่างๆที่ถูกออกแบบขึ้นมา ในระดับที่สูงขึ้นตัวภาษานั้นสามารถที่จะออกแบบระบบโดยไม่คำนึงกระบวนการหรือวิธีการของวงจร เพราะวงจรจะถูกสร้างในรูปแบบของฟังก์ชั่น เราจะพิจารณาเพียงจุดมุ่งหมายของการออกแบบวงจรเท่านั้น แต่ถึงอย่างไรก็ตามการที่จะทำให้เราทราบถึงการทำงานและขอบเขตของจุดมุ่งหมายอย่างแท้จริงแล้ว เราจำเป็นจะต้องเข้าใจและคุ้นเคยกับโครงสร้างของวงจรนั้น และที่สำคัญอีกอย่างก็คือ ตัวภาษาที่ใช้ในการออกแบบวงจรเหล่านั้นด้วย ดังนั้นในบทความนี้จะกล่าวถึงหลักการและโครงสร้างต่างๆในการเขียนภาษา VHDL รวมไปถึงเทคนิคที่สำคัญในการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 องค์ประกอบที่สำคัญของ VHDL

VHDL มีองค์ประกอบที่สำคัญอยู่ 2 ส่วนคือ Entity และ Architecture

1) Entity เป็นเสมือน Block ที่เราสร้างขึ้นมาเพื่อจะบอกถึงจุดเชื่อมต่อภายนอก (I/O) ว่ามีอะไรบ้าง โดยไม่ต้องมีการอธิบายโครงสร้างภายใน

2) Architecture ใช้ในการอธิบายโครงสร้างภายในของ Entity

4.3.3 รูปแบบการเขียน Architecture ของ VHDL

VHDL มีรูปแบบการเขียนอยู่ 3 รูปแบบ

1) Structural Model การอธิบายวงจรโดยใช้โมดูลต่างๆมาเชื่อมต่อกันให้เห็นโครงสร้างภายใน

2) Behavioral Model การอธิบายการทำงานของวงจรในระดับลอจิกเกท

3) Sequential Model การอธิบายขั้นตอนการทำงานโดยใช้ Sequential Statement ทำงานทีละขั้นตอน

สำหรับรูปแบบการเขียนที่เราจะเลือกใช้นั้น ขึ้นอยู่กับความเหมาะสมของฮาร์ดแวร์ เราสามารถที่จะใช้รูป แบบต่างๆมาร่วมกันได้อยู่ใน Architecture ตัวเดียวกันก็ขึ้นอยู่กับความเหมาะสมเช่นกัน จะเห็นว่าในแต่ละแบบก็มีจุดเด่นของตัวมันเอง แต่ทั้ง 3 แบบก็ให้ผลลัพธ์เหมือนกัน

Sequential มีการอธิบาย Architecture เป็นลำดับขั้นตอนที่ชัดเจน เข้าใจง่าย โดยปกติแล้วโปรแกรมที่อยู่ภายใน Architecture นั้นจะมีการ process พร้อมๆกันทุกบรรทัด (เหมือนวงจรจริงที่มีการเชื่อมต่อสัญญาณกันทั้งวงจร เมื่อมีสัญญาณ input ตัวใดเปลี่ยน output ก็จะมีการเปลี่ยนตามทันที จะมี delay ก็เพียงเล็กน้อย) แต่แบบ Sequential สามารถจะมองเป็นขั้นตอนได้โดยใส่ Process Statement คร่อมเข้าไปในโปรแกรม ส่วน Structural ก็มีการดึง Component NAND มาจะมองเห็นการเชื่อมต่อที่ชัดเจน ส่วน Behavioral นั้นมีการอธิบายคุณสมบัติระดับลอจิกเกท

4.3.4 ข้อดีของ VHDL

1) รูปแบบของภาษาที่เข้าใจได้ง่าย

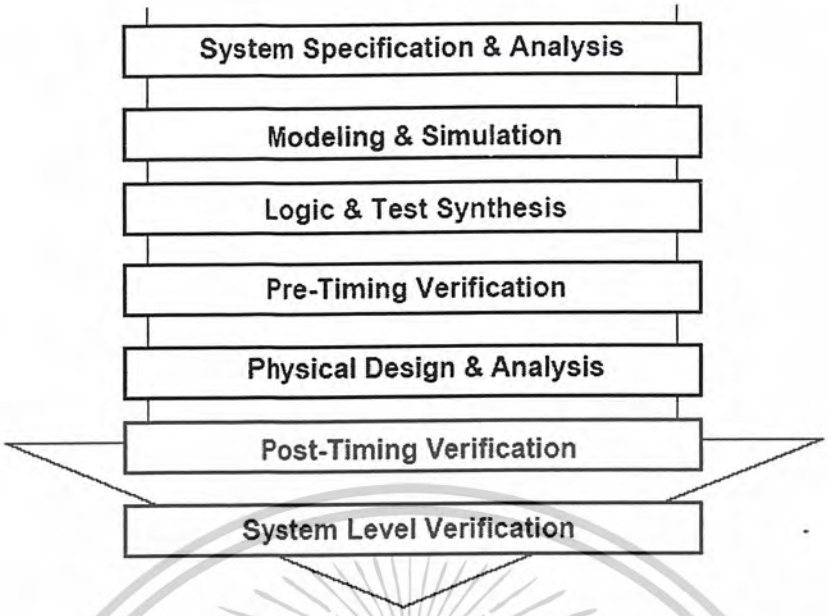
2) มีโครงสร้างของภาษาที่สามารถปรับเปลี่ยนให้เข้ากับ ฮาร์ดแวร์ (Hardware) ได้ง่าย

3) มี สเตทเม้นท์ (Statement) ให้ใช้งานอยู่หลายตัว

4) มีไลบรารี่ (library) ต่างๆให้เลือกใช้งานได้ทำให้ออกแบบได้ง่ายขึ้น เป็นประโยชน์ต่อโปรแกรมเมอร์ (Programmer)

5) สามารถออกแบบในรูปแบบ top-down design ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-8 แสดงขั้นตอน Top-Down Design

4.3.5 VHDL ช่วยในการออกแบบได้อย่างไรบ้าง

1) การออกแบบวงจรสามารถทำได้โดยง่าย

2) เราไม่จำเป็นต้องรู้ถึงโครงสร้างภายในของชิพ เพียงแต่รู้ขั้นตอนการออกแบบลอจิกก็พอ

3) ภาษามีความยืดหยุ่นสูง เพียงแต่กำหนดลักษณะการทำงานให้มัน

4) ภาษา VHDL สามารถใช้ได้กับซอฟท์แวร์ทุกตัว ดังนั้นจึงสามารถทำได้กับชิพทุกตัว ทุกบริษัท

5) เมื่อมีการแก้ไข หรือ เปลี่ยนแปลงวงจรสามารถทำได้โดยง่าย เพียงโปรแกรมวงจรใหม่ลงไป ไม่ต้องมีการเปลี่ยนแปลงตัว ฮาร์ดแวร์

6) การทดสอบไม่จำเป็นต้องทดลองกับวงจรจริง สามารถใช้การ ซิมูเลท (Simulate) ทดลองดูผลลัพธ์ของวงจรที่เราออกแบบได้ว่าถูกต้องหรือไม่

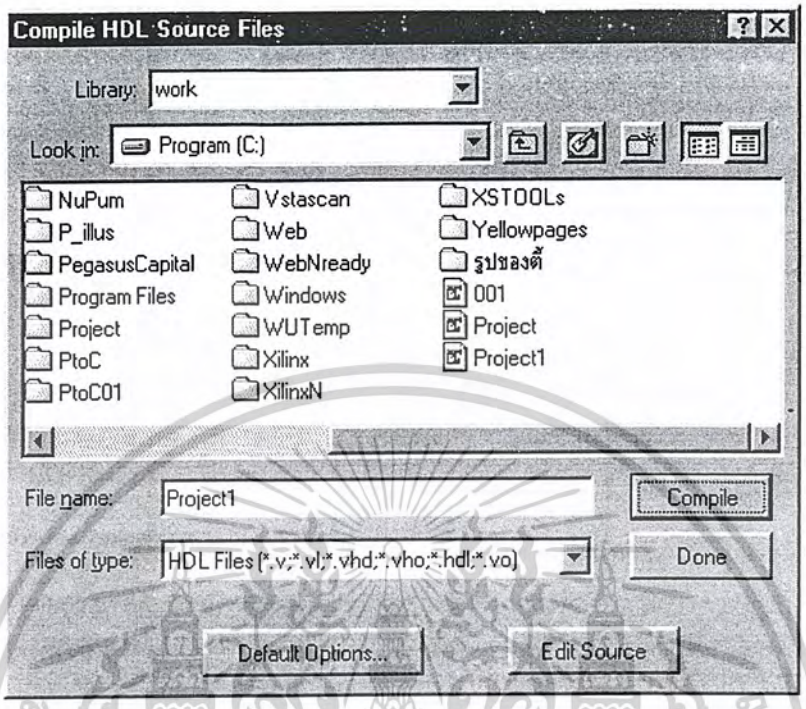
มีขั้นตอนการออกแบบดังนี้

1) วิเคราะห์ความต้องการของระบบ และนำมาวิเคราะห์และหาแนวความคิดและหลักการในการแก้ไขปัญหา และนำมาเขียนเป็น รูปแบบ ของระบบ ตลอดจนวางแผนการทดสอบว่าจะสามารถทดสอบความถูกต้องของวงจรที่ได้จากการออกแบบอย่างไร ซึ่งรวมไปถึงการออกแบบวิธีการทดสอบ เป็นต้น

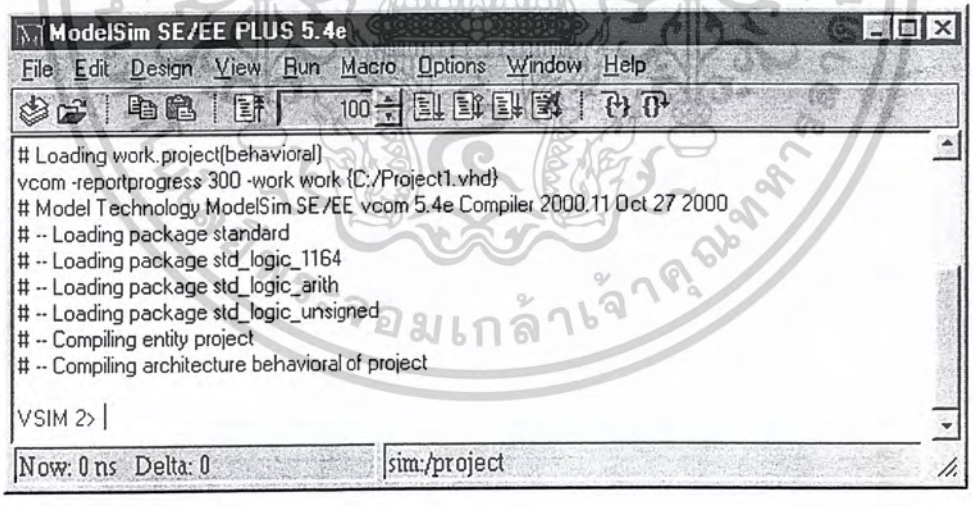
2) นำเอารูปแบบ ซึ่งได้จากขั้นตอนแรก มาเขียนเป็นภาษา VHDL เพื่อบรรยายวงจรให้อยู่ในรูป hardware-description language ในการเขียนวงจรด้วยภาษา VHDL นี้สามารถเขียนได้ในหลายรูปแบบด้วยกัน เช่น แบบ structure, แบบ behavior, แบบ RTL หรือใช้ทั้ง 3 รูปแบบร่วมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) นำโค้ด VHDL ที่เขียนขึ้นมาไปทดสอบด้วยโปรแกรมซิมูเลเตอร์ซึ่งจะใช้โปรแกรม ModelSim PE/Plus เป็นตัวทดสอบการทำงาน

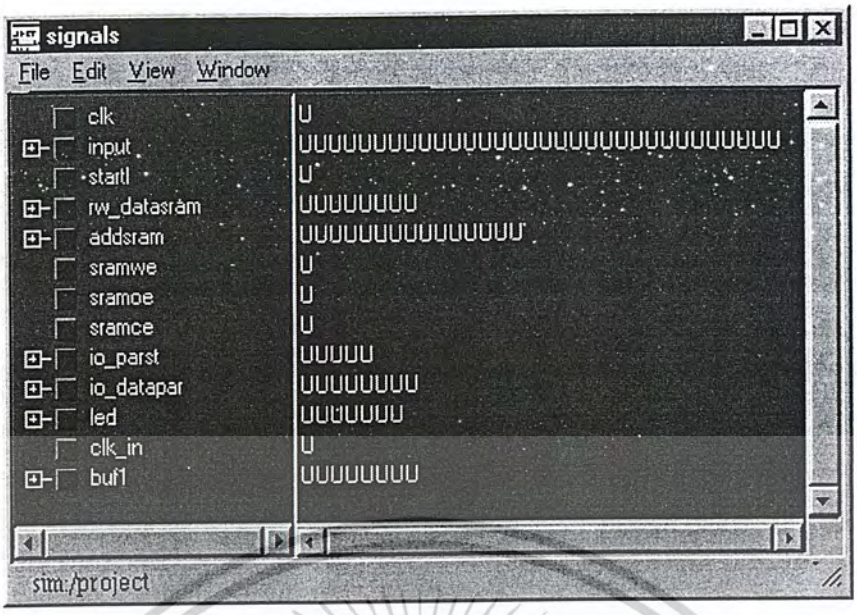


รูปที่ 4-9 การเลือกไฟล์ VHDL

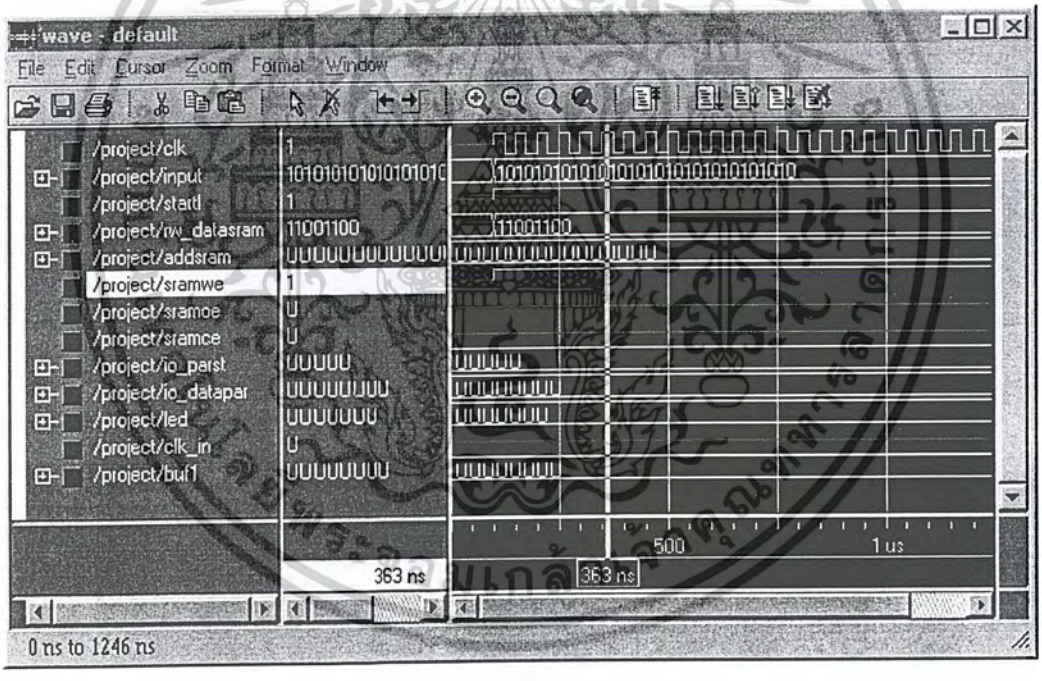


รูปที่ 4-10 แสดงการโหลดโค๊ด VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-11 แสดงสัญญาณต่างๆในโปรแกรม

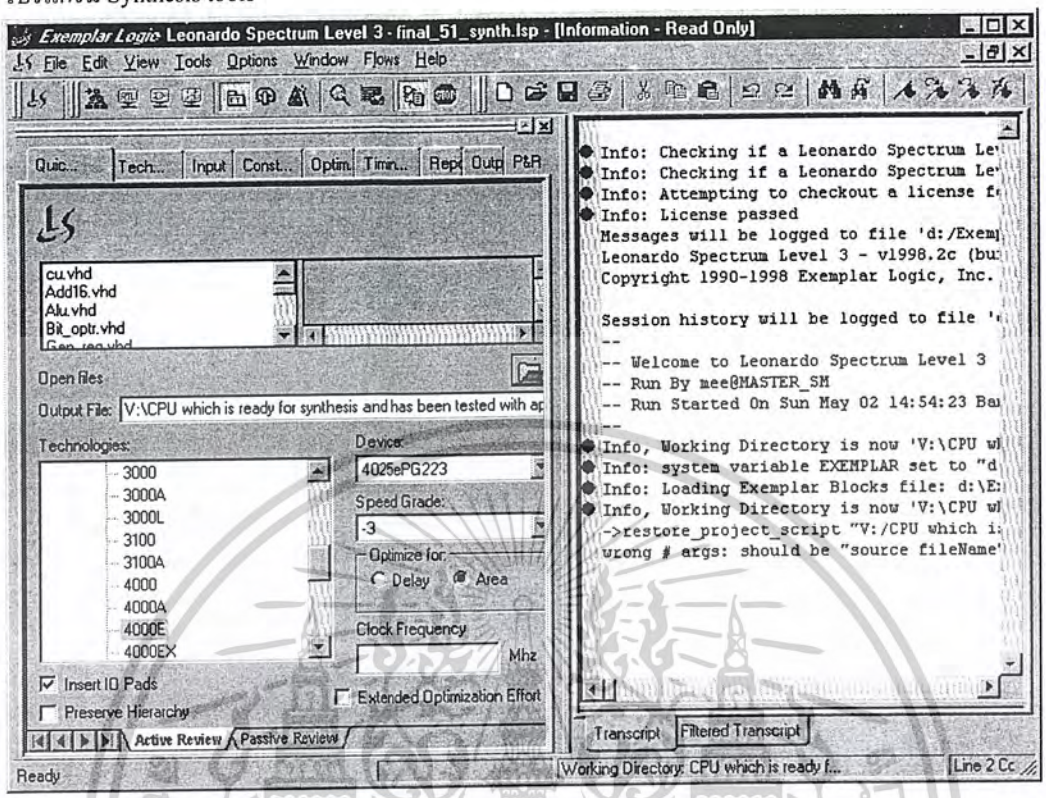


รูปที่ 4-12 แสดงการทำงานของโปรแกรม

4) การแปลงโค๊ด VHDL ที่ผ่านการตรวจสอบความถูกต้องของการทำงานแล้วให้ได้วงจรในระดับ Gate-level ซึ่งจะเป็น Netlist ของเกทต่างๆ ในวงจร ในการ Synthesis นั้นผู้ออกแบบจะต้องระบุถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีที่ใช้, ชื่อผู้ผลิตฮาร์ดแวร์ และหมายเลขรุ่นของฮาร์ดแวร์ที่ใช้ด้วยการป้อนค่าพารามิเตอร์ให้กับโปรแกรม Synthesis tools

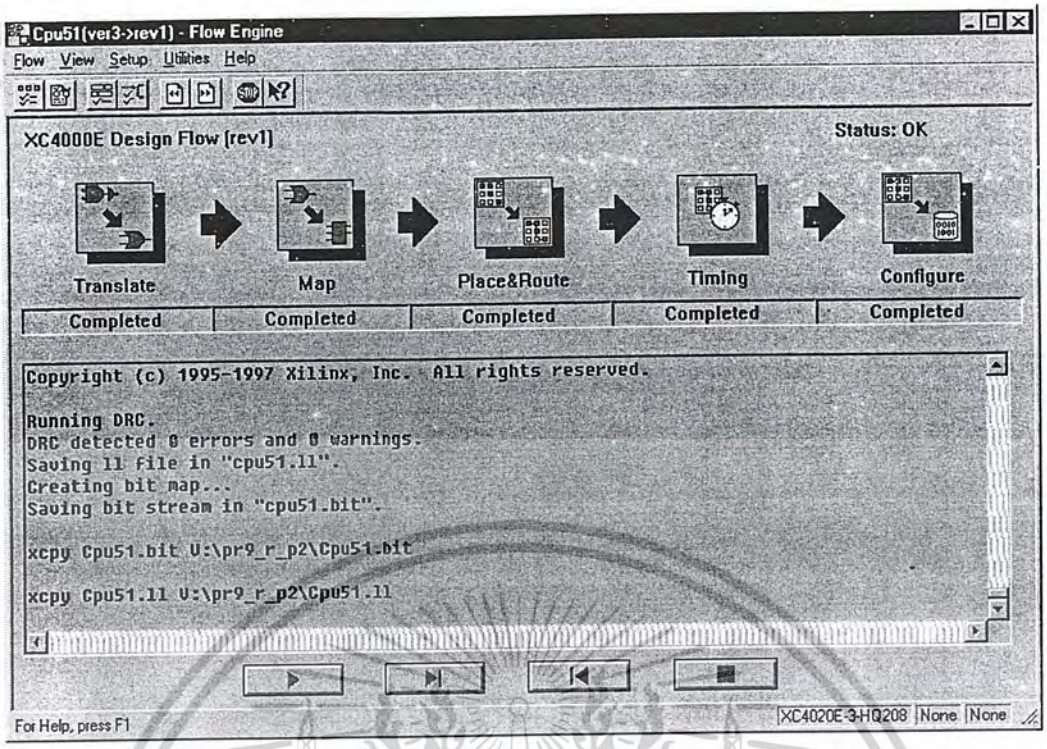


รูปที่ 4-13 การแปลงโค๊ด VHDL ให้ได้วงจรในระดับ Gate-level

5) แบ่งวงจรที่ได้จากการ synthesis เป็นส่วนย่อยๆ และแปลงลงบนบล็อกภายในของอุปกรณ์ FPGA แล้วทำการวางวงจรว่าควรจะอยู่ตำแหน่งใดภายใน FPGA จึงจะมีความเหมาะสมที่สุดแล้วทำการเชื่อมต่อ (route) สัญญาณต่างๆ ภายในอุปกรณ์ FPGA เข้าด้วยกัน ซึ่งขั้นตอนทั้งหมดดังที่กล่าวมาข้างต้นนี้จะถูกดำเนินการโดยการใช้ซอฟท์แวร์ทูลล์เข้าช่วย

6) บันทึกโปรแกรมลงบนอุปกรณ์ FPGA โดยนำไฟล์ บิตสตรีมที่ได้จากการทำการเชื่อมต่ออุปกรณ์ ต่อบอร์ดทดลองเป็นแบบ Static RAM based หากวงจรไม่ได้รับไฟเลี้ยงแล้วโปรแกรมที่บันทึกอยู่จะหายไป ผู้ออกแบบจะต้องบันทึกโปรแกรมลงบนอุปกรณ์ประเภท Serial PROM เก็บเอาไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-14 การบันทึกโปรแกรมลงบนอุปกรณ์ FPGA

4.3.6 การทำงานของโปรโตคอลอจิกอนาไลเซอร์

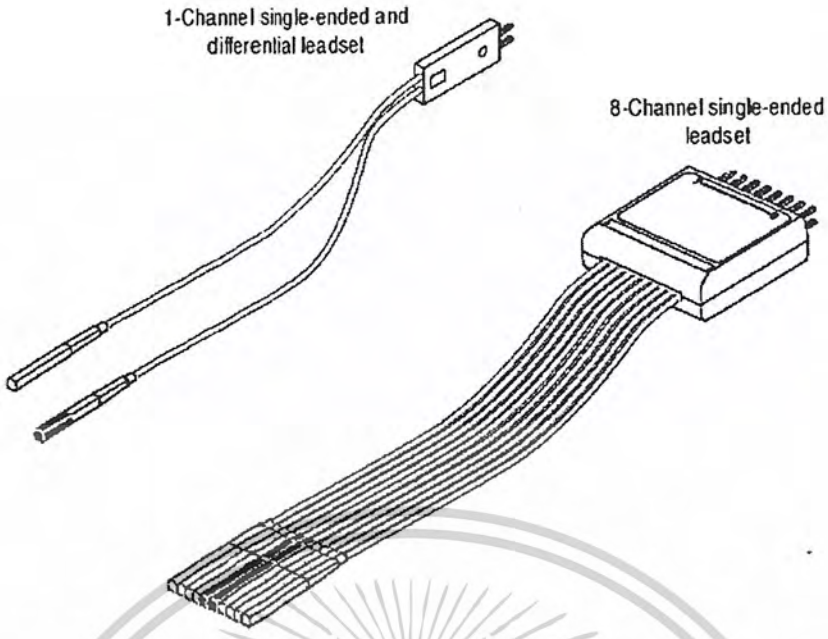
- เลือกรูปแบบการทำงาน ว่าจะเริ่มทำการจับสัญญาณ หรือดึงข้อมูลการจับสัญญาณที่เสร็จเรียบร้อยดีแล้วไปที่คอมพิวเตอร์

- การจับสัญญาณจะเริ่มโดยส่งการตั้งค่าต่างๆ จากคอมพิวเตอร์ไปที่บอร์ดทดลอง เช่น จำนวนพอร์ตของการจับสัญญาณ, ความถี่ของสัญญาณนาฬิกา เป็นต้น

- จับสัญญาณ โดยนำข้อมูลการจับสัญญาณที่ได้ไปเก็บไว้ที่ เอสแรม ของบอร์ดทดลองก่อน

- การดึงข้อมูลการจับสัญญาณที่เสร็จเรียบร้อยดีแล้ว จะเริ่มทำงานโดยดึงข้อมูลจาก เอสแรม ไปประมวลผลด้วยโปรแกรมที่ทำงานบนคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-15 สายจับสัญญาณ TTL

4.3.7 รายละเอียดของลอจิกอนาไลเซอร์

1) ควบคุมการทำงานด้วยคอมพิวเตอร์ผ่านทาง พอร์ตขนาน

2) ความถี่ที่ใช้ในการ แซมปิ้ง (Sampling) สัญญาณสามารถปรับค่าได้ โดยความถี่สูงสุดคือ100 เมกกะเฮิร์ท นอกจากนั้นยังสามารถใช้สัญญาณนาฬิกาจากภายนอกเป็นความถี่ของการแซมปิ้ง สัญญาณได้ด้วย

3) สามารถจับสัญญาณได้พร้อมกัน 32 ช่องสัญญาณ หรือเลือกจับสัญญาณเพียงบางช่องสัญญาณ เช่น 24, 16, 8 ช่องสัญญาณ โดยจำนวนช่องสัญญาณยิ่งน้อย จำนวนหน่วยความจำต่อช่องสัญญาณจะยิ่งมากขึ้น

4) สามารถ แซมปิ้ง สัญญาณได้ 8กิโลไบท์ ต่อหนึ่ง ช่องสัญญาณ

5) ลอจิกอนาไลเซอร์ สามารถ แซมปิ้ง สัญญาณที่เป็น TTL หรือ CMOS (3.3 หรือ 5 โวลต์)เท่านั้น โดยความต่างศักย์ต้องไม่มากกว่า 5 โวลต์ และไม่น้อยกว่า 0 โวลต์

6) ลอจิกอนาไลเซอร์ ทำงานแบบ ไทมิ่งอนาไลซิส (Timing Analysis) คือทำการ แซมปิ้ง ข้อมูลตามจังหวะของสัญญาณนาฬิกา ไม่ได้ทำงานแบบ สเตทอนาไลซิส (State Analysis) คือ แบบที่เก็บข้อมูลเฉพาะเมื่อเกิด ทรานซิชั่น (Transition) เท่านั้น

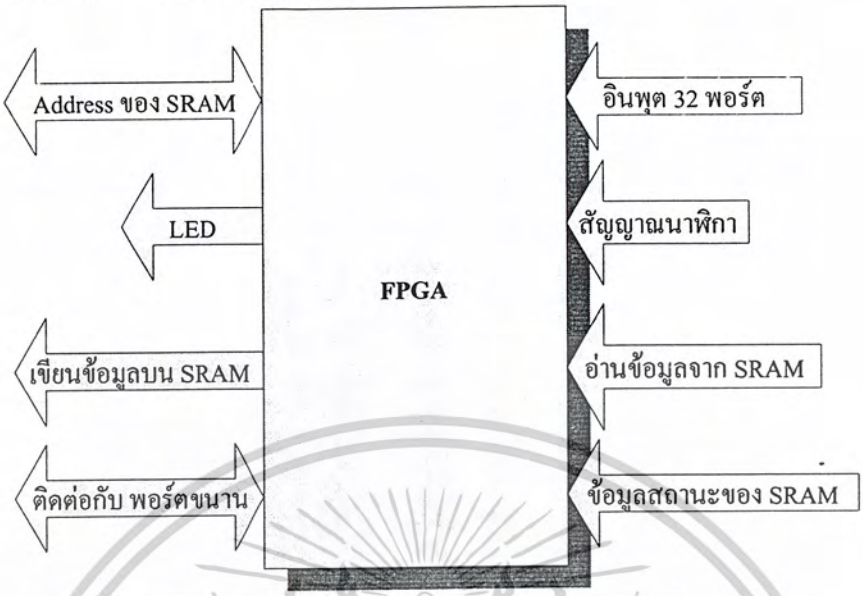
4.3.8 การออกแบบโปรแกรม

การออกแบบโปรแกรมควบคุม ลอจิกอนาไลเซอร์นี้ เราใช้หลักการของ โปรเซส (Process) เข้าช่วยในการออกแบบ โดยจะมีอยู่ทั้งหมด 3 โปรเซสหลักดังนี้

- โปรเซสช่วยจัดการ หน่วยความจำ ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรเซสช่วยจัดการ พอร์ตขนาน
- โปรเซสช่วยจัดการ สัญญาณนาฬิกา



รูปที่ 4-16 แสดงลักษณะการทำงานของอินพุต/เอาต์พุต

Z	Z	Z	Z	Z	Z	Line"1"	Line"0"

รูปที่ 4-17 แสดงรายละเอียดชุดบิตควบคุม ที่แอดเดรส "0000000000000000"

Line1"7"	Line1"6"	Line1"5"	Line1"4"	Line1"3"	Line1"2"	Line1"1"	Line1"0"

รูปที่ 4-18 แสดงรายละเอียดชุดบิตควบคุมสัญญาณนาฬิกา ที่แอดเดรส "0000000000000001"

Line"1"	Line"0"	CH
0	0	32 พอร์ต
0	1	24 พอร์ต
1	0	16 พอร์ต
1	1	8 พอร์ต

รูปที่ 4-19 แสดงรายละเอียดบิตควบคุมจำนวนพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 แผนการดำเนินงาน

การดำเนินงานในภาคเรียนที่ 1 มีดังตารางต่อไปนี้

No.	การดำเนินงาน	เริ่ม	สิ้นสุด	Jun 2003	Jul 2003	Aug 2003
				8/6	6/7	3/8
1	สำรวจและศึกษาผลงานที่ใกล้เคียงทั้งในและนอกประเทศ	2/6/2003	31/8/2003			
2	กำหนดขอบเขตและคุณสมบัติโดยละเอียดของโปรเจคให้ชัดเจนและครบถ้วน	22/6/2003	5/9/2003			
3	ศึกษาความรู้และเทคโนโลยีที่จำเป็นต้องใช้ประกอบการทำโปรเจค	14/6/2003	31/8/2003			
4	วางโครงสร้างหลักและย่อยของโปรเจคพร้อมแบ่งความรับผิดชอบ	9/6/2003	8/7/2003			
5	ดำเนินงานและพบปะเพื่อแก้ปัญหาที่อาจจะเกิดขึ้นเป็นระยะๆ และรายงานความคืบหน้า	2/6/2003	12/9/2003			
6	จัดทำรายงานและพรีเซนเทชันเพื่อนำเสนอความคืบหน้า	26/8/2003	12/9/2003			

ตารางที่ 4-1 การดำเนินงานในภาคเรียนที่ 1

การดำเนินงานในภาคเรียนที่ 2 มีดังตารางต่อไปนี้

No.	การดำเนินงาน	เริ่ม	สิ้นสุด	Nov 2003	Dec 2003	Jan 2004	Feb 2004
1	พัฒนาส่วนซอฟต์แวร์ของโครงการ	3/11/2003	15/2/2004				
2	พัฒนาส่วนฮาร์ดแวร์ของโครงการ	3/11/2003	15/2/2004				
3	ตรวจสอบและแก้ไขข้อผิดพลาดทั้งส่วนซอฟต์แวร์และฮาร์ดแวร์	1/12/2003	22/2/2004				
4	ดำเนินงานและพบปะเพื่อแก้ปัญหาที่อาจจะเกิดขึ้นเป็นระยะๆ และรายงานความคืบหน้า	3/11/2003	2/3/2004				
5	จัดทำปฏิญญานิพนธ์และพรีเซนเทชันเพื่อนำเสนอครั้งสุดท้าย	9/2/2004	7/3/2004				

ตารางที่ 4-2 การดำเนินงานในภาคเรียนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
รายละเอียดการพัฒนาและผลการทดสอบ

5.1 การพัฒนาและการทดสอบในส่วนซอฟต์แวร์

อธิบายการพัฒนาโดยแบ่งเป็นส่วน ดังนี้

1) ส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

2) ส่วนบันทึกข้อมูลลงไฟล์ชนิด Value Change Dump (VCD)

3) ส่วนแสดงผลและวิเคราะห์ข้อมูล

4) ส่วนวิเคราะห์โปรโตคอล

โดยแต่ละส่วนมีดังรูปต่อไปนี้

รูปที่ 5-1 แผนผังส่วนต่างๆ ของโปรแกรมที่พัฒนา

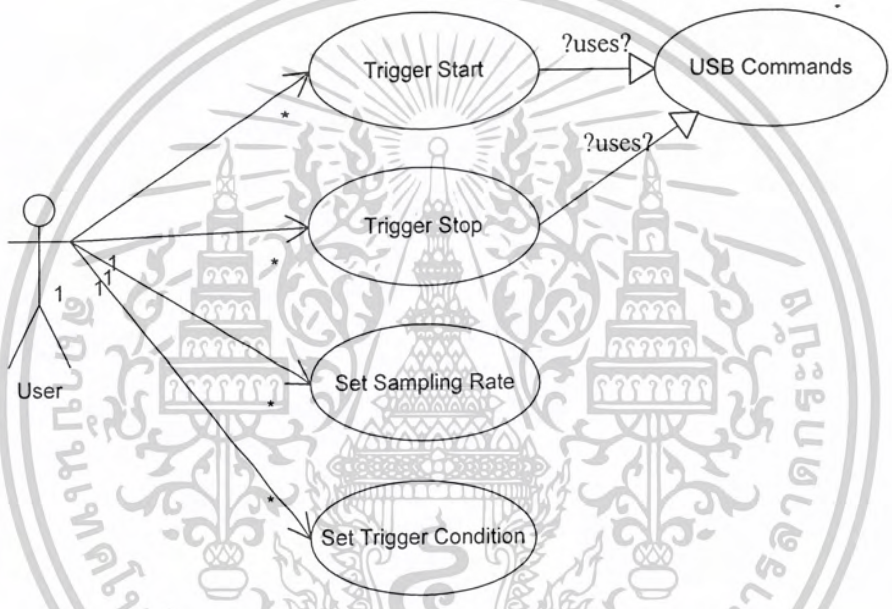
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 การพัฒนาในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

1) ความสามารถของโปรแกรมในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

- สามารถติดต่อกับฮาร์ดแวร์ผ่านทางพอร์ต USB
- สามารถเช็คได้ว่าฮาร์ดแวร์เชื่อมต่อกับพอร์ต USB
- สามารถตั้งค่าความถี่ในการสุ่มสัญญาณ (Sampling Rate)
- สามารถตั้งค่าเงื่อนไขในการเริ่มรับสัญญาณข้อมูลได้
- สามารถสั่งให้เริ่มหรือหยุดรับสัญญาณข้อมูลได้
- สามารถรับข้อมูลที่ฮาร์ดแวร์ทำการสุ่มสัญญาณ (Sampling)

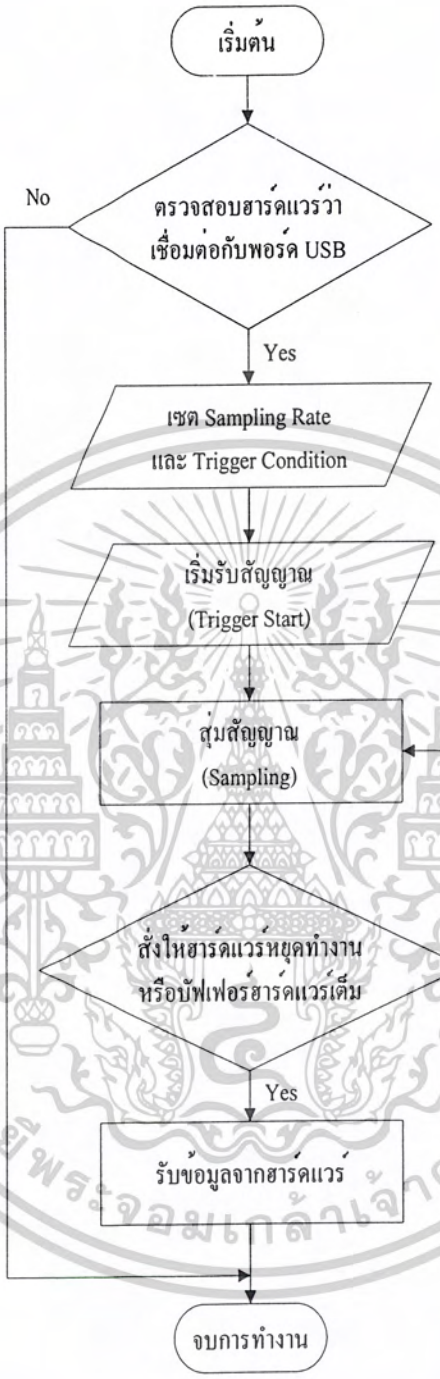
2) Use Case Diagram ของส่วนติดต่อกับอุปกรณ์วัดสัญญาณ



รูปที่ 5-2 Use Case Diagram ของส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

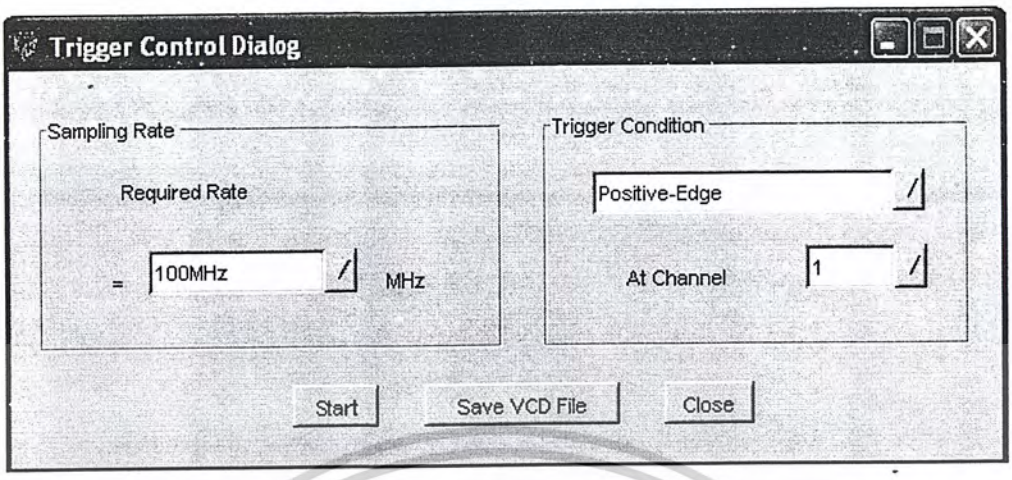
3) Flow Chart ของส่วนติดต่อกับอุปกรณ์วัดสัญญาณ



รูปที่ 5-3 Flow Chart ของส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

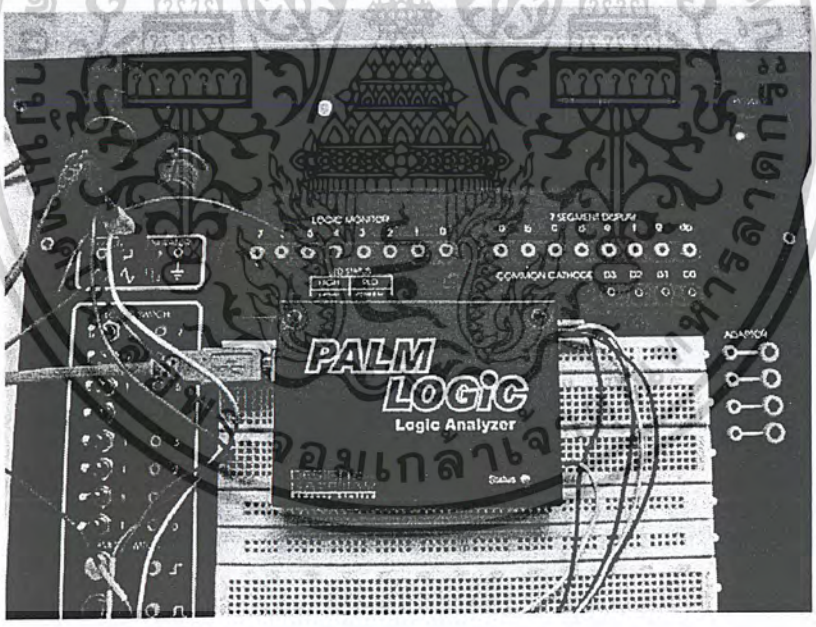
4) หน้าจอโปรแกรมส่วนติดต่อกับอุปกรณ์วัดสัญญาณ



รูปที่ 5-4 หน้าจอโปรแกรมส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

5) การทดสอบโปรแกรมในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

- ทำการทดสอบโดยการวัดค่าสัญญาณความถี่จากเครื่องกำเนิดความถี่ ดังรูป



รูปที่ 5-5 การทดสอบโดยการวัดค่าสัญญาณความถี่จากเครื่องกำเนิดความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมสั่งให้ Palm Logic ทำการสุ่มจับสัญญาณ (Sampling) จากเครื่องกำเนิดความถี่ และข้อมูลในบัฟเฟอร์รับข้อมูลมา ได้ดังนี้

```
00070000h: 00 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00 ; ............
00070010h: 00 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00 ; ............
00070020h: 00 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00 ; ............
00070030h: 00 00 00 00 00 00 00 00 01 00 00 00 01 00 00 00 ; ............
00070040h: 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ............
00070050h: 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ............
00070060h: 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ............
00070070h: 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ............
00070080h: 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ............
00070090h: 01 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ; ............
000700a0h: 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ; ............
000700b0h: 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 ; ............
000700c0h: 01 00 00 00 01 00 00 00 00 00 00 00 00 00 00 00 ; ............
```

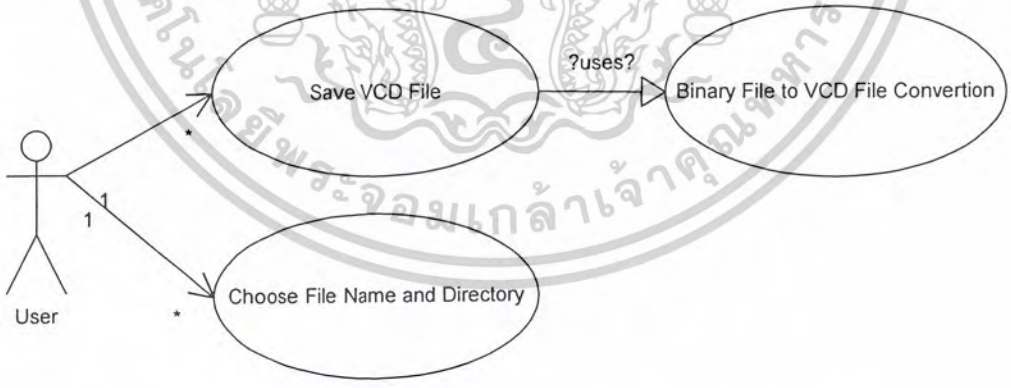
รูปที่ 5-6 ข้อมูลในบัฟเฟอร์ที่ได้จากการสุ่มจับสัญญาณ

5.1.2 การพัฒนาในส่วนบันทึกข้อมูลลงไฟล์ชนิด Value Change Dump (VCD)

1) ความสามารถของโปรแกรมในส่วนบันทึกข้อมูลลงไฟล์

- สามารถบันทึกข้อมูลในบัฟเฟอร์ที่รับมาจากฮาร์ดแวร์ลงไฟล์ชนิด Value Change Dump (VCD) รายละเอียดมาตรฐานของไฟล์ชนิดนี้อยู่ในผนวก ข
- สามารถเลือกชื่อไฟล์ที่ทำการบันทึกได้

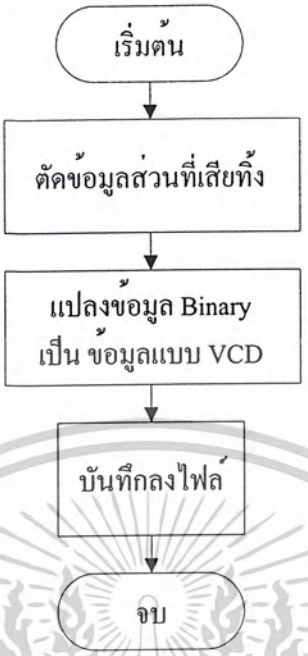
2) Use Case Diagram ของส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD



รูปที่ 5-7 Use Case Diagram ของส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD

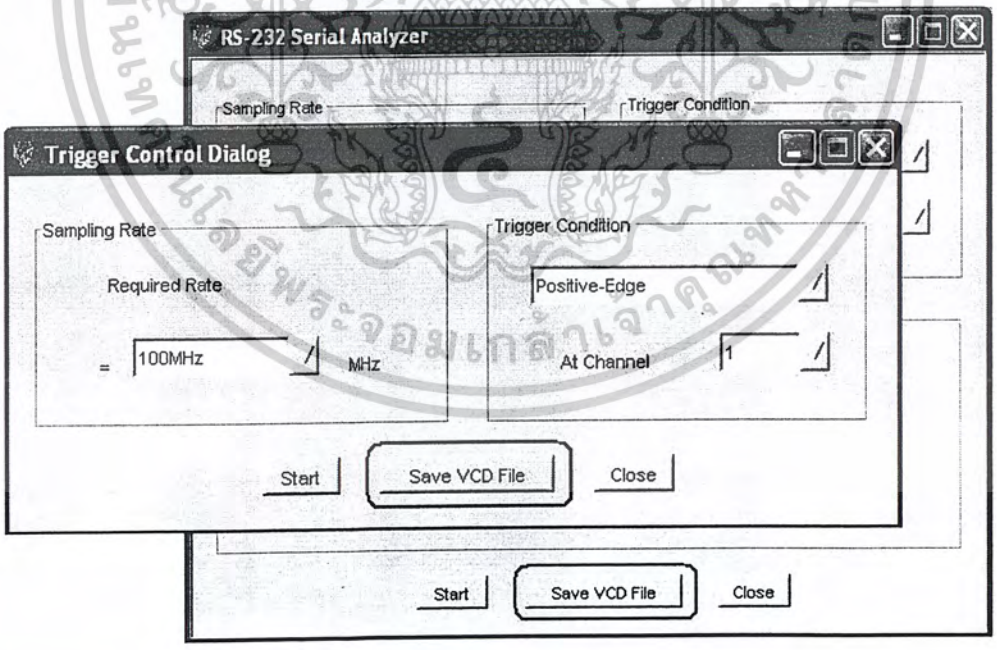
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) Flow Chart ของส่วนส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD



รูปที่ 5-8 Flow Chart ของส่วนส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD

4) หน้าจอโปรแกรมส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD



รูปที่ 5-9 หน้าจอโปรแกรมส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การทดสอบโปรแกรมในส่วนบันทึกข้อมูลลงไฟล์ชนิด VCD

ทำการบันทึกข้อมูลในบัฟเฟอร์แล้วได้ไฟล์ชนิด VCD ดังนี้

```
$date
   13 January 2004        05:56:40
$end
$version
   Protocologic Analyzer 1.0
$end
$timescale
    1ns
$end
$scope module Protocologic $end
        $var reg     1 A   Channel1   $end
        $var reg     1 B   Channel2   $end
$upscope $end
$enddefinitions $end
#0
$dumpvars
0A
0B
$end
1A
#30.00
0A
#50.00
1A
#70.00
0A
```

รูปที่ 5-10 ข้อมูลในไฟล์ชนิด VCD

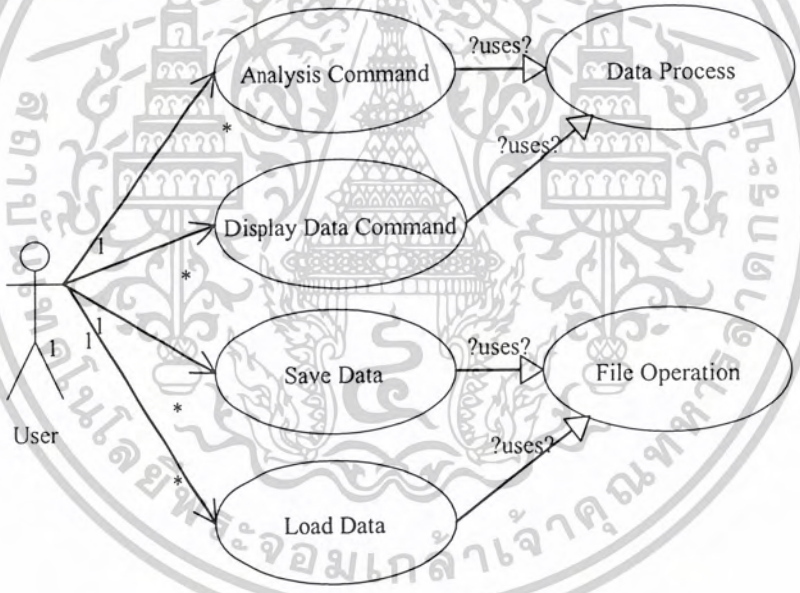
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 การพัฒนาในส่วนแสดงผลและวิเคราะห์ข้อมูล

1) ความสามารถของโปรแกรมในส่วนแสดงผลและวิเคราะห์ข้อมูล

- จัดกลุ่มของสัญญาณที่แสดงผล (Group define for channels)
- แสดงผลแบบตัวเลข (View as number)
- ย้ายตำแหน่งหน้าจอแสดงผลไปที่ตำแหน่งต่างๆ (Move display window)
- แสดงผลแบบเวฟฟอร์ม (View as waveform)
- แสดงค่าข้อมูล ณ ตำแหน่งของเคอร์เซอร์ (View cursor values)
- ย่อหรือขยายเวฟฟอร์ม (Waveform zoom)
- ค้นหาแพ็ทเทิร์ทที่ต้องการ (Search pattern)
- เพิ่มหรือลดสัญญาณที่หน้าจอแสดงผล (Insert or Delete channel)
- เพิ่มเคอร์เซอร์เข้าไปที่หน้าจอแสดงผล (Add cursor)
- เรียงลำดับชื่อของสัญญาณที่หน้าจอแสดงผล (Sorting view)

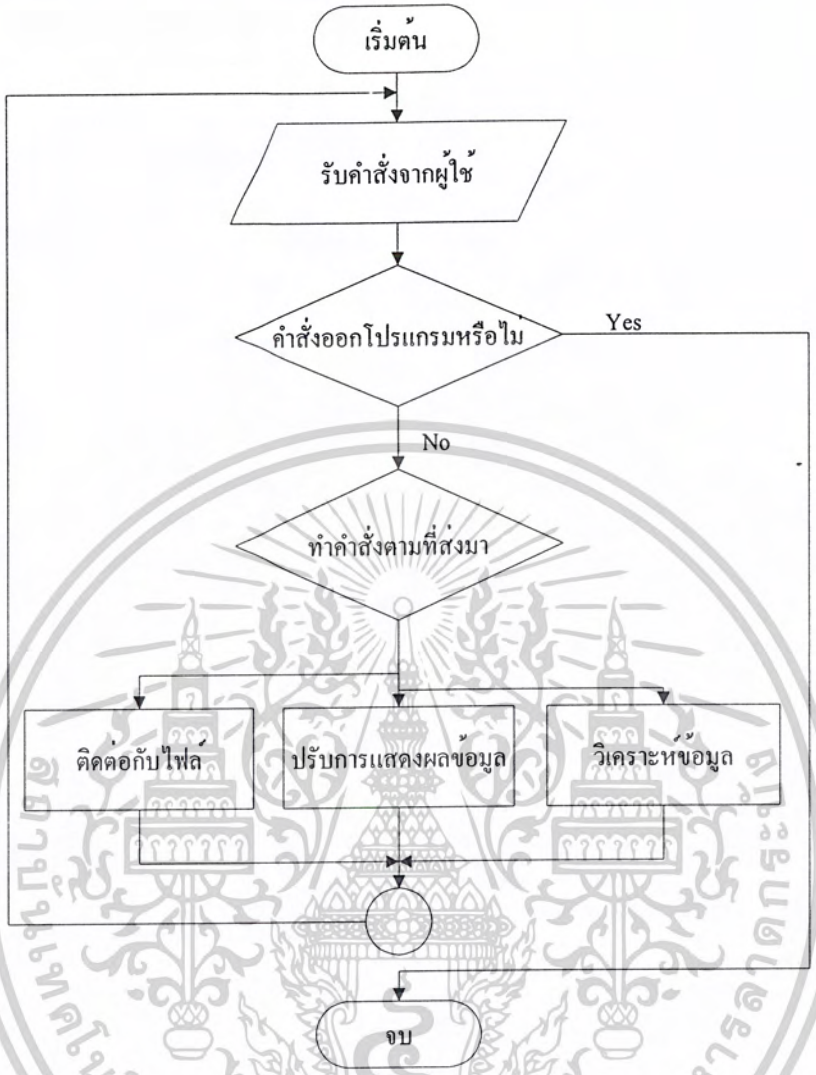
2) Use Case Diagram ของส่วนแสดงผลและวิเคราะห์ข้อมูล



รูปที่ 5-11 Use Case Diagram ของส่วนแสดงผลและวิเคราะห์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

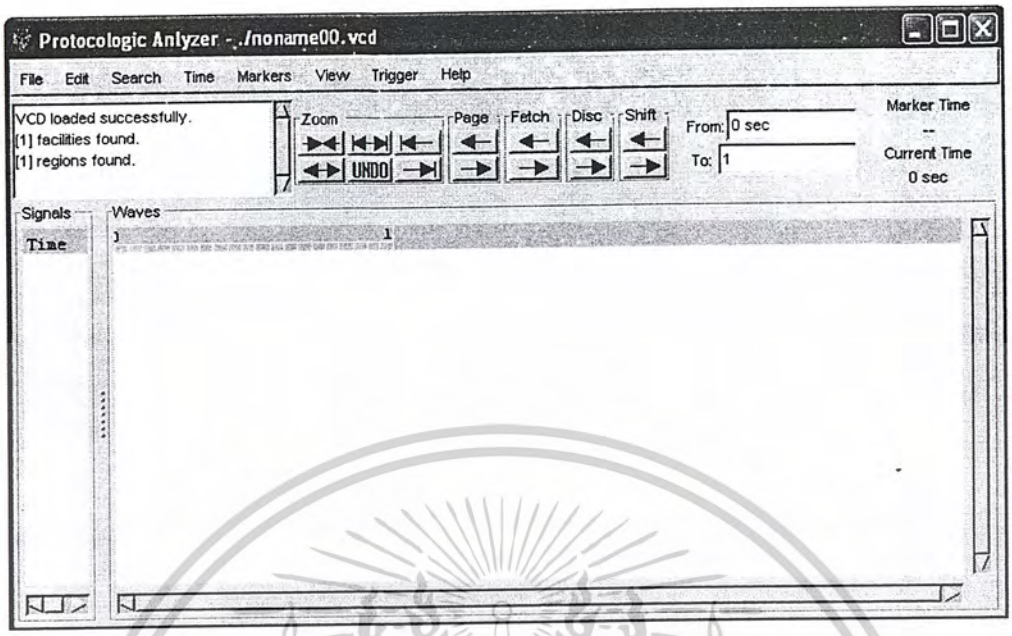
3) Flow Chart ของส่วนแสดงผลและวิเคราะห์ข้อมูล



รูปที่ 5-12 Flow Chart ของส่วนแสดงผลและวิเคราะห์ข้อมูล

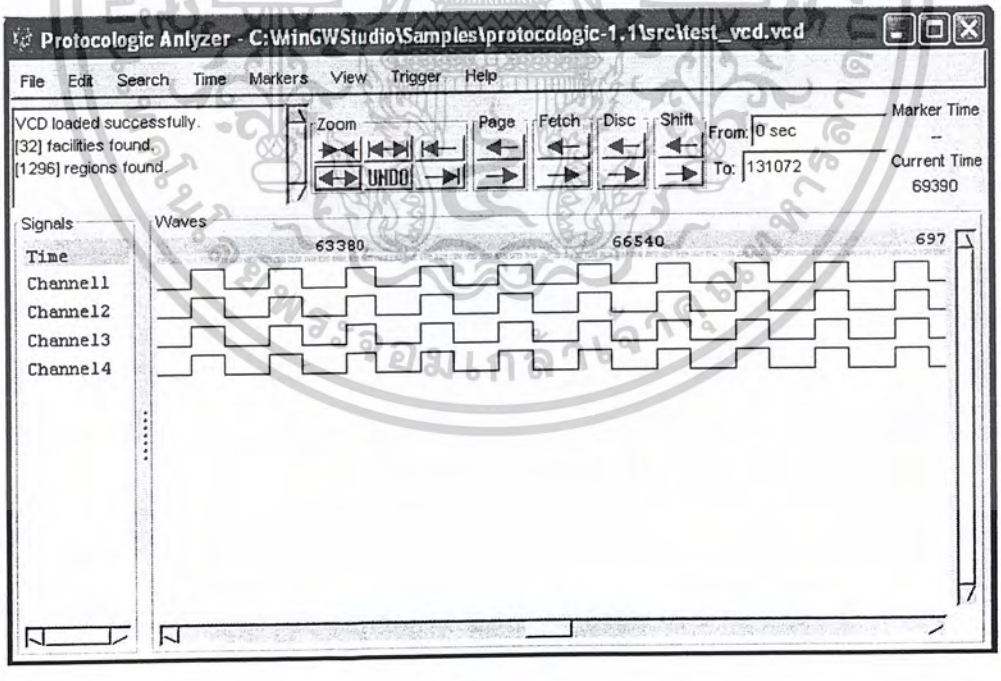
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) หน้าจอโปรแกรมส่วนแสดงผลและวิเคราะห์ข้อมูล



รูปที่ 5-13 หน้าจอโปรแกรมส่วนแสดงผลและวิเคราะห์ข้อมูล

5) การทดสอบโปรแกรมในส่วนแสดงผลและวิเคราะห์ข้อมูล



รูปที่ 5-14 การทดสอบโปรแกรมในส่วนแสดงผลและวิเคราะห์ข้อมูล

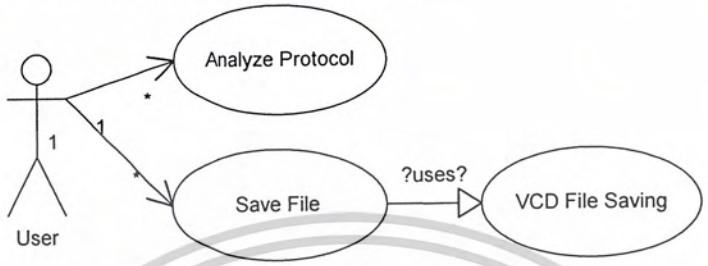
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.4 การพัฒนาในส่วนวิเคราะห์โปรโตคอล

1) ความสามารถของโปรแกรมในส่วนวิเคราะห์โปรโตคอล

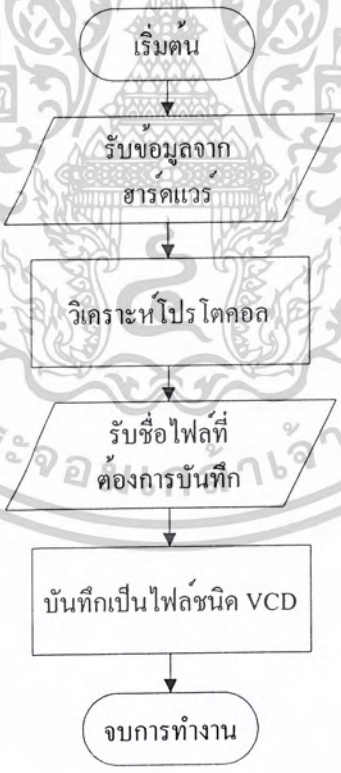
- สามารถวิเคราะห์โปรโตคอล RS-232 Serial
- สามารถบันทึกผลการวิเคราะห์ได้

2) Use Case Diagram ของส่วนวิเคราะห์โปรโตคอล



รูปที่ 5-15 Use Case Diagram ของส่วนวิเคราะห์โปรโตคอล

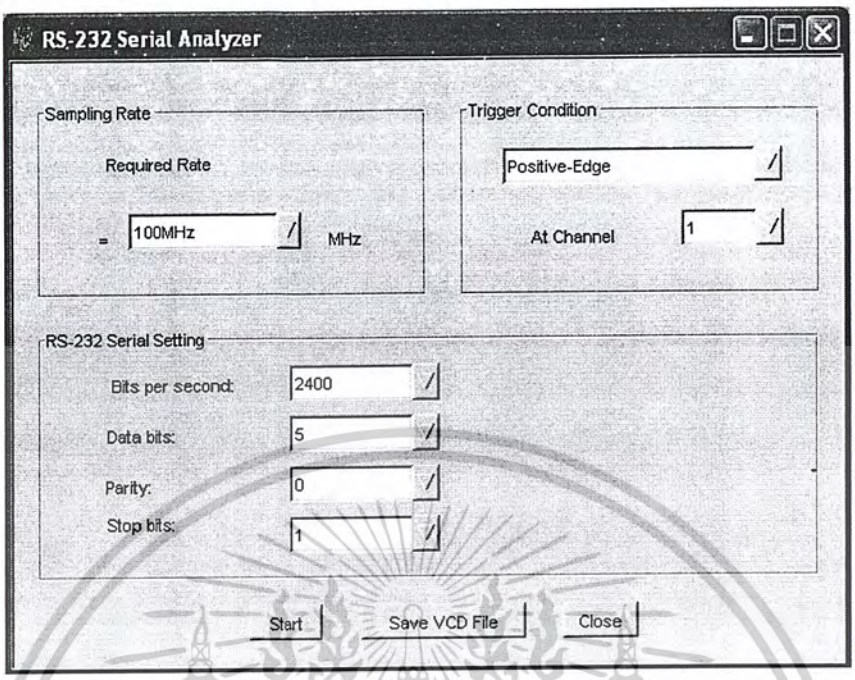
3) Flow Chart ของส่วนวิเคราะห์โปรโตคอล



รูปที่ 5-16 Flow Chart ของส่วนวิเคราะห์โปรโตคอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

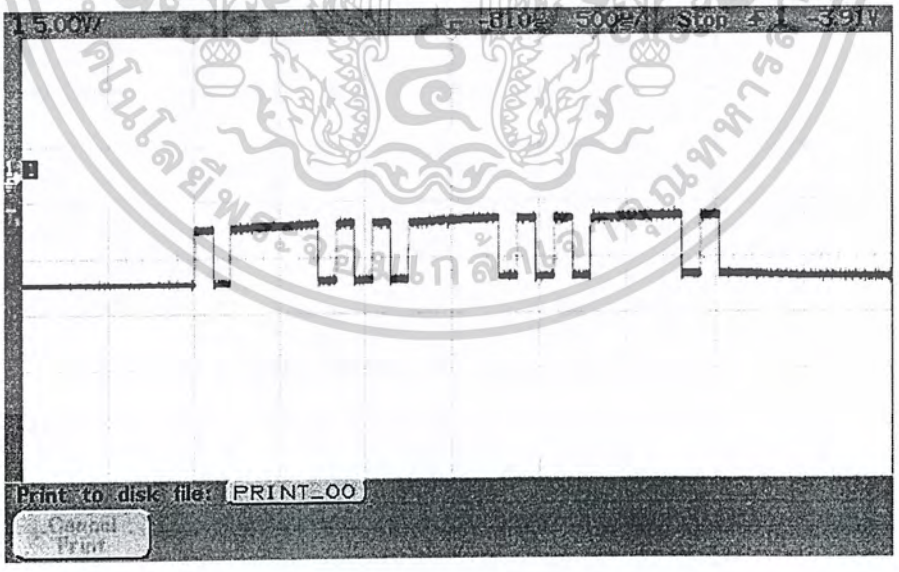
4) หน้าจอโปรแกรมส่วนวิเคราะห์โปรโตคอล



รูปที่ 5-17 หน้าจอโปรแกรมส่วนวิเคราะห์โปรโตคอล

5) การทดสอบโปรแกรมในส่วนวิเคราะห์โปรโตคอล

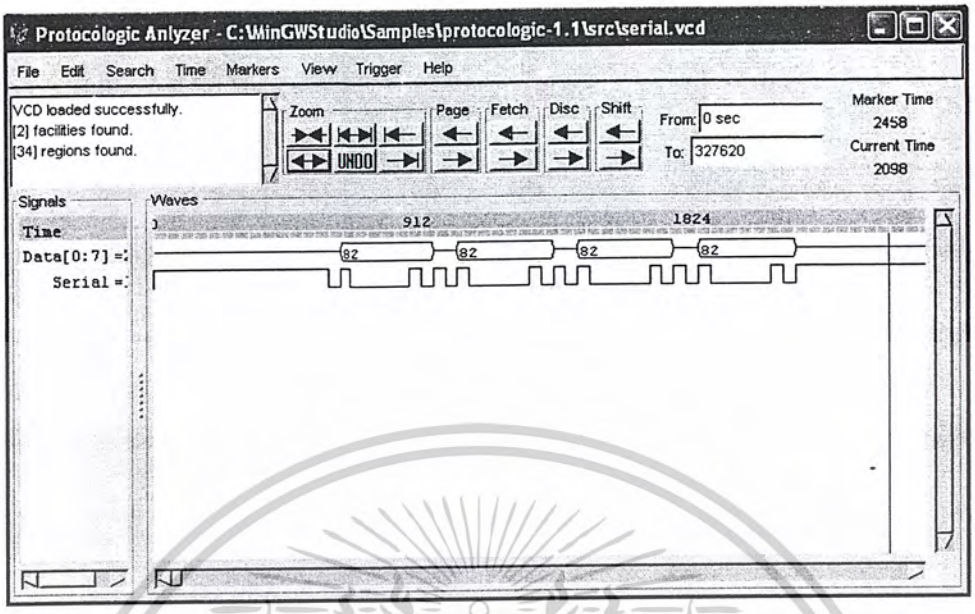
- วัดสัญญาณในสารส่ง RS-323 Serial ด้วยออสซิลโลสโคป และบันทึกภาพได้ดังนี้



รูปที่ 5-18 สัญญาณในสายส่งข้อมูลแบบ RS-232 Serial ที่วัดด้วยออสซิลโลสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข้อมูลที่แสดงหลังจากทำการวิเคราะห์โปรโตคอลแล้ว

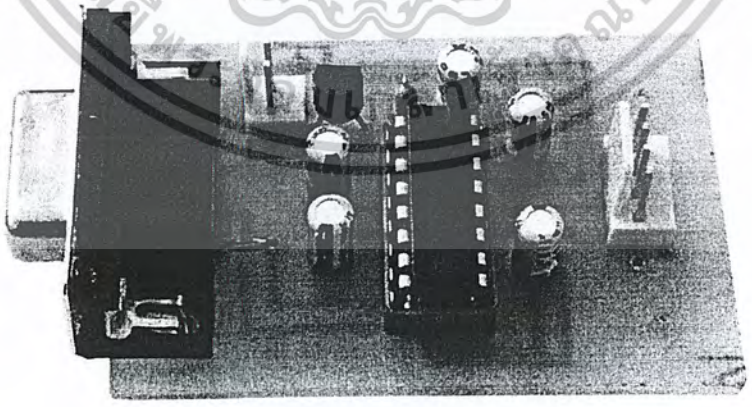


รูปที่ 5-19 ข้อมูลที่แสดงหลังจากทำการวิเคราะห์โปรโตคอล

5.2 การพัฒนาและการทดสอบในส่วนฮาร์ดแวร์

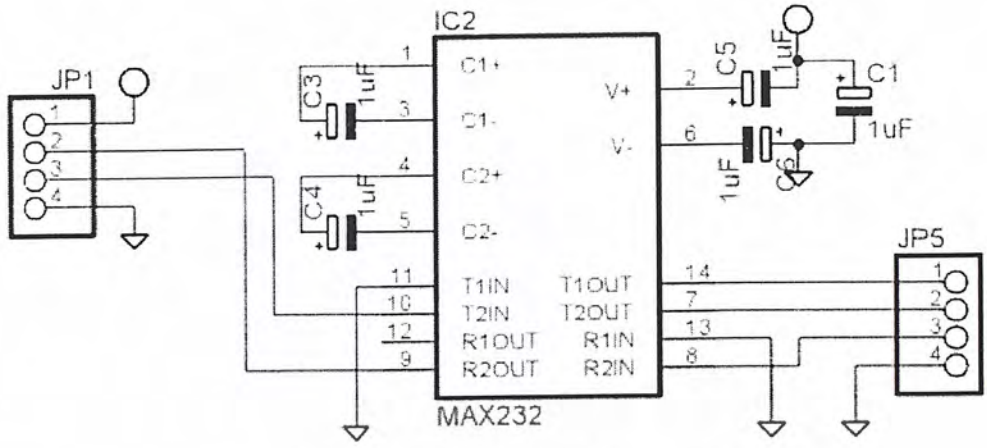
5.2.1 บอร์ดแปลงพอร์ตอนุกรมเป็น TTL

บอร์ดแปลงพอร์ตอนุกรมเป็น TTL นี้เราสร้างขึ้นเพื่อ แปลงสัญญาณจากพอร์ตอนุกรมซึ่ง มีระดับแรงดัน -8 ถึง +8 โวลต์ เป็น TTL ที่มีระดับแรงดัน 5 โวลต์ โดยใช้ชิพ MAX232 ในการแปลงระดับแรงดัน



รูปที่ 5-20 บอร์ดแปลงพอร์ตอนุกรมเป็น TTL (รูปแทน)

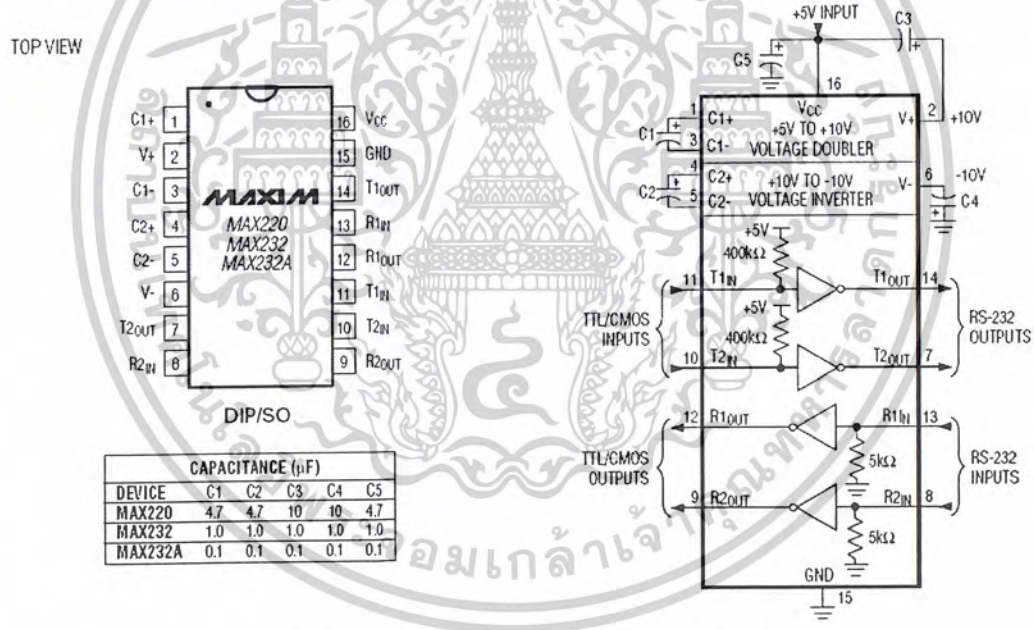
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-21 โครงสร้างของบอร์ดแปลงพอร์ตอนุกรมเป็น TTL

5.2.2 ชิพ MAX232

ชิพ MAX232 ใช้ไฟเลี้ยงขนาด 5 โวลต์สามารถแปลงพอร์ตอนุกรมได้ 2 พอร์ต มีความเร็วในการส่งข้อมูลที่ 120 กิโลบิต ต่อวินาที มีขาทั้งหมด 16 พิน



CAPACITANCE (μF)					
DEVICE	C1	C2	C3	C4	C5
MAX220	4.7	4.7	10	10	4.7
MAX232	1.0	1.0	1.0	1.0	1.0
MAX232A	0.1	0.1	0.1	0.1	0.1

รูปที่ 5-22 สถาปัตยกรรมภายในของชิพ MAX232

5.2.3 พอร์ตอนุกรม

พอร์ตสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ มีโครงสร้างเป็นแบบฟูลดูเพล็กซ์ ซึ่งรับ และส่งข้อมูลในเวลาเดียวกันได้ มีรีจิสเตอร์ SBUF (Serial Data Buffer) เป็นบัฟเฟอร์สำหรับการรับส่งข้อมูลอนุกรม โดยเริ่มต้นเมื่อมีการเขียนข้อมูลเก็บไว้ในรีจิสเตอร์ SBUF หลังจากนั้นข้อมูลจะถูกจัดการโดยวิธี

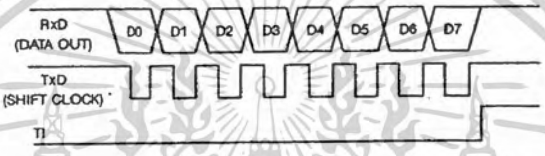
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางฮาร์ดแวร์ในการเลื่อนบิตเพื่อส่งสัญญาณออกไปภายนอก หลังจากมีการส่งข้อมูลออกไปจนครบแล้ว จึงจะเซตบิตโดยกำหนดค่าของแฟลก TI ในรีจิสเตอร์ SCON ให้เป็นสถานะ "1" เพื่อแจ้งว่ารีจิสเตอร์ SBUF ว่างแล้ว และพร้อมที่จะส่งข้อมูลไบต์ต่อไปได้

การรับข้อมูลจากพอร์ตอนุกรมจะต้อง เริ่มต้นโดยกำหนดค่าของบิต REN ที่อยู่ในรีจิสเตอร์ SCON ให้มีค่าเป็นสถานะ "1" หลังจากนั้นเมื่อมีการรับข้อมูลเข้ามาจากภายนอกก็จะเลื่อนข้อมูลไปโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนบิตเข้ามาเรียบร้อยแล้ว ข้อมูลจะถูกย้ายมาเก็บไว้ที่นรจิสเตอร์ SBUF และจะเซตบิต RI ให้เป็นสถานะ "1" ซึ่งส่งผลให้เกิดการอินเตอร์รัปต์โปรแกรมขึ้น

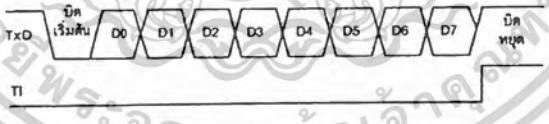
5.2.4 โหมดการติดต่อทางพอร์ตอนุกรม

การสื่อสารอนุกรมของไมโครคอนโทรลเลอร์ จะแบ่งออกได้เป็น 4 โหมดด้วยกัน และในแต่ละโหมดจะสามารถสรุปหน้าที่ได้ดังนี้



รูปที่ 5-23 การส่งข้อมูลแบบอนุกรม

โหมด 0 จะเป็นการรับส่งข้อมูลขนาด 8 บิตแบบอนุกรมการส่งข้อมูลจะเลื่อนออกไปทีละบิต โดยจะใช้งานของขา RxD เพียงขาเดียวและไม่มีการส่งบิตเริ่มต้น (Start bit) ส่วนขา TxD จะใช้เป็นขาของสัญญาณนาฬิกาในการให้จังหวะการเลื่อนข้อมูลกับวงจรภายนอก (Shift clock) อัตราการรับส่งข้อมูล (Baud rate) จะเป็น 1/12 เท่าของสัญญาณนาฬิกา

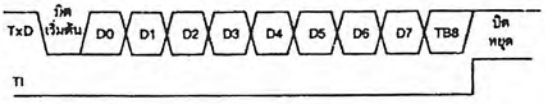


รูปที่ 5-24 การทำงานในโหมด 0

โหมด 1 จะเป็นการรับ และส่งข้อมูลขนาด 10 บิตแบบ UART (Universal Asynchronous Receiver Transmitters) สามารถใช้ติดต่อสื่อสารอนุกรมกับมาตรฐานของ RS-232C ของไมโครคอมพิวเตอร์ได้ ซึ่งข้อมูลอนุกรม 10 บิตจะเข้ามาทางขา RxD และส่งข้อมูลออกแบบอนุกรมทางขา TxD โดยจะประกอบด้วย 1 บิตแรกเป็นบิตเริ่มต้น (Start bit ค่า 0), 8 บิตต่อมาจะเป็นบิตของข้อมูล (การรับ/ส่งจะเริ่มจากบิตต่ำก่อน) และบิตหยุดอีก 1 บิต (Stop bit ค่า 1) ส่วนทางด้านรับข้อมูลจะนำค่าบิตหยุดที่รับเข้ามาได้นำไปเก็บไว้ในบิต RB8 ที่อยู่ในรีจิสเตอร์ SCON และความเร็วของการส่งข้อมูลในโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะขึ้นอยู่กับบิต SMOD ที่อยู่ในรีจิสเตอร์ PCON และอัตราโอเวอร์โฟลว์ของไทเมอร์ 1 ซึ่งอัตราการรับส่งข้อมูลในโหมดนี้สามารถกำหนดได้ตามต้องการ



รูปที่ 5-25 การทำงานในโหมด 1

โหมด 2 จะเป็นการรับ และส่งข้อมูลขนาด 11 บิตแบบ UART ข้อมูลแบบอนุกรมจะถูกรับเข้ามาทางขา RxD และส่งข้อมูลออกไปทางขา TxD ซึ่งข้อมูล 11 บิต ประกอบด้วยบิตแรกจะเป็นบิตเริ่มต้น, 9 บิตต่อมาจะเป็นบิตของข้อมูล และบิตสุดท้ายจะเป็นบิตหยุด 1 บิต สำหรับข้อมูลในบิตที่ 9 จะกำหนดไว้ใน TB8 ที่อยู่ในรีจิสเตอร์ SCON ซึ่งสามารถกำหนดเป็น 1 หรือ 0 ก็ได้ นิยมนำมาใช้ในการส่งบิตเพื่อตรวจสอบการส่งข้อมูล (Parity bit)

โหมด 3 เป็นการรับส่งข้อมูลแบบ 11 บิตแบบ UART เหมือนกับโหมด 2 แต่ในโหมด 3 สามารถกำหนดอัตราความเร็วในการรับ และส่งข้อมูลได้ตามความต้องการ

หมายเหตุ UART เป็นหารส่งข้อมูลแบบอนุกรม โดยขึ้นอยู่กับความพร้อมของทางด้านส่ง และด้านรับ เป็นการส่งข้อมูลโดยเพิ่มเติมข้อมูลบางอย่างเข้าไป (Start bit, Stop bit, Parity bit) เพื่อให้การรับและการส่งข้อมูลสามารถจะทำงานให้มีความถูกต้องของข้อมูลมากยิ่งขึ้น

การสื่อสารเพื่อเชื่อมต่อไมโครคอนโทรลเลอร์ในการรับ และส่งข้อมูลทางอนุกรมมีอยู่ด้วยกัน 2 ระบบคือ

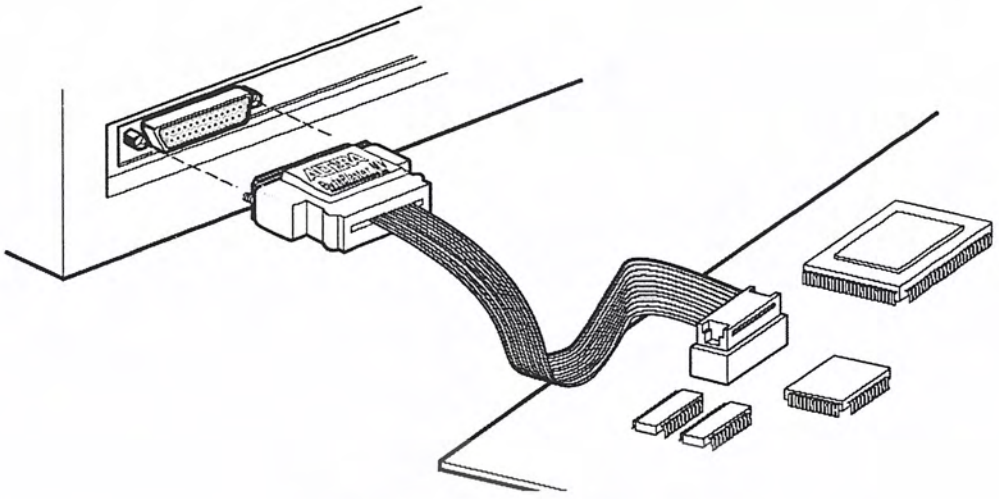
1) Single Processor System คือระบบการสื่อสารโดยใช้ไมโครคอนโทรลเลอร์ 2 ตัวเชื่อมต่อหากัน

2) Multi Processor System คือระบบการสื่อสารแบบมัลติโปรเซสเซอร์โดยใช้ไมโครคอนโทรลเลอร์ 1 ตัว เป็นตัวแม่ (Master) และสามารถที่จะเชื่อมต่อกับไมโครคอนโทรลเลอร์ ที่เป็นตัวลูก (Slave) ได้อีกเป็นจำนวนหลายๆ ตัว

5.2.5 พอร์ตขนาน

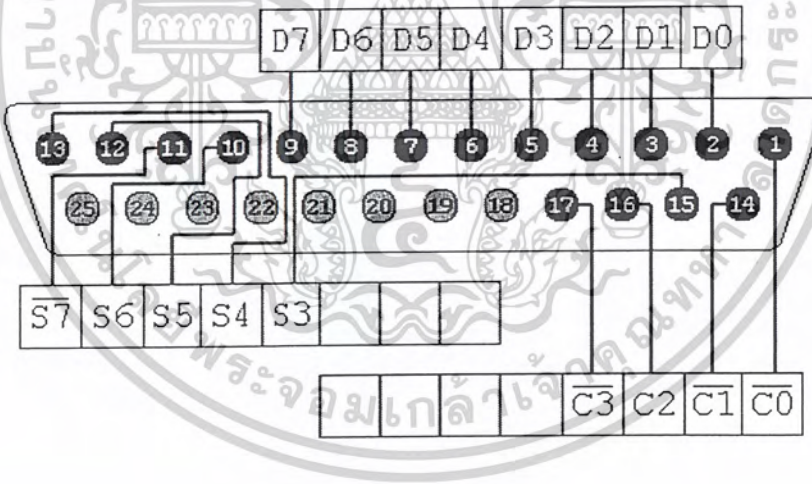
มีการรับส่งข้อมูลเป็นแบบสองทาง (Bidirectional) โดยข้อมูลที่ทำการรับส่งเป็นข้อมูล แบบ TTL แรงดัน 5 โวลต์ โดยที่ใช้งานอยู่ทั่วๆ ไปเป็นแบบ 25 พิน D-type

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5-26 แสดงการติดต่อกับพอร์ตขนาน

- มีพอร์ตเอาต์พุตข้อมูล 8 พอร์ต
- มีพอร์ตอินพุตสถานะของพอร์ต 5 พอร์ต
- มีพอร์ตเอาต์พุตสำหรับควบคุมพอร์ต 4 พอร์ต
- มีกราวนด์ 8 พอร์ต



รูปที่ 5-27 แสดงรายละเอียดของ พอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
บทวิจารณ์และสรุป

6.1 สรุปการดำเนินงาน

สรุปการดำเนินงานโดยอธิบายเป็นส่วนๆ ดังนี้

1) ส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

2) ส่วนบันทึกข้อมูลลงไฟล์ชนิด Value Change Dump

3) ส่วนแสดงผลและวิเคราะห์ข้อมูล

4) ส่วนวิเคราะห์โปรโตคอล

6.1.1 สรุปการดำเนินงานส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

1) เป้าหมายแรกที่พัฒนา

ต้องการพัฒนาโปรแกรมที่สามารถติดต่อกับอุปกรณ์วัดสัญญาณ โดยผ่านทางพอร์ต USB บนระบบปฏิบัติการลีนุกซ์ (Linux)

2) ปัญหาที่พบ

มีปัญหาเกี่ยวกับไดร์ฟเวอร์ (Driver) ของอุปกรณ์รับสัญญาณ ซึ่งไม่สามารถทำงานได้บนระบบปฏิบัติการลีนุกซ์

3) การแก้ปัญหา

หลังจากพยายามแก้ปัญหาเรื่องไดร์ฟเวอร์ของอุปกรณ์รับสัญญาณให้ทำงานได้บนระบบปฏิบัติการลีนุกซ์ แต่ไม่ประสบผลสำเร็จเนื่องจากว่า ผู้พัฒนาไม่ได้เป็นสร้างไดร์ฟเวอร์ขึ้นมาเอง ดังนั้นจึงทดลองเปลี่ยนไปพัฒนาบนระบบปฏิบัติการวินโดวส์ และสามารถทำงานได้

4) สรุปผลลัพธ์ที่ได้

ส่วนของโปรแกรมที่ติดต่อกับอุปกรณ์วัดสัญญาณทำงานอยู่บนระบบปฏิบัติการวินโดวส์

6.1.2 สรุปการดำเนินงานส่วนบันทึกข้อมูลลงไฟล์ชนิด Value Change Dump

1) เป้าหมายที่พัฒนา

ต้องการสร้างไฟล์ที่มีการบันทึกข้อมูลที่เป็นมาตรฐาน โดยพิจารณาจากโปรแกรมต้นแบบสามารถรองรับได้ และมีเอกสารอธิบายถึงมาตรฐานของไฟล์ จึงใช้ไฟล์ชนิด Value Change Dump ส่วนมาตรฐานของไฟล์ชนิดนี้อธิบายไว้ในผนวก ข.

2) สรุปผลลัพธ์ที่ได้

ส่วนของโปแกรมที่บันทึกข้อมูลลงไฟล์ชนิด Value Change Dump ทำงานอยู่บนระบบปฏิบัติงานวินโดวส์ และไม่ได้ใช้ API ช่วยในการพัฒนา จึงไม่ต้องการไฟล์ไลบารี่ใดๆ ในการใช้งานโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.3 สรุปการดำเนินงานส่วนแสดงผลและวิเคราะห์ข้อมูล

1) เป้าหมายแรกที่พัฒนา

ต้องการพัฒนาให้โปรแกรมในส่วนแสดงผลและวิเคราะห์ข้อมูลสามารถทำงานได้บนระบบปฏิบัติการลีนุกซ์

2) ปัญหาที่พบ

เนื่องจากส่วนของโปรแกรมที่ติดต่อกับอุปกรณ์วัดสัญญาณไม่สามารถทำงานได้บนระบบปฏิบัติการลีนุกซ์ ดังนั้นจึงทดลองแปลงโปรแกรมในส่วนนี้ให้มาทำงานบนระบบปฏิบัติการวินโดวส์

3) การแก้ปัญหาและผลสรุป

ส่วนของโปรแกรมที่แสดงผลและวิเคราะห์ข้อมูลทำงานอยู่บนระบบปฏิบัติการวินโดวส์ และตัดบางส่วนของโปรแกรมที่ไม่สามารถแปลงได้ เนื่องจากมีการเรียกใช้ฟังชันก์บางฟังชันก์ที่ไม่มีไลบาร์รีรองรับบนระบบปฏิบัติการวินโดวส์

6.1.4 สรุปการดำเนินงานส่วนวิเคราะห์โปรโตคอล

1) เป้าหมายแรกที่พัฒนา

ต้องการโปรแกรมที่สามารถวิเคราะห์โปรโตคอล RS-232 Serial และ USB 1.0 โดยวิธีการวัดสัญญาณในสายส่งข้อมูล

2) ปัญหาที่พบ

การวัดสัญญาณในสารส่งข้อมูล (Physical Layer) ไม่เหมาะสมในการวิเคราะห์โปรโตคอล USB 1.0 เนื่องจากจะมีความยุ่งยากมากในเรื่องของการหาแพ็คเก็ต (Packet) และการตีความหมายของแพ็คเก็ตซึ่งสามารถเป็นโครงงานได้อีก 1 โครงงาน

3) การแก้ปัญหาและผลสรุป

ส่วนของโปรแกรมที่วิเคราะห์โปรโตคอล สามารถวิเคราะห์โปรโตคอลได้เฉพาะ RS-232 Serial

6.2 แนวทางในการพัฒนาต่อไป

พัฒนาให้โปรแกรมสามารถแสดงผลตามข้อมูลที่เปลี่ยนแปลงทันที (Real time) ซึ่งต้องพัฒนาส่วนฮาร์ดแวร์ให้รองรับการทำงานนี้ด้วย และพัฒนาส่วนวิเคราะห์โปรโตคอลให้สามารถรองรับโปรโตคอลได้หลากหลายมากขึ้น ทั้งนี้ต้องพัฒนาส่วนของฮาร์ดแวร์ควบคู่กันไปด้วย เนื่องจากว่าแต่ละโปรโตคอลมีการส่งข้อมูลที่ไม่เหมือนกัน ต้องทำอุปกรณ์มารองรับในการดักจับสัญญาณในสายส่งข้อมูลตามแต่ละโปรโตคอลไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ซอร์สโค้ด (Source Code)

ส่วนติดต่อกับฮาร์ดแวร์

```
#include "stdafx.h"
#include "TriggerControl.h"
#include "TriggerControlDlg.h"
#include <winioctl.h>
s32 runing;
CTriggerControlDlg::CTriggerControlDlg(CWnd* pParent /*=NULL*/)
	: CDialog(CTriggerControlDlg::IDD, pParent)
{
	//{{AFX_DATA_INIT(CTriggerControlDlg)
	//}}AFX_DATA_INIT
	// Note that LoadIcon does not require a subsequent DestroyIcon in Win32
	m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
void CTriggerControlDlg::DoDataExchange(CDataExchange* pDX)
{
	CDialog::DoDataExchange(pDX);
	//{{AFX_DATA_MAP(CTriggerControlDlg)
	DDX_Control(pDX, IDC_SAVE, m_save);
	DDX_Control(pDX, IDC_LTRIGGERCONDITION, m_trigger);
	DDX_Control(pDX, IDC_LCHANNEL, m_channel);
	DDX_Control(pDX, IDC_LRATE, m_rate);
	DDX_Control(pDX, IDSTART, m_start);
	//}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CTriggerControlDlg, CDialog)
	//{{AFX_MSG_MAP(CTriggerControlDlg)
	ON_WM_PAINT()
	ON_WM_QUERYDRAGICON()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ON_BN_CLICKED(IDSTART, OnStart)
	ON_WM_TIMER()
	ON_BN_CLICKED(IDC_SAVE, OnSave)
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()
BOOL CTriggerControlDlg::OnInitDialog()
{
	CDialog::OnInitDialog();
	SetIcon(m_hIcon, TRUE);
	SetIcon(m_hIcon, FALSE);
	m_rate.SetCurSel(0);
	m_trigger.SetCurSel(0);
	m_channel.SetCurSel(0);
	runing=0;
	SetTimer(1,1600,NULL);
	return TRUE;  // return TRUE  unless you set the focus to a control
}
void CTriggerControlDlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting
		SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(), 0);
		int cxIcon = GetSystemMetrics(SM_CXICON);
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
		GetClientRect(&rect);
		int x = (rect.Width() - cxIcon + 1) / 2;
		int y = (rect.Height() - cyIcon + 1) / 2;
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
	{
		CDialog::OnPaint();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
}
HCURSOR CTriggerControlDlg::OnQueryDragIcon()
{
        return (HCURSOR) m_hIcon;
}
void CTriggerControlDlg::OnStart()
{
        if(runing==0)
        {
                clk_speed=m_rate.GetCurSel();
                trg_type=0;
                trg_cond=m_trigger.GetCurSel();
                trg_sel=m_channel.GetCurSel();
                probe1();
                runing=1;
                m_start.SetWindowText("Stop");
        }
        else
        {
                runing=0;
                m_start.SetWindowText("Start");
        }
}
void CTriggerControlDlg::OnTimer(UINT nIDEvent)
{
        if(runing)
        {
                probe2();
                runing=0;
                m_start.SetWindowText("START");
        }
        CDialog::OnTimer(nIDEvent);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอพีไอมาตรฐานที่ใช้ในการพัฒนาโปรแกรม

(Standard API Specification)

รูปแบบไฟล์ชนิด Value Change Dump (VCD)

ไฟล์ชนิด Value Change Dump (VCD) ใช้บันทึกการเปลี่ยนแปลงค่าของตัวแปร เพื่อจำลองการทำงานของระบบ เช่น ค่าของสัญญาณในวงจรดิจิตอล เป็นต้น ไฟล์ชนิด VCD สามารถจัดเก็บการแสดงผลแบบกราฟ หรือผลของการวิเคราะห์หลังจากประมาลผลข้อมูลได้

ไฟล์ชนิด VCD มีขนาดใหญ่พอสำหรับงานออกแบบที่มีขนาดใหญ่และสามารถบันทึกผลการทำงานของระบบได้นาน ขนาดของไฟล์ชนิด VCD ถูกจำกัดโดยจำนวนสัญลักษณ์ที่นำมาใช้ซึ่งมีจำนวน 94^3 (830,584)

รูปแบบค่าของตัวแปร

ตัวแปรอาจเป็นแบบสเกล่าร์หรือเวคเตอร์ ซึ่งตัวแปรแต่ละชนิดถูกเก็บในรูปแบบที่ต่างกัน การเขียนแสดงการเปลี่ยนแปลงค่าของข้อมูลของตัวแปรแบบสเกล่าร์ต้องไม่มีช่องว่างระหว่างค่าของข้อมูลและตัวแปร การเขียนแสดงการเปลี่ยนแปลงค่าของข้อมูลของตัวแปรแบบเวคเตอร์ต้องไม่มีช่องว่างระหว่างตัวอักษรแสดงเลขฐานกับค่าของข้อมูล แต่ต้องมีช่องว่างระหว่างค่าของข้อมูลกับตัวแปร

รูปแบบของผลลัพธ์ของแต่ละค่าข้อมูลจะมองจากขวาไปซ้าย ค่าของเวคเตอร์สามารถเขียนให้สั้นที่สุดได้ เนื่องจากค่าซ้ายสุดของบิตข้อมูลจะถูกเติมเต็มตามจำนวนบิตที่มากที่สุด ซึ่งกฎของการเติมเต็มค่าด้านซ้ายของเวคเตอร์มีดังนี้

ค่าของข้อมูล	ค่าบิตด้านซ้ายที่เติมเต็ม
1	0
0	0
Z	Z
X	X

ตารางที่ ข-1 แสดงการเติมเต็มบิตด้านซ้ายของเวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขฐาน 2	การเติมให้เต็ม 4 บิต	ค่าที่เขียนในไฟล์ VCD
10	0010	b10
X10	XX10	bX1
ZX0	ZZX0	bZX0
0X10	0X10	b0X10

ตารางที่ ข-2 แสดงการเขียนค่าในรูปแบบย่อ

ข้อมูลถูกจัดเก็บในรูปแบบสเกล่าร์ เช่น 1*% จากตัวอย่าง 1 หมายถึงค่าข้อมูล *% หมายถึงตัวแปร ซึ่งแสดงถึงการเปลี่ยนแปลงของข้อมูลตามเวลาที่ระบุ

ตัวอย่าง

1*@ ต้องไม่มีช่องว่างระหว่าง 1 และตัวแปร *@

b1100x01z (k ต้องไม่มีช่องว่างระหว่าง b และ 1100x01z แต่ต้องมีช่องว่างระหว่าง b1100x01z และ (k

คำสั่ง

ไฟล์ VCD ประกอบไปด้วยส่วนต่างๆ ตามคำสั่ง ซึ่งใช้อธิบายความหมายของข้อมูลในไฟล์ ซึ่งประกอบด้วยคำสั่งต่างๆ ดังนี้

คำสั่งประเภทประกาศ

$comment
$date
$enddefinitions
$scope
$timescale
$upscope
$var
$version

คำสั่งประเภทอธิบายข้อมูล

$dumpall
$dumpoff
$dumpon
$dumpvars

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารประกอบอธิบายความหมายของคำสั่งต่างๆ ในไฟล์ VCD

$comment

The $comment section provides a means of inserting a comment in the VCD file.

Syntax:

$comment

comment_text

$end

Examples:

$comment

This is a single-line comment

$end

$date

The $date section indicates the date on which the VCD file was generated.

Syntax:

$date

date_text

$end

Example:

$date

June 25, 1989 09:24:35

$end

$enddefinitions

The $enddefinitions section marks the end of the header information and definitions.

Syntax:

$enddefinitions $end

$scope

The $scope section defines the scope of the variables being dumped.

Syntax:

$scope

scope_type *scope*_identifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$end

scope_type ::= module
| task
| function
| begin
| fork

The scope type indicates one of the following scopes:

Module top-level module and module instances

task tasks

function functions

begin named sequential blocks

fork named parallel blocks

Example:

$scope
module top
$end

$timescale

The $timescale keyword specifies what timescale was used for the simulation.

Syntax:

$timescale
number time_unit
$end

number ::= 1
| 10
| 100

time_unit ::= s
| ms
| us
| ns
| ps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| fs

Example:

$timescale

10 ns

$end

$upscope

The $upscope section indicates a change of scope to the next higher level in the design hierarchy.

Syntax:

$upscope $end

$version

The $version section indicates which version of the VCD writer was used to produce the VCD file.

Syntax:

$version

version_text

$end

Example:

$version

VERILOG-SIMULATOR 1.0a

$end

$var

The $var section prints the names and identifier codes of the variables being dumped.

Syntax:

$var

var_type size identifier_code reference

$end

var_type ::= event

| integer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| parameter

| real

| reg

| supply0

| supply1

| time

| tri

| triand

| trior

| trireg

| tri0

| tri1

| wand

| wire

| wor

size ::= decimal_number

reference ::= identifier| identifier [*bit_select*_index]

| identifier [*msb*_index : *lsb*_index]

index ::= decimal_number

Size specifies how many bits are in the variable.

The identifier code specifies the name of the variable using printable ASCII characters, as previously described.

a) The msb index indicates the most significant index; lsb index indicates the least significant index.

b) More than one reference name may be mapped to the same identifier code. For example, net10 and net15 may be interconnected in the circuit, and therefore may have the same identifier code.

c) The individual bits of vector nets may be dumped individually.

d) The identifier is the name of the variable being dumped in the model .

Example:

$var

integer 32 (2 index

$end

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$dumpall

The $dumpall keyword specifies current values of all variables dumped.

Syntax:

```
$dumpall
        { value_changes }
$end
```

Example:

```
$dumpall
        1*@
        x*#
        0*$
        bx  (k
$end
```

$dumpoff

The $dumpoff keyword indicates all variables dumped with X values.

Syntax:

```
$dumpoff
        { value_changes }
$end
```

Example:

```
$dumpoff
        x*@
        x*#
        x*$
        bx  (k
$end
```

$dumpon

The $dumpon keyword indicates resuming of dumping and lists current values of all variables dumped.

Syntax:

```
$dumpon
        { value_changes }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$end

Example:

$dumpon

x*@

0*#

x*$

b1 (k

$end

$dumpvars

The section beginning with $dumpvars keyword lists initial values of all variables dumped.

Syntax:

$dumpvars

{ value_changes }

$end

Example:

$dumpvars

x*@

z*$

b0 (k

$end

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
ตัวอักษรย่อ

API (Application Program Interface)

ฟังก์ชันหรือไลบารี่ที่ช่วยในการเขียนโปรแกรมให้สะดวกมากขึ้น

CMOS (Complementary Metal-Oxide Semiconductor)

วงจรที่ออกแบบไว้เป็นพิเศษ เพื่อการประหยัดกระแสไฟ วงจรซีมอสนี้มักจะใช้ในของชิ้นเล็กๆ เช่น นาฬิกาข้อมือ เครื่องคิดเลข คอมพิวเตอร์

DLL (Dynamic Link Library)

มักใช้เป็นนามสกุลของไฟล์ในระบบปฏิบัติการวินโดวส์ และโอเอส/ทู เป็นส่วนที่มีโปรแกรมตั้งแต่ 2 โปรแกรมขึ้นไปแบ่งกันใช้หรือใช้ร่วมกันได้

FPGA (Field Programable Gate Array)

เป็นอุปกรณ์อิเล็คทรอนิคส์ตัวหนึ่งที่สามารถเขียนโปรแกรมให้มีการทำงานตามที่ต้องการให้อุปกรณ์ชนิดนี้คล้ายกับไอซี แต่สามารถโปรแกรมการทำงานได้ ว่าให้มีการทำงานอย่างไร มีอินพุตหรือเอาท์พุตคืออะไร

GUI (Graphical User Interface)

เป็นวิธีการให้ความสะดวกแก่ผู้ใช้คอมพิวเตอร์ ให้ติดต่อสื่อสารกับเครื่องคอมพิวเตอร์โดยผ่านทางภาพ เช่น ใช้เมาส์กดเลือกไอคอนแทนการพิมพ์คำสั่ง หรือการเลือกคำสั่งตามรายการเลือกที่เรียกว่าเมนู

PLD (Programable Ligic Device)

เป็นอุปกรณ์อิเล็คทรอนิคส์ตัวหนึ่งที่สามารถเขียนโปรแกรมให้มีการทำงานตามที่ต้องการให้อุปกรณ์ชนิดนี้คล้ายกับไอซี แต่สามารถโปรแกรมการทำงานได้ ว่าให้มีการทำงานอย่างไร มีอินพุตหรือเอาท์พุตคืออะไร

VCD (Value change dump)

เป็นมาตรฐานของไฟล์แบบหนึ่ง ที่ใช้อธิบายถึงการเปลี่ยนนแปลงของข้อมูล ณ เวลาใดๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง
คู่มือการใช้งานโปรแกรม

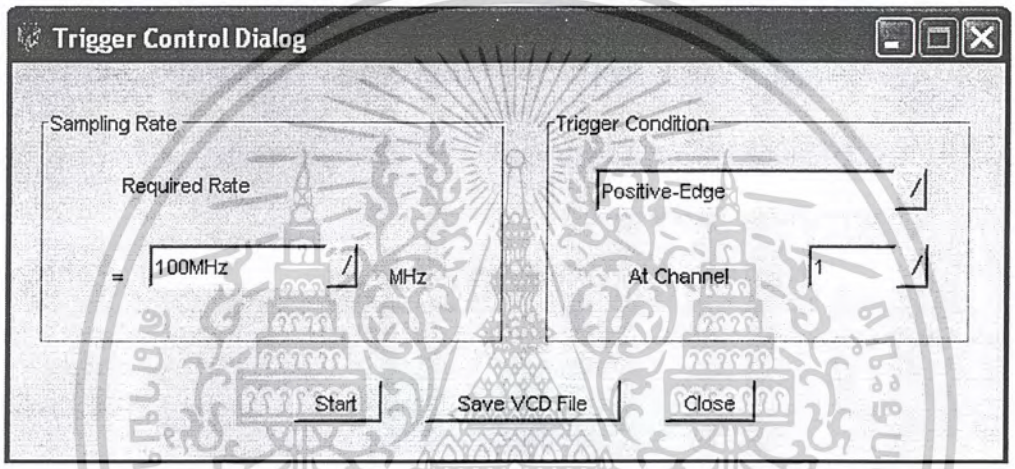
คู่มือการใช้งานประกอบด้วยส่วนต่างๆ ดังนี้

1) คู่มือการใช้งานในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

2) คู่มือการใช้งานในส่วนส่วนวิเคราะห์โปรโตคอล

3) คู่มือการใช้งานในส่วนแสดงผลและวิเคราะห์ข้อมูล

ง-1 คู่มือการใช้งานในส่วนติดต่อกับอุปกรณ์วัดสัญญาณ



รูปที่ ง-1 หน้าจอส่วนติดต่อกับอุปกรณ์วัดสัญญาณ

- Sampling Rate

  ต้องการให้ฮาร์ดแวร์สุ่มสัญญาณด้วยความถี่เท่าไหร่

- Trigger Condition

  ต้องการให้ฮาร์ดแวร์เริ่มทำงานตามเงื่อนไขที่ตั้งไว้

- Start

  สั่งให้ฮาร์ดแวร์เริ่มทำงาน

- Save VCD File

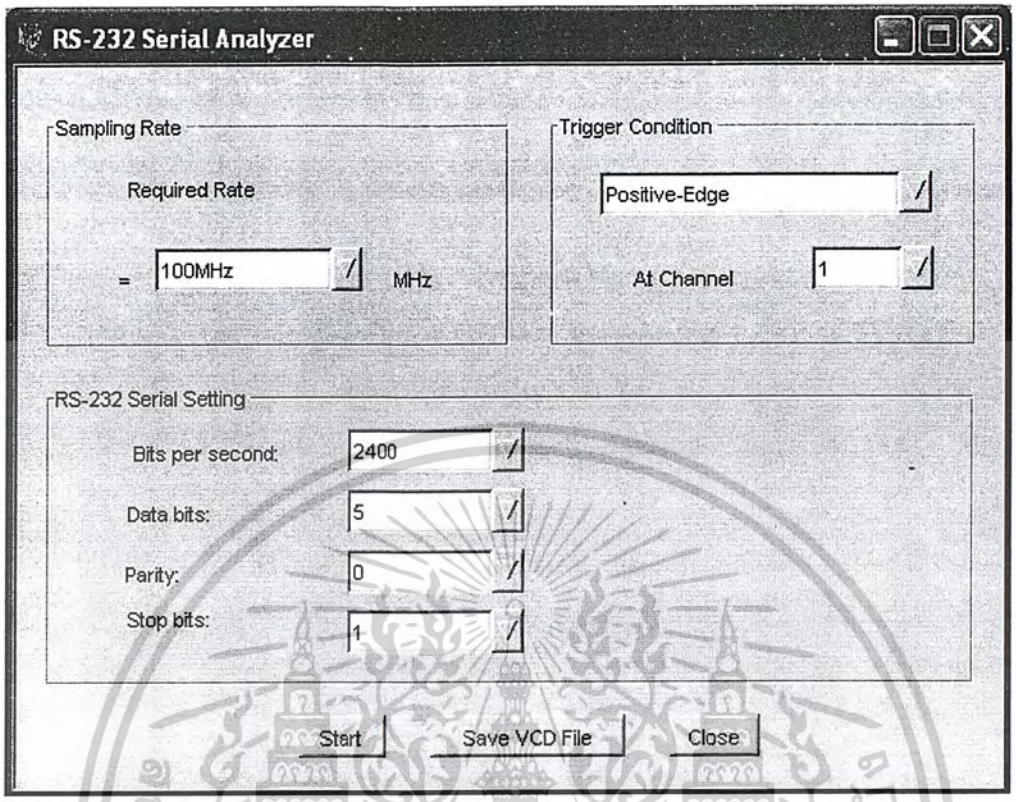
  บันทึกข้อมูลลงไฟล์ชนิด VCD

- Close

  ปิดหน้าต่าง Trigger Control Dialog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง-2 คู่มือการใช้งานในส่วนวิเคราะห์โปรโตคอล



รูปที่ ง-2 หน้าจอส่วนวิเคราะห์โปรโตคอล

- Sampling Rate

ต้องการให้ฮาร์ดแวร์สุ่มสัญญาณด้วยความถี่เท่าไหร่

- Trigger Condition

ต้องการให้ฮาร์ดแวร์เริ่มทำงานตามเงื่อนไขที่ตั้งไว้

- RS-232 Serial Setting

ตั้งค่าของสัญญาณข้อมูลของ RS-232 Serial ที่ใช้ในการส่งข้อมูล

- Start

สั่งให้ฮาร์ดแวร์เริ่มทำงาน

- Save VCD File

บันทึกข้อมูลลงไฟล์ชนิด VCD

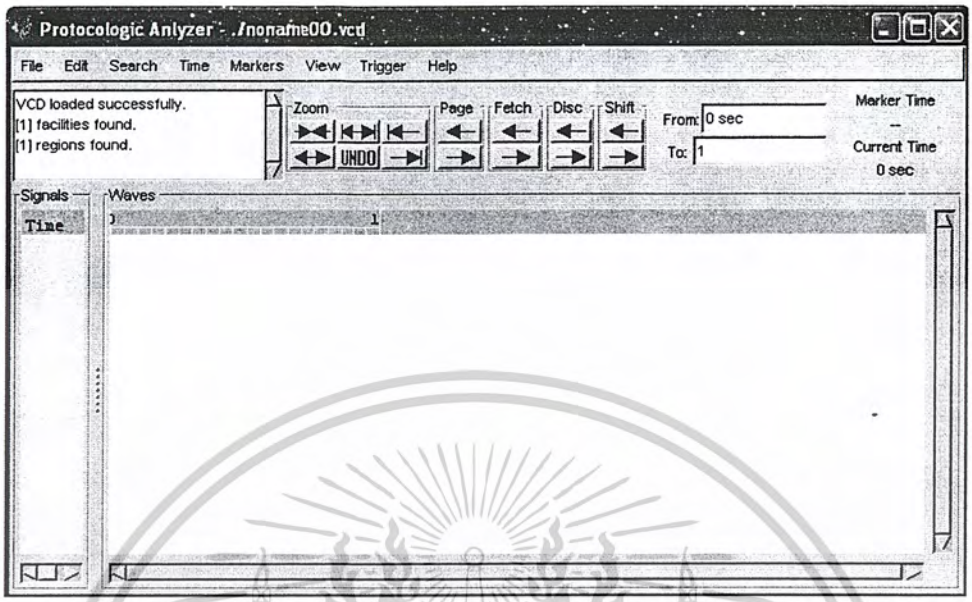
- Close

ปิดหน้าต่าง Trigger Control Dialog

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

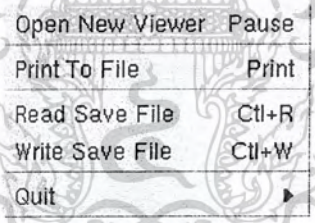
ง-3 คู่มือการใช้งานในส่วนแสดงผลและวิเคราะห์ข้อมูล

หน้าตาของโปรแกรม มีลักษณะดังนี้



รูปที่ ง-3 หน้าจอส่วนแสดงผลและวิเคราะห์ข้อมูล

1) เมนู File



รูปที่ ง-4 เมนู File

- Open New Viewer

ทำการเปิดหน้าต่างของโปรแกรมใหม่ ซึ่งโปรแกรมจะให้เลือกไฟล์ที่ต้องการให้แสดงผลในหน้าต่างที่ปิดใหม่นั้น

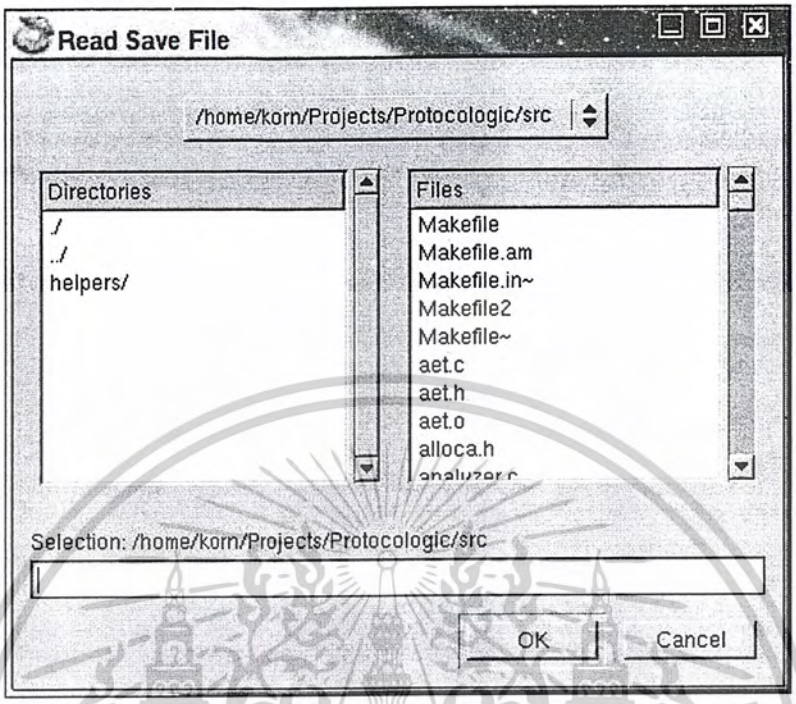
- Print To File

ทำการพิมพ์ข้อมูลลงไฟล์ในรูปแบบ PS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Read Save File

ทำการเปิดไฟล์ที่เซฟไว้ขึ้นมาแสดงผล โดยมี Dialog ดังนี้



รูปที่ ง-5 หน้าจอ Read Save File

- Write Save File

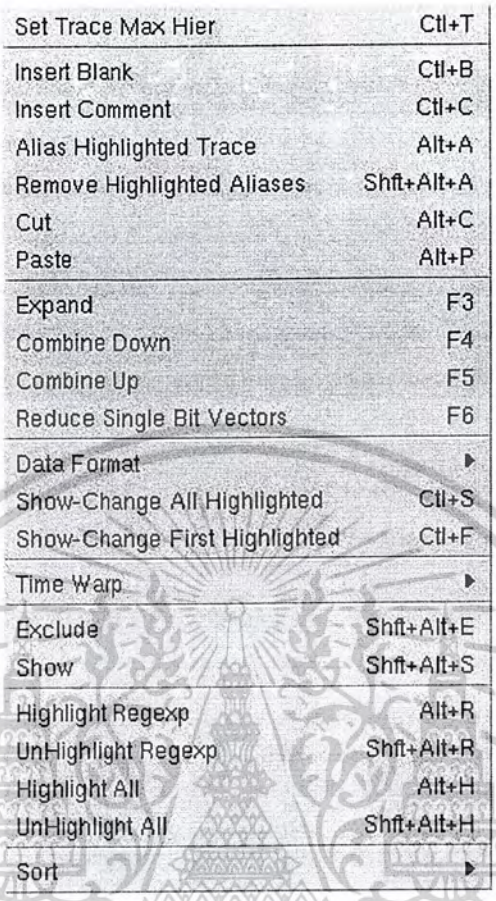
ทำการบันทึกข้อมูลที่แสดงผลลงไฟล์

- Quit

ออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) เมนู Edit



รูปที่ ง-6 เมนู Edit

- Set Trace Max Hier

ทำการเซตชื่อของข้อมูลที่แสดงว่าจะให้มีลำดับชั้นของชื่อข้อมูลมากน้อยแค่ไหน

- Insert Blank

แทรกบรรทัดว่างลงไปล่าง trace ที่เลือกอยู่

- Insert Comment

แทรก Comment ลงไปล่าง trace ที่เลือก

- Alias Highlighted Trace

เปลี่ยนชื่อของ trace ข้อมูลที่เลือก

- Remove Highlighted Aliases

ยกเลิกการชื่อของ trace ข้อมูลที่เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Cut

ทำการตัด trace ข้อมูลที่เลือกออก และเก็บไว้ใน clip board

- Paste

ทำการวาง trace ข้อมูลจาก clip board ล่าง trace ที่เลือก

- Expand

ทำการขยายข้อมูลที่รวมกลุ่มกันอยู่

- Combine Down

ทำการรวมกลุ่มของสัญญาณที่เลือก โดยเรียงจากบนลงล่าง

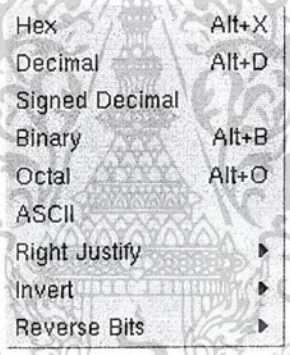
- Combine Up

ทำการรวมกลุ่มของสัญญาณที่เลือก โดยเรียงจากล่างขึ้นบน

- Reduce Single Bit Vectors

ทำการขยายการรวมกลุ่มของข้อมูล ถ้า trace ที่เลือกมีข้อมูลอยู่ 1 บิต

2.1) เมนูย่อย Data Format



รูปที่ ง-7 เมนูย่อย Data Format

- Hex

แสดงข้อมูลของ trace ที่เลือก ให้เป็นตัวเลขฐาน 16

- Decimal

แสดงข้อมูลของ trace ที่เลือก ให้เป็นตัวเลขฐาน 10

- Signed Decimal

แสดงข้อมูลของ trace ที่เลือก ให้เป็นตัวเลขฐาน 10 แบบมีเครื่องหมาย

- Binary

แสดงข้อมูลของ trace ที่เลือก ให้เป็นตัวเลขฐาน 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Octal

แสดงข้อมูลของ trace ที่เลือก ให้เป็นตัวเลขฐาน 8

- ASCII

แสดงข้อมูลของ trace ที่เลือก ให้เป็นตัว ASCII

- Right Justify

แสดงข้อมูลของ trace ที่เลือก ให้เป็นแบบ Right Justify

- Invert

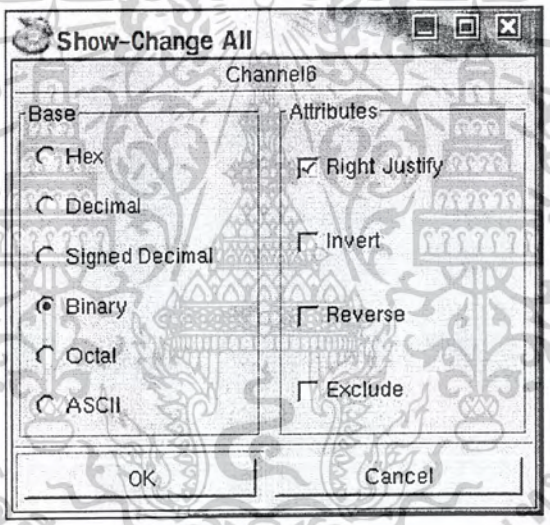
แสดงข้อมูลของ trace ที่เลือก โดยทำการกลับค่า 1 เป็น 0 และ 0 เป็น 1

- Reverse Bits

แสดงข้อมูลของ trace ที่เลือก โดยเรียงลำดับของบิตข้อมูลกลับด้าน

- Show-Change All Highlighted

ทำการปรับเปลี่ยนการแสดงผลของ trace ข้อมูลที่เลือก โดยจะขึ้น Dialog ดังนี้



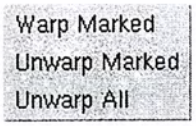
รูปที่ ง-8 หน้าจอ Show-Change All Highlighted

- Show-Change First Highlighted

ทำการปรับเปลี่ยนการแสดงผลเฉพาะ trace ข้อมูลที่เลือกลำดับแรก โดยจะขึ้น Dialog คล้ายกับ Show-Change All Highlighted

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2) เมนูย่อย Time Warp



รูปที่ ง-9 เมนูย่อย Time Warp

- Warp Marked

ตั้งค่าการแสดงผลของ trace ข้อมูลที่เลือก ว่าให้เริ่มแสดงผลตั้งแต่เวลาเท่าไหร่

- Unwarp Marked

ยกเลิกการตั้งค่าเวลาเริ่มต้นการแสดงข้อมูล ของ trace ที่เลือก

- Unwarp All

ยกเลิกการตั้งค่าเวลาเริ่มต้นการแสดงข้อมูล ทั้งหมด

- Exclude

ซ่อน ไม่ให้แสดงข้อมูลของ trace ข้อมูลที่เลือก

- Show

แสดงข้อมูลของ trace ข้อมูลที่เลือก ที่ถูกซ่อนไว้

- Highlight Regexp

ทำการเลือก trace ข้อมูล โดยขึ้นอยู่กับเงื่อนไขตาม Regular Expression

- UnHighlight Regexp

ยกเลิกการเลือก trace ข้อมูล โดยขึ้นอยู่กับเงื่อนไขตาม Regular Expression

- Highlight All

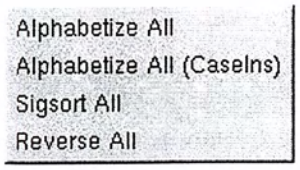
ทำการเลือก trace ข้อมูลทั้งหมด

- UnHighlight All

ยกเลือกการเลือก trace ข้อมูลทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3) เมนูย่อย Sort



รูปที่ ง-10 เมนูย่อย Sort

- Alphabetize All

เรียงชื่อ trace ข้อมูล ตามตัวอักษรในพจนานุกรม ช่องว่างจะอยู่ท้ายสุด

- Alphabetize All (CaseIns)

เรียงชื่อ trace ข้อมูล ตามตัวอักษรในพจนานุกรม แล้วไม่สนใจตัวอักษรตัวเล็ก ตัวใหญ่ในภาษาอังกฤษ ช่องว่างจะอยู่ท้ายสุด

- Sigsort All

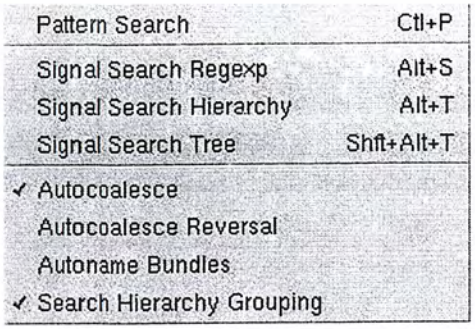
เรียงชื่อ trace ข้อมูล ตามตัวอักษร และตามลำดับของตัวเลข

- Reverse All

เรียงชื่อ trace ข้อมูลที่แสดงผล กลับด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

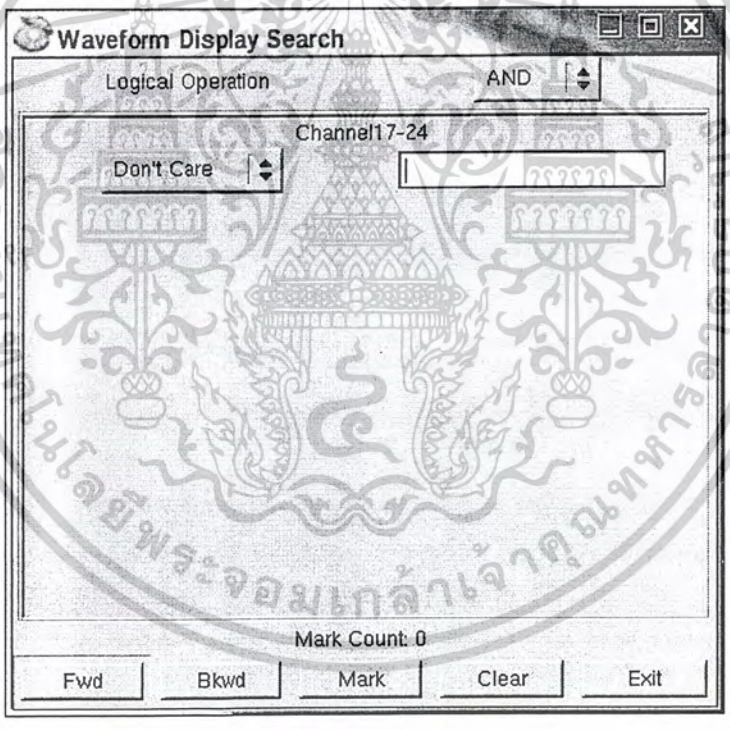
3) เมนู Search



รูปที่ ง-11 เมนู Search

- Pattern Search

ทำการค้นหาข้อมูลตามลักษณะที่ต้องการ โดยจะมี Dialog ดังนี้

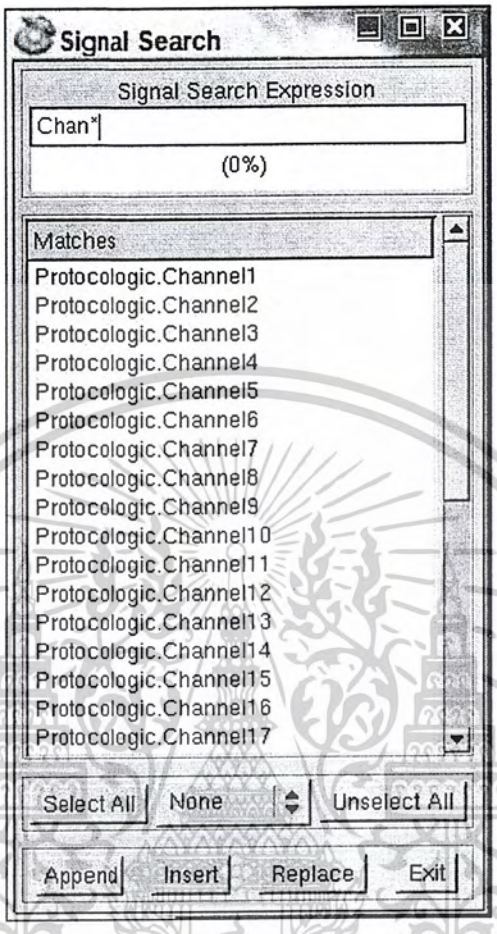


รูปที่ ง-12 หน้าจอ Pattern Search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Signal Search Regexp

ทำการค้นหาสัญญาณข้อมูล มีเงื่อนไขตาม Regular Expression โดยจะมี Dialog ดังนี้

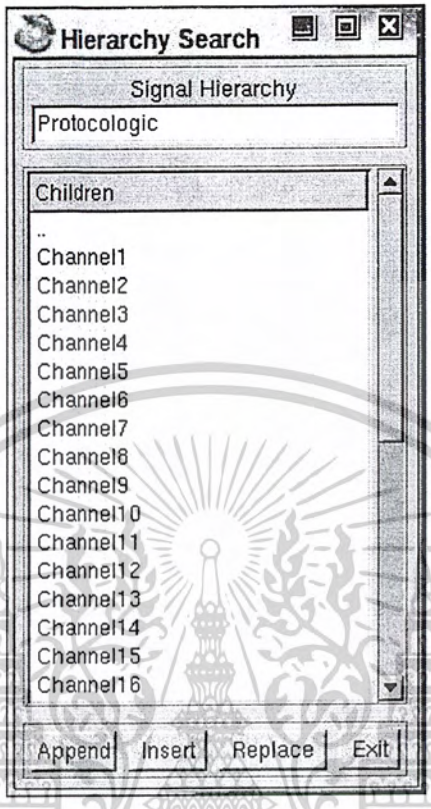


รูปที่ ง-13 หน้าจอ Signal Search Regexp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Signal Search Hierarchy

ทำการค้นหาสัญญาณข้อมูล ตามลำดับชั้นของสัญญาณ โดยมี Dialog ดังนี้

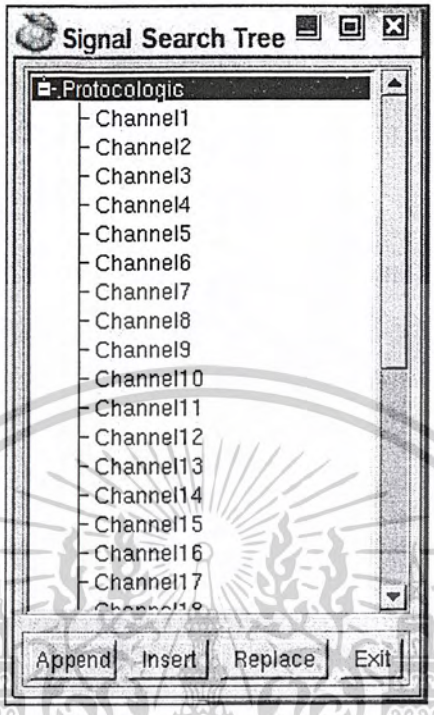


รูปที่ ง-14 หน้าจอ Signal Search Hierarchy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Signal Search Tree

ทำการค้นหาสัญญาณข้อมูล แบบ Tree โดยมี Dialog ดังนี้



รูปที่ ง-15 หน้าจอ Signal Search Tree

- Autocoalesce

เมื่อถูกเลือก จะทำให้โปรแกรมสามารถรวมกลุ่ม จากกลุ่มสัญญาณที่ถูกแยกออกจากกันมาก่อนได้

- Autocoalesce Reversal

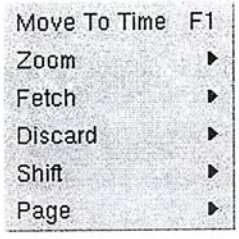
เมื่อถูกเลือก จะทำให้โปรแกรมสามารถรวมกลุ่ม จากกลุ่มสัญญาณที่ถูกแยกออกจากกันมาก่อนได้ โดยเรียงกลับด้านกัน และ Autocoalesce ต้องถูกเลือกอยู่

- Search Hierarchy Grouping

เมื่อเลือกไว้ จะทำให้ข้อมูลใหม่ (เมื่อเปลี่ยนแปลง) ถูกนำเข้ามายัง Tree Search และ Hierarchy Search

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

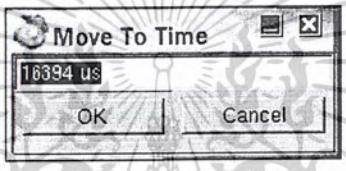
4) เมนู Time



รูปที่ ง-16 เมนู Time

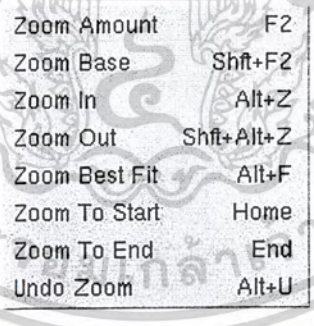
- Move To Time

ทำการย้ายจอแสดงผล ให้ไปแสดง ณ เวลาที่เลือก โดยมี Dialog ดังนี้



รูปที่ ง-17 หน้าจอ Move To Time

4.1) เมนูย่อย Zoom



รูปที่ ง-18 เมนูย่อย Zoom

- Zoom Amount

ทำการย่อ/ขยาย การแสดงผลข้อมูลตามตัวเลยที่ระบุ โดยถ้าเป็น + จะขยาย ถ้าเป็น - จะย่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Zoom Base

ทำการย่อ/ขยาย การแสดงผลข้อมูลเป็นจำนวนเท่า (ลักษณะคล้ายแว่นขยาย) โดยมีค่าระหว่าง 1.5 - 10

- Zoom In

ทำการขยาย การแสดงผลข้อมูล

- Zoom Out

ทำการย่อ การแสดงผลข้อมูล

- Zoom Best Fit

แสดงผลข้อมูลทั้งหมด ภายให้หน้าเดียว

- Zoom To Start

แสดงผลข้อมูล ณ เวลาเริ่มต้น

- Zoom To End

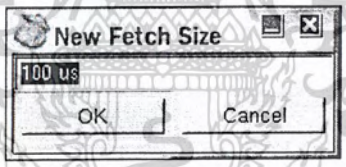
แสดงผลข้อมูล ณ เวลาสุดท้าย

- Undo Zoom

ยกเลิกการย่อ/ขยายครั้งล่าสุด

- Fetch Size

ทำการกำหนดค่า สำหรับใช้ fetch และ discard โดยมี



รูปที่ ง-19 หน้าจอ Fetch Size

- Fetch Right

เพิ่มค่าสิ้นสุดของเวลาที่แสดงผลข้อมูล

- Fetch Left

ลดค่าเริ่มต้นของเวลาที่แสดงผลข้อมูล

- Discard Right

ลดค่าสิ้นสุดของเวลาที่แสดงผลข้อมูล

- Discard Left

เพิ่มค่าเริ่มต้นของเวลาที่แสดงผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Shift Right

เลื่อนหน้าจอแสดงผลข้อมูลไปทางขวาทีละ 1 หน่วย

- Shift Left

เลื่อนหน้าจอแสดงผลข้อมูลไปทางซ้ายทีละ 1 หน่วย

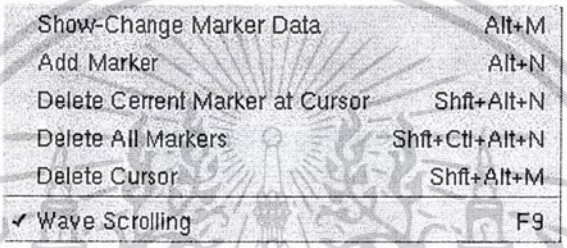
- Page Right

เลื่อนหน้าจอแสดงผลข้อมูลไปทางขวาทีละหน้าจอ

- Page Left

เลื่อนหน้าจอแสดงผลข้อมูลไปทางซ้ายทีละหน้าจอ

5) เมนู Markers



รูปที่ ง-20 เมนู Markers

- Show-Change Marker Data

แสดงหรือเปลี่ยนข้อมูล ของ Marker ต่างๆ

- Add Marker

เพิ่ม Marker ณ ตำแหน่งของ Cursor ปัจจุบัน

- Delete Cerrent Marker at Cursor

ลบ Marker ณ ตำแหน่งของ Cursor ปัจจุบัน

- Delete All Markers

ลบ Marker ทั้งหมด

- Delete Cursor

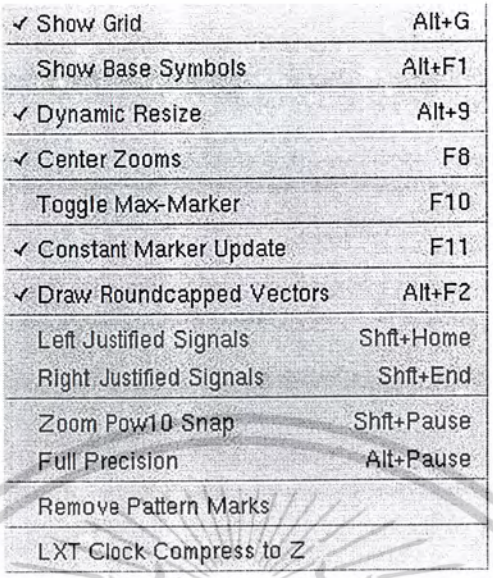
ลบ Cursor

- Wave Scrolling

เปิด/ปิด ให้หน้าจอเลื่อนตาม Cursor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) เมนู View



รูปที่ ง-21 เมนู View

- Show Grid

  แสดง/ซ่อน Grid

- Show Base Symbols

  แสดง/ซ่อน สัญลักษณ์แสดงฐานของเลข

- Dynamic Resize

  อนุญาต/ไม่อนุญาต ให้ทำ Dynamic Resize

- Center Zooms

  อนุญาต/ไม่อนุญาต เมื่อทำการย่อ/ขยาย แล้วจะเอาตำแหน่งของ Cursor อยู่ตรงกลางจอภาพ

- Toggle Max-Marker

  สลับการแสดงผลระหว่าง Maximum Time และ Marker Time

- Constant Marker Update

  อนุญาต/ไม่อนุญาต ให้แสดงค่าของข้อมูล ณ ตำแหน่งของ Cursor เมื่อกำลัง Drag Mouse

- Draw Roundcapped Vectors

  อนุญาต/ไม่อนุญาต ให้แสดงข้อมูลที่เป็นกลุ่มๆ แบบสี่เหลี่ยม

- Left Justified Zooms

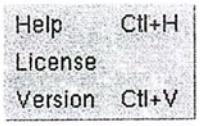
  จัดให้ชื่อของ trace ข้อมูลอยู่ชิดซ้าย

- Right Justified Zooms

  จัดให้ชื่อของ trace ข้อมูลอยู่ชิดขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8) เมนู Help



รูปที่ ง-24 หน้าจอ เมนู Help

- Help

เปิด Dialog ของ Help

- License

แสดง License Agreement

- Version

แสดง Version ของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] "USB Analyzer", http://www.ellisys.com/products/tracker110/index.php

[2] "GTKWave", http://www.cs.man.ac.uk/apt/tools/gtkwave/index.html

[3] "Palm Logic", http://www.design-gateway.com

[4] "Value change dump file", http://www-ee.eng.hawaii.edu/~msmith/ASICs/HTML/Verilog /LRM/HTML/15/ch15.htm

[5] "Universal Serial Bus", http://www.beyondlogic.org

[6] สุรศักดิ์ สงวนพงษ์ และ เอกพล โรจน์รัตนาวิชัย, "ไวยากรณ์นำและนิยามสนับสนุนตัววิเคราะห์-โปรโตคอลแบบสากล", 22nd Electrical Engineering Conference (EECON-22), Bangkok December,1999

[7] นัฐกร ชาติประทีป, "เครื่องวัดและวิเคราะห์สัญญาณดิจิตอลลอจิก", 2543

[8] ณัฐพล สุทธินนท์, นิวาสน์ บุญหมื่น, มนต์ชัย แช่ช้อย, "โลจิกอนาไลเซอร์", 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้